DEEP LEARNING FRAMEWORK OF ABSTRACTIVE SUMMARIZATION BASED ON SEMANTIC ROLE LABELLING OF TELUGU TEXT

Aluri. Lakshmi
Research Scholar,
Department of Computer Science and Engineering
Adikavi Nannaya University
Andhra Pradesh, India.
ORCID ID: 0000-0003-3403-5175

Dr. D. Latha
Assistant Professor
Department of Computer Science and Engineering
Adikavi Nannaya University
Andhra Pradesh, India.
ORCID ID: 0009-0003-0063-1902

ABSTRACT

Telugu, being a widely spoken language, presents the need for effective text summarization techniques to enhance accessibility and information management. This study aims to develop an abstractive summarization model specifically tailored for Telugu language documents. The research focuses on exploring natural language processing techniques and deep learning approaches to generate concise and coherent summaries that capture the essence of the original content. This study uses the suggested Hunter Sail Fish Optimizer (HSFO), a hybrid optimisation technique, leading to an abstractive summary. The obtained document is now suitable for Semantic Role Labelling (SRL), where Predicate Argument Structures (PAS) are extracted using the Stanza tool. In addition to SRL, Wave-Hedges metrics are used to compute semantic similarity and provide optimised features. Additionally, Bayesian Fuzzy Clustering (BFC) is used to cluster the semantic features of PAS. The Long Short-Term Memory-Convolutional Neural Network (LSTM-CNN) performs abstractive summarization after generating the feature score using the HSFO for parameter selection. Here, Hunter-Prey Optimizer (HPO) and Sail Fish Optimizer (SFO) are combined to create HSFO. Telugu dataset was employed in this study, and a text document in sentence form was extracted from it. HSFO_LSTM-CNN performance is finally evaluated using four performance metrics: precision, recall, F-measure, and Rouge.

Keywords: *Semantic Role Labeling (SRL), Predicate Argument Structures (PAS), Bayesian Fuzzy Clustering (BFC), Hunter Sail Fish Optimizer (HSFO), Wave hedge Metrics.*

INTRODUCTION

Text summarization is crucial for Telugu language documents for several reasons. Firstly, it enhances accessibility by making information more readily available to a wider audience. By summarizing Telugu text, individuals who are not proficient in the language or non-Telugu speakers can still understand the main points and essence of the content. This promotes inclusivity and ensures that valuable information is not limited to a specific linguistic group.Summarizing Telugu text documents saves time for users. Rather than reading lengthy and detailed documents, individuals can quickly grasp the key ideas and important information through a concise summary [31].

This is especially beneficial in today's fast-paced society where time is a precious resource. By providing a condensed version of the text, summarization enables users to efficiently process and absorb the main content without extensive reading. Furthermore, text

summarization of Telugu documents aids in information organization and management. It helps researchers, scholars, and professionals to sift through vast amounts of information and identify relevant material more efficiently. Summaries act as valuable references that allow users to quickly revisit key points without having to go through the entire document again. This facilitates effective information retrieval and knowledge extraction from Telugu text resources.

Telugu text summarization contributes to language processing and natural language understanding research. By developing robust and accurate summarization models specifically for Telugu, researchers can advance the field and improve the overall quality of automated summarization techniques. This, in turn, benefits various natural language processing applications, such as machine translation, information retrieval, and content recommendation systems, enhancing the overall language technology ecosystem for Telugu language users.

One of the most challenging professions in NLP is ATS because of the difficulty of the input text. Recently, Deep Learning (DL) has become one of the most effective and promising methodologies. It is now used in many different fields, including image processing, computer vision (CV), natural language processing (NLP), and more. A few of the DL networks that are used in ATS are Deep Neural Networks (DNNs), Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and Graph Neural Networks (GNNs) [17].. This study is concerned with single document abstractive summarization utilising the DL model, LSTM-CNN. The dataset used in this work is the Telugu dataset, from which text documents in sentence form are obtained.

Main contribution of this paper is involved with:

- **Developmentof HSFO_LSTM-CNN for abstractive summarization:** Abstractive summarization is carried out with LSTM-CNN, where parameter selection for feature score generation is done with HSFO. This HSFO is formed by combination of HPO as well as SFO, where this combination is very supportive to resolve real-world issues like abstractive summarization.

Balance work involves, section 1 includes motivation, literature section and challenges of single document abstractive summarization. Section 2 comprises of proposed methodology which consists steps regarding semantic role labelling, wave hedge sentence similarity score, HSFO for parameter selection in feature score generation and LSTM-CNN for abstractive summarization. Section 3 involves expected output for the given Telugu sentence and concluded in section 4.

1. Motivation

The exponential growth in the amount of textual data made available online has created new difficulties for accurately and rapidly accessing information. By giving the main points of the text, summarization enables users to accomplish this aim while saving them time and effort. Manual summarizing is a process that is currently used, however it is

exceedingly expensive, time-consuming, and impractical. To address this issue and enable users to get the information they need right away, ATS approaches are being explored. This section brings out the literature assessment and challenges on single document abstractive summarization.

1.1. Literature assessment

Moratanch, N. and Chitrakala, S., [1] developed Joint Model of Predicate Sense Disambiguation and SRL (PSD + SRL) to capture semantic representation of text. This method worked well for creating an abstract summary with excellent clarity. However, this method was unable to be moulded into a domain-specific application, like a summarizer for medical records.

Khan, A., *et al.* [2] designed Argument Structure_ Genetic Algorithm_ SRL (AS_GA_SRL) for abstractive summarization of multi-documents. The summary generated by this method showed control over the structure and content of the summaries generated, and it was more similar to how humans produce a summary. Yet, this plan was unable to generate a better amount of precision.

Mohamed, M. and Oussalah, M., [3] used SRL-Explicit Semantic Analysis (SRL-ESA) for text summarization. The evaluation data size did not affect the performance of this approach because it was scalable. When compared to the approach to other summary tasks, such as opinion, product or service evaluation, and guided summarization, this strategy proved ineffective.

Sudha, D.N. and Latha, Y.M.,[4] enabled RNN for multi-document abstractive text summarization via semantic similarity matrix for Telugu language. The strategy was effective in eliminating repetition and handling lengthy text summaries. To boost generalizability, this technique should have included several kinds of multi-document datasets.

Gabriel, S., *et al.* [5] introduced Cooperative Generator – Discriminator Networks (Co-opNet) for discourse understanding and factual consistency in abstractive summarization. By using this method, created abstracts could become more abstract while yet retaining higher degrees of factual consistency. This strategy, meanwhile, occasionally favored copying from the introduction, losing the narrative structure in the process.

Mandal, S., *et al.* [6] used Cuckoo Search (CS) algorithm for single document text summarization. For text summarization, this algorithm had the best readability, coherence, and non-redundancy. However, this method did not take sentiment analysis into account while abstracting text to improve the summary.

Wang, Q. and Ren, J., [7] introduced Summary-aware attention for social media short text abstractive summarization. This technique effectively raised quality of summary, which increased the fluency and adequacy scores, but accuracy was improved by significant improvement over time.

Balachandran, V., *et al.* [8] designed StructSum framework for abstractive summarization. This technique reduced layout biases and increased the coverage of generated summaries. Nevertheless, this method did not look at how document structures affect language models that have already been trained.

1.2. Challenges

Challenges for single document abstractive summarization by existing methods based on SRL are described as follows,

- PSD+SRL in [1] was introduced for semantic oriented abstractive text summary, and it proved to be highly helpful for students who wanted to finish a book quickly. The technique does not, however, take into account using a voice recognition system to condense lengthy speeches.
- AS_GA_SRL [2] was used for multi-document abstractive summarization, but it was unable to combine the graph with SRL to create a semantic graph that significantly enhanced the summarization outcome.
- SRL-ESA in [3] failed to consider as guided summarization, which entails retrieving a summary answer to an event described in a user query, was the primary problem it encountered for generic single and multi-document summarization.
- Summary-aware attention was suggested in [7] for abstractive summarization of social media short texts, and the method was effective in increasing the weight of related content and decreasing the weight of noise. The method had a high computational cost, though, and it overlooked the possibility of skipping summary-aware attention in order to cut down on pointless calculation.
- Despite the fact that a variety of techniques have been put forth for single document abstractive summarization, these techniques are hampered by the absence of semantic representation of original text. This representation of original text will be appropriate since abstractive summarization necessitates in-depth text analysis.

2. Proposed Methodology

Main objective in research regarding text summarization is the abstractive summarization technique, which involves some kind of natural language generation and results in the final summary using new words that are not found in the vocabulary of the source data. The fact that there is inevitably overlap in information contained in various documents presents particular challenge for single document summarization by current methods. To address this problem, an efficient single document abstractive summarization is proposed, which is implemented in the following manner. At first, input Telugu text document comprising various sentences is acquired from database [23] and it is subjected to SRL, wherein Stanza tool [26] is used to extract PAS from the contents of input documents. Input sentence is taken for SRL that is carried out by Stanza tool [26]. Purpose of SRL [2] is to identify the syntactic components or arguments of a sentence in relation to the sentence predicates, as well as their semantic functions and supplementary arguments. Finding the semantic relationship that a predicate has with its participants or components is main aim of SRL. For extracting PAS

structure from sentences in the document collection, SRL is used, as abstractive summarization necessitates a more in-depth semantic examination of the text.

Then, semantic similarity or PAS is computed using Wave-Hedges [27] to compute the sentence similarity score for optimized feature generation. Hereafter, semantic feature clustering of PAS is performed using BFC [18]. After that, the feature score is generated based on optimized features. The features gained are length of PAS, PAS to PAS similarity, position of PAS, proper nouns, numerical data, number of nouns and verbs, and temporal features [2]. The final predicate selection using HSFO. Optimal solution is attained based on HSFO, formed by combining both HPO [19] and SFO [20]. HPO [19] is new population-based optimization algorithm that draws its inspiration from the behavior of prey species like deer and gazelle as well as predator animals like lions, leopards, and wolves. The key driving force for the development of this optimization algorithm was its distinctive properties, like pursuing prey outside of group and advancing prey in front of group towards the leader. The adaptive parameter lessens the harshness of prey and hunter movement during iterations, ensuring convergence of HPO algorithm. The SFO [20] optimization algorithm was inspired by a group of sailfish hunters. Sailfish are the fastest fish in the water, with top speeds exceeding 100 km/hr. They are quite capable of hunting and attacking. SFO can easily be used to address complex technical problems without requiring structural changes. Thus, HSFO is used for real world problems that help in resolving those problems in an easy way.

Step 1: Initialization

Initially, the population is set to $\overleftarrow{(N)} = \{\vec{N}_1, \vec{N}_2, \ldots, \vec{N}_n\}$ and its objective function is indicated as $(Ob) = \{Ob_1, Ob_2, \ldots, Ob_n\}$ for all population members. The position of each member in population is randomly produced by,

$$N_g = rand(1,h)*(U_b - L_b) + L_b \quad (1)$$

where, N_g is prey or position of hunter, h is number of variables, L_b is lower boundary, as well as U_b is upper boundary.

Step 2: Exploration and exploitation

To direct search agents to ideal position, a search means needs to be established and repeated numerous times. Exploration and exploitation are often the first two steps in the search process. Exploration is algorithm's propensity for very erratic behaviors causing solutions to alter frequently. Exploitation is process of decreasing random behaviors after promising regions is identified so that algorithm explore promising regions. This is illustrated in below formula,

$$N_{g,i}(q+1) = N_{g,i}(q) + 0.5\left[(2BCD_{pos(i)} - N_{g,i}(q)) + (2(1-B)C_{(i)} - N_{g,i}(q))\right] \quad (2)$$

where, $N(q)$ is current position of hunter, $N(q+1)$ is next position of hunter, D_{pos} is position of prey, l is mean of every position, as well as C is adaptive parameter. Here, D and C are evaluated by,

$$D = G_1 < B; ind = (D == 0) \tag{3}$$

$$C = G_2 \otimes ind + G_3 \otimes (\sim ind) \tag{4}$$

where, G_1, G_2, and G_3 are random variables ranging $(0,1)$, ind is index numbers of vector G_1, B is balance parameter among exploitation and exploration, which is indicated as,

$$B = 1 - iter\left(\frac{0.98}{Maxiter}\right) \tag{5}$$

where, $Max\,iter$ is maximal number of iterations. Here, distance of every search agent from mean position is indicated by,

$$l = \frac{1}{n}\sum_{g=1}^{n} \vec{N}_g \tag{6}$$

Moreover, search agent having maximal distance from the mean positions is indicated by below formula,

$$\vec{D}_{pos} = \vec{N}_g \,|\, g\ is\ index\ of\ Max(end)\ sort(F_{ec}) \tag{7}$$

Here,

$$F_{ec(g)} = \left(\sum_{i=1}^{k} \left(N_{g,i} - l_i\right)^2\right)^{\frac{1}{2}} \tag{8}$$

Step 3: Hunting scenario

Based on hunting state, when hunter takes prey, it dies and then, hunter moves to next prey. This is solved by decreasing mechanism, which is given as,

$$abest = round(B \times J) \tag{9}$$

where, J is count of search agents. Now, position of prey is formed as,

$$\vec{D}_{pos} = \vec{N}_g \,|\, g\ is\ sorted\ F_{ec}(abest) \tag{10}$$

Step 4: Best safe position

Optimal global position is best safe position and hunter choose another prey, giving the prey better chance of survival and hence prey position is updated as,

$$N_{g,i}(q+1) = M_{pos(i)} + BC\cos(2\pi G_4) \times (M_{pos(i)} - N_{g,i}(q)) \quad (11)$$

where, $N(q)$ is current prey position, $N(q+1)$ is next prey position, M_{pos} is optimal global position, C is adaptive parameter, B is balance parameter, and G_4 is random number ranging $[-1,1]$.

$$N_{g,i}(q+1) = M_{pos(i)} + BC\cos(2\pi G_4) M_{pos(i)} - BC\cos(2\pi G_4) N_{g,i}(q) \quad (12)$$

$$N_{g,i}(q+1) = M_{pos(i)}[1 + BC\cos(2\pi G_4)] - BC\cos(2\pi G_4) N_{g,i}(q) \quad (13)$$

The basic equation of SFO is indicated by,

$$N^{g}_{new\_S} \left(= d \times N^{g}_{elite\_SF} - N^{g}_{old_S} + A_p\right) \quad (14)$$

where, $N^{g}_{elite\_SF}$ is best position of elite sailfish, $N^{g}_{old_S}$ is sardine's current position, d is random numbers ranging 0 and 1, and A_p is sailfish attack power at every iteration.

Let, $N^{g}_{new\_S} = N_{g,i}(q+1)$, $N^{g}_{old_S} = N_{g,i}(q)$, $N^{g}_{elite\_SF} = N^{best}_{g,i}(q)$

By substituting the above considerations, equation (26) becomes,

$$N_{g,i}(q+1) = d \times \left(N^{best}_{g,i}(q) - N_{g,i}(q) + A_p\right) \quad (15)$$

$$N_{g,i}(q) == \frac{d \times N^{best}_{g,i}(q) - N_{g,i}(q+1) + A_p d}{d} \quad (16)$$

Substitute equation (28) in equation (25), forming hybridization of SFO in HPO,

$$N_{g,i}(q+1) = M_{pos(i)}[1 + BC\cos(2\pi G_4)] - BC\cos(2\pi G_4)\left\lfloor \frac{d \times N^{best}_{g,i}(q) - N_{g,i}(q+1) + A_p d}{d} \right\rfloor \quad (17)$$

$$N_{g,i}(q+1) + \frac{N_{g,i}(q+1)}{d} = M_{pos(i)}[1 + BC\cos(2\pi G_4)] - BC\cos(2\pi G_4)\left\lfloor \frac{d \times N^{best}_{g,i}(q) + A_p d}{d} \right\rfloor \quad (18)$$

$$\frac{(d+1)N_{g,i}(q+1)}{d} = \frac{dM_{pos(i)}[1 + BC\cos(2\pi G_4)] - BC\cos(2\pi G_4)\left(d \times N^{best}_{g,i}(q) + A_p d\right)}{d} \quad (19)$$

$$N_{g,i}(q+1) = \frac{dM_{pos(i)}[1 + BC\cos(2\pi G_4)] - BC\cos(2\pi G_4)\left(d \times N^{best}_{g,i}(q) + A_p d\right)}{(d+1)} \quad (20)$$

where, $M_{pos(i)}$ is optimal global position. This forms the basic equation of HSFO.

Step 5: Updated prey position

Next prey position is updated at global optimal various angles and radials, and thus performance of exploitation is increased. This is given as in below formula,

$$N_g(q+1) = \begin{cases} N_g(q) + 0.5\left[(2BCD_{pos} - N_g(q)) + (2(1-B)C - N_g(q))\right] & if\ G_5 < \nu, \ (a) \\ M_{pos} + BC\cos(2\pi G_4) \times (M_{pos} - N_g(q)) & else, \ (b) \end{cases} \quad (21)$$

where, ν is regulatory parameter=0.1.

Step 6: End

Iteration process is continued depending on fitness equation (12) and then termination is carried out until maximal solution is attained. This optimal solution forms best solution for abstractive summarization by HSFO. Algorithm 1 enumerates pseudo code of HSFO.

Algorithm 1. Pseudo code of HSFO

Sl. No.	Pseudo code ofHSFO
1	***Input:*** Maximal iteration $Max\,iter$ and n population
2	***Output:*** Optimum solution $N_{g,i}(q+1)$
3	**Start HSFO**
4	Initialization of population in random manner by Eq. (13)
5	Evaluate fitness function as in Eq. (12)
6	Find M_{pos}
7	Update B with Eq. (17)
8	Find C with Eq. (16)
9	**If** $G_5 < \nu$ **then**
10	Evaluate D_{pos} with Eq. (22)
11	Update position with Eq. (33a)
12	**Else**
13	Update position with Eq. (33b)
14	Find M_{pos}

15	**Hybridization of SFO in HPO,**
16	Basic equation of HSFO is given in Eq. (32)
17	Reevaluate by fitness as per Eq. (12)
18	Find best solution
19	**Terminate HSFO**

Finally, abstractive summarization is carried out using LSTM-CNN [21] [30]. Abstractive text summarization is process of creating summary sentences using combining information from several source sentences as well as compressing it to more concise representation when maintaining the material's overall meaning. This is done based on LSTM-CNN. Figure 1 depicted the graphical user interface for abstractive text summarization.

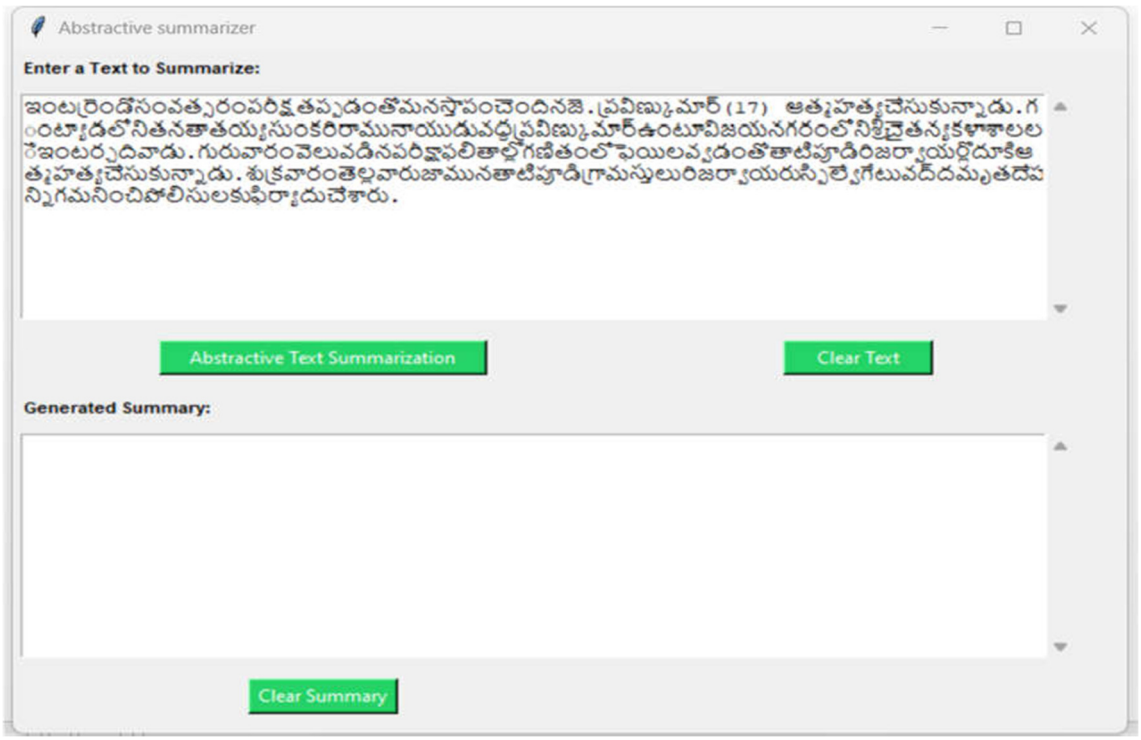

Fig. 1. Abstractive Text Summarization User Interface

Input to the application is given in the form of telugu text and when the user clicks Abstractive Text Summarization button, the text is generated in abstract format and shown in other Text area.

3. EXPECTED OUTPUT

The proposed system should be able to accurately summarize for the given Telugu sentences with new phrases and words. Precision, recall, F-measure, and Rouge scores should be

maximum, and the performance should be increased with the increased training data and number of iterations in the algorithm.

Expected Input:

ꢀ ꢁ 4 ꢂꢃ ꢄ ꢅ yꢆ ꢄ & ꢇꢃ gꢈ ꢉ ꢄ ꢄ &ꢃ ?.ꢄ æey➢ð(17) ꢊ ꢅ ęꢋ ꢅ ş 3 h g.ꢌ

şꢈ å9 ꢃꢍ ꢅ ꢎ ꢏ ş ꢐ i ð £ ꢄ ꢑ ꢄ æey➢ð ꢒ æꢓ ꢏ ꢄ ꢌ ꢔ å9 æ ꢅ ꢄ şꢐ

æꢕ â ꢀ ꢁ ꢆ şæ . æ ð ꢄ æꢃ ꢄ & ꢖ æ â ꢌ &ꢅ â eꢕꢃ gꢈ

& æ iꢓ gꢏ & 9 ꢗ ęꢋ ꢅ ş 3 h g . ꢐ æ Tꢕ æ wðꢃ & æ e ꢉ

iꢓ gꢏ ig@g 2 ꢄ ꢑ ꢉ ðå9gꢌ ꢉ 9 9æ ꢕ h ş 3æ .

Expected Output:

ꢀ ꢁ ig ꢏ ꢄ ꢅ yꢆ ꢖ æ åꢌ &ꢅ åæꢐꢕ ꢉ ꢉ ?.ꢄ æey➢ð(17) & æꢓ æꢐ ꢏ å

9 ꢗ ꢅ ęꢋꢅ ş3 h g .

ig@g2 ꢄ ꢑ ꢉ ðå9gꢌꢉ 9 9ꢃ e ꢉ &æ ꢕ h ş 3æ .

4. CONCLUSION

Abstractive summarization is a technique used to create summaries from the semantic representation of source documents instead of sentences from the source documents. This study combines two techniques, HPO and SFO to create a HSFO. The input document is then processed using Semantic Role Labeling (SRL) to extract the Predicate-Argument Structure (PAS) using the Stanza tool, allowing for the creation of an accurate abstractive summary. This study will show that that HSFO is a promising approach to abstractive summarization.

REFERENCES

1. Moratanch, N. and Chitrakala, S., "A novel framework for semantic oriented abstractive text summarization", Journal of Web Engineering, vol. 17, no. 8, pp. 675-716, 2018.

2. Khan, A., Salim, N. and Kumar, Y.J., "A framework for multi-document abstractive summarization based on semantic role labeling", Applied Soft Computing, vol. 30, pp. 737-747, 2015.

3. Mohamed, M. and Oussalah, M., "SRL-ESA-TextSum: A text summarization approach based on semantic role labeling and explicit semantic analysis", Information Processing & Management, vol. 56, no. 4, pp. 1356-1372, 2019.

4. Sudha, D.N. and Latha, Y.M., "Multi-document abstractive text summarization through semantic similarity matrix for Telugu language", Int J Adv Sci Technol, pp. 513-521, 2020.

5. Gabriel, S., Bosselut, A., Da, J., Holtzman, A., Buys, J., Lo, K., Celikyilmaz, A. and Choi, Y., "Discourse understanding and factual consistency in abstractive summarization", arXiv preprint arXiv:1907.01272, 2019.

6. Mandal, S., Singh, G.K. and Pal, A., "Single document text summarization technique using optimal combination of cuckoo search algorithm, sentence scoring and sentiment score", International Journal of Information Technology, vol. 13, pp. 1805-1813, 2021.

7. Wang, Q. and Ren, J., "Summary-aware attention for social media short text abstractive summarization", Neurocomputing, vol. 425, pp. 290-299, 2021.

8. Balachandran, V., Pagnoni, A., Lee, J.Y., Rajagopal, D., Carbonell, J. and Tsvetkov, Y., "StructSum: Summarization via structured representations", arXiv preprint arXiv:2003.00576, 2020.

9. Munot, N. and Govilkar, S.S., "Comparative study of text summarization methods", International Journal of Computer Applications, vol. 102, no. 12, 2014.

10. Fortuna, B., Grobelnik, M. and Mladenic, D., "Visualization of text document corpus", Informatica, vol. 29, no. 4, 2005.

11. Huang, L., Cao, S., Parulian, N., Ji, H. and Wang, L., "Efficient attentions for long document summarization", arXiv preprint arXiv:2104.02112, 2021.

12. Yadav, C.S. and Sharan, A., "Hybrid approach for single text document summarization using statistical and sentiment features", International Journal of Information Retrieval Research (IJIRR), vol.5, no.4, pp.46-70, 2015.

13. Mohd, M., Jan, R. and Shah, M., "Text document summarization using word embedding", Expert Systems with Applications, vol. 143, pp. 112958, 2020.

14. Yadav, C.S., Sharan, A., Kumar, R. and Biswas, P., "A new approach for single text document summarization", In Proceedings of the Second International Conference on Computer and Communication Technologies: IC3T 2015, Vol.2, pp. 401-411,2016.

15. Nagalavi, D., Hanumanthappa, M. and Ravikumar, K., "An Improved Attention Layer assisted Recurrent Convolutional Neural Network Model for Abstractive Text Summarization", INFOCOMP Journal of Computer Science, vol. 18, no. 2, pp. 36-47, 2019.

16. Barros, C., Lloret, E., Saquete, E. and Navarro-Colorado, B., "NATSUM: Narrative abstractive summarization through cross-document timeline generation", Information Processing & Management, vol. 56, no. 5, pp. 1775-1793, 2019.

17. Zhang, M., Zhou, G., Yu, W., Huang, N. and Liu, W., "A Comprehensive Survey of Abstractive Text Summarization Based on Deep Learning", Computational Intelligence and Neuroscience, 2022.

18. Glenn, T.C., Zare, A. and Gader, P.D., "Bayesian fuzzy clustering", IEEE Transactions on Fuzzy Systems, vol. 23, no. 5, pp. 1545-1561, 2014.

19. Naruei, I., Keynia, F. and Sabbagh Molahosseini, A., "Hunter–prey optimization: Algorithm and applications", Soft Computing, vol. 26, no. 3, pp. 1279-1314, 2022.

20. Shadravan, S., Naji, H.R. and Bardsiri, V.K., "The Sailfish Optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems", Engineering Applications of Artificial Intelligence, vol. 80, pp. 20-34, 2019.

21. Song, S., Huang, H. and Ruan, T., "Abstractive text summarization using LSTM-CNN based deep learning", Multimedia Tools and Applications, vol. 78, pp. 857-875, 2019.

22. Zhang, M., Zhou, G., Yu, W., Huang, N. and Liu, W., "A Comprehensive Survey of Abstractive Text Summarization Based on Deep Learning", Computational Intelligence and Neuroscience, 2022.

23. Telugu dataset is taken from, "https://github.com/csebuetnlp/xl-sum", accessed on February 2022.

24. Liu, Y., Titov, I. and Lapata, M., "Single document summarization as tree induction", In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 1745-1755, June 2019.

25. Agarwal, S., Singh, N.K. and Meel, P., "Single-document summarization using sentence embeddings and k-means clustering", In proceedings of 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), IEEE, pp. 162-165, October 2018.

26. Stanza tool is taken from, "https://stanfordnlp.github.io/stanza/", accessed on August 2022.

27. Hatzigiorgaki, M. and Skodras, A.N., "Compressed domain image retrieval: a comparative study of similarity metrics", In Visual Communications and Image Processing 2003, vol. 5150, pp. 439-448, SPIE, June 2003.

28. Gandomi, A.H., Yang, X.S. and Alavi, A.H., "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems", Engineering with computers, vol. 29, pp. 17-35, 2013.

29. Abualigah, L., Yousri, D., Abd Elaziz, M., Ewees, A.A., Al-Qaness, M.A. and Gandomi, A.H., "Aquila optimizer: a novel meta-heuristic optimization algorithm", Computers & Industrial Engineering, vol. 157, pp. 107250, 2021.

30. Xia, K., Huang, J. and Wang, H., "LSTM-CNN architecture for human activity recognition", IEEE Access, vol. 8, pp. 56855-56866, 2020.

31. A. Lakshmi and D. Latha., "Automatic Text Summarization for Telugu Language," 2021 4th International Conference on Recent Trends in Computer Science and Technology (ICRTCST), Jamshedpur, India, 2022, pp. 223-227, doi: 10.1109/ICRTCST54752.2022.9781921.