International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 17s, 2025
https://theaspd.com/index.php

A Probabilistic Character Computing Approach For Telugu Ocr Post-Processing

Dumpal Koteswararao[1], Dr. Nagaratna P Hedge[2]
[1]Research Scholar, Department of CSE, Osmania University.
[2]Professor, Department of CSE, Vasavi College of Engineering
[1]dumpal.koteswararao@gmail.com, [2]nagaratnaph@staff.vce.ac.in

Abstract:-
Recent innovative technologies are highly attracted by users and reduce complicated tasks most efficiently. Conversion of digital Image documentation to editable text conversion becomes an essential tendency. OCR (Optical Character Recognition) was utilized to convert images into text. In the process of transformation, accuracy plays a vital role. In a language like Telegu, OCR fails accuracy to generate a correct word. To improve the accuracy of a word by detecting and correcting the errors, many schemes support the N-Gram model but, restricted to unigram, bigram, and trigram words. Thus, the paper proposes a novel approach called Probabilistic character computing, which computes each character of a word efficiently.
Keywords: *- OCR, Probabilistic character computing, unigram, N-gram*

1. INTRODUCTION

Optical Character Recognition is a popular term for extracting text from the images where the image includes any language like English, Hindi, Urdu, Telegu, etc. Extracting South Indian document images is still a problem because many errors occur, such as character segmentation and post-processing steps involved in OCR to improve tshe accuracy. Telugu is one of the famous South Indian Languages. In Telugu, each character is a combination of Vowels (V) and Consonants (C). To extract Telugu document images into editable text, the OCR model follows different steps. These steps involved pre-processing, which is used for noise reduction, and improve the picture quality, segmentation for selecting regions occupied by each character, classification for comparing features and giving related information, and post-processing to improve accuracy. Since Telugu characters are combinations of any vowels and consonants, It is impossible to store all the possible sequences in one place.

Nowadays, due to advanced technologies, the habit of converting text documents into scanned images is increasing. By using The OCR engine again, these scanned images are converting into editable text documents. In this post-processing of OCR plays a vital role; based on this only, the accuracy of editable papers varies. Post-processing of English documents is very easy because of the involvement of only 26 characters, and all these characters are unique symbols. But, in Telugu, there is an ambiguity due to different symbols participation. Rules for Telugu characters came from the ancient writing system called Brahmi script. All the structural characteristics specified in Indic scripts. The complexity of the post-processing stage depends upon the structure of each character or group of characters.

Error detection and correction play an essential task in the post-processing OCR engine— spell checking is the primary goal used to find and flag misspelled words. Corpus database, the structure of languages, and OCR intermediate stages must be available accurately; any disturbance causes errors in recovering text in the final step of OCR. After spell-checking the document, the spell-correcting, swap the misspelled word or character with the suitable word or character. For spell correcting, there are many applications available, like word processing tools and word alternative tools.

The available word errors and non-available word errors are part of spelling errors. Suitable dictionary words used instead of available wrong spelled words. Many times, finding non-available words in the corpus dictionary database was difficult.

Typographic errors, cognitive errors, and phonetic errors are the different non-available word errors. Typographic errors cause due to typing mistakes. (example: “పరుగు” – “పరుగ”), due to misunderstandings, cognitive errors may cause (example: “వాడు బాగా పెరుగు తింటాడు” – “వాడు బాగా పరుగు తింటాడు”) and phonetic errors cause by the wrong pronunciation (example: “బలం” – “

భలం"). Section II presents the Literature survey, the proposed work explained in section III, section IV result analysis discussed, and finally, section V concludes.

2. LITERATURE REVIEW

Many algorithms are available for spell-checking systems—the latest techniques developed for Indic scripts discussed in this section. Dictionary lookups and Statistical methods used for spell checking and correction for the Punjabi language discussed by Baljeet Kaur [1]. J. Bharathi, P. Chandrasekar Reddy [2] proposed combining script-level properties and structural properties to identify partial touching characters. Unicode approximation Model (UAM) introduced by N Shobha Rani [3]; using UAM, segmentation and pre-processing errors were solved and achieved 96% accuracy. Grigori Sidorov [4] used Tree Edit Distance (TED) for computing text similarity and, by using edit mapping, swapped misspelled words with suitable words. To extract noisy Telugu script images, K Mohana Lakshmi [5] proposed a SURF descriptor method. Using the word spotting technique Nagasudha D [6] described a keyword substitute method for framing words in Telugu document images.

The detection and correction of errors in OCR were discussed by V S Vinitha [7] by using statistical language model (SLM) and dictionary-based methods. Instead of the Unicode form, the Akshara method produces good results. M Priya [8] proposed a "Hybrid optimization algorithm using N-gram based edit distance, to handle rule generation. This hybrid algorithm produced good results compared to the N-gram model and Edit distance model. By using "Segmentation Edit Distance (SED)" Daniel Pucher [9], measured the distance between two words.

3. Proposed Work

OCR approach generates the text from an image but fails to provide the correct linguistic word due to image quality. OCR generally recognizes incorrectly in two ways, 1) Available word 2) non-available word. If the OCR converts the image with the word 'curd' into the text as 'card,' card is a valid word, and this type of error is called an available word. If the OCR converts the image with the word "curd" into the text as "cird," which is not a valid word, this type of error is called a non-available word. Available word corrections are undetectable until unless have the full context.

OCR alone unable to improve the performance, need additional support to compute misspellings. This research paper proposes a novel approach called Probabilistic character computing, which calculates each character of a word in an efficient way to correct the misspelling.

In the proposed research paper, we consider a Telegu dictionary to map the correctness of output word from OCR with dictionary word. The proposed novel approach designed to compute the probabilistic of each character of a Telegu word.

The architecture of a proposed novel approach shown in Figure 1. A novel approach requires an OCR f or image conversion into text. The output of OCR is comparing with the available Telegu dictionary for correctness. The comparison process done by using a predefined method using edit distance [15]. In Eng lish, string each character as a count of 1, whereas in Telugu, it counts based on the consonants and vow el combination present in the computed character. For example, the word in English, i.e., JNTU count = 4 in Telugu జేఎన్టీయూ count = 11 (జ ే ఎ న ్ ట ీ య ూ) but by viewing the count = 5.

This method computes efficiently whether the retrieved Telegu text is correct or not but for misspell correction does not work efficiently. To correct the misspell word and identify the incorrect character position requires additional support, i.e., Probabilistic Character Computing (PCC). This proposed method computes each character probability by assuming that the word exists in the Telugu dictionary.

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 17s, 2025
https://theaspd.com/index.php

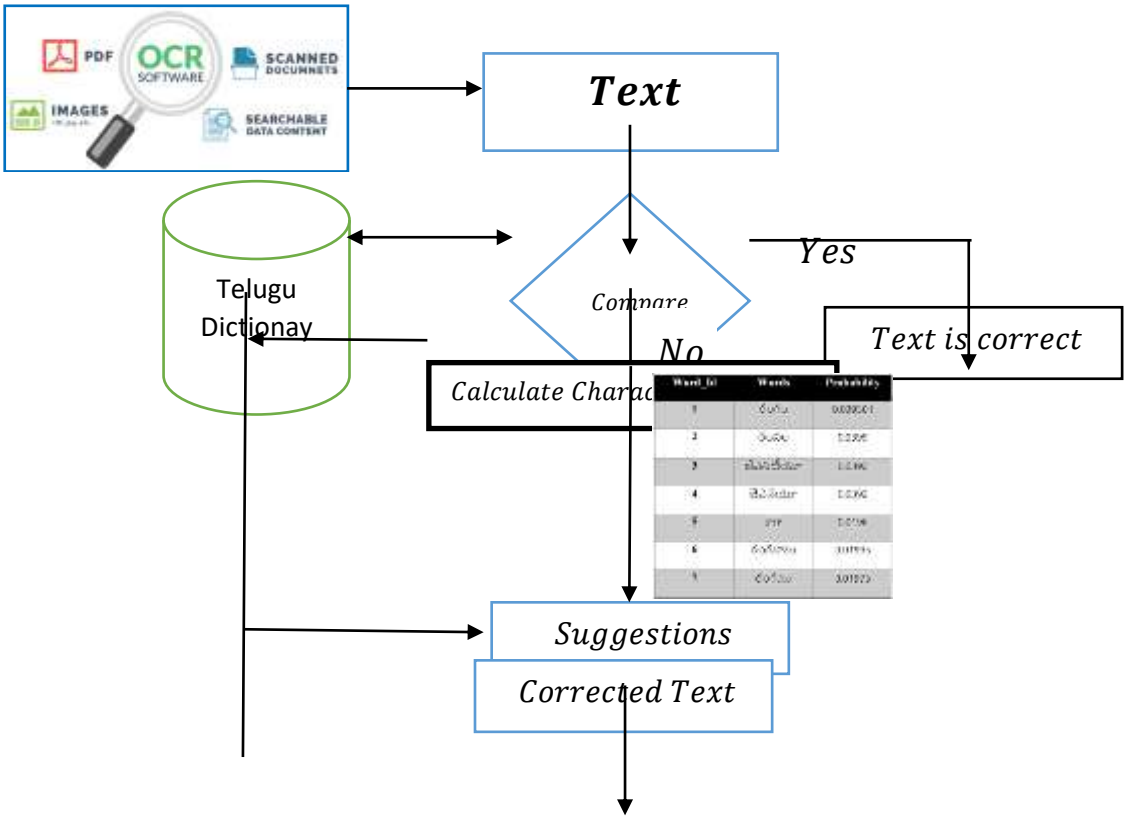

Figure.1. Efficient misspell computing architecture.

The figure 1, architecture explains the implementation process of a system, and if OCR fails to generate correct text, then the proposed scheme computes the probability of each Telugu character by using an equation,

$$P_{char}(C_m|C_{m-1}) = \frac{Count(C_{m-1}, C_m)}{Count(C_{m-1})} \quad \text{.....(1)}$$

C_m = Current calculating character, C_{m-1} = Previous computed character

If a character gets some value while computing the probability of a character, then concludes the character is in the correct position and available in the dictionary, if a character value is equal to 0 while the computing probability of a character then concludes that the character is not available in the dictionary and identifies the misspell position. This approach provides suggestions with appropriate characters based on position to replace with the misspelled character.

Algorithm: Misspell Correction using PCC

Input : $OCR\ output$
Output : $Probabilistic\ Character$
$OCR\ conversion = Words$
for w_i from w_1 to w_n
for character C_j from C_1 to C_n do
for $i = 1\ to\ n\ do$
Probability of $Words$ array $\boldsymbol{Pw}[P_w]$
for j$= 1\ to\ n\ do$
$P_{char}(C_m|C_{m-1})$ = Probability of characters of $Word\ w_i$
$P_{char}(C_m|C_{m-1}) = \frac{Count(C_{m-1},C_m)}{Count(C_{m-1})}$
end for
end for
Total Probability $Word$ = P_{word}
$P_{word} = \prod_{j=1}^{n} P_j(C_m|C_{m-1})$

$$Telugu\ Dictionar = P_{word}$$

end for
end for

Algorithm 1. Misspell Correction Algorithm using PCC

The total probability of Telugu words can be calculated as follows:

$P_{word} = \prod_{j=1}^{m} P_j(C_m|C_{m-1})$ ---------- (2)

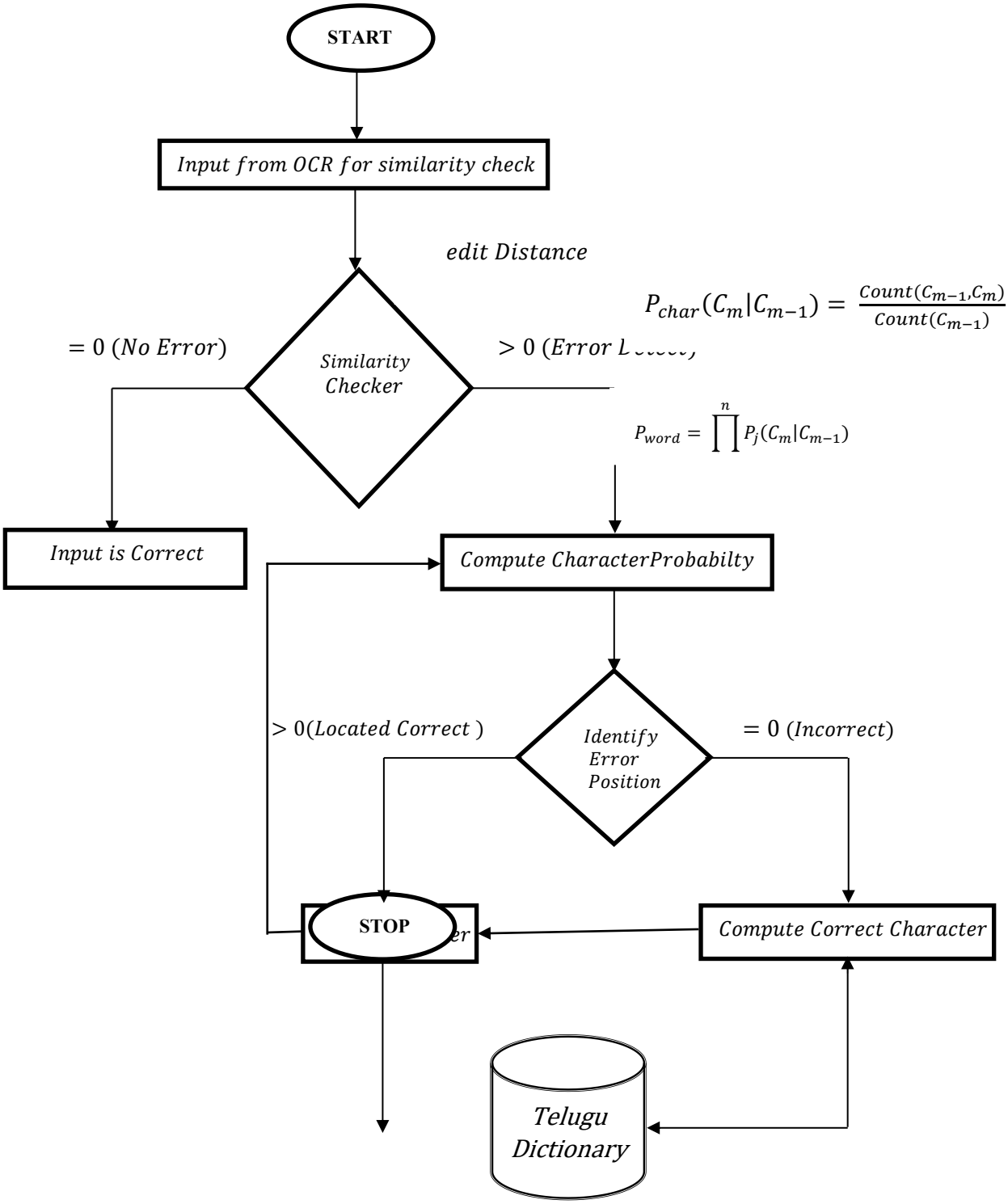

Figure.2. Workflow for Misspell Correction.

4. Performance Analysis

To compute the performance of a novel Probabilistic Character Computing approach by using Python language on a windows 10 server with core i5 CPU @ 1.80GHz, 64-bit operating system. For experiment, consider Telugu dictionary dataset to check similarity with the outcome of OCR. Provide OCR output to PCC along with other two existing methods Unicode approximation model (UAM), and N-Gram based edit distance (NED) for comparison. Spell checking test is performed on three algorithms PCC corrected 99% whereas UAM 96.20% and NED 94%.

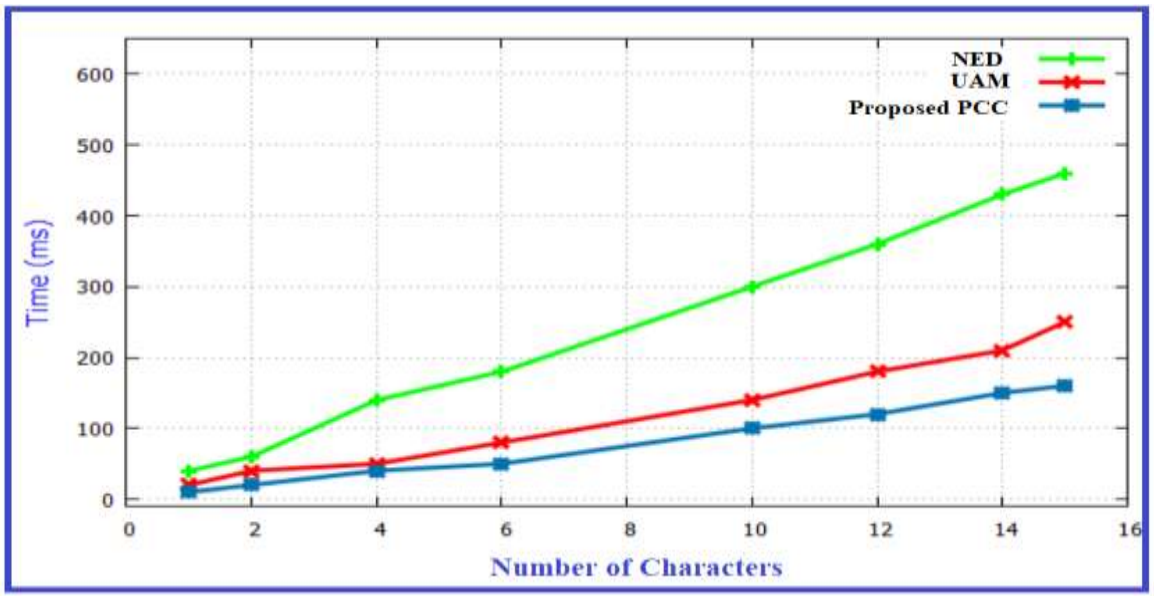

Fig.3. Computation time vs number of Characters

Figure 3 shows the comparison between the computation time and number of characters with in the predefined time. The result clearly indicates that the proposed work data retrieval time is lesser than the existing algorithms.

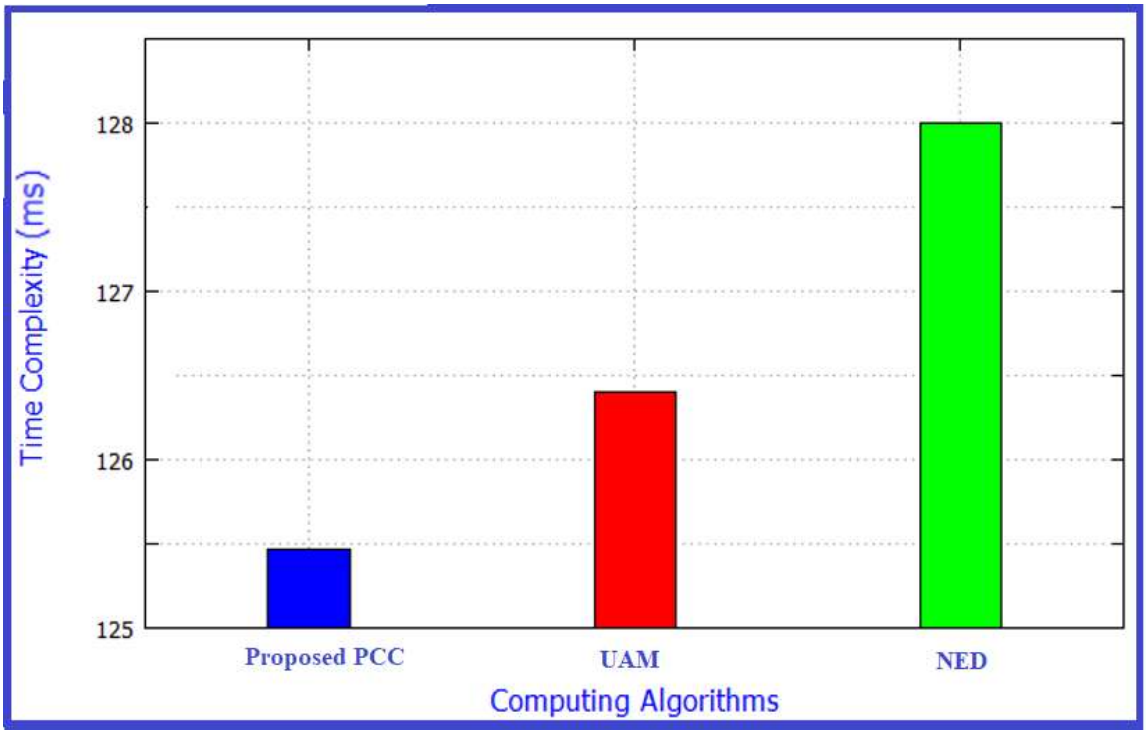

Fig.4. Average time complexity of the computing algorithms

Figure 4 shows the time require to correct the misspelled character. The result indicates that the proposed work data correct time is lesser than the existing algorithms. It is clear from the result that the proposed

work accuracy is better than the existing computing algorithms. The results indicate that the proposed scheme produces better results than the current methods of Computation Time.

5. CONCLUSION

In the present work, proposed PCC method for improving Telugu OCR. Enhanced edit distance method used to collect the number of spelling mistakes in each word by comparing with Telugu dictionary database. PCC method discussed to handle these misspelled words and achieved good results compared to other methods like UAM and NED. Sometimes, if any correct word is not available in the dictionary, the PCC method considers it a misspelled word. In further, the proposed work extended to dynamic updating of the database.

Conflict of interest

On behalf of all authors, the corresponding author states that there is no conflict of interest.

REFERENCES

[1] Kaur, Baljeet. "Review on error detection and error correction techniques in NLP." Int. J. Adv. Res. Comput. Sci. Software Eng 4 (2014): 851-853.
[2] Bharathi, J., and P. Chandrasekar Reddy. "Segmentation of Touching Conjunct Consonants in Telugu using Minimum Area Bounding Boxes." Int. J. Soft Comput. Eng 3.3 (2013): 260-264.
[3] Rani, N. Shobha, and T. Vasudev. "Post-processing methodology for word level Telugu character recognition systems using Unicode Approximation Models." 2015 International Conference on Trends in Automation, Communications and Computing Technology (I-TACT-15). IEEE, 2015.
[4] Sidorov, Grigori, et al. "Computing text similarity using tree edit distance." 2015 Annual Conference of the North American Fuzzy Information Processing Society (NAFIPS) held jointly with 2015 5th World Conference on Soft Computing (WConSC). IEEE, 2015.
[5] Lakshmi, K. Mohana, and T. Ranga Babu. "Searching for Telugu script in noisy images using SURF descriptors." 2016 IEEE 6th International Conference on Advanced Computing (IACC). IEEE, 2016.
[6] Lakshmi, K. Mohana, and T. Ranga Babu. "A new hybrid algorithm for Telugu word retrieval and recognition." nternational Journal of Intelligent Engineering and Systems 11.4 (2018).
[7] Vinitha, V. S. Error detection and correction in Indic OCRs. Diss. International Institute of Information Technology Hyderabad, 2017.
[8] Priya, M., R. Kalpana, and T. Srisupriya. "Hybrid optimization algorithm using N-gram based edit distance." 2017 International Conference on Communication and Signal Processing (ICCSP). IEEE, 2017.
[9] Pucher, Daniel, and Walter G. Kropatsch. "Segmentation edit distance." 2018 24th International Conference on Pattern Recognition (ICPR). IEEE, 2018.
[10] Sindhu, D. V., and B. M. Sagar. "Dictionary based machine translation from Kannada to Telugu." IOP conference series: materials science and engineering. Vol. 225. No. 1. IOP Publishing, 2017.
[11] Chakravarthi, Bharathi Raja, et al. "Bilingual lexicon induction across orthographically-distinct under-resourced Dravidian languages." (2020).
[12] Soujanya, B., and T. Sitamahalakshmi. "Optimization with ADAM and RMSprop in Convolution neural Network (CNN): A Case study for Telugu Handwritten Characters." International Journal 8.9 (2020).
[13] Singh, Shashank, and Shailendra Singh. "HINDIA: a deep-learning-based model for spell-checking of Hindi language." Neural Computing and Applications 33.8 (2021): 3825-3840.
[14] Prasad, Palanati Durga, K. V. N. Sunitha, and B. Padmaja Rani. "Word N-gram based approach for word sense disambiguation in Telugu natural language processing." Int. J. Rec. Technol. Eng 7 (2019): 686-690.
[15] Pucher, Daniel, and Walter G. Kropatsch. "Segmentation edit distance." 2018 24th International Conference on Pattern Recognition (ICPR). IEEE, 2018.
[16] Madhuri, G., Modali NL Kashyap, and Atul Negi. "Telugu OCR using Dictionary Learning and Multi-Layer Perceptrons." 2019 International Conference on Computing, Power and Communication Technologies (GUCON). IEEE, 2019.
[17] Lakshmi, C. Vasantha, Sarika Singh, and C. Patvardhan. "Determination of optimal features database for OCR of printed Telugu text." 2015 39th National Systems Conference (NSC). IEEE, 2015.