SPELLING ERROR DETECTION AND CORRECTION METHODS FOR INDIAN LANGUAGES - A STUDY

Abstract

Spelling error detection techniques play an important role for any language. While composing the written document, spelling errors appear. Newspapers, information retrieval, search engines, and other applications require user input. Languages such as English are extremely strong and can tolerate any form of spelling mistake. but efficient spelling checkers are not available for Indian language [1]. The authors conducted studies on the developing approaches and responsibilities of spell checkers in various applications based on Indian languages.

Keywords: N-gram, Information retrieval Text, Machine translation, Spell Checker, Error Detection, Error Correction.

Authors

Padmadhar Mishra
Research Scholar
(CSVTU, Bhilai)
Durg, India
padmadhar.mishra@bitdurg.ac

Jyothi Pillai
Professor
(BIT, Durg)
Durg, India
jyothi.pillai@bitdurg.ac.in

Ani Thomas
Professor
(BIT, Durg)
Durg, India
ani.thomas@bitdurg.ac.in

I. INTRODUCTION

Spelling correction is the process of substituting an erroneously spelt word with the most likely intended one in a document produced in any Natural Language. [18]. It has been a focus of ongoing study in the field of Natural Language Processing (NLP)[18]. The NLP group's mission is to create and construct software that will analyse, comprehend, and generate natural human languages. [15].Various applications are used in NLP like Text Summarization, Question Answering, Machine Translation, Parsing, Information Retrieval and Optical Recognition.

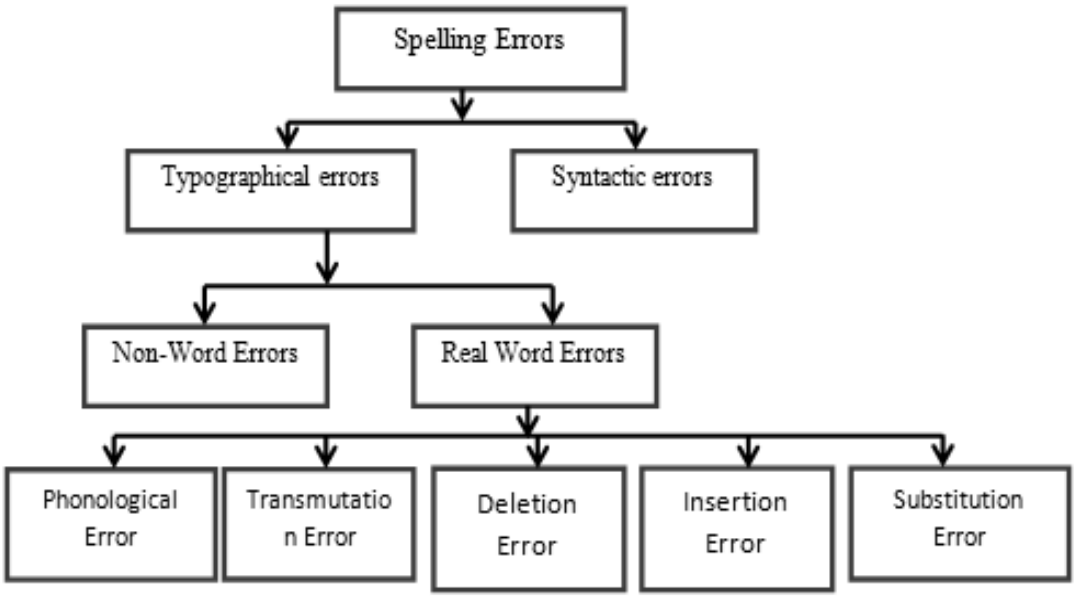
The basic aim of spell checking is to discover problems in written text. Dictionary lookup and N-gram analysis are two ways for error detection. The technique for detecting errors often entails determining whether or not an input string is a legitimate dictionary word. For detecting such errors, efficient approaches have been developed. Dictionary lookup and n-gram approaches are used by most spellcheckers. When a word in a written document is discovered as a mistake, spelling correction procedures are used to correct the word or provide right options. For text error correction, many techniques such as rule-based techniques, edit distance techniques, N-gram techniques, and deep learning techniques are available.

Therefore, study of automated word correction may be seen as focused on three progressively large topics for descriptive purposes: 1) Detection of non-word errors 2) Correction of isolated word errors and 3) Word correction based on the context. From the beginning of the 1970s to 1980s, work on the first issue listed. During that period, the majority of the work was devoted to investigating fast string comparison and pattern matching approaches for determining if a given The input string is found in a dictionary or vocabulary list [34]. From the 1970s to the present, more time was spent working on the second problem. At that time, a variety of general and specialized methods for correcting spelling errors were developed, some of which were used in combination with instances of spelling mistake patterns. In the early on 1980s, the creation of automatic NLP models started concern in the third challenge, and the development of statistical language models rekindled it. [34].

1. **Types of Spelling Errors;** Various strategies produced on the basis of spelling errors and trends, often known as error patterns, are summarised in this section [15]. The most important things are the studies carried out by various researchers. These studies classify spelling errors into two categories: Typographical and Syntactic errors.

2. **Typographical Errors:** Typographical errors are made by humans while writing text. This kind of error can be caused by typing carelessly, not knowing how to spell, pressing the wrong key accidentally or pressing the keys in the wrong order[11]. The typographic errors can further divide into two subcategories. The first type of error is known as a “non-word error” and it occurs when a string of letters lacks any meaning. For example, there may be 24 possible combinations for the letters L, I, O, and N, but the correct word is only “LION”. Apart from all 23 permutations will result in non-word error. Real-word error, on the other hand, result in a string of letters that are in the word dictionary but do not belong in the phrase. The terms "शंकर" and "संकर" have the same pronunciation. Both of these are appropriate terms, but if they are not used in the right context, they

might lead to mistakes. Five categories of real world are further subdivided. A) Phonological Error: when two words sound identical but have very distinct meanings.eg. "शंकर" and "संकर". B) Transmutation Error: When two consecutive letter positions are changed, a new word is created that is likewise grammatically correct. eg. "कलम" and "कमल". C) Deletion Error: These are the errors made when a letter is unintentionally erased.eg. "शैलजा" and "शैल". D) Insertion Error: This error was caused by accidentally adding one or more letters.eg. "पुत्र"and "कुपुत्र". E) Substitution Error: These kinds of mistakes are caused by replacing one or more letters with a different set of letters.eg. "बाग" and "राग".



3. **Syntactic Errors:** A Syntactic error is an error in utilizing a language that includes coordinating words and expressions that don't seem OK. Essentially, syntax reveals the structure and wording of a sentence, which is easy to misunderstand. For example,अध्यापकविद्यार्थीबुलाएbut the correct is अध्यापकनेविद्यार्थीकोबुलाए।

II. DETECTION OF NON-WORD TEXT ERRORS

Dictionary lookup and n-gram analysis are two main approaches investigated for the purpose of identifying non-word errors [12]. N-grams are subsets of words or strings formed up n letters, where n is typically one, two or three. Unigrams or monograms are names for one-letter n-grams. Bi-grams or di-grams are the names given to n-grams with two letters, furthermore Trigrams are n-grams that consist of three letters. In most cases, error detection in n-grams methods work by finding through an input string for each n-gram and checking to see if it exists or how frequently it appears in a pre-compiled n-gram statistics table. To preassemble an n-gram table using n-gram approaches, generally a lexicon or a large corpus of texts, are required. Dictionary lookup techniques simply determine whether an input text is present in a lexicon before working [34]. i.e. a collection of appropriate words. If not, a misspelt word alert is raised for the string. The creation of a helpful vocabulary for a spelling correction application involves subtle issues.

Spelling checkers have traditionally relied on dictionary search techniques, whereas systems that recognise texts have a tendency to use n-gram approaches for error detection [34]. Run-on or split words come from mistakes that cross word boundaries in both situations [34].

1. **N-gram Analysis Techniques:** Typically one of two text modes is the primary focus of text recognition systems. Handwritten text (sometimes known as cursive script) and text printed by a machine. Devices for optical character recognition (OCR) can process two modes [4]. OCR devices typically recognize individual characters within words using feature analysis. The count of a character's Features includes lines that are horizontal, curving, vertical and crossing. Due to the fact that these errors frequently produce implausible n-grams, n-gram analysis has proven effective for identifying them. Errors made by OCR devices typically involve Characters having similar properties, such as O and 0, S and 5, or t and f, are misread.[34].

   n-gram tables can take on various structures. The least difficulties are twofold bi-gram exhibit, this is a 26 X 26 two-layered cluster, whose components address all conceivable two letter blends of the letters in order [34]. Depending on whether the bi-gram appears in at least one word in a lexicon or dictionary that has been predetermined, each arrays element's value is set to either 0 or 1. The dimensions of a binary tri-gram arrays are three. The two aforementioned exhibitions are mentioned are alluded to as non-positional two-fold n-gram clusters since they do not display the n-gram's location within a word. A large portion of the dictionary's creation might be collected by a set of positional paired n-gram clusters. For instance, the i, j and kth elements of a positional binary trigram array would be set to 1 if and only if there is at least one term in the lexicon with the letters l, m, and n in locations z, j, and k. The increased storage necessary to reflect more of the lexicon's structure, however, comes at the price of this trade-off. The amount of space necessary for the full collection of arrays with positions. By looking up its related term, the word may be checked for mistakes entries in binary n-gram arrays and verifying that they are all one-digit integers. Make a trigram frequency table based on the text in order to check for spelling problems, Shree Devi and Srinivasa [2017]. Based on the word's trigram frequencies, they calculate an index of oddity for each distinct word in the text. The words are then arranged in decreasing order of strangeness. [3]. They predict that terms with typos will be closer to the top of the list. They note that one author of a 108-page article only needed ten minutes to examine the output list and spot misspelt terms, and that 23 of the 30 misspelt words in the text were in the first 100 words of the list. [4].

2. **Search Methods for Dictionaries:** A dictionary search is a straightforward procedure. In any case, reaction time turns into an issue when word reference size surpasses two or three hundred words. In archive handling and data recovery, the quantity of word reference passages can go from 25,000 to in excess of 250,000 words. This issue has been handled in three ways, by means of productive word reference query and additionally design matching calculations, through word reference dividing plans, and by means of morphological-handling procedures [34]. The most well-known method for acquiring quick admittance to a word reference for the utilization of a hash table [Knuth 1992]. To analyse a string of information, just locate its hash address and retrieve the word kept there in the pre-built hash table.[20]. In some cases, a chance of crash may happen during

development of the hash table. An incorrect spelling is displayed if the word stored at the hash position does not match the info string or is invalid [20].

3. **Dictionary creation challenges:** A lexicon must be carefully adjusted to the target area of conversation for a spell checker or Text detection programme. Unacceptably high numbers of false acceptances, or real errors that were missed because they occurred to be considered valid low-frequency or additional domain terms (e.g.fen, lave, veery etc.) can result from lexicons that are too small or too large, respectively. However, there are certain complications in the link between word frequencies and misspellings [21].

   However, Maysand Dameraudisagree with this advice [21]. Using a corpus of more than 22 million words of text from various genres, they observed that by increasing the size of their frequency rank-ordered word list from 50,000 to 60,000 words, they were able to eliminate 1,348 erroneous rejections while incurring just 23 extra incorrect acceptances. [20]. They recommend using larger lexicons since the 50-to-1 difference in text error rate suggests a momentous gain in corrected accuracy.

4. **The Issue of Word Boundaries:** White space characters define word boundaries (such as carriage returns, blanks, tabs, etc.) for almost all spelling mistake detection and correction methods [20]. This assumption seems to be incorrect since a large percentage of text mistakes include separating a one word (e.g. spent thebook) or putting two or more words together, sometimes with inherent faults (e.g. ofthe, understandhme). In a corpus of 40,000 words of typed textual conversations, Kukich [20] discovered that 15% of all non-wordspelling errors were of this type (i.e., 2% were split words and13% were run-on words), Mitton [22] observed that run-ons and splits typically produce at lowest one legitimate word (e.g. in form → inform and forgot → forgot)as a result, one or both of these errors might occur [22].Although certain spelling error correction programmes make an explicit effort to address some run-on and split-word problems, no spelling error detection programme differentiates between word boundary breaches and other errors.

5. **Rule-based Techniques:** These methods use morphology-based heuristics, part of speech, proper, etc., which have other properties the word that does the spell check. Monisha Das et al.[24] Assamese language designed speller based on morphological and lexical search approach to find and correct errors [24]. Dhanabalan et al.[21] recommended Tamil spell check using morphological analysis for error detection and correction [25]. Many other changes to the spelling of languages were later proposed using morphological analysis. Secondly Fossati et al [20] suggested a technique for spell checking that is based on rules. where they suggested using a part of speech (POS) tag for English spell check. Aside from these few texts, the Hidden Markov model was used to increase spell check efficiency [28]. The most significant disadvantage Such approaches need a variety of heuristic principles as well as information particular to the language in concern.

6. **A Techniques Based on Statistical Analysis:** A specific language is not necessary to use statistical procedures. Spell-checking techniques such as finite state automata-based spell-checkers,n-gram-based and frequency-based work on word counts and word properties. Abdullah et al[28] identified and corrected spelling errors. based on the finite-state representation (FSR) and state table technique, developed a spell-checker for Bengali

Naseem et al. [29] presented an Urdu spell-checker that used the word-frequency and edit-distance methods. Iqbal et al. suggested another Urdu spell-checker that integrates the finite state automata (FSA) and reverse edit-distance method (REDM).Manohar et al[30] employed finite-state automata. to spell-check Malayalam. Additionally, P. H. Hema et al [32]. designed a Malayalam spelling checker utilising the N-gram and minimal edit distance methods [31]. The statistical method has the benefit of considerably improving performance while requiring minimal knowledge of the specific language. These systems have a fault in that they spell-check using characteristics, frequency, word counts, etc., when certain spelling errors need knowledge of the target language. Many academics used rule-based and statistical approaches to solve this challenge. To get around the problems, a hybrid model combines statistical and rule-based approaches [2].

7. **Deep-learning-based techniques:** Although statistical and rule-based approaches to spell-checking are effective, deep learning (DL) approaches have the potential to improve performance even further. These deep-learning techniques excel in detecting real-word mistakes, which depend on the word's context in relation to the sentence. The first researchers to employ deep learning techniques for mistake correction were Ghosh and Kristensson [35]. They proposed an English text-correction approach. While this was going on, word recognition and spelling correction capabilities of the semi character recurrent neural network (SCRNN) were investigated by Keisuke Sakaguchi et al [31].The trials showed that the SCRNN works better than several other current spelling checkers. Language processing using deep learning is still an emerging area of study. When it comes to regional languages, the only language for which the deep-learning-based spell-checker is available is Malayala M Sooraj et al [32]. developed a Malayalam spell-checker using an LSTM network. The network of this spell-checker has been taught to recognise spelling problems and pinpoint their locations.

III. RESEARCH ON ISOLATED WORD ERROR CORRECTION

It may be adequate for certain apps to only identify text problems, but this is not the case for the majority of them. For instance, output mistakes need to be found and fixed as text recognition software aims to faithfully copy input text[34]. Similarly, people increasingly a spellchecker is expected to give corrections for any non-words they discover. In fact some applications for spelled correction, including speech synthesis using text, require that error being recognised and fixed automatically. To solve the issue of fixing words in text, many isolated-word mistake correction algorithms have been created.

The design of isolated-text error correctors is subject to a variety of limitations depending on the features of the application in question, and Several efficient rectification methods have been developed by taking use of these constraints and qualities. Before getting into the specifics of each approach, it is worthwhile to analyse a few isolated-word mistake repair applications and their features.

The three main challenges that affect the majority of application-specific design considerations are (1) lexical concerns (2) computer human interface issues and (3) spelling mistake pattern issues. Issues with lexicons include topics like lexical size and coverage, rates of new word entry, and if morphological processing, such as handling of affixes, is required. These were covered in the section on problems with dictionary construction that came before.

Table1: Some Isolated-Word Spelling Correction MethodsAccuracy

Methods	521 -Word Lexicon	1142-Word Lexicon	1872-Word Lexicon
Levenshtein distance	**64%**	**62%**	**60%**
Similarity Key	**80%**	**78%**	**75%**
Simple N-gram Vector Distance • scalar product • Hamming space • Cosine Distance	**58%** **69%** **76%**	**54%** **68%** **75%**	**52%** **67%** **74%**
SVD N-gram Vector Distance	**81%**	**76%**	**74%**
Probabilistic	**-**	**78%**	**-**
Neural Net	**75%**	**75%**	**-**

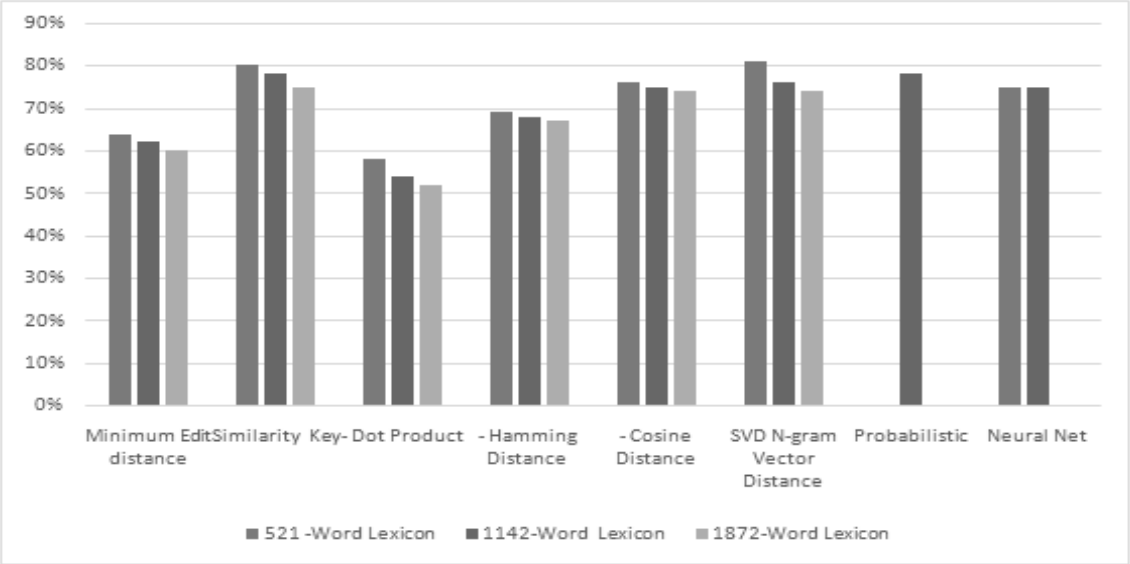


Figure 1: Some Isolated-Word Spelling Correction Methods Accuracy

IV. RESEARCH ON CONTEXT-DEPENDENT TEXT CORRECTIONS

There will always be a subset of faults that isolated-word error correction methods cannot handle, despite the advancements achieved in the field. When one correctly spelt word is used in place of another, it falls under the category of real-word mistakes. Example of real-word error (कलमकीचड़मेंखिलताहै|)is an acceptable word "कलम" in this statement, although it was not meant. This sentence's correct word is"कमल" which is both acceptable and intended.

All of these problems appear to require contextual information to be detected and corrected. Contextual information might be beneficial for enhancing the detection of non-word errors during repair. All of these problems appear to require contextual information to be detected and corrected. Additionally, contextual information would help to increase the

accuracy of non-word error correction. All of these problems appear to require contextual information to be detected and corrected. Additionally, contextual information would help to increase the accuracy of non-word error correction. All of these problems appear to require contextual information to be detected and corrected. Additionally, contextual information would help to increase the accuracy of non-word error correction.

Constructing context-sensitive text correction tools has remained a difficult task. This is mostly because of the problem's seeming intractability, which seems to demand fully developed natural language processing (NLP) skills, such as robust natural language parsing, semantic comprehension, pragmatic modelling, and discourse structure modelling. Successful NLP systems up to this point have been limited to a small number of discourse domains, and while some of these systems have addressed the need to handle input that is not well-formed, none of them were intended for widespread application. However, recent developments in robust syntactic parsing have resulted in the creation of at least two broad writing assistance programs that can identify and fix mistakes brought on by a few syntactic-constraint violations. Additionally, improvements in statistical language modelling and a consistent rise in.

Review of context-dependent word correction research's historical development. There are four subsections in it. The first examines research on real-word mistakes' frequency and classification. The second examines prototype NLP systems that deal with the issue of handling improperly formatted input, including two tools for writing assistance that are based on syntactic rules and contain grammar and spelling checks. The third discusses current research on the use of statistical language models for text-dependent spelling error detection and repair. The final gives a summary of the spelling correction job that is depending on context.

Table 2: Spelling Error Distribution(from Atwell & Elliott)

Text Source	Total no of errors	% Non-Word Errors	% Local Syntactic Errors	% Global Syntactic Errors	% Semantic Errors
Published Texts	50	52%	28%	8%	12%
Student Essay	50	36%	38%	16%	10%
Non-Native Text	50	4%	48%	12%	36%

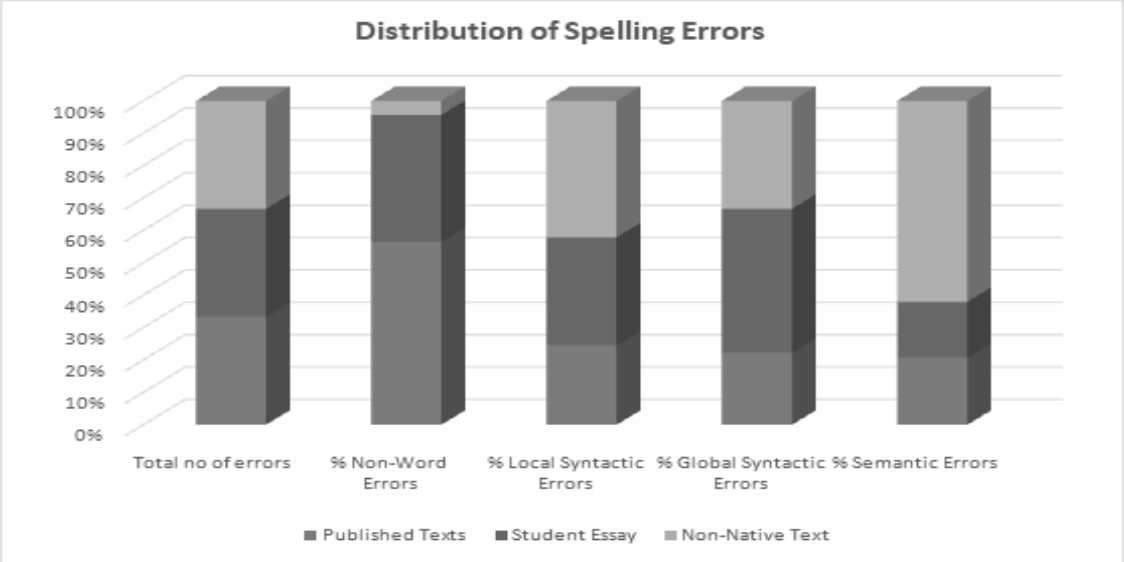


Figure 2: Distribution of Spelling Errors

Although the prevalence of errors in various text forms vary greatly. (Table 2)[20]; yet, they do imply that a substantial portion of mistakes may be recognised as local syntactic violations. Atwell and Elliott developed a prototype error-detection system.

V. CONCLUSION

In the review paper, the author suggested methods for identifying and fixing Indian language non-word spelling mistakes. Here, it is examined why approaches for English error detection and repair couldn't be used directly for Indian language. In addition to Hindi vowels and consonants, it also includes several types of symbols like the "vowel sign" "matrass," half letters, and halant, among others. The research can also be directed towards developing spell-checkers that use a tiny amount of data while still performing well. Further study is required to discover grammatical errors in Indian Languages data using deep-learning.

REFERENCES

[1] Singh, S., & Singh, S. (2021). HINDIA: a deep-learning-based model for spell-checking of Hindi language. Neural Computing and Applications, 33(8), 3825-3840.
[2] Caryappa, B. C., Hulipalled, V. R., & Simha, J. B. (2020, October). Kannada Grammar Checker Using LSTM Neural Network. In 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE) (pp. 332-337). IEEE.
[3] Etoori, P., Chinnakotla, M., & Mamidi, R. (2018, July). Automatic spelling correction for resource-scarce languages using deep learning. In Proceedings of ACL 2018, Student Research Workshop (pp. 146-152).
[4] Bopche, L., Dhopavkar, G., & Kshirsagar, M. (2011, December). Grammar checking system using rule based morphological process for an indian language. In International Conference on Computing and Communication Systems (pp. 524-531). Springer, Berlin, Heidelberg.
[5] Islam, S., Sarkar, M. F., Hussain, T., Hasan, M. M., Farid, D. M., & Shatabda, S. (2018, December). Bangla sentence correction using deep neural network based sequence to sequence learning. In 2018 21st International Conference of Computer and Information Technology (ICCIT) (pp. 1-6). IEEE.
[6] Islam, S., Sarkar, M. F., Hussain, T., Hasan, M. M., Farid, D. M., & Shatabda, S. (2018, December). Bangla sentence correction using deep neural network based sequence to sequence learning. In 2018 21st International Conference of Computer and Information Technology (ICCIT) (pp. 1-6). IEEE.

[7] Dereza, O. (2018, October). Lemmatization for Ancient Languages: Rules or Neural Networks?. In Conference on Artificial Intelligence and Natural Language (pp. 35-47). Springer, Cham.
[8] Li, H., Wang, Y., Liu, X., Sheng, Z., & Wei, S. (2018). Spelling error correction using a nested rnn model and pseudo training data.
[9] Hu, Y., Jing, X., Ko, Y., & Rayz, J. T. (2020, September). Misspelling Correction with Pre-trained Contextual Language Model. In 2020 IEEE 19th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC) (pp. 144-149). IEEE.
[10] EL Atawy, S. M., & Ahmed, H. M. (2021). Spelling Checker for Dyslexic Second Language Arab Learners. Journal of Theoretical and Applied Information Technology, 99(2).
[11] Singh, S., & Singh, S. (2018, March). Review of real-word error detection and correction methods in text documents. In 2018 second international conference on electronics, communication and aerospace technology (ICECA) (pp. 1076-1081). IEEE.
[12] Jain, A., Jain, M., Jain, G., & Tayal, D. K. (2018). "UTTAM" An Efficient Spelling Correction System for Hindi Language Based on Supervised Learning. ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), 18(1), 1-26.
[13] Kumar, R., Bala, M., & Sourabh, K. (2018). A study of spell checking techniques for indian languages. JK Research Journal in Mathematics and Computer Sciences, 1(1).
[14] Etoori, P., Chinnakotla, M., & Mamidi, R. (2018, July). Automatic spelling correction for resource-scarce languages using deep learning. In Proceedings of ACL 2018, Student Research Workshop (pp. 146-152).
[15] Jain, A., & Jain, M. (2014, September). Detection and correction of non word spelling errors in Hindi language. In 2014 International Conference on Data Mining and Intelligent Computing (ICDMIC) (pp. 1-5). IEEE.
[16] Jain, N., Singla, K., Tammewar, A., & Jain, S. (2012, December). Two-stage approach for hindi dependency parsing using maltparser. In Proceedings of the Workshop on Machine Translation and Parsing in Indian Languages (pp. 163-170).
[17] Shalini, M., & Indira, B. (2017). Implementation of Hindi Word Recognitionand Classification System Using Artificial Neural Network. International Journal of Pure and Applied Mathematics, 117(15), 557-565.
[18] Faili, H., Ehsan, N., Montazery, M., & Pilehvar, M. T. (2016). Vafa spell-checker for detecting spelling, grammatical, and real-word errors of Persian language. Digital Scholarship in the Humanities, 31(1), 95-117.
[19] Gupta, P. (2020, February). A context-sensitive real-time Spell Checker with language adaptability. In 2020 IEEE 14th International Conference on Semantic Computing (ICSC) (pp. 116-122). IEEE.
[20] Kukich, K. (1992). Techniques for automatically correcting words in text. ACM computing surveys (CSUR), 24(4), 377-439.
[21] Damerau, F. J., & Mays, E. (1989). An examination of undetected typing errors. Information Processing & Management, 25(6), 659-664.
[22] Mitton, R. (1987). Spelling checkers, spelling correctors and the misspellings of poor spellers. Information processing & management, 23(5), 495-505.
[23] Atwell, E., & Elliott, S. (1987). Dealing with ill-formed English text, The Computational Analysis of English. A Corpus-Based Approach, 12, 120-138.
[24] Das M, Borgohain S, Gogoi J, Nair SB (2002) Design and implementation of a spell checker for assamese. In: Language engineering conference, proceedings IEEE, pp 156–162.
[25] Dhanabalan T, Parthasarathi R, Geetha TV (2003) Tamil spellchecker. In: Sixth tamil internet conference, Chennai, Tamilnadu,India, pp 18–27.
[26] Fossati D, Di Eugenio B (2007) I Saw TREE trees in the park : how to correct real-word spelling mistakes. In: LREC, pp 896–901 Jain U, Kaur J (2015) Text chunker for Punjabi. Int J Curr Eng Technol 5(5):3349–3353.
[27] Abdullah M, Islam Z, Khan M (2007) Error-tolerant finite-state recognizer and string pattern similarity based spelling-checker for Bangla. In: Proceeding of 5th international conference on natural language processing (ICON).
[28] Naseem T, Hussain S (2007) A Novel approach for ranking spelling error corrections for Urdu. Lang Resour Eval 41(2):117–128.
[29] Manohar N, Lekshmipriya PT, Jayan V, Bhadran VK (2015) Spellchecker for Malayalam using finite state transition models. In: IEEE recent advances in intelligent computational systems, RAICS 2015, pp 157–161.

[30] Sakaguchi K, Duh K, Post M, Van Durme B (2017) Robsut wrod reocginiton via semi-character recurrent neural network. In: Thirty-first AAAI conference on artificial intelligence, pp 3281–3287.
[31] Sooraj S, Manjusha K, Anand Kumar M, Soman KP (2018) Deep learning based spell checker for malayalam language. J Intell Fuzzy Syst 34(3):1427–1434
[32] [33]. Gumaei A, Hassan MM, Alelaiwi A, Alsalman H (2019) A hybrid deep learning model for human activity recognition using multimodal body sensing data. IEEE Access 7:99152–99160.
[33] Kukich, K. (1992). Techniques for automatically correcting words in text. ACM computing surveys (CSUR), 24(4), 377-439.
[34] Ghosh, S., & Kristensson, P. O. (2017). Neural networks for text correction and completion in keyboard decoding. arXiv preprint arXiv:1709.06429.

