Hybrid Context Aware Gujarati Spell Correction Using Norvig Algorithm, GRU, and IndicBERT

Brijeshkumar Y. Panchal[1,2,*], Apurva Shah[1]
[1]Computer Science and Engineering Department, Faculty of Technology and Engineering, The Maharaja Sayajirao University of Baroda, Vadodara, Gujarat, India
[2]Computer Engineering Department, Sardar Vallabhbhai Patel Institute of Technology (SVIT)-Vasad, Gujarat Technological University (GTU), Anand, Gujarat, India
E-mail: panchalbrijesh02@gmail.com
[*]Corresponding author

Keywords: Natural Language Processing (NLP), Gujarati, Grammar Checker, Spelling Correction/Checker, IndicBERT, Peter Norvig, GRU

Received: June 23, 2025

Numerous applications in the domain of Natural Language Processing (NLP) rely on spelling and grammatical checks, including email, opinion mining, text summarization, chatbots, and countless more. An individual's credibility, cybersecurity efforts, legal ambiguities, and NLP application performance can all take a hit if they make a mistake when dealing with regional languages such as Assamese, Gujarati, Hindi, etc. In order to lessen the frequency of spelling errors, this article examines and concentrates on Gujarati. In addition to a thorough examination of issues related to the Gujarati language, this article provides up-to-date strategies for fixing spelling mistakes based on context of the word. A novel hybrid approach ensures top-notch Gujarati context aware spelling verification. After thoroughly considering all the suggestions, we used a two-layer GRU network and the IndicBERTv2-SS model, which was fine-tuned only on our curated Gujarati dataset of about 20,000 sentences (70/15/15 split into training, validation, and test), to choose the best correction while keeping the context in mind. Normalization for Gujarati (diacritics, compound characters, and numbers), regex-based tokenization, and edit-distance candidate creation were all part of preprocessing. We used accuracy, precision, and recall to assess the test split. Our proposed IndicBERT-GUJBRIJAPU tool got 93.49% accuracy, 94.46% precision, and 91.59% recall, which is much better than other approaches for context-aware correction.

Povzetek: Članek predstavi hibridno kontekstno preverjanje pravopisa za gudžaratščino, ki združuje Norvigov algoritem, dvoslojni GRU in izboljšani IndicBERTv2-SS na 20.000 stavkih.

1 Introduction

India boasts a wealth of literature in a variety of regional languages, including Gujarati, Hindi, Tamil, Assamese and many more. For speakers of other languages inside the nation, these languages nevertheless can remain unintelligible. In languages such as Gujarati and Hindi, the context of a statement is quite important in imparting its intended meaning. Natural Language Processing (NLP) discipline seeks to apply grammatical principles and linguistic structures to analyze and comprehend natural languages. By means of natural language in both speech and text, NLP research investigates how computers might understand, analyze, and modify it, hence bridging the distance between humans and technology [1]. The term "language" denotes the natural languages including Gujarati, Hindi, and English in NLP. Preprocessing, an essential part of natural language processing, involves examining the text for spelling errors in order to improve its quality by associating words with their accurate meanings [2].

Many NLP systems use grammar and spelling correction to remove textual data mistakes. These mistakes provide noise that influences syntactic and semantic understanding, therefore influencing the performance of NLP-based systems [3]. For spell-checking and grammar correction, Deep Learning is quite successful since it lets machines learn from past data. From SMS texting to social media (Facebook, Twitter, WhatsApp), text-based data is exploding exponentially and today digital government records and e-newspapers are expanding [4] [5]. Essential components of text processing, spell-checkers help to fix written language mistakes [4]. They point out two main kinds of mistakes: Words not found in the dictionary (e.g., "Gujarti" rather than "Gujarati")- Non-word errors. Real-world Errors: Dictionary words used wrongly in context (e.g., "Their going to the market" instead of "They’re going to the market"). Although spell-checkers for Latin and Western languages have been extensively developed, the great linguistic variety and complicated grammatical structures

of Indian regional languages mean that research on them is still in its early years.

Gujarati is kind similar to Hindi out of the Indo-Aryan branch of the Indo-European language family. Among the 22 officially recognized languages of India, it has over 55.5 million native speakers—4.5% of the nation's total population as per the 2011 census [1]. Though some NLP tools [6] [7] [8] [5] [9] [10] — stemmers, lemmatizers, tiny corpora—exist—the language currently lacks thorough tools for spelling and grammatical correction [3]. Gujarati is widely used, although it lacks basic NLP tools—especially in relation to spell and grammar checking [11]. Available for Gujarati now, the "Saras" spell checker detects spelling mistakes using Directed Acyclic Word Graphs (DAWG). It does not, however, consider prefixes, suffixes, or inflections, therefore creating fresh research prospects for sophisticated spell-checking methods. Gujarati grammar, adhering to rigorous guidelines [12], encompasses Jodani (જોડણી) for correct spelling, Sandhi (સંધિ) for word joining, and Samas (સમાસ) for compound word formation.

The aim of this study is to solve context aware spelling mistakes in Gujarati language. To solve the shortcomings of current spell-checkers and raise Gujarati NLP application accuracy, novel and hybrid approaches are developed and implemented. In this article, the challenges related to context aware spelling checking with Gujarati language is focused and reviewed which offers valuable insights for researchers, programmers, and language technology enthusiasts who are interested in improving current models. Using NLP methods and deep learning, the proposed models seek to:

- Handle grammatical norms and morphological variances;
- Improve error identification and correction for Gujarati text.
- Improve accuracy and precision for Gujarati.
- Improve Gujarati NLP applications' language processing efficiency

The upcoming session addresses the difficulties associated with Gujarati language spelling correction, exploring several methodologies including rule-based, statistical, and deep learning approaches to rectify spelling and grammatical problems in Gujarati. The subsequent lesson presents two models designed to identify and rectify context aware

spelling mistakes in Gujarati language sentences. The initial model employs Peter Norvig's algorithm and a Gated Recurrent Units (GRU) model, trained using a Gujarati word dictionary, to detect and activate errors while analyzing phrase context. Another model employs IndicBERT to finetune the Peter Norvig-based model, enhancing efficiency and accuracy by concentrating on omitted words inside the statement and assigning scores to forecast sentence correctness. The comparative analysis with respect to accuracy, precision, recall and F1 score for correct and incorrect statements with one of the existing tools has been covered in next section.

2 Related work and background theory

2.1 Characteristics and challenges of gujarati language

Gujarati is a language that is rich in vocabulary. There are a number of inflections for adjectives, verbs, and nouns. The language has 12 vowels and 34consonants, as well as the ideas of matras and half consonants [3] [13]. Gujarati has several characters with practically same phonetic characteristics. Matras' sounds match those of a vowel: આ, ઇ, ઈ, ઉ, ઊ, એ, ઐ, ઓ, ઔ, અં, and અઃ. All consonants possess inherent vowels. Furthermore, In Gujarati, vowels and constants can stand on their own or be accompanied by one or more matras, which are seen as distinct characters after the vowels and constants. A total of twelve possible word usages exists for each constant, as it is possible for it to appear with each of the eleven matras. In this manner, a total of 374 permutations of constant and matra are generated by combining all 34 constants with 11 matras [3]. Therefore, matras must be handled with due diligence during computer processing [13]. In the Gujarati language, the phonemes િ (e) and ી (ee) are identical, differing only in their degree of extension. The situation is analogous for ુ (u) and ૂ (oo). Consequently, words containing these characters are frequently misspelled. For instance, both પૂજા and પુજા are pronounced as '{pooja}' and signify ' worship'. In contrast to English, prepositions such as in and to can take on suffix inflections within the word, and words can even have several inflections complicates the process of spelling error detection and correction. As a highly inflected language, it is challenging to compile all potential word forms in a lexical dictionary for a spelling checker for the Gujarati language.

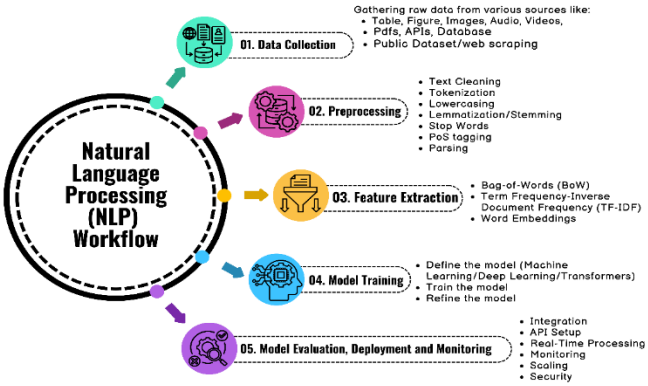

Figure 1: Natural Language Processing (NLP) workflow

2.2 Various spelling and grammar checking approaches

Creating a humanoid—the most intelligent machine ever—is the ultimate objective of artificial intelligence.

An important consideration in this development process is the interaction between humans and computers for which the language tools developed that can comprehend communication languages across all technological dimensions need to be considered. To ensure that words are spelt correctly, the spell-checker consults the language's dictionary and lexicon. An easy way to understand a spellchecker is that it uses a spelling detector to look for words that are not in base form in a document and a spelling corrector to replace them with the most likely term from a database or corpus.

Preprocessing, feature extraction, and modeling are the three primary steps that make up the Natural Language Processing (NLP) pipeline as shown in Figure 1 [14]. Every step of the process changes the text in some manner and generates a result that is needed by the following step. Sometimes, there are non-linear steps in the NLP pipeline. Going back and forth between the various stages is often essential in practice. For instance, if the modeling stage yields unsatisfactory results, it might be required to revisit the pre-processing or feature extraction stage in order to enhance the data's quality.

There are primarily three stages to spellchecking which include thorough pre-processing, spelling check, and creation of recommendation lists. Some of the steps involved in preprocessing include stemming, tokenization, and normalization [15] [16]. The spelling-checking module uses several dictionary lookup techniques to verify the candidate words' authenticity, while the recommendation list building module flags the list of possible suggestions for misspelled words. Suggestions are ranked by the spell-checker. This part ranks the ideas according to how necessary they are for the sentences. The primary stages of a spellchecker are as shown in Figure 2. After running the sentence through spellcheck, it returns a corrected version containing the correct term.

2.2.1 Syntax based

Each sentence obtains a parse tree created according to the base language's grammar. Text is inaccurate if full parsing fails. So, the parser should be as thorough as possible to reduce false alerts. The main advantage of this method is that the grammar checker will detect all errors if the grammar is complete and covers all possible syntactic rules.

2.2.2 Rule based

When checking for spelling errors, these systems use heuristics derived from various word properties, including as morphology, part-of-speech, stemming, and more [4]. A rule-based spell-checker using part-of-speech (POS) tagging for English language spell-checking is also being used. Additionally, text chunkers were created using the Hidden Markov Model to improve spell-checker speed [4]. This technique allows for progressive system expansion by starting with one rule and adding more [17].

2.2.3 Statistical based

The ability to speak a specific language is not necessary to understand statistical procedures. Examples of spellcheckers that use word counts and word characteristics include those that are frequency-based, n-gram-based, and finite state automata-based [4]. The statistical method greatly enhances performance without requiring knowledge of the particular language, which is a major advantage. One problem with these approaches is that they rely on metrics like word count, frequency, and characteristics to do spellchecking, yet processing certain spelling mistakes necessitates familiarity with the target language. Many academics employed a combination of rule-based and statistical approaches to address this type of problem. To get past the problems, a hybrid model combines rule-based and statistical approaches [4] [18].

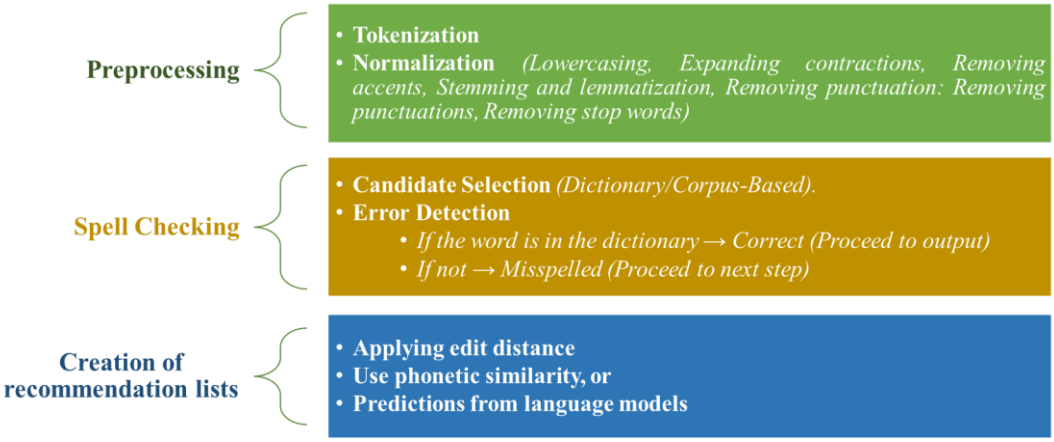

Figure 2: The primary stages of Spelling checker using NLP

Table 1: Comparison of various spell checker and grammar checker for low resource languages

Refere	Year	Language	Research Focus	Methodology/Approach	Key Findings & Accuracy
[24]	2002	Assamese	Dictionary-based spellchecker	Dictionary lookup, bigram search, Soundex code integration	Adequate results with over 5000 words, integration with Assamese-English dictionary in progress
[26]	2012	Kashmiri	Spellchecker development	Standalone application, non-word error correction	80% error detection, 85% correct suggestions, plans for real-word error handling
[25]	2013	Urdu	Spellchecker evaluation	Reverse edit distance method	High complexity (86n + 41 comparisons), needs improved methods for better accuracy
[27]	2015	Hindi	HINSPELL spellchecker	Error detection, repair, substitution	83.2% detection, 77.9% correction, future focus on grammatical errors
[28]	2015	Tamil	Morphological analyzer	Linguistic analysis, POS tagging	Efficiency between 60-97%, useful for NLP tasks like MT, lemmatization, parsing
[29]	2016	Kashmiri	Improved spellchecker	Standalone application, lexicon development	80% detection, 85% correct recommendations, integration with OpenOffice needed
[30]	2016	Tamil	Hybrid spellchecker	N-gram, stemming, tree-based algorithm	91% accuracy, tree-based method better for error detection
[31]	2016	Telugu	Spellchecker with sandhi analysis	Morphophonemic, external sandhi handling	Addresses Telugu's complex linguistic features
[19]	2018	Malayalam	Deep learning-based spellchecker	LSTM neural networks, error detection & correction	Outperforms Unicode splitting, limited by computational resources
[32]	2019	Bengali	Spellchecker development	Hybrid of edit distance, Soundex	Adapts existing methods for better Bengali spell correction
[33]	2020	Multilingual	Comprehensive spellchecker review	Literature analysis, NLP methods (rule-based, statistical, deep learning)	Categorizes spellcheckers, compares performance across languages
[25]	2020	Tamil	Alternative spellchecking methods	Bloom-filter, Symspell, LSTM	Symspell is fast but lacks accuracy; LSTM is promising but underexplored
[3]	2021	Gujarati	Jodani spellchecker	Root word-based, Levenshtein distance	91.56% accuracy, plans to improve character assumption handling
[23]	2022	Dogri	Hybrid spellchecker	Hybrid methodology for detection & correction	First known attempt for Dogri spellchecking
[34]	2023	Gujarati	Enhancing ASR Performance with Improved Spell Corrector	Combination of MFCC and CQCC features, GRU-based DeepSpeech2 architecture, and enhanced spell corrector	Improved Word Error Rate by 17.46% compared to the model without post-processing
[11]	2024	Gujarati	Spell Checker Using Norvig Algorithm	Implementation of Norvig's algorithm with a dataset of 16,937 distinct Gujarati words	Achieved 80–90% accuracy in identifying and correcting misspelled words

Table 2: Comparison of spell correction models designed for multilingual and low-resource languages

Model	**Language Coverage**	**Architecture**	**Error Handling (Morph., Contextual)**	**Computational Efficiency**	**Novelty/Remarks**
IndicBERT [35]	12 Indic + English	ALBERT (Transformer-based)	Limited analysis; strong recall	Efficient (ALBERT backbone)	First shared Indic ALBERT model
MuRIL (Google) [36]	17+ Indian Languages	Multilingual BERT variant	Better contextual handling (cross-lingual pretraining)	Moderate to high	Strong cross-lingual transfer
XLM-R [37]	100+ languages	RoBERTa-based	Strong contextual handling	High resource requirements	Robust multilingual performance
Adapter-BERT for Indic [38]	Varies	BERT + Adapters	Task-specific error modeling possible	High efficiency (modular)	Scalable low-resource adaptation
L3Cube-IndicSBERT [39]	10 Indic Languages	Multilingual SBERT	Enhanced sentence-level contextual embeddings	Efficient fine-tuning	First multilingual sentence representation model for Indic languages
ArabicCorrectionCntxt [40]	Arabic	Levenshtein + Context Probabilities	Handles contextual errors via paragraph-level keyword-based context detection	Efficient (simple lexical + context)	Uses paragraph context and keyword frequency to re-rank corrections
Icelandic Contextual Spell Corrector [41]	Icelandic	ML Classifiers + Morphological Tags	Strong contextual disambiguation; affected by rich morphology	Moderate (due to tag sparsity)	Contextual confusion-set disambiguation with lemmatized and PoS features
Context-Free ML Spell Corrector [42]	English (demonstrated); extendable	Supervised ML (e.g., Naive Bayes)	No context used; character/word/token-based input features	Efficient for standalone terms	Context-free, character-level input with multiple ML classifiers
Amazon Multilingual Spell Checker [43]	24 Languages (Indic included)	N-gram-based + SymSpell Ranking	Context-aware using n-gram conditional probabilities	Real-time capable (optimized Trie)	Real-time, extendable to new languages via Wikipedia + subtitle corpora

2.2.4 Deep learning based

Deep learning is the specialty of artificial neural networks, or ML algorithms. Deep learning algorithms have been widely employed and effective lately. Deep learning approaches' success is partly due to the freedom of architecture selection. Deep learning methods were used in ML research for natural language processing [17]. While the rule-based and statistical methods demonstrate significant effectiveness, the performance of spell-checking can be further improved through the application of deep-learning techniques. Regarding regional languages, the deep-learning-based spell-checker is currently available for the Malayalam and Tamil language which utilizes an LSTM network [19] [20]. This spell-checker involves a network that is both trained and tested to detect spelling errors and pinpoint their locations [4] [21] [22] [44] [45] [46] [47].

3 Comparative analysis of Spell checker for various regional languages of India

Natural language processing (NLP) depends much on spell and grammar checkers since they find and fix textual data mistakes. Although a lot of study has been done on English and other generally spoken languages, regional languages of India provide special difficulties because of their rich morphology, complicated phonetic structures, and different scripts. The spell-checking methods created for several Indian languages—including

Hindi, Bengali, Tamil, Telugu, Gujarati, Dogri, Malyalam and Assamese—are compared in this work [23] [33]. Examining several approaches including rule-based, statistical, hybrid, and deep learning-based spell-checkers, the paper assesses their performance in managing orthographic variants, phonetic mistakes, and real-word errors. Particularly languages with strong inflectional morphology, like Tamil and Telugu, call for more complex methods like sandhi-based and morphological analyzers; languages like Hindi and Bengali gain from hybrid approaches combining edit distance and phonetic algorithms [6] [21] [24] [25]. Emphasizing the importance of language-specific optimizations, the study shows the benefits and restrictions of every technique. Future lines of research include using transformer-based models such IndicBERT and GRU to better contextual spell-checking, hence improving accuracy over several languages. Through tackling these difficulties, our work hopes to help to create more strong and effective spell-checking systems for India's linguistically varied terrain. Table 1 shows the differences between spell checkers for languages with few resources. Table 2 looks at spell correction models that work in more than one language. These models, such as IndicBERT, MuRIL, and Amazon's spell checker, use contextual and probabilistic methodologies. Others deal with linguistic and morphological issues.

Previously studies examined spellchecking in Indic languages by dictionary lookup [24], root-based matching [3], hybrid linguistic methodologies [30], and deep learning [19]; nevertheless, these systems exhibit notable deficiencies. Rule-based systems like Jodani for Gujarati or HINSPELL for Hindi do a good job of finding errors, but they don't take context into account, so they can't tell the difference between real-word errors (such homophones or words that look similar). Hybrid approaches created for Tamil and Dogri enhance identification via stemming or POS-based criteria; nonetheless, they are still language-specific and do not adapt well to the orthographic complexity of Gujarati (diacritics, compound characters). Neural methods, like LSTM-based rectification for Malayalam or GRU-based ASR augmentation for Gujarati [34], show gains in context, but they are constrained by datasets that are particular to a certain field, a lack of generalization, and high computing needs.

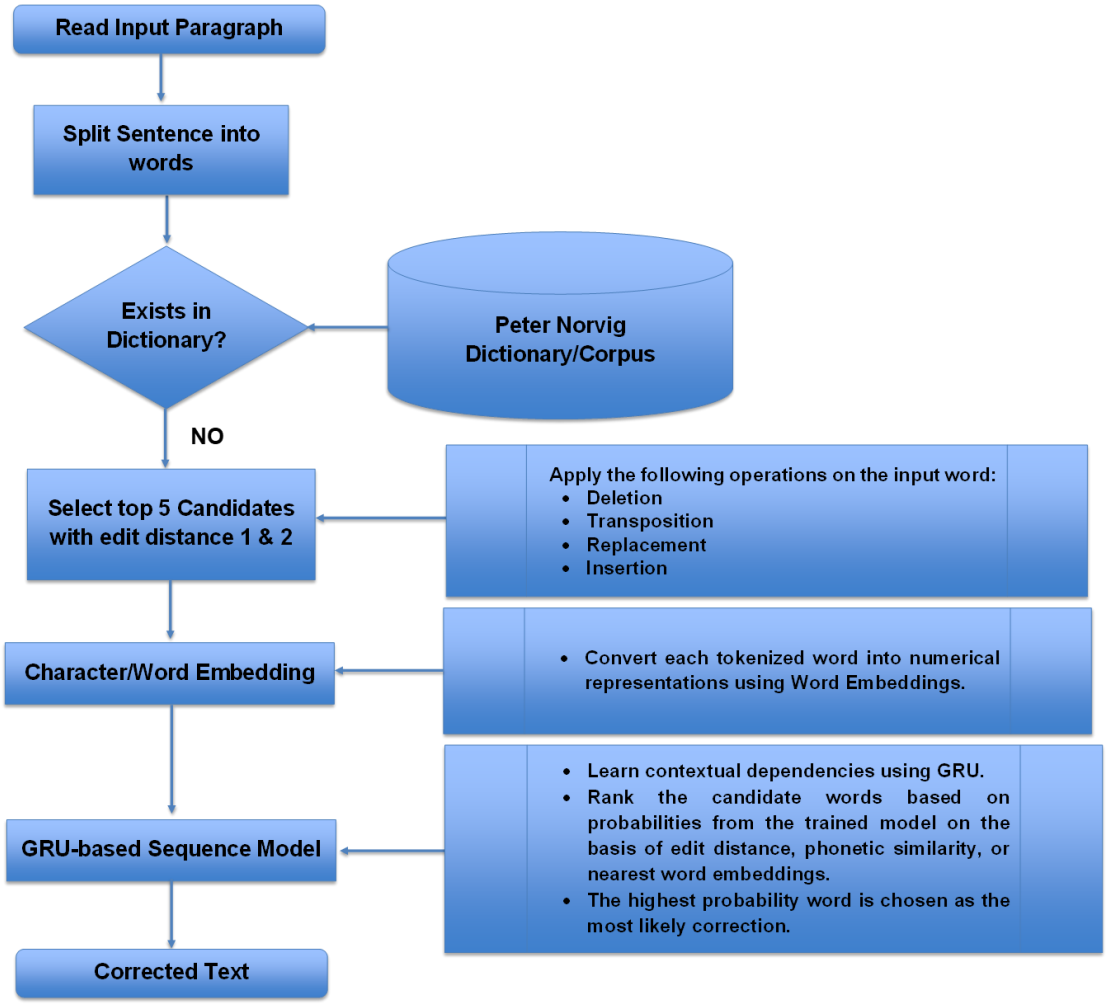

Figure 3: Novel and hybrid Spelling and Grammar Error Correction approaches for Gujarati Language using Peter Norvig with GRU – GUJAPUBRIJ Model

Multilingual transformer models (IndicBERT [35], MuRIL [36], XLM-R [37]) provide strong contextual embeddings, but they have two problems when it comes to correcting Gujarati spelling: (i) they are pre-trained on mixed Indic corpora without domain adaptation, which makes them bad at Gujarati morphology, and (ii) they need too many resources to be useful for lightweight deployment. These gaps highlight the need for a context-aware, Gujarati-specialized spellchecking approach that balances linguistic coverage with computational efficiency.

4 Proposed GUJAPUBRIJ and GUJBRIJAPU Models

The proposed Gujarati spell checker models are founded on Peter Norvig's algorithm, which applies probabilistic methods and edit-distance operations for error correction using a Gujarati lexicon. Although effective in detecting typographical errors within a predefined dictionary, this approach is limited by its lack of contextual awareness. To overcome this limitation, two enhanced models were developed: the GUJAPUBRIJ model, incorporating a GRU-based neural network, and the GUJBRIJAPU model, leveraging IndicBERT for contextual analysis. For ease of reference and novel identification, the models were named after the contributing researchers, Apurva and Brijehkumar. This approach represents a significant advancement in context-aware spell checking for Gujarati and contributes to strengthening NLP resources for low-resource Indic languages by delivering improved accuracy, reliability, and applicability.

4.1 Novel and hybrid spelling and grammar error correction approaches for Gujarati language using Peter Norvig with GRU – GUJAPUBRIJ model

The GUJAPUBRIJ model combines Peter Norvig's probabilistic spell correction with a GRU-based neural network to make it more accurate and aware of the context. A carefully chosen Gujarati vocabulary was used as the reference dictionary to make sure that any adjustments made through insertions, deletions, substitutions, and transpositions were correct from a language point of view.

TensorFlow's Tokenizer split the input text into tokens (using <UNK> for OOV words) and added

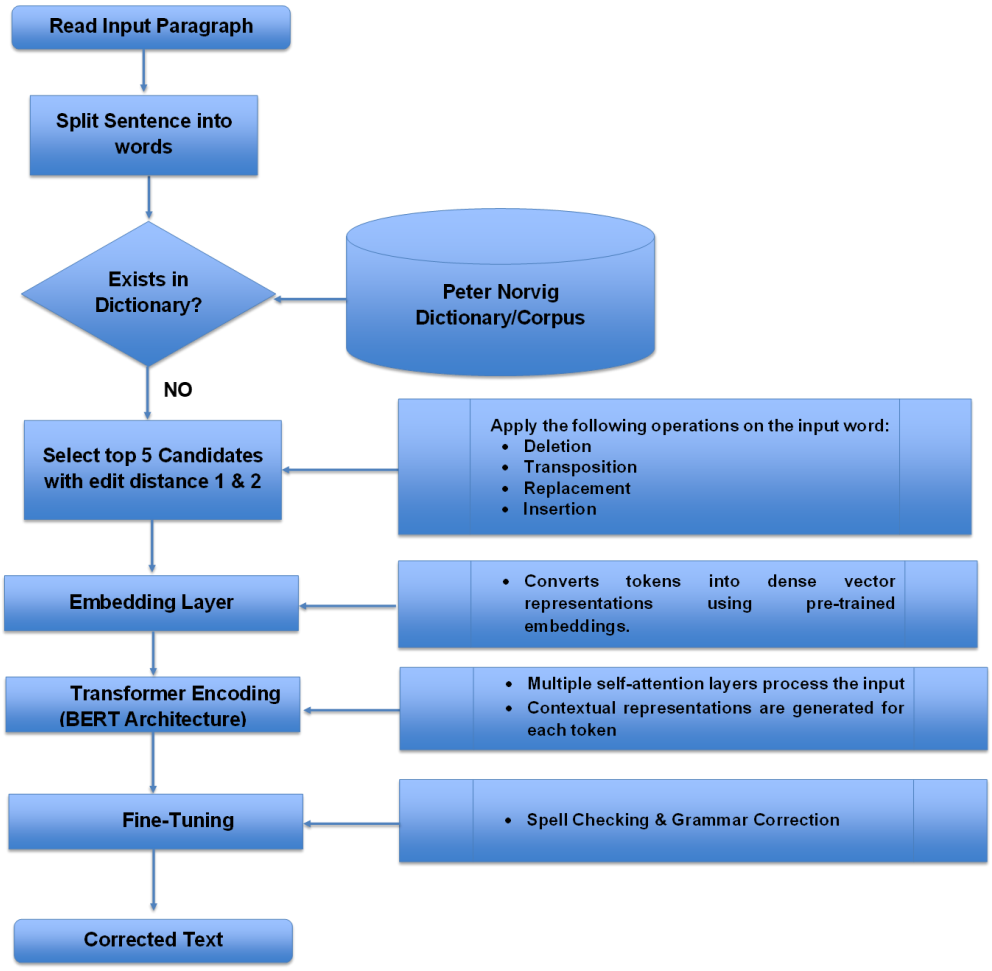

Figure 4: Novel and hybrid Spelling and Grammar Error Correction approaches for Gujarati Language using Peter Norvig with IndicBERT- GUJBRIJAPU Model

padding to make them all the same size. The GRU network then took these sequences and used them to choose the most appropriate correction from the top five choices that were within edit distance 1–2. This preparation pipeline made sure that both the spelling and the meaning were correct.

The proposed GUJAPUBRIJ novel model for the Gujarati text spelling checker based on Peter Norvig and GRU are depicted in Figure 3.

Peter Norvig's approach fixes mistakes by utilizing edit distance and probability. This works well for typos within words, but it needs a defined lexicon and doesn't know what the context is. On the other hand, GRU-based neural networks work with sequential language input, taking into account how words are related to each other in context to improve grammar checks and deal with real-world mistakes that Norvig's method can't fix.

4.2 Novel and hybrid Spelling and grammar error correction approaches for Gujarati Language using Peter Norvig with IndicBERT – GUJBRIJAPU model

The GUJBRIJAPU model builds a Gujarati spell checker by moving from dictionary-based methods to a BERT-based framework that can rank phrases based on how well they fit in with the context as shown in Figure 4. The preprocessing pipeline uses lexical normalization and semantic validation. It starts with a cleaned Gujarati corpus to make the reference vocabulary.

The input text is broken up into tokens, and for each token, Peter Norvig's probabilistic method creates possible repairs by using edit-distance operations (insertions, deletions, replacements, transpositions) that have been changed for Gujarati. Candidates that follow spelling guidelines are ranked, and the top five (with an edit distance of 1–2) are chosen for further testing.

To add context, candidate-corrected sentences were run using IndicBERT, a multilingual model that has already been trained on Indian languages, to make contextual embeddings. A scoring system (masked language modeling or sentence-level probability) selected choices based on how well they fit semantically. The sentence with the best score was chosen as the correction. This two-step technique worked well for both real and non-word errors by combining lexical correction with contextual relevance. Tokens were turned into dense embeddings and improved by IndicBERT's self-attention layers, which made the grammar and spelling more accurate overall. The mMML model based on BERT was used to make the spell checker more accurate and faster. The Sentence Scoring Model (which checks the validity of sentences) was chosen for Gujarati because it does a good job of ranking candidate sentences by accuracy.

5 Experimental evaluation and Analysis of the proposed model

To develop a Gujarati language dataset for the study, data was sourced from publicly available resources provided by the Ekatra Foundation, accessible via https://www.ekatrafoundation.org/. Additionally, a curated Google Drive folder containing extensive Gujarati textual data was utilized (https://drive.google.com/drive/folders/17gskNhAGgzOpncOh2VsAKC4Fc0ju5GaC?usp=sharing). Over 100,000 sentences were initially collected from these sources. The preprocessing of the dataset includes converting to lowercase Unicode NFC, managing compound characters and diacritics specifically for the Gujarati language (ZWJ/ZWNJ), removing duplicates, stripping punctuation, and selecting sentences based on length. Peter Norvig's edit-distance approach was modified for the Gujarati script and diacritics, and it was used for candidate generation. For training, the Hugging Face IndicBERTv2-SS AutoTokenizer could handle sentences up to 128 tokens in length, and for inference, it could handle sentences up to 512 tokens in length, therefore it could handle lengthier candidate sentences. After performing a thorough data cleaning and preprocessing process to ensure quality and relevance, a final dataset comprising 20,000 sentences was created. This refined dataset includes both correct and erroneous sentence pairs, making it suitable for tasks such as spell error correction and language model training. A dataset of 20,000 sentences has been divided into one of two groups:

1. Sentences that are grammatically and orthographically correct.
2. Incorrect sentences that have common spelling and grammar mistakes that are common in Gujarati.

To make supervised learning easier, sentences were given probability-based labels: 0.9 for right and 0.1 for wrong. This rating helped the model learn how to tell the difference between valid and invalid text using regression. To make sure that the evaluation was strong, the dataset was split into 16,816 training samples and 4,204 validation samples. The GUJAPUBRIJ (Peter Norvig + GRU) and GUJBRIJAPU (Peter Norvig + IndicBERT) models were trained using the dataset that had been built up. The training process was mostly about regression-based fine-tuning, which meant that the model gave each potential sentence a likelihood score. Then, a rating system was used:

1. Predicted scores were used to rank the candidate sentences.
2. The sentence with the greatest probability score was chosen as the one that was most likely to be correct.

This improvement made the GUJBRIJAPU model better at correcting context by letting it tell the difference between small spelling problems and big grammar issues. The GUJAPUBRIJ model converged quickly, getting about 100% training accuracy in just two epochs. The validation accuracy also stabilized close to 100%. This meant that the model not only learned the training patterns by heart, but it also worked well with new validation data. This indicates that the model is not Over Fitting; instead, it is showing that it can generalize well on the validation set.

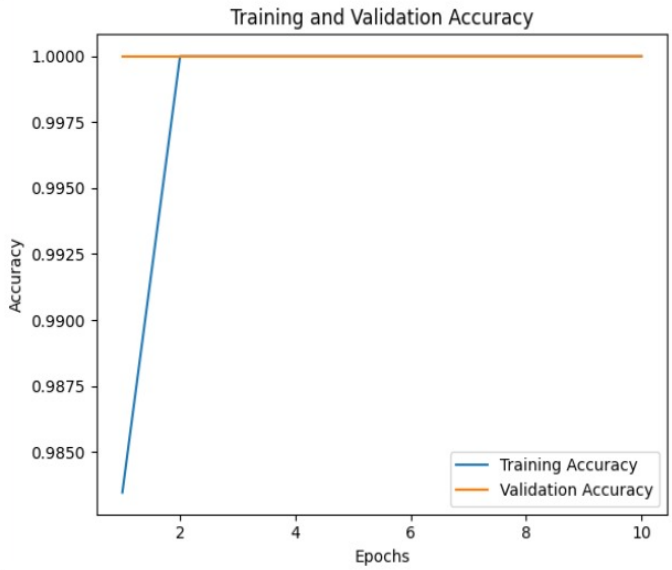

Figure 5: Training and Testing accuracy for Novel and hybrid Spelling Error Correction approaches for Gujarati Language using Peter Norvig with GRU – GUJAPUBRIJ Model

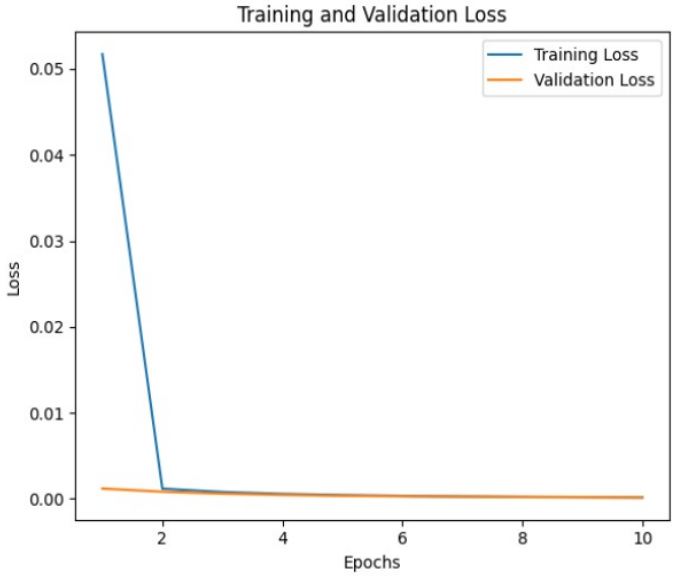

Figure 6: Training and validation loss for Novel and hybrid spelling error correction approaches for Gujarati Language using Peter Norvig with GRU- GUJBRIJAPU Model

Table 3: Performance analysis of Novel and hybrid spelling correction approaches for Gujarati language GUJAPUBRIJ and GUJBRIJAPU Model

Evaluation Parameters	**Peter Norvig with GRU – GUJAPUBRIJ Model**		**Peter Norvig with IndicBERT – GUJBRIJAPU Model**	
	Incorrect sentences (in %)	**Correct Sentences (in %)**	**Incorrect sentences (in %)**	**Correct Sentences (in %)**
Accuracy	71.00	85.00	84.79	93.49
Precision	71.00	84.00	86.21	94.46
Recall	71.00	85.00	83.54	90.13
F1 Score	70.00	84.00	85.74	91.59

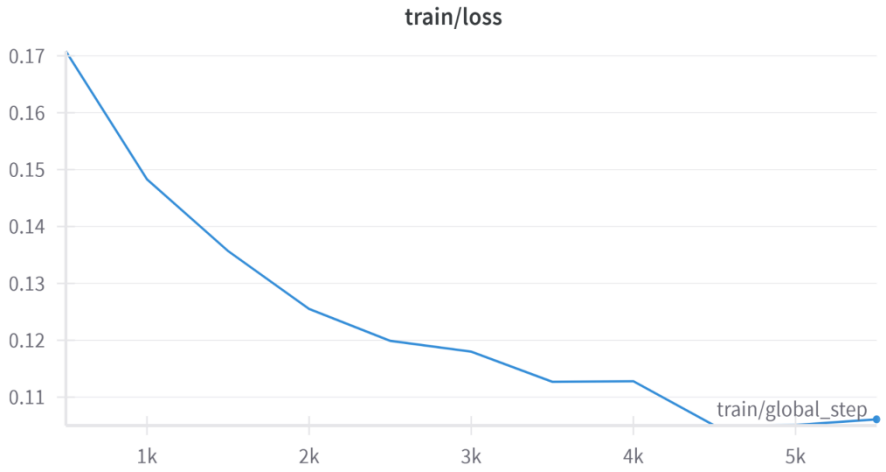

Figure 7: train/loss plot for Novel and hybrid Spelling Error Correction approaches for Gujarati Language using Peter Norvig with IndicBERT – GUJBRIJAPU Model

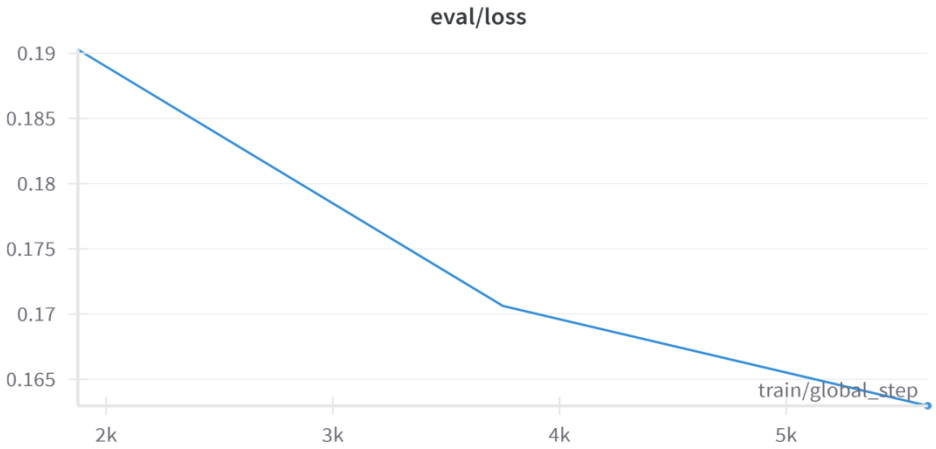

Figure 8: eval/loss plot for Novel and hybrid Spelling Error Correction approaches for Gujarati Language using Peter Norvig with IndicBERT – GUJBRIJAPU Model

But the validation accuracy being so high could also be due to the dataset itself, like its small size or the fact that the training and validation samples are quite comparable. This is something to keep in mind when looking at these results. The Table 3 shows accuracy statistics were achieved for correct and incorrect sentences for both the proposed models.

The Figure 5 shows the graph for training and validation accuracy while Figure 6 shows the graph for training and validation loss.

Training epochs and results for the GUJBRIJAPU Model based on Peter Norvig with IndicBERT are plotted as shown in Figure 7 and Figure 8.

For the GRU-based GUJAPUBRIJ model, we used Random Search with Keras Tuner to find the best hyperparameters in the following space: embedding dimensions {64, 96, 128, 160, 192, 224, 256}, GRU hidden units {32, 64, 96, 128}, dropout {0.1–0.5}, and learning rates {1e-3, 1e-4, 1e-5}. The batch size was fixed at 32 and the maximum number of trials was 10. The best setup, which was determined based on the validation F1-score, was: Embedding dimension = 128 (adding more dimensions made the model fit too well without improving performance), GRU hidden units = 64 (enough power without making training take longer or increasing the danger of overfitting), Dropout = 0.3 (kept training stable while keeping it from underfitting at lower rates), Learning rate = 1e-3 (this made the process go faster than 1e-4 or 1e-5), with 10 epochs and 32 batches.

This system found a good balance between being complicated and being able to apply to a lot of different situations. To reduce the amount of processing power needed, each configuration was only performed once (executions_per_trial=1). We recognize that random initialization creates variation; in subsequent research,

Table 4: Hyperparameter tuning for the Novel and hybrid Spelling Correction approach for Gujarati Language using Peter Norvig with GRU- GUJAPUBRIJ Model

Hyperparameter	**Range / Value**	**Description**
embedding_dim	64 to 256 (step=32)	Size of embedding vectors, tuned using Keras Tuner
gru_units	32 to 128 (step=32)	Number of units in GRU layer, tuned using Keras Tuner
input_length	max_seq_length from data	Length of padded sequences
Optimizer	Adam	Optimization algorithm used for training
Loss	Binary Crossentropy	Loss function for binary classification
Metrics	Accuracy	Evaluation metric during training and validation
tuning method	Random Search	Keras Tuner with 10 trials and validation accuracy goal

Table 5: Hyperparameter tuning for the Novel and hybrid Spelling Correction approach for Gujarati Language using Peter Norvig with IndicBERT – GUJBRIJAPU Model

Parameter	**Value**	**Description**
Model	ai4bharat/IndicBERTv2-SS	Pretrained Indic language transformer model
Number of Labels	1	Binary classification setup
Tokenizer	AutoTokenizer (from model)	Handles subword tokenization using the same model
Optimizer	Adam	Adaptive Moment Estimation (default in Trainer)
Learning Rate	2e-5	Fine-tuning rate for BERT parameters
Batch Size	16	Per-device mini-batch size
Number of Epochs	3	Number of full passes over training data
Weight Decay	0.01	L2 regularization strength
Evaluation Strategy	Epoch	Evaluate after every epoch
Loss Function	Binary Cross Entropy (via Trainer)	Used for sentence-level scoring
Compute Metrics	Custom scoring (e.g., accuracy or ranking)	Contextual ranking of candidates

multiple seeds will be employed for reliable significance reporting.

The AI4Bharat/IndicBERTv2-SS model served as the basis for the GUJBRIJAPU model, which was then fine-tuned on the 20,000-sentence Gujarati dataset we talked about earlier. To help with regression-based training, each sentence was given a probability score (0.1 for wrong, 0.9 for right). We used AdamW to improve the model. The learning rate was 2e-5, the batch size was 16, there were three epochs, the weight decay was 0.01, and the binary classification loss was changed to work with regression-style output. This made sure that the model could score potential sentences based on how well they fit in with the context. We used general-purpose IndicBERT, but we note that using bigger Gujarati-only corpora for domain adaptation could boost rectification performance even more and is something we will explore on in the future. The Table 4 and Table 5 shows the hyperparameters details for both the proposed model.

The entire study was conducted on Google Colab with an NVIDIA Tesla T4 GPU (16GB). The software environment included Python 3.10, PyTorch 2.0.1+cu118, Hugging Face Transformers 4.30.2, Datasets 2.13.0, and CUDA 11.8, as well as Weights & Biases (wandb), pandas, and numpy. We used the Hugging Face Trainer API with AdamW optimization, a weight decay of 0.01, and a binary classification loss that was changed to work with regression-style outputs for training. You can use PyTorch, NumPy, and Python's random module to set random seeds so that your results are the same every time you run the program. You can also turn on deterministic cuDNN mode.

6 Discussion and comparative evaluation of the proposed GUJAPUBRIJ and GUJBRIJAPU model

Two context-aware Gujarati spell-checking models: the GRU-based GUJAPUBRIJ (blue solid line with triangles) and the IndicBERT-based GUJBRIJAPU (green dashed line with squares) are compared in the Figure 9 and Figure 10 graph on the basis of Accuracy, precision, recall and F1 score with the metric values displayed on the y-axis (percentage scale) and the metric types arranged on the x-axis for incorrect sentences and correct sentences respectively.

The comparative assessment demonstrates that the IndicBERT-based GUJBRIJAPU Model substantially outperforms the GRU-based GUJAPUBRIJ Model across all principal performance metrics as shown in Table 7.

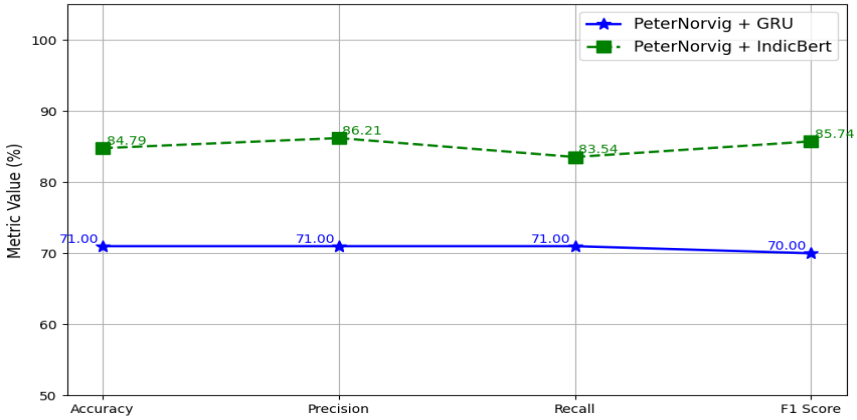

Figure 9 Comparative analysis of both the proposed model on incorrect sentences

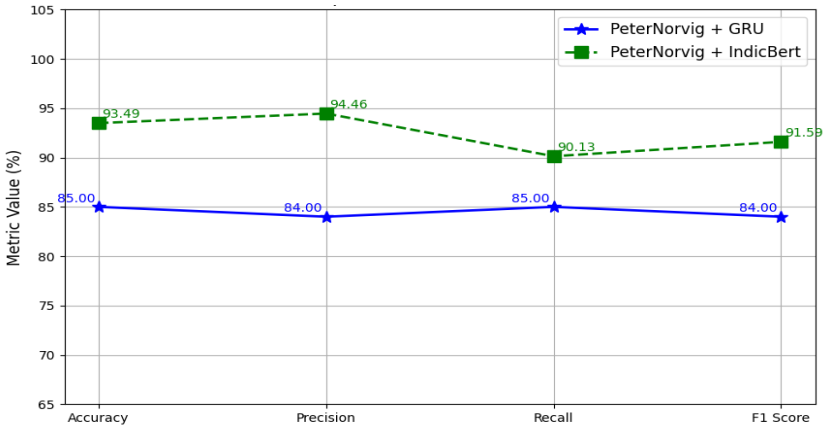

Figure 10: Comparative analysis of both the proposed models for correct sentences

Table 6: Comparative analysis of proposed GUJAPUBRIJ/GUJBRIJAPU model with Jodani [3]

Feature	**Peter Norvig with GRU / IndicBERT– GUJAPUBRIJ/GUJBRIJAPU**	**Jodani [3]**
Spelling Correction	Deep- learning based approach	Rule based
Grammar Checking	Yes	No
Contextual Understanding	Yes	No
Efficiency & Speed	More Computation power	Faster and lightweight
Scalability & Learning	Can be fine-tuned	Predefined rules are used.
Use case	NLP applications (Chatbots, Translation, Gujarati Grammar Checking)	Word Processing, Typo Correction

Table 7: Performance analysis of proposed model with Jodani [3]

Metric	**Peter Norvig with GRU-GUJAPUBRIJ**	**Peter Norvig with IndicBERT-GUJBRIJAPU**	**Jodani [3]**
Accuracy	85.00%	93.49%	~87-90%
Precision	84.00%	94.46%	~89.00%
Recall	85.00%	90.13%	~86.00%
F1 Score	84.00%	91.59%	~87.00%

The GUJBRIJAPU model is more reliable when it comes to handling both real-world and non-word errors since it has higher accuracy (93.49% vs. 85.00%), precision (94.46% vs. 84.00%), recall (90.13% vs. 85.00%), and F1-score (91.59% vs. 84.00%). It is important to note that it lowers the number of false positives and misclassifications of wrong sentences, which leads to more context-aware and linguistically correct repairs. These results show that GUJBRIJAPU is a strong Gujarati spell and grammar checker, which sets the stage for further improvements and changes for specific fields.

The Novel Peter Norvig and GRU/IndicBERT based GUJAPUBRIJ/GUJBRIJAPU Model context aware spelling checker for Gujarati Text is compared with Jodani which a spellchecker tool for Gujarati that finds mistakes and fixes them while also suggesting ways to be more accurate. Jodani boosts accuracy by combining rule-based and statistical methods and uses a predetermined Gujarati vocabulary to identify misspelled words using edit distance and phonetic comparableness. It deals with orthographic mistakes such as missing characters, transposition, addition, and substitution using phonetic similarities, N-gram models, and Peter Norvig's technique for correcting spelling to enhance precision.

The IndicBERT-based GUJBRIJAPU model outperforms both the GRU-based GUJAPUBRIJ and the rule-based Jodani on all important measures. Its higher accuracy (93.49%) and precision (94.46%) show that it as fewer false positives and more reliable corrections. Its higher recall (90.13%) shows that it is better at finding real-world and non-word errors. GUJBRIJAPU is better than Jodani because it can check grammar and grasp context, while Jodani can only fix spelling mistakes using edit distance and phonetic similarity. It can find and fix grammar-sensitive mistakes by using contextual embeddings, which makes it better for NLP tasks like grammar checking, translation, and conversational systems. The key trade-off is the cost of computing: GUJBRIJAPU is slower and uses more resources than Jodani, which is still light and useful for speedier programs like word processors. Some unusual and code-mixed terms were misclassified, which means that more training and domain adaptation are needed. This technique is new since it combines IndicBERT for understanding context, Peter Norvig's algorithm for generating candidates, and neural sequence modeling for grammar sensitivity. This integration goes beyond small changes and offers a unified, context-aware spelling and grammar checker for Gujarati. This is better than current systems that only fix spelling based on rules. Table 6 compares the proposed models with Jodani [3], The researcher’s proposed work is based on deep learning and includes a comparison with a rule-based approach. Currently, there is no existing spell checker developed at an individual level for direct comparison. Therefore, to clearly demonstrate the novelty of the research, a comparative analysis with the older rule-based system has been included. Although this comparison may not be technically perfect, it helps in better understanding the improvement and effectiveness of the proposed model. From this comparison, researchers demonstrate the novelty and effectiveness of their proposed work.

7 Conclusion

Spell and grammar checkers are vital to natural language processing (NLP) because they detect and fix mistakes in textual input. Although a lot of study has been done on English and other generally spoken languages, the regional languages of India provide special difficulties because of their complicated morphology, sophisticated phonetic patterns, and different scripts. The study carried out in this article emphasizes how difficult context aware spell checking in Indian regional languages especially Gujarati is given their intricate linguistic systems. The novel and hybrid error detection and correction approach based on Peter Norvig's spelling correction algorithm in collaboration with GRU (GUJAPUBRIJ Model) neural networks and IndicBERT (GUJBRIJAPU Model) are proposed and assessed. With outstanding accuracy, precision, recall, and F1-score, IndicBERT regularly exceeded GRU among the evaluated models. For all important criteria, Peter Norvig's method combined with IndicBERT means GUJBRIJAPU Model showed the best performance;
Thus, it is the most dependable method for context aware grammar correction in Gujarati literature. Jodani, rule-based spell checker for Gujarati, suffers with grammatical precision and lacks contextual knowledge even if it offers a quick rule-based fix for spelling errors. On the other hand, the Peter Norvig with IndicBERT GUJBRIJAPU Model is perfect for uses needing excellent contextual understanding since it uses deep learning to precisely identify spelling and grammatical mistakes. Still, Jodani is a good choice in situations requiring speedier processing with reduced computing burden. With improved accuracy and contextual awareness, the Peter Norvig with IndicBERT GUJBRIJAPU Model eventually seems to be the most solid approach for Gujarati language processing. Its capacity to lower false positives, increase classification accuracy, and offer thorough error correction makes it a useful tool for many NLP uses where linguistic accuracy is crucial. The computation overload can be reduced in future to improve the performance of the Peter Norving with IndicBERT GUJBRIJAPU model. Moreover, the proposed models are unable to identify the error related to punctuations which can be improved in future.

References

[1] N. G. Patel and D. D. B. Patel, "Research review of Rule Based Gujarati Grammar Implementation with the Concepts of Natural Language Processing (NLP)," Journal of Emerging Technologies and Innovative Research (JETIR), vol. 5, no. 9, 2018. https://doi.org/10.6084/m9.jetir.JETIRA006276

[2] N. P. Desai and V. K. Dabhi, "Resources and components for Gujarati NLP systems: a survey.," Artificial Intelligence Review, vol. 55, pp. 1-19, 2022. https://doi.org/10.1007/s10462-021-10120-1

[3] H. Patel, B. Patel and K. Lad, "Jodani: A spell checking and suggesting tool for Gujarati language," in 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2021. https://doi.org/10.1109/Confluence51648.2021.9377072

[4] S. Singh and S. Singh., "HINDIA: a deep-learning-based model for spell-checking of Hindi language," Neural Computing and Applications, vol. 33, no. 8, pp. 3825-3840, 2021. https://doi.org/10.1007/s00521-020-05207-9

[5] M. Gokani and R. Mamidi, "GSAC: A Gujarati Sentiment Analysis Corpus from Twitter," in Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, Association for Computational Linguistics, 2023. https://doi.org/10.18653/v1/2023.wassa-1.12

[6] S. Bhuva and D. Mishra, "Gujarati Optical Character Recognition Using Efficient Text Feature Extraction Approaches.," Informatica, vol. 49, no. 28, 2025. https://doi.org/10.31449/inf.v49i28.8341

[7] J. Baxi and B. Bhatt., "GujMORPH-ADatasetforCreatingGujaratiMorphological Analyzer," in ProceedingsoftheThirteenthLanguageResourcesandEvaluationConference, 2022. https://aclanthology.org/2022.lrec-1.767/

[8] A. Desai, "Gujarati handwritten numeral optical character reorganization through neural network.," Pattern recognition, vol. 43, no. 7, pp. 2582-2589, 2010. https://doi.org/10.1016/j.patcog.2010.01.008

[9] S. Antani and L. Agnihotri, "Gujarati character recognition," in Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR '99, Bangalore, India, 1999. https://doi.org/10.1109/ICDAR.1999.791813

[10] Tailor, C., Patel, B."Chunker for Gujarati Language Using Hybrid Approach," in Rising Threats in Expert Applications and Solutions. Advances in Intelligent Systems and Computing, 2021. https://doi.org/10.1007/978-981-15-6014-9_10

[11] K. Suba, D. Jiandani and P. Bhattacharyya, "Hybrid inflectional stemmer and rule-based derivational stemmer for gujarati.," in Proceedings of the 2nd workshop on south southeast Asian natural language processing (WSSANLP), 2011. https://aclanthology.org/W11-3001

[12] B. K. Y. Panchal and A. Shah, "Spell Checker Using Norvig Algorithm for Gujarati Language," in nternational Conference on Smart Data Intelligence. Singapore, Singapore, 2024. https://doi.org/10.1007/978-981-97-3191-6_21

[13] N. Patel and D. Patel, "Implementation Approach of Indian Language Gujarati Grammar's Concept “sandhi” using the Concepts of Rule-based NLP," in 8th International Conference on Computing for Sustainable Global Development (INDIACom)., 2021. https://doi.org/10.1109/INDIACom51348.2021.00085.

[14] J. Sheth and B. C. Patel., "Gujarati phonetics and Levenshtein based string similarity measure for Gujarati language.," in 5th National Conference on Indian Language Computing., 2015. https://www.researchgate.net/publication/314153559

[15] T. A. Gal. "Natural Language Processing (NLP) Pipeline." Medium, 23 Oct 2023. [Online]. Available: https://medium.com/@theaveragegal/natural-language-processing-nlp-pipeline-e766d832a1e5

[16] P. Patel, K. Popat and P. Bhattacharyya, "Hybrid stemmer for Gujarati," in Proceedings of the 1st Workshop on South and Southeast Asian Natural Language Processing, 2010. https://aclanthology.org/W10-3607

[17] M. Parikh and A. Desai, "Recognition of Handwritten Gujarati Conjuncts Using the Convolutional Neural Network Architectures: AlexNet, GoogLeNet, Inception V3, and ResNet50," in Advances in Computing and Data Sciences: 6th International Conference, ICACDS2022, Kurnool, India, 2022. https://doi.org/10.1007/978-3-031-12641-3_24.

[18] B. K. Y. Panchal and A. Shah, "NLP-Based Spellchecker and Grammar Checker for Indic Languages.," in Natural Language Processing for Software Engineering, Scrivener Publishing LLC, 2025, pp. 43-70. https://doi.org/10.1002/9781394272464.ch4

[19] C. Tailor and B. Patel, "Sentence Tokenization Using Statistical Unsupervised Machine LearningandRule-BasedApproachforRunningTextinGujaratiLanguage, " in Emerging Trends in Expert Applications andSecurity.AdvancesinIntelligent SystemsandComputing, 2018. https://doi.org/10.1007/978-981-13-2285-3_38

[20] S. Sooraj, K. Manjusha, M. A. Kumar and K. P. Soman, "Deep learning-based spell checker for Malayalam language," Journal of Intelligent & Fuzzy Systems, vol. 34, no. 3, pp. 1427-1434, 2018. https://doi.org/10.3233/JIFS-169438

[21] S. Murugan, T. A. Bakthavatchalam and M. Sankarasubbu, "Symspell and lstm based spell-checkers for tamil," in Tamil Internet Conference, 2020. https://www.researchgate.net/publication/3499249

[22] N. Hossain, M. H. Bijoy, S. Islam and S. Shatabda, "Panini: a transformer-based grammatical error correction method for Bangla," Neural Computing and Applications, vol. 36, pp. 3463-3477, 2024. https://doi.org/10.1007/s00521-023-09211-7

[23] R. Phukan, M. Neog and N. Baruah, "A Deep Learning Based Approach For Spelling Error Detection In The Assamese Language," in 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), Delhi, India, 2023. https://doi.org/10.1109/ICCCNT56998.2023.10306972

[24] S. S. Jamwal and P. Gupta., "A Novel Hybrid Approach for the Designing and Implementation of Dogri Spell Checker," in Data, Engineering and Applications: Select Proceedings of IDEA 2021, Singapore, 2022. https://doi.org/10.1007/978-981-19-4687-5_53

[25] S. Singh and S. Singh, "Systematic review of spell-checkers for highly inflectional languages," Artificial Intelligence Review, vol. 53, no. 6, pp. 4051-4092, 2020. https://doi.org/10.1007/s10462-019-09787-4

[26] M. Das, S. Borgohain, J. Gogoi and S. Nair, "Design and implementation of a spell checker for Assamese," in Language Engineering Conference, 2002. Proceedings, 2002. https://doi.org/10.1109/LEC.2002.1182303

[27] S. Iqbal, W. Anwar, U. I. Bajwa and Z. Rehman., "Urdu spell checking: Reverse edit distance approach," in In Proceedings of the 4th workshop on south and southeast asian natural language processing, 2013. https://aclanthology.org/W13-4707

[28] R. Sakuntharaj and S. Mahesan, "A novel hybrid approach to detect and correct spelling in Tamil text," in 2016 IEEE international conference on information and automation for sustainability (ICIAfS), 2016. https://doi.org/10.1109/ICIAFS.2016.7946522

[29] B. Bhagat and M. Dua, "Enhancing performance of end-to-end gujarati language asr using combination of integrated feature extraction and improved spell corrector algorithm," in ITM Web of Conferences, 2023. https://doi.org/10.1051/itmconf/20235401016

[30] D. Kakwani, A. Kunchukuttan, S. Golla, G. NC, A. Bhattacharyya, M. M. Khapra and P. Kumar., "IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages," In Findings of the association for computational linguistics: EMNLP 2020, pp. 4948-4961, 2020. https://doi.org/10.18653/v1/2020.findings-emnlp.445

[31] S. Khanuja, D. Bansal, S. Mehtani, S. Khosla, A. Dey, B. Gopalan, D. Margam, P. Aggarwal, R. Nagipogu, S. Dave and S. Gupta, "Muril: Multilingual representations for indian languages.," arXiv preprint arXiv:2103, p. 10730, 2021. https://doi.org/10.48550/arXiv.2103.10730

[32] Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer and V. Stoyanov., "Unsupervised cross-lingual representation learning at scale," arXiv preprint arXiv:1911.02116, 2019.

[33] A. Lawaye and B. S. Purkayastha, "KASHMIRI SPELL CHECKER AND SUGGESTION SYSTEM," THE COMMUNICATIONS, vol. 21, no. 2, p. 123, 2012. https://ddeku.edu.in/Files/2cfa4584-5afe-43ce-aa4b-ad936cc9d3be/Journal/6bb36225-ee44-4d4c-9d3d-0905436082e8.pdf

[34] Kaur and H. Singh, "Design and implementation of HINSPELL—Hindi spell checker using hybrid approach," International Journal of scientific research and management, vol. 3, no. 2, pp. 2058-2062, 2015. https://ijsrm.net/index.php/ijsrm/article/view/102

[35] R. Sankaravelayuthan, "Spell and grammar checker for Tamil.," Developing computing tools for Tamil, vol. 5, no. 23, pp. 52-64, 2015. https://doi.org/10.13140/RG.2.1.3700.6803

[36] A. Lawaye and B. S. Purkayastha, "Design and implementation of spell checker for Kashmiri," International Journal of Scientific Research, vol. 5, no. 7, 2016. https://www.researchgate.net/publication/321906322

[37] U. M. G. Rao, A. P. Kulkarni and a. P. K. Christopher Mala, "Telugu Spell-Checker," Vaagartha, 2012. https://sanskrit.uohyd.ac.in/faculty/amba/PUBLICATIONS/papers/ITIC-ss.pdf

[38] S. Saha, F. Tabassum, K. Saha, Akter. and Marjana, "Bangla Spell Checker and Suggestion Generator," (Dissertation, United International University), 2019. https://www.academia.edu/96829901/

[39] J. A. R. C. P. Pfeiffer, A. Kamath, I. Vulić, S. Ruder, K. Cho and I. Gurevych, "Adapterhub: A framework for adapting transformers," arXiv preprint arXiv:2007.07779, 2020. https://doi.org/10.48550/arXiv.2007.07779

[40] S. Deode, J. Gadre, A. Kajale, A. Joshi and R. Joshi, "L3Cube-IndicSBERT: A simple approach for learning cross-lingual sentence representations using multilingual BERT."," arXiv preprint arXiv:2304.11434, 2023. https://doi.org/10.48550/arXiv.2304.11434

[41] M. Nejja and A. Yousfi., "The context in automatic spell correction," Procedia Computer Science, vol. 73, pp. 109-114, 2015. https://doi.org/10.1016/j.procs.2015.12.055

[42] K. Ingason, S. B. Jóhannsson, E. Rögnvaldsson, H. Loftsson and S. Helgadóttir., "Context-sensitive spelling correction and rich morphology.," in Proceedings of the 17th Nordic Conference of Computational Linguistics (NODALIDA 2009), 2009. https://aclanthology.org/W09-4634.pdf

[43] Yunus and M. Masum., "A context free spell correction method using supervised machine learning algorithms," International Journal of Computer Applications, vol. 176, no. 27, pp. 36-41, 2020. https://doi.org/10.5120/ijca2020920288

[44] P. Gupta, "A context-sensitive real-time Spell Checker with language adaptability," in 2020 IEEE 14th International Conference on Semantic Computing (ICSC), 2020. https://doi.org/10.48550/arXiv.1910.11242

[45] Priya, M.C.S., Renuka, D.K., Kumar, L.A. et al. Multilingual low resource Indian language speech recognition and spell correction using Indic BERT. Sādhanā 47, 227 (2022). https://doi.org/10.1007/s12046-022-01973-5

[46] Parida, S. et al. (2022). BertOdia: BERT Pre-training for Low Resource Odia Language. In: Dehuri, S., Prasad Mishra, B.S., Mallick, P.K., Cho, SB. (eds) Biologically Inspired Techniques in Many Criteria Decision Making. Smart Innovation, Systems and Technologies, vol 271. Springer, Singapore. https://doi.org/10.1007/978-981-16-8739-6_32

[47] Dashti, S.M.S., Khatibi Bardsiri, A. & Jafari Shahbazzadeh, M. PERCORE: A Deep Learning-Based Framework for Persian Spelling Correction with Phonetic Analysis. Int J Comput Intell Syst 17, 114 (2024). https://doi.org/10.1007/s44196-024-00459-y