UtilitasMathematica

ISSN 0315-3681 Volume 122, 2025

TEXT NORMALIZATION AND SPELL CORRECTION OF PUNJABI TEXT

Jagdish Kaur[1]
Research Scholar
Department of Computer Science
Punjabi University
Patiala, Punjab, India
jagdishkaurmehrok@gmail.com

C P Kamboj[2]
Assistant Professor
Department of Punjabi Computer Help Center
Punjabi University
Patiala, Punjab, India
cpk@pbi.ac.in

Abstract

Text Normalization is the practice of mapping non-standardized words into standardized and canonical form.Training a language model of Punjabi language for Grammar Checker is very tedious task as notplentifulcorrect dataset for Punjabi linguistic is available. Collecting data from different sources may include noisy text, spelling errors and unwanted text etc. which require text normalization to make these data more suitable for training language model. In this paper we look at various texts' normalization methods including spelling correction and highlight our framework for normalizing the Punjabi text.We treat text normalization of Punjabi text with neural machine translation approach. In this paper we propose ahybrid approach using deep learning-based encoder-decoder model using fine tuning of transformer with copy input method to do the task of text normalization and spelling correction of Punjabi language misspelled words and statistical technique which is highlighted and could be used as pre-processing or post-processing for enhancing the performance of our proposed model architecture. We trained and evaluated our proposed model on prepared Punjabi language parallel dataset consisting of correct-incorrect words.The experiments reveal that our proposed model touches significant performance on various semiotic classes and outperforms other existing models in terms of the accuracy.

Keywords: Tokenization, Normalization, NeuralNetworks, Transformer, Deep Learning

1. Introduction

Punjabi language is the most widely spoken language in Punjab of India and Pakistan. Around 88.9 native speakers in India, 31.1 million in Pakistan and 150 million approximately in the world of Punjabi language. Despite this, Punjabi language is characterized by a complexity in its grammatical and spelling rules. There is necessity of automatic grammar checker which can ensure the correctness of language usage including spelling. As grammar checker is one of major application of natural language processing, to develop such model demand huge amount of correct dataset. Despite this, data which is available on the internet of Punjabi language have lot of disparities including grammar errors, spelling errors, and containing noise text including alphanumeric characters, hash tags, emojis, URL tags and other languages characters that cause hindrance in understanding. Thus, there is requirement of text normalization of this unstructured data prior to processing or training by deep learning model. Text normalization tasks are applied to overcome the dataset from such irregular properties.

2. Literature Review

2.1 Text Normalization

In Natural Language Processing text normalization play a significant role. It plays avital role in various applications. To convert the non-standard words corresponding to their standard formats text normalization is required. For example, the word “ਵੇਖਨਾ“can be transformed to “ਵੇਖਣਾ”, in its canonical form. This is very beneficialin preparing text for later processing. By text normalization, other operations will able to work with the standard data.Non-Standard Words may also contain phone numbers, hash tags, dates, currencies, addresses, acronyms etc. and it became required to convert them into standard format.Mostly noisy texts appear in social media contents, text messages and comments given by people to blog posts where misspellings,use of out-of-vocabulary words (OOV) and abbreviationsare prevalent, text normalization is very important. The normalization process can improve text matching. Before using the text data in training NLP model, it's necessary to pre-processed to a suitable form. During text preprocessing phase text normalization is mainly required before training the model. Text normalization is the modeling process and it can enhance the model's performance. Text normalization comprisesvarious stages includingcase normalization, stemming,Lemmatization, Removal of ***punctuations***, numbers and non-textual elements and Spelling Correction etc. depend upon requirements of application. Case normalizationtechnique is mainly helpful for text which is combination of lowercase and uppercase letters especially in English language text and to convert that text in one standardized form. Stemming is thetechniquethatlessen inflected words to their stem, base or root words. Strip suffixes. For example, ‘ਖੇਡ' is stemmingterm of ਖੇਡਿਆand ਖੇਡਣਾwords. Stemming is a simpleand faster methodto apply in comparision to Lemmatization**.**Removal of ***punctuations***, numbers and non-textual elements normalization is also included to remove punctuation mark, non-textual elements like url tags, email addresses, special characters, numbers, symbols, emojis and devanagari characters from text.Lemmatizationis the mechanism used to diminish surface forms to their root form. Tokenization is the mechanism to split a given text (words and sentences) to its possible smallest morpheme like sentence into words or word into characters is known as token. A token might be of character, number, symbol, n-gram and word type in text. The tokenizer receives the text after Pre-processing phase and generate tokens. For example:ਜਾਦੂਗਰਨੇਜਾਦੂਵਿਖਾਇਆ।This sentence tokenizes as: “ਜਾਦੂਗਰ “,” ਨੇ “,” ਜਾਦੂ “,” ਵਿਖਾਇਆ “**.** Spelling Correction is useful to correct spelling errors under preprocessing phase to improve the accuracy of model.There are various models existing based on rule based, statistical and deep learning for such task.

2.1.1 Rule- based Approach

A set of rules are predefined set for text normalization.These rules are created by expert who has linguistic knowledge and methods used to create such rules may include regular expressions, pattern matching and heuristic based techniques. For example, there may a rule which can replace all alphanumeric values from text with blank space. Ariffin and Tiun[1] proposed a text normalizer for Malay language using rule-based method. They trained their model on tweets dataset which consist of Malay words and other non-standard words. Those modelswere designed to normalize the Malay, English and Romanized Arabic words from tweets. Osama and Asim Karim [2] proposed a rule –based model for normalizing SMS texts which are written in Romanized Urdu and English.Nana Mulyana Maghfur and Muhammad Okky Ibrohim et.al;[3] proposed a rule-based approach to normalize Indonesian text dataset and its corresponding spoken form to enhance the performance of their Indonesian TTS(Text to Speech) model. Dang & Phan [4] presented text normalization of Non-Standard Words (NSWs) into their spoken version using a set of pre-defined rule classes.

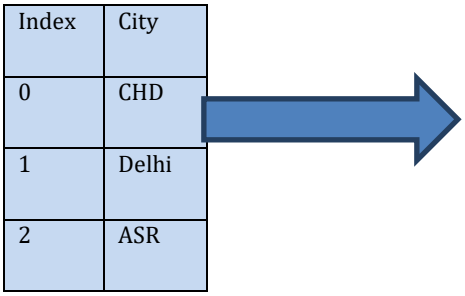
2.1.2 Statistical Based

Statistical based models are also used for normalizing text by applying statistical rules which are generated by learned from data. These models use machine algorithm to identify pattern from input large dataset. Various statistical based techniques are existing for text normalization task. Levenshtein Distance and N-gram models are famous techniques for such task.Levenshtein Distance is used to calculate the distance between input word and known words of dictionary to find closest match and correct spelling or typo errors of that input word. N-gram based model is used to discover the probability of assumed n-gram from each word occurring in given sequence or text in language.N-gram based model is specificallyapplied to predict

next word in sequence particularly to correct grammar errors in input text. Tim, Chenfei, Daniel and Tanja [5] proposed statistical based machine translation method for text normalization.They trained their model on parallel dataset of normalized and non-normalized text to translate non-normalized text into normalized text.Aw, Zhang, Xiao, & Su [10] presented a phrase-based statistical machine translation model to normalize SMS text by converting it into standard form in English language after removing its irregularities. In another similar research Scannell [38] proposed statistical based method for text normalization and machine translation from Scottish Gaelic to Irish, which is also then used for normalizing pre-standard Irish texts.

2.1.3 Neural Network Based Model

In present time, Neural Network has gained everyone 's attention to itself. Such type of network is consisting of neurons that resemble human brain. Its deep layered structure makes it deep learning model which trained on large dataset of text to identify the complex pattern of text [11] and dependencies or contextual relationship among words in text.These models are then automatically applied their rules on input text which they learned from data during training stage for the task of text normalization. With some fine-tuning in such models, they can be very effective to transform non -standard text into standardized form. Deep learning models in the techniques like Keras and tensorflow, basicallyneedsevery input and output variables to be numeric. This means that text data must encode to numbers before it can fit and evaluate a model under text normalizationphase. Neural Networks offers encoding techniquessuch as One Hot Encoding and Word Embedding for vector representation in Natural Language Processing.***One Hot Encoding***: A one hot encoding is suitable for uncompromising data where no relationship exists between categories. Afresh binary variables are known as ***Dummy variables*.** The number of dummy variables depends on the levels exist in the uncompromising variable. Suppose we have a dataset with a category having different cities like Chandigarh, Delhi, Amritsar, Ludhiana, and Patiala representing one hot encoding in figure 2. Word embedding is another technique of turning texts into numbers because machine learning algorithms can only understand numbers not plain texts. It also used to contain the meaning of the sentence from the words and after training on a large data set, it can even find words that are not existed in vector representation from sentences. It can help the model for reasonably small unlabeled dataset that can be useful for a model to train a larger dataset and use transfer learning. Word Embedding is useful to use One Hot vector in tasks for dedicated large datasets because it does not work exceptionally well on those tasks. Logically this techniqueis hard to implement as computationally it is more expensive than One Hot vector. **Word2Vec** was created by Google in 2013, a two-layer neural network that processes text by "vectorizing" words. **Word2Vec**'s input is a text from corpus and output is a set of vectors.This technique turns text into a numerical form which deep neural networks can understand. Word2vec is used to group the vectors of similar words together in vector space. The vectors used to signify words. This technique is known as ***neural word embeddings***. Word2vec is analogous to an auto encoder, encoding each word in a vector, but rather than training against the input, it also trains words against other words that neighbour them in the input corpus. It normally performed by using two techniques, either using context to predict a target word or using a word to predict a target context, popularly known as skip-gram.Ajay Shankar and Sudip Kuar Naskar [12] proposed deep learning-based text normalization using encoder-decoder model with RNN and LSTM of social media text like facebook, whatsapp and twitter etc. They also spoke the role of synthetic dataset in transfer learning approach and achieved state of art 0.9098 F1 score which outperform previous model.



Index	City
0	CHD
1	Delhi
2	ASR

3	LDH
4	Patiala

Index	CHD	Delhi	ASR	Patiala	LDH
0	1	0	0	0	0
1	0	1	0	0	0
2	0	0	1	0	0
3	0	0	0	0	1
4	0	0	0	1	0

Figure 1: One Hot Encoding

Table 1: One Hot Encoding

Index	**Chandigarh**	**Delhi**	**Amritsar**	**Ludhiana**	**Patiala**
0	1.0	0.0	0.0	0.0	0.0
1	0.0	1.0	0.0	0.0	0.0
2	0.0	0.0	1.0	0.0	0.0
3	0.0	0.0	0.0	1.0	0.0
4	0.0	0.0	0.0	0.0	1.0
5	1.0	0.0	0.0	0.0	0.0

2.1.4 Hybrid Model

Hybrid technique is the combination of both rule-based and machine learning or deep learning technique. Rule based technique is very helpful in solving some type of errors and machine or deep learning approach is better in solving some other types of errors. So, in hybrid system each part should be implemented according to its 'competence'. In this hybrid approach, corpus of text can be used to train the system for identifying correct pattern of words and the output can be filtered by applying some hand-crafted rules. Hybrid technique is helpful in solving complex type errors.

2.2 Spelling Correction

A Spelling Correction is a process to find the correct spelling of word if it is found misspelled. A word is misspelled word in a text if it is not a valid word of that language and typically not found in dictionary of the corresponding language [6] and seeks for the best possible suggestion. If a word is present in dictionary it supposed to be a correct word. There are various fields like machine translation, searching, information retrieval etc. where spell correction is required. Spell checker has three main components .i.e. error detection, candidate spelling generator and error correction. Error detection is a procedure to detect error or misspelled word. Candidate spelling generator is used to provide spelling suggestions for detected errors and error correction is used to correct errors. N-Gram analysis and dictionary lookup are different techniques available for error detection. For spelling correction there are various techniques available like rule based, edit distance, similarity keys, N-Gram and neural network based. Before studying about error detection and correction techniques, it is very important to know how these errors occur. The Spelling and typing mistakes are very common in given input text made by human. Detecting the spelling errors and rectifying them automatically is an enormous research challenge. There are two types of errors i.e. real word errors and non-real word errors [18]***Typographic Errors***are non-real word errors are occurring when the word is mistyped

by mistake even if the user knows the correct spellings of the word. There are various types of typographic errors. ***Insertion error***appears when at least one additional character is inserted in the word typed by human. ਸੇਵਾਦਾਰ>ਸੇਵਾਦਧਾਰ***Deletion error***arises when at least one character is deleted from the desired word by human. ਸੇਵਾਦਾਰ>ਸੇਵਾਦਰ***Substitution error*** occurs when at least one desired character is replaced by the other character. ਸੇਵਾਦਾਰ>ਸੇਵਾਡਾਰ***Transposition error***occurs when two adjacent characters transposed in desired word.ਸੇਵਾਦਾਰ>ਸੇਵਾਦਰਾ***Run on error*** are when space is missed between two words then this error occurred. ਅੰਮ੍ਰਿਤਬਾਣੀ>ਅੰਮ੍ਰਿਤ ਬਾਣੀ***Split word error*** arises when some additional space is inserted between parts of a word. This is opposite of run on error. ਸੇਵਾਦਾਰ>ਸੇਵਾਦਾਰ***Cognitive Errors*** are noticed when the user typed word but does not know the correct spellings of that word. These are also known as real word errors.

2.2.1 Spelling Errors Detection Techniques

For error detection input text is tokenized in words by using a tokenizer. The word is checked for its validity. The word is considered to be a valid word if it has a meaning otherwise it is a non-word. There are commonly two techniques available for error detection i.e. dictionary lookup approach and N-gram analysis. ***Dictionary lookup*** is a spell-checking technique. Dictionary is a lexical source that containing list of valid words of corresponding language. This approach compares the token with dictionary values to check if it is correct or not. If token (word) is available in dictionary it is assumed to be correct otherwise it is incorrect. For creating dictionary there are various resources like Punjabi text books, newspapers, Punjabi websites etc. being used. The demerits of this approach are difficulties in keeping it up to date and to cover all the words in text. ***N-Gram*** technique is used to search misspelled words in given input text. Instead of matching each input word with text in dictionary, just n-grams are verified. An n-gram is a set of consecutive characters taken from a string with a length of whatever n is set to [7]. N-dimensional matrix is used for checks where real n-gram frequencies are stored. The word is labeled as misspelled word if a non-existent n-gram is found otherwise not. This method is language independent and does not require the knowledge of the language for which it is used [7].

2.2.3 Spelling Errors Correction Techniques

In ***Rule Based Approach***set of rules are created for common misspelling and typing errors. Rule is applied on each input token and compared with it. With the help of this approach corresponding results can be produced. Each rule in this approach fixes one kind of error. ***Similarity Keys***In this approach a key is assigned to each dictionary word. This key compared with the key computed for the non-word. Those words only selected as suggestions for which the keys are most similar. These approaches take less time i.e. speed effectual as only the words with similar keys have to be handled [7].***Edit-Distance Approach***In this technique misspelled words compared to every word in the dictionary [8].Minimum set of operations required to transform a non-word to a dictionary word. The operations are insertion, deletion and substitutions. This approach is useful for correct keyboard errors.***N-Gram*** technique can be used with dictionary or without dictionary. Without dictionary N-gram technique used to search the position in misspelled word where the error occurs and use valid n-gram to change the misspelled word. This method is easy and does not need any dictionary. With a dictionary, N-gram is used to measure distance between words, but the words are ever analyzed against dictionary. For example: Verify how many n-gram the incorrect word and a dictionary word which are common, weighted by length of the words [7]. ***Probabilistic Techniques***are based on statistical features of the language. There are two methods of probabilistic techniques i.e. transition probabilities and confusion probabilities. ***Transition probabilities*****:** A letter is followed by another given letter. By collecting N-gram frequency statistic on a large corpus of text to estimate transition probabilities.***Confusion probabilities*****:** How a letter is substituted or mistaken for another letter. These probabilities are source dependent because different OCR devices are using different techniques and features to recognize characters. Each device will have a unique confusion probability distribution [9].***Neural Networks***are interesting and promising techniques for spelling correction currently because they are excelling in pattern recognition due to their learning ability from data. Various types of neural networks existing such as recurrent neural networks, LSTM, CNN, deep neural network for this purpose.

3. Methodology

In this phase the study includes data collection, model architecture and its parameters. To accomplish such task of text normalization and spelling error correction asequence-to-sequencedeep learning-basedencoder-decoder model using transformer with copy input method is proposed and applied. The model is trained onneural machine translation based prepared parallel dataset comprising of correct and incorrect Punjabi words The proposed model is gone through various text normalization stagesduring training and testing in which rule based approach is used to clean the noise from dataset in preprocessing phase and deep learning-based methods are applied for further tasks including tokenization, encoding,padding in text normalization and spelling corrections.

3.1 Dataset

In my research, I have train and test my model on two types of different dataset comprising of parallel dataset of correct -incorrect sentences and another parallel corpus of correct-incorrect words in Punjabi language.For preparing a corpus of correct unique Punjabi words and sentences is not simple task as there is not large repository of grammatically and spelling wise correct Punjabi languagetext available on any source on the internet to train our model. Thus corpora of correct unique Punjabi Sentences is prepared by collecting Punjabi text from various domains like essay[14], stories, books, articles in newspapers andother sources on internet including Kaggle dataset[13].After the collection phase,text is passed through code to split this collected Punjabi text in sentence wise to prepare a text file of correct Punjabi sentences and then this text file is divided in words and save in different file to prepare Punjabi words dataset. Then another code is produced to find unique words from this file and make a new file of correct unique Punjabi words. After these both created text documents are also manually checked to ensure its correctness.A dictionary of approximately 60,000no. of words is prepared which will act as a parallel corpus of correct Punjabi words in our proposed model during training.Our statistical tool generate the output by calculating distance between input word and words stored in this Punjabi dictionary.In order to prepare a parallel corpora of Incorrect Punjabi sentences and words some artificial noise is added using rule -based methods include replacing a character with another character, delete a character from word, substitution of extra character in word etc. in dataset of correct Punjabi words in order to make an incorrect word corresponding to its correct form in another dataset.

Table 2: Detail of Dataset

Collected Dataset	Total No. of Punjabi Sentences Collected	6 Lakhs in sentences
Training Data Set	Total No. of Correct Punjabi Words Extracted	60000 in words
	Total no. of Incorrect Punjabi Words prepared after injecting artificial noise in correct dataset	60000 in words
Punjabi Dictionary	Total no. of correct words	60000 words

3.2Proposed Model Architecture

In this research author proposed a hybrid spell correction model using user interface. In first phase, the input text which will be given by user, it will first go through cleaning phase in which all noisy unwanted data from input will be removed using rule-based method, then it will pass to transformer for tokenization, embedding, positional encoding and further steps. Proposed transformer based deep learning model was developed for spelling corrections of misspelled Punjabi words by fine tuning of Vaswani model [24] by modifying encoder and decoder layers of transformer and added a special layer at end of transformer to copy input tokens in decoder's last output to enhance the probability of correct word output prediction. If user is not satisfied with output which was produced by deep learning model, then user can pass output generate by deep learning model to statistical based tool for post processing spell correction. We developed this tool using Levenshtein distance algorithm to correct wrong Punjabi words entered by user in interface if it was not fully corrected by deep learning-based tool. This statistical tool could be also used as pre-processing of input text to enhance the performance of our deep learning spell correction model. The basic purpose to add this tool along with deep learning-based spell correction tool in single interface is to satisfy user to get maximum correct results in case if user is not getting correct word of any specific misspelled word using deep learning model, he/she can use

this statistical based tool to get correct result. Its user choice if he/she wants to use it as pre or post processing when as required. But in our experiment rarely 2 or 3% we need to use our second statistical tool.

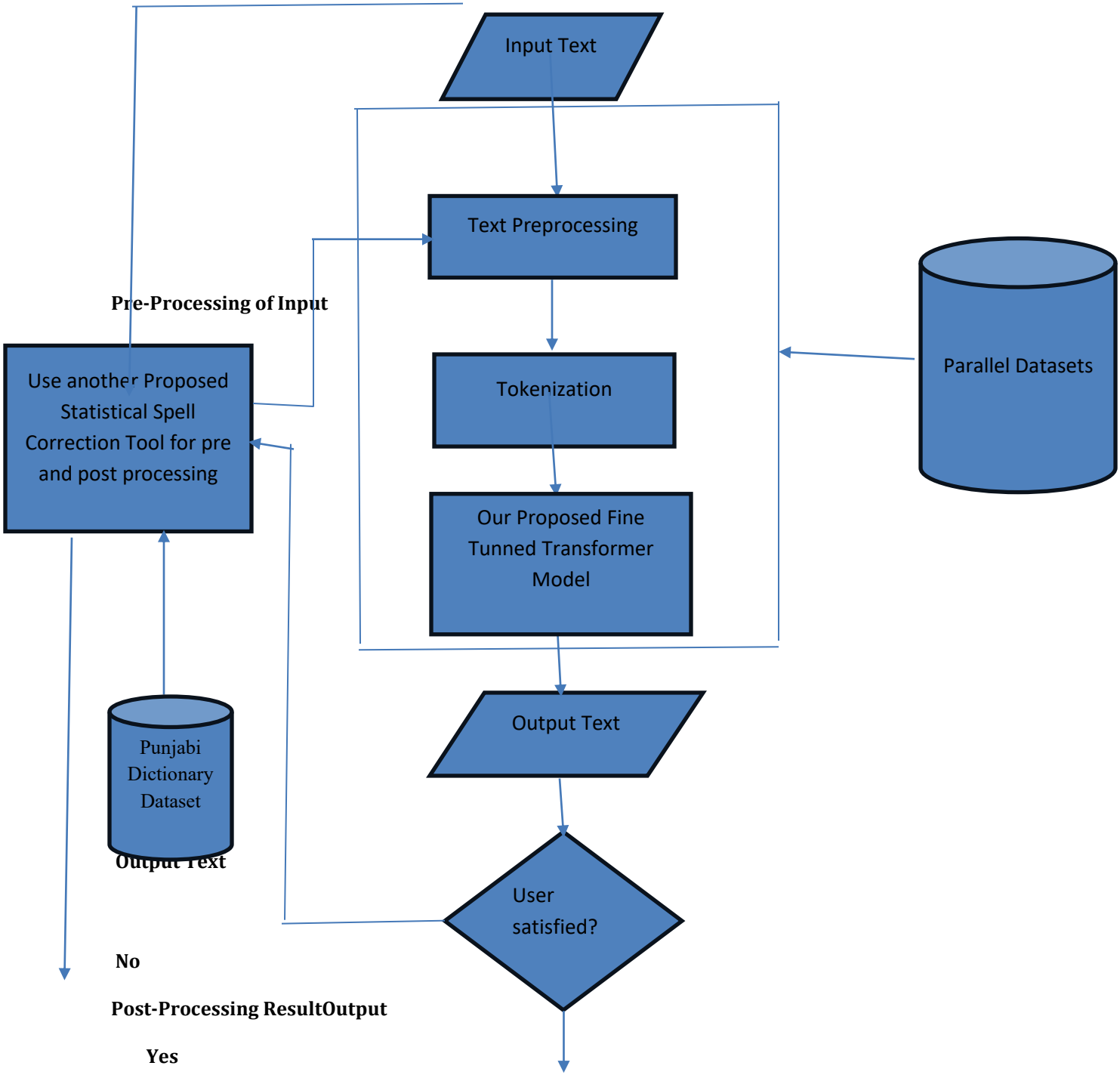


Figure 2: Proposed Architecture of Spell Correction Model

3.3 Model Parameters

We had implanted our proposed transformer based deep learning spell correction model in Google Colab Python3 using V28 TPUat backend. Experiment was done by training the proposed model on parallel dataset which consist of 60,000words in both files. One is correct words file and the other one stored corresponding misspelled words of correct Punjabi words which was created by adding artificial noise in correct words file using rule-based approach. Both text files read through Panda in data frames to train the model. To maintain the length of every word to maximum word length, zero post padding technique is used. Proposed model was trained for 10 epochs and then saved in google drive so that model could be loaded again and tested. We tested our model on different test cases and the results are given in table:

Table 3: Hyper parameters

Hyper parameters	**Values**
Batch Size	32
Epochs	10
Learning rate	0.1
Optimizer	Adam
Activation Function	Softmax
Dropout	0.1

4. Experiments

This section is introducing our text normalization framework which includes text preprocessing and spelling correction processes and results of our implementation work.

4.1 Cleaning of Punjabi Text Dataset

Under the preprocessing phase the cleaning of collected data has been done which includes removal of noise from raw data so that computers can easily detect the patterns in data. Text data like Wikipedia contain noise like URL tags, bullets, numbers, alphabets, hash tags, punctuation, email addresses, devanagari numbers and special characters etc. All of which are very challenging for computers to understand if they are existing in the data.Processingof data is required to remove all these non-textual elements and unwanted elements. The cleaning of data has been done using Python as the Programming language.

For example:

Text = re. sub ("[-+@#^/|*() {} $~<>=_%:]","“, text)

4.2 Tokenization: Tokenization is the process of splitting a text (sentences and words) to its possible smallest morpheme known as token. Morpheme is the smallest possible unit after which it cannot be broken further. A token can be a word, character, number, symbol, or n-gram. The tokenizer receives the text after

the Pre-processing phase and generates tokens. The tokenizer takes the text generated by the pre-processing phase as input. It is a method of separating the text into smaller units called tokens so that computers can easily understand it. Tokenization can be done on character, word and n-gram characters based. For implementing our grammar correction model, tokenization has been done on the basis of words by splitting the sentence text in words and returning the token. For tokenizing the text in grammar checking algorithm text_to_sequence class of Keras has been used.

For example1:[“ਮੈਨੂੰਇੱਕਦਿਨਦੀਛੱਟੀਦਿੱਤੀਜਾਵੇ। **“]**

Tokens: [” ਮੈਨੂੰ“, “ਇੱਕ “, “ਦਿਨ “,” ਦੀ “,”ਛੱਟੀ “,”ਦਿੱਤੀ”,”ਜਾਵੇ”,”। **“]**

For example2:[“ਜਾਦੂਗਰਨੇਜਾਦੂਵਿਖਾਇਆ “]

Tokens: [“ਜਾਦੂਗਰ “,”ਨੇ “,”ਜਾਦੂ “,”ਵਿਖਾਇਆ “]

4.3 Encoding Technique

For implementing our framework, Keras Tokenizer class has been used for encoding. There are various Keras Tokenizer classes available like Fit_on_texts, texts_to_sequences, Texts_to_matrix, Sequences_to_matrix. For tokenizing the Punjabi sentences, text_to_sequence class of Keras has been used. This technique is used to assign integer value to every word in text. Every word will be identified as index

Results:
{'bos': 1, 'eos': 2, 'ਦੇ': 3, 'ਦੀ': 4, 'ਨੂੰ': 5, 'ਦਾ': 6, 'ਵਿੱਚ': 7, 'ਹੈ': 8, 'ਵੀ': 9, 'ਕੇ': 10, 'ਨੇ': 11, 'ਤੋਂ': 12, 'ਹੈ': 13, 'ਹੀ': 14, 'ਨਾਲ': 15, 'ਉਸ': 16, 'ਇਸ': 17, 'ਤੇ': 18, 'ਅਤੇ': 19, 'ਹਨ': 20, 'ਵਿਚ': 21, 'ਤਾਂ': 22, 'ਉਹ': 23, 'ਕਿ': 24, 'ਸੀ': 25, 'ਲਈ': 26, 'ਨਹੀਂ': 27, 'ਦੀਆਂ': 28, 'ਇਹ': 29, 'ਪੰਜਾਬੀ': 30, 'ਹੋ': 31, '।': 32, ''ਤੇ': 33, 'ਹਨ': 34, 'ਆਪਣੇ': 35, 'ਮੈਂ': 36, 'ਉਨ੍ਹਾਂ': 37, 'ਸੀ': 38, 'ਇਕ': 39, 'ਜੀ': 40, 'ਕਰਨ': 41, 'ਨਾ': 42, 'ਕੋਈ': 43, 'ਇੱਕ': 44, 'ਜਾ': 45, 'ਪਰ': 46, 'ਕਰ': 47, 'ਕੰਮ': 48, 'ਸਨ।': 49, 'ਦਿਨ': 50, 'ਹੋਰ': 51, 'ਸਮੇਂ': 52, 'ਪੰਜਾਬ': 53, 'ਕਿਸੇ': 54, 'ਆ': 55, 'ਮੈਨੂੰ': 56, 'ਬਹੁਤ': 57, 'ਸਕੂਲ': 58, 'ਵਾਲੇ': 59, 'ਆਪਣੀ': 60, 'ਕੁਝ': 61, 'ਸਿੰਘ': 62, 'ਰਹੀ': 63, 'ਰਿਹਾ': 64, 'ਘਰ': 65, 'ਸਾਡੇ': 66, 'ਮੇਰੇ': 67, 'ਸਰਕਾਰ': 68, 'ਜਾਂਦਾ': 69, 'ਕਈ': 70, 'ਗਿਆ': 71, 'ਜੋ': 72, 'ਗਈ': 73, '’’': 74, 'ਜਿਸ': 75, 'ਹੁਣ': 76, 'ਅੱਜ': 77, 'ਹਰ': 78, 'ਕਰਕੇ': 79, 'ਕੀਤਾ': 80, ''ਚ': 81, 'ਸਾਰੇ': 82, 'ਜਾਂਦੀ': 83, 'ਜਦੋਂ': 84, 'ਬਾਅਦ': 85, 'ਰਹੇ': 86, 'ਵਰਮਾ': 87, 'ਆਪ': 88, 'ਲੋਕ': 89, 'ਈ': 90, 'ਲੈ': 91, 'ਲੋਕਾਂ': 92, 'ਭਾਸ਼ਾ': 93, 'ਅਸੀਂ': 94, 'ਹੋਏ': 95, 'ਕਰਦੇ': 96, 'ਸਨ': 97, 'ਸਾਨੂੰ': 98, 'ਕੋਲ': 99, 'ਵੱਲ': 100, 'ਹੋਣ': 101, 'ਬੋਲੀ': 102, 'ਹੁੰਦਾ': 103, 'ਗਏ': 104, 'ਦੇਸ਼': 105, 'ਕੀਤੀ': 106, 'ਜਿਵੇਂ': 107, 'ਗੱਲ': 108, 'ਜਾਂ': 109, "'ਚ": 110, 'ਗਿਆ': 111, 'ਕਾਰਨ': 112, 'ਗਈ।': 113, 'ਕੀ': 114, 'ਹੋਈ': 115, 'ਕਿਹਾ': 116, 'ਮਾਤਭਾਸ਼ਾ': 117, 'ਸਭ': 118, 'ਦੋ': 119, 'ਬੱਚਿਆਂ': 120, 'ਜਾਣ': 121, 'ਤੱਕ': 122, 'ਆਈ': 123, 'ਵਾਰ': 124, 'ਭਾਰਤ': 125, 'ਕਾਂ': 126, 'ਉੱਤੇ': 127, 'ਕਰਨਾ etc.}

4.4 Padding

In Natural Language Processing (NLP), text preprocessing which include: Tokenization, Sequencing, padding text data can be considered as sequence of character, sequence of words or sequence of sentences. For spell checker, text data has been considered in sequence of characters. As deep learning models do not understand text, we need to convert text into numerical representation. For this purpose, the first step is Tokenization. The *Tokenizer* splits sentences into words and words into characters and encodes these characters into integers. *For example, t*here are 4 words including 1 with a maximum length of 5. Then convert all characters to integer index and text represented as sequence.

Text= [“ਕਮਲ “,” ਐਬ “,” ਆਇਆ “]

Texts are represented by sequences as for example,

“ਕਮਲ “= [9, 32, 35]

” ਐਬ “= [6, 30]

” ਆਇਆ “= [1, 2, 1]

“ਗਈ” = [11, 3]

Types of Padding

In any text data, naturally there may be different lengths of words. Neural networks in deep learning require having inputs with the same size. For this purpose, padding needs to be done. Padding can be based on pre and post. Pre padding means to add 0 before each sequence and post will pad after each sequence. Padding is done according to the maximum length of the longest word in a sentence.

Pre Padding

Sequence= [[2,6,3,7,8], [1,3,9,2], [6,30, [11,35]]

Padded_seq = [[2 6 3 7 8]

[0 1 3 9 2]

[0 0 0 6 30] ----------0 Padded at the beginning

[0 0 0 11 35]]

Post Padding

Sequence= [[2,6,3,7,8], [1,3,9,2], [6,30, [11,35]]

Padded seq = [[2 6 3 7 8]

[1 3 9 2 0]

[6 30 0 0 0] ----------0 Padded at the End

[11 35 0 0 0]]

Post padding technique has been used in grammar correction algorithms after encoding.

Encoding and Padding Results of followings sentences in our dataset:

Encoding:

Sequence 1

Input: ਮੈਨੂੰਇੱਕਦਿਨਦੀਛੁੱਟੀਦਿੱਤੀਜਾਵੇ।
Output: [56, 44, 50, 4, 338,363,981]
Sequence 2
Input: ਆਪਜੀਦਾਧੰਨਵਾਦੀਹੋਵਾਂਗਾ
Output: [88, 40, 6,780,467]
Before Padding Result

Input: [56, 44, 50, 4, 338,363,981]
Input: [88, 40, 6,780,467]

After Padding Result
Output: [56 44 50 4 338 363 981]
Output: [88 40 6 780 467 0 0]

Table 3: Tokenization and Padding in Neural Network

Table 4: Text Normalization

Text	[” ਮੈਨੂੰਇੱਕਦਿਨਦੀਛੁੱਟੀਦਿੱਤੀਜਾਵੇ”] [“ਆਪਜੀਦਾਧੰਨਵਾਦੀਹੋਵਾਂਗਾ”]
Tokens	[” ਮੈਨੂੰ “, “ਇੱਕ “, “ਦਿਨ “,” ਦੀ “,”ਛੁੱਟੀ “,”ਦਿੱਤੀ”,”ਜਾਵੇ”,”] [“ਆਪ“,”ਜੀ”,”ਦਾ“,”ਧੰਨਵਾਦੀ”,”ਹੋਵਾਂਗਾ”]
Word Index	[” ਮੈਨੂੰ “:56, “ਇੱਕ “:44, “ਦਿਨ “:50,” ਦੀ “:4,”ਛੁੱਟੀ “:338,”ਦਿੱਤੀ”:363,”ਜਾਵੇ”:981] [“ਆਪ“:88,”ਜੀ”:40,”ਦਾ“:6,”ਧੰਨਵਾਦੀ”:780,”ਹੋਵਾਂਗਾ”:467]
Encoded Sequences	[56, 44, 50, 4, 338,363,981] [88, 40, 6,780,467]
Padded Sequences	[56, 44, 50, 4, 338,363,981] [88, 40, 6,780, 467, 0, 0]

	Noisy Text
Text before Normalization	ਪੰਜਾਬਦੇਲੋਕ-ਨਾਚ ਲੋਕ-ਨਾਚਲੋਕ-ਸਮੂਹ@@ ਦੀਸਿਰਜਨ-ਕਲਾਦੀਇੱਕ .....ਮਹੱਤਵਪੂਰਨਵੰਨਗੀਹੈ, ਜੋਲੋਕ-ਮਾਨਸਦੇਸਮੁਚੇਹਾਵਾਂ- ਭਾਵਾਂਨੂੰਸਰੀਰਕਮੁਦਰਾਵਾਂਦੇਪ੍ਰਗਟਾਉ-ਸੰਦਰਭਰਾਹੀਂਪੇਸ਼ਕਰਦੀਹੈ|% ##ਲੋਕਨਾਚਵੀਇਸੇਪ੍ਰਕਾਰਦੀਇੱਕਲੋਕ-ਕਲਾਹੈ|% ਲੋਕ-ਨਾਚ ***ਮੰਨੋਰੰਜਨਦਾਸਾਧਨਹੀਨਹੀਂ...., ਇਹਕਿਸੇਖਿੱਤੇਦੇਜਨ-ਸਮੂਹਦੀਸਮਾਜਿਕ, ਸੱਭਿਆਚਾਰਿਕ, ਨੈਤਿਕ, &&ਧਾਰਮਿਕ, ਰਾਜਸੀਅਤੇਇਤਹਾਸਿਕਜੀਵਨ-ਤੋਰ&^ ਦੀਆਂਵਿਭਿੰਨਪਰਤਾਂਦਾਸਰੀਰਕਮੁਦਰਾਵਾਂ&&ਦੇਮਾਧਿਅਮਰਾਹੀਂ** ਆਪ-ਮੁਹਾਰਾ, ਸਧਾਰਨ, ਖੁਸ਼ੀਆਂ- ਖੇੜਿਆਂਭਰਪੂਰਅੰਦਰੂਨੀਭਾਵਨਾਵਾਂਦਾਬਾਹਰੀਪ੍ਰਗਟਾਵਾਵੀਹੈ| ##ਪੰਜਾਬਦੇਲੋਕ-ਨਾਚਪੰਜਾਬੀਆਂਦੇਜਨ- ਜੀਵਨਦਾਮਹੱਤਪੂਰਨਅੰਗਹਨ।## ਪੰਜਾਬਦੇਲੋਕ-ਨਾਚਾਂ: ਦਾਵਰਗੀਕਰਨਦੋਪੱਧਰਾਂਤੇਕੀਤਾਜਾਸਕਦਾਹੈ$$ ਇਸਤਰੀਆਂਦੇਲੋਕ-ਨਾਚ.... ਮਰਦਾਂਦੇਲੋਕ-ਨਾਚ.....
	Cleaned and Normalized Text
Text after Normalization	ਪੰਜਾਬਦੇਲੋਕ-ਨਾਚ ਲੋਕ-ਨਾਚਲੋਕ-ਸਮੂਹਦੀਸਿਰਜਨ-ਕਲਾਦੀਇੱਕਮਹੱਤਵਪੂਰਨਵੰਨਗੀਹੈ, ਜੋਲੋਕ-ਮਾਨਸਦੇਸਮੁਚੇਹਾਵਾਂ-ਭਾਵਾਂਨੂੰਸਰੀਰਕਮੁਦਰਾਵਾਂਦੇਪ੍ਰਗਟਾਉ- ਸੰਦਰਭਰਾਹੀਂਪੇਸ਼ਕਰਦੀਹੈ| ਲੋਕਨਾਚਵੀਇਸੇਪ੍ਰਕਾਰਦੀਇੱਕਲੋਕ-ਕਲਾਹੈ| ਲੋਕ-ਨਾਚਮੰਨੋਰੰਜਨਦਾਸਾਧਨਹੀਨਹੀਂ , ਇਹਕਿਸੇਖਿੱਤੇਦੇਜਨ- ਸਮੂਹਦੀਸਮਾਜਿਕ, ਸੱਭਿਆਚਾਰਿਕ, ਨੈਤਿਕ, ਧਾਰਮਿਕ, ਰਾਜਸੀਅਤੇਇਤਹਾਸਿਕਜੀਵਨ- ਤੋਰਦੀਆਂਵਿਭਿੰਨਪਰਤਾਂਦਾਸਰੀਰਕਮੁਦਰਾਵਾਂਦੇਮਾਧਿਅਮਰਾਹੀਂਆਪ- ਮੁਹਾਰਾ, ਸਧਾਰਨ, ਖੁਸ਼ੀਆਂ- ਖੇੜਿਆਂਭਰਪੂਰਅੰਦਰੂਨੀਭਾਵਨਾਵਾਂਦਾਬਾਹਰੀਪ੍ਰਗਟਾਵਾਵੀਹੈ| ਪੰਜਾਬਦੇਲੋਕ-ਨਾਚਪੰਜਾਬੀਆਂਦੇਜਨ- ਜੀਵਨਦਾਮਹੱਤਪੂਰਨਅੰਗਹਨ।ਪੰਜਾਬਦੇਲੋਕ- ਨਾਚਾਂਦਾਵਰਗੀਕਰਨਦੋਪੱਧਰਾਂਤੇਕੀਤਾਜਾਸਕਦਾਹੈਇਸਤਰੀਆਂਦੇਲੋਕ- ਨਾਚ ਮਰਦਾਂਦੇਲੋਕ-ਨਾਚ

5. Evaluation of Experimental Results

We had tested some 50 misspelled Punjabi words on existing online rule-basedmodel [23], ChatGPT [21], Grok3 [22] and our proposed architecture model manually and its output also saved manually.The results of tested models are given in following table 5.

Table5: Comparison of Results of Our Proposed Spelling Correction Model with other models on sameTesting Data set

Sr. No.	Input Word	Rule Based Model Sodhak [23]	Our Proposed Model	ChatGPT Model [21]	Grok [22]
1	ਸਾਹੀਤ	ਸਾਹਿਤ	ਸਾਹਿਤ	|ਸਾਹਿਤ	ਸਾਹਿਤ
2	ਆਗਸਤ	ਅਗਸਤ	ਅਗਸਤ	ਅਗਸਤ	ਅਗਸਤ
3	ਚਗੀ	ਚੰਗੀ	ਚੰਗੀ	ਚੰਗੀ	ਚੰਗੀ
4	ਬਚਿਆਂ	Noprediction	ਬੱਚਿਆਂ	ਬੱਚਿਆਂ	ਬੱਚਿਆਂ
5	ਧਾਮਰੀਕ	ਧਾਰਮਿਕ	ਧਾਰਮਿਕ	ਧਾਰਮਿਕ	ਧਾਰਮਿਕ
6	ਜੂਲਾਈ	ਜੁਲਾਈ	ਜੁਲਾਈ	ਜੁਲਾਈ	ਜੁਲਾਈ
7	ਕੌਸੀਸ	ਕਸੀਸ	ਕੋਸ਼ਿਸ਼	ਕੋਸ਼ਿਸ਼	ਕੋਸ਼ਿਸ਼
8	ਆਮ੍ਰੀਤਸਰ	No Suggestions Available	ਅੰਮ੍ਰਿਤਸਰ	ਅੰਮ੍ਰਿਤਸਰ	ਅੰਮ੍ਰਿਤਸਰ
9	ਮੂਢਲੀ	ਮੁੱਢਲੀ	ਮੁੱਢਲੀ	ਮੂਲਦੀ	ਮੁੱਢਲੀ
10	ਗਰਿਬ	ਗ਼ਰੀਬ	ਗਰੀਬ	ਗਰੀਬ	ਗਰੀਬ
11	ਇਗਲੇਂਡ	ਇੰਗਲੈਂਡ	ਇੰਗਲੈਂਡ	ਇੰਗਲੈਂਡ	ਇੰਗਲੈਂਡ

12	ਯੂਨੀਵਰਸੀਟੀ	ਯੂਨੀਵਰਸਿਟੀ	ਯੂਨੀਵਰਸਿਟੀ	ਯੂਨੀਵਰਸਿਟੀ	ਯੂਨੀਵਰਸਿਟੀ
13	ਇਟਰਵੀਊ	ਇੰਟਰਵਿਊ	ਇੰਟਰਵਿਊ	ਇੰਟਰਵਿਊ	ਇੰਟਰਵਿਊ
14	ਭਾਸਾ	ਭਾਸ਼ਾ	ਭਾਸ਼ਾ	ਭਾਸ਼ਾ	ਭਾਸ਼ਾ
15	ਦੇਸ	No Prediction	ਦੇਸ਼	ਦੇਸ਼	ਦੇਸ਼
16	ਪਾਨੀ	ਪਾਣੀ	ਪਾਣੀ	ਪਾਣੀ	ਪਾਣੀ
17	ਗੁਰੁ	ਗੁਰੂ	ਗੁਰੂ	ਗੁਰੂ	ਗੁਰੂ
18	ਸਕੁਲ	ਸਕੂਲ	ਸਕੂਲ	ਸਕੂਲ	ਸਕੂਲ
19	ਰੋਟੀ	ਰੋਟੀ	ਰੋਟੀ	ਰੋਟੀ	ਰੋਟੀ
20	ਗਡੀ	ਗੱਡੀ	ਗੱਡੀ	ਗੱਡੀ	ਗੱਡੀ
21	ਖੁਸੀ	ਖ਼ੁਸ਼ੀ	ਖੁਸ਼ੀ	ਖੁਸ਼ੀ	ਖੁਸ਼ੀ
22	ਰੁਖ	ਰੁੱਖ	ਰੁੱਖ	ਰੁੱਖ	ਰੁੱਖ
23	ਅਗ	ਅੱਗ	ਅੱਗ	ਅੱਗ	ਅੱਗ
24	ਕਵਿ	ਕਵੀ	ਕਵੀ	ਕਵੀ	ਕਵੀ
25	ਨਦਿ	ਨਦੀ	ਨਦੀ	ਨਦੀ	ਨਦੀ
26	ਆਪਨਾ	ਆਪਣਾ	ਆਪਣਾ	ਆਪਣਾ	ਆਪਣਾ
27	ਚੋਲ	ਚੌਲ	ਚੌਲ	ਚੌਲ	ਚੋਲ਼
28	ਦੁਸ਼ਮਨ	ਦੁਸ਼ਮਣ	ਦੁਸ਼ਮਣ	ਦੁਸ਼ਮਣ	ਦੁਸ਼ਮਣ
29	ਪੋੜੀ	ਪੌੜੀ	ਪੌੜੀ	ਪੌੜੀ	ਪੌੜੀ
30	ਕਨਕ	ਕਣਕ	ਕਣਕ	ਕਣਕ	ਕਣਕ
31	ਕੁੱਭਾ	ਕੁੱਤਾ	ਕੁੱਤਾ	ਕੁੱਛ,ਕੁੱਬਾ	ਕੁਝ
32	ਬੁਕ	ਬੁੱਕ	ਬਕ	ਕਿਤਾਬ ,ਬੁੱਕ	ਬੁੱਕ

33	ਵੇਖਨਾ	ਵੇਖਣਾ	ਵੇਖਣਾ	ਵੇਖਣਾ	ਵੇਖਨਾ
34	ਸਮਜਦਾਰ	ਸਮਝਦਾਰ	ਸਮਝਦਾਰ	ਸਮਝਦਾਰ	ਸਮਝਦਾਰ
35	ਗਯਾਨ	ਗਾਣ	ਗਯਾਨ	ਗਿਆਨ	ਗਿਆਨ
36	ਦਹੀ	ਦਹੀਂ	ਦਹੀ	ਦਹੀਂ	ਦਹੀਂ
37	ਸੰਘਨਾ	ਸੰਘਣਾ	ਮਹਾਰਤ,ਸੰਘਾ	ਸੂੰਘਣਾ	ਸੰਗਠਨ
38	ਸਕੁਲ	ਸਕੂਲ	ਸਕੂਲ	ਸਕੂਲ	ਸਕੂਲ
39	ਬਜਾਰ	ਬਾਜ਼ਾਰ	ਬਜ਼ਾਰ	ਬਜ਼ਾਰ	ਬਾਜ਼ਾਰ
40	ਰਾਜਨੀਤਿ	ਰਾਜਨੀਤੀ	ਰਾਜਨੀਤੀ	ਰਾਜਨੀਤੀ	ਰਾਜਨੀਤੀ
41	ਪਰਸਿੱਧਿ	ਪਰਸਿੱਧੀ	ਪਰਸਿੱਧ	ਪ੍ਰਸਿੱਧੀ	ਪ੍ਰਸਿੱਧੀ
42	ਸੱਬਿਆਚਾਰ	ਸਭਿਆਚਾਰ	ਸੱਭਿਆਚਾਰ	ਸਭਿਆਚਾਰ	ਸਭਿਆਚਾਰ
43	ਮਨੋਰਜਨ	ਮਨੋਰੰਜਨ	ਮਨੋਰੰਜਨ	ਮਨੋਰੰਜਨ	ਮਨੋਰੰਜਨ
44	ਅਨੇਖਾ	ਅਨੇਖਾ	ਅਨੇਖਾ	ਅਨੇਖਾ	ਅਨੇਖਾ
45	ਸਿਪਾਈ	ਸਿਪਾਈ	ਸਿਲਾਈ	ਸਿਪਾਹੀ	ਸਿਪਾਹੀ
46	ਗੋਬੀ	ਗੋਭੀ	ਗੋਦੀ	ਗੋਭੀ	ਗੋਭੀ
47	ਤੁਰਣਾ	ਤੁਰਨਾ	ਤੁਰਨਾ	ਤੁਰਣਾ	ਤੁਰਨਾ
48	ਲਾਬ	ਲਾਬ	ਲਾਭ	ਲਾਭ	ਲਾਭ
49	ਇਕ	ਇੱਕ	ਇੱਕ	ਇੱਕ	ਇੱਕ
50	ਚਿਠੀ	ਚਿੱਠੀ	ਚਿੱਠੀ	ਚਿੱਠੀ	ਚਿੱਠੀ

6. Test Results and Discussion

After training, we validate the performance of our proposed model by testing on same real test dataset which we had tested on other existing models. It revealed from tables 5 and 6

that our model has shown more accuracy as compared to rule-based model and close to achieve good performance as ChatGPT and Grok3 models.

Table 6: Analysis of Our Proposed model with other existing models based on tested words

Models	**Total No. of Spelling Error Words Tested**	**Total No. of Words Corrected CFE**	**Total No. of words not Corrected FWE**	**Accuracy (Precision)**
Rule Based Punjabi Grammar Checker [23]	**50**	**41**	**09**	**0.82**
ChatGPT [21]	**50**	**45**	**05**	**0.90**
Grok 3 [22]	**50**	**47**	**03**	**0.94**
Our Proposed Model	**50**	**45**	**05**	**0.90**

7. Conclusion and Future scope

In this research paper we had developed deep learning based spell correction tool by fine tunned base transformer model proposed by Vaswani [24]. The developed model was tested on different 50 incorrect Punjabi words which has shown precision of 0.90 which is near as tested other AI models. We developed user interface with hybrid approach using our two different developed tools Deep Spell Checker and Statistical spell checker.Statistical tool is only required when user is not getting right output from deep spell checker or in case of post processing to fully correct the spelling of output which was generated by deep spell checker. User can also use this tool as pre-processing by first pass input to this tool and generated output will pass to deep learning model to get outstanding results. In our testing only 2-3% we required to use our second tool. Results analysis which had shown in table 6 reveal that our proposed model is performing nicely with small training dataset. In future this proposed and developed fine tunned transformer based deep learning model can be extended by training on large dataset and also to cover all types of errors while preparing incorrected dataset. Suchdeveloped model could also be fine tunned with different hyper parameters so that it can surpass accuracy of others popular AI models like ChatGPT, GROK3 etc.

References

[1] Ariffin, S.N.A.N.,Tiun, S.: Rule-based text normalization for Malay social media texts.International Journal of Advanced Computer Science and Applications. (2020)

[2] Khan,O.A.,Karim, A.:A Rule-Based Model for Normalization of SMS Text.2012 IEEE 24th International Conference on Tools with Artificial Intelligence. (2012)https://doi.org/10.1109/ICTAI.2012.91

[3] Maghfur,N.M.,Ibrohim,M.O.:Text Normalization for Indonesian Text-to-Speech (TTS) using Rule-Based Approach: A Dataset and Preliminary Study.2021 4th International Conference of Computer and Informatics Engineering (IC2IE).129-134(2021)

[4] Dang, H.-T., Phan, X.-H.: Non-Standard Vietnamese Word Detection and Normalization for Text–to–Speechin 2022 14th International Conference on Knowledge and Systems Engineering (KSE), IEEE. (2022)

[5] Schlippe, T., Zhu, C., Lemcke, D.,Schultz, T.:Statistical machine translation based text normalization with crowdsourcing", 2013 IEEE International Conference on Acoustics, Speech and Signal Processing.(2013)https://ieeexplore.ieee.org/document/6639305

[6] Mandal, P.,Hossain, B.M.M.:Clustering-based Bangla spell checker", IEEE International Conference on Imaging, Vision & Pattern Recognition (ICIVPR).(2017)

[7] Sharma, M., Nagineni, S.K.: Spell detection and correction in independent system. International Journal of Advances in Electronics and Computer Sciences ISSN: 2393-2835, Vol.5, Issue 6. (2018)

[8] Kaur, A., Singh, P.,Rani,S.: Spell Checking and error correcting system for text paragraphs written in Punjabi language using hybrid approach. International Journal of Advanced Research in Science, Engineering and Technology.vol.2, No.11, 1031-1038. (2015)

[9]Mishra, M.,Kaur, N.: Survey of Spelling Error Detection and Correction Techniques. In International Journal of Computer Trends and Technology, Vol.4, Issue.3. (2013)

[10] Aw, A., Zhang, M., Xiao, J., Su,J.:A Phrase-Based Statistical Model for SMS Text Normalization .Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions. 33–40, Sydney, July 2006ⓒ2006 Association for Computational Linguistic.(2006) https://aclanthology.org/P06-2005/

[11] Satapathy, R.,Guerreiro, C., Chaturvedi,I.,Cambria, E.: Phonetic-based microtext normalization for twitter sentiment analysis.In 2017 IEEE international conference on data mining workshops (ICDMW). (2017)
DOI 10.1109/ICDMW.2017.59

[12] Tiwari, A.S.,&Naskar, S.K.:Normalization of social media Text using Deep Neural Networks.In proceedings of 14th International Conference on Natural Language Processing,312-321.(2017)

[13]https://www.kaggle.com/dataset

[14]https://www.punjabigrammar.com/p/punjabi-grammar.html

[15]https://machinelearningmastery.com/data-preparation-variable-length-input-sequences-sequence-prediction

[16]https://www.analyticsvidhya.com/blog/2021/03/tokenization-and-text-normalization/

[17]https://www.analyticsvidhya.com/blog/2020/08/types-of-categorical-data-encoding/

[18] Lehal, G.S, Bhagat, M.: Spelling Error Pattern Analysis of Punjabi Typed Text", In Proceeding of the 2007 International Symposium on Machine Translation, NLP and TSS, 128-141.(2007)

[19] Kaur, G., Kaur, K., Singh,P.: Spell Checker for Punjabi Language using deep Neural Network", IEEE 5th International Conference on Advanced Computing & Communication System. (2019)

[20] Bijoy, M. H., Hossain, N., Islam, S.,Shatabda, S.: A transformer-based spelling errorcorrection framework for Bangla and resource scarce Indic languages.Computer Speech & Language,Volume 89,2025,101703,ISSN 0885-2308. (2025)

https://doi.org/10.1016/j.csl.2024.101703.

[21]https://chatgpt.com/

[22] https://grok.com/?referrer=website

[23] Sodhak::Gurmukhi Unicode Typing Pad, Spell Checker and Font Converter

[24]Vaswani,A.,Shazeer,N.,Parmar,N.,Uszkoreit,J.,Jones, L.,Gomez,A.N.,Kaiser, L.,Polosukhin, I.:Attention Is All You Need", In proceedings of 31st Conference on Neural Information Processing Systems, Long Beach, CA, USA, 6000-6010. (2017)