Enhancing Extractive Summarization for Low Resource Indian Languages using TF-IDF and SVD

Sangita Singh[1,*,†], Jyoti Prakash Singh[1,†], Akshay Deepak[1,†] and Supriya[1,†]

[1]National Institute of Technology Patna, Patna, 800005, Bihar

Abstract

Text summarization is one of the well-known issues in natural language processing (NLP) task in recent years. A combination of the term frequency-inverse document frequency (TF-IDF) with a dimension reduction technique named as singular value decomposition (SVD) has shown promising results for extractive text summarization based on different Indian languages. Our main goal is to produce an extractive summary of a text document that is succinct, fluid, and stable. In this regard, we have used the Indian Language Summarization (ILSUM)-2024 datasets, which is the third additional task shared by the Forum for Information Retrieval Evaluation (FIRE-2024). Our team, Sangita_NIT_Patna, achieved third place for Bengali and Gujarati languages in task 1. For Hindi and Telugu languages, we secured fourth place, and for Tamil, we ranked fifth. We have used article descriptions as our input data and generated a simple summary of that article description as an output.

Keywords

TF-IDF, SVD, Extractive Text Summarization, ILSUM-2024 Datasets,

1. Introduction

In the modern information era, we are surrounded by an overwhelming amount of textual content, such as news articles, novels, legal documents, scientific papers, and more, all of which continue to grow at an accelerated pace. However, searching through this vast amount of information to extract specific details can be a time-consuming task, and the results may not always be precise [1]. This is where automatic text summarization (ATS) comes in, helping us quickly identify relevant content. The ATS is indeed one of the most challenging tasks in Natural Language Processing (NLP) and Artificial Intelligence (AI). There are two primary approaches to automatic text summarization (ATS) tasks: (i) Abstractive approach: This method involves generating a concise summary by paraphrasing and condensing the original text, often requiring a deep understanding of the content. (ii) Extractive approach: This method involves selecting and combining key sentences or phrases from the original text to create a summary, focusing on identifying crucial information while preserving the original wording. The Natural Language Processing (NLP) research community has shown a remarkably high level of interest in automatic text summarization for Indian languages, dedicating significant efforts to developing effective summarization techniques for these languages. Although large-scale datasets are available for languages such as English, Chinese, French, and German, there is a notable absence of similar datasets for Indian languages. So, forum for information retrieval evaluation (FIRE) [2] will bridge the existing gap by creating reusable corpora for Indian language summarization through this collaborative task. They provide datasets in seven major Indian languages for this task: Hindi, Indian English, Gujarati, Tamil, Telugu, Kannada and Bengali.

The FIRE hosted a competition task for Indian Language Summarization (ILSUM) 2024, in which the dataset consists of articles and headline pairs from several of the country's most prominent newspapers. Each language receives over 15,000 in news articles. The task for each article is to write a meaningful fixed-length summary, either extractive or abstractive. In this regard, we examine five Indian languages (Bengali, Hindi, Gujarati, Tamil, and telugu) provided by the organisers of ILSUM-2024 [3]. For this

Forum for Information Retrieval Evaluation, December 12-15, 2024, India

sangitas.ph21.cs@nitp.ac.in (S. Singh); jps@nitp.ac.in (J. P. Singh); akshayd@nitp.ac.in (A. Deepak); supriya.phd20.cs@nitp.ac.in (Supriya)

0000-0002-9318-2186 (S. Singh); 0000-0002-3742-7484 (J. P. Singh); 0000-0001-6854-8599 (A. Deepak); 0009-0006-9763-4641 (Supriya)

task, we developed a single document extractive text summarization framework using TF-IDF and SVD techniques for various Indian languages.

The remaining of the paper is formatted as follows. Section 2 provides a synopsis of the related works. Section 3 presents our proposed framework for ILSUM-2022. Section 4 presents the proposed systems discovery and analysis of the results. Finally, in Section 5, we conclude the paper.

2. Related work

Text summarization is a vibrant research area in Natural Language Processing (NLP), with a focus on automatically condensing text into concise summaries. This article provides an overview of the numerous studies conducted in the area of text summarization, such as Kumar et al. [4] designed an extractive text summarization framework, which involves multiple text features including position, length, similarity, frequent words, and sentence numbers in ILSUM task at FIRE 2022. These features are then combined with optimized weights, determined using Genetic Algorithm (GA), to rank sentences. They achieved an F-score of 0.3843 for ROUGE-1, 0.2584 for ROUGE-2, 0.1997 for ROUGE-3, and 0.2190 for ROUGE-4 in the best run, submitted along with two other runs. Singh et al. [5] used PSO-based technique with ROUGE-1 recall cost function in a supervised manner for single-document extractive text summarization task. They also preduced new feature “incorrect word” in this work. Agarwal et al. [6] employed the IndicBART model to generate text summaries on the provided Hindi dataset for ILSUM-2022. IndicBART is a multilingual sequence-to-sequence pre-trained model that supports 11 Indian languages. By leveraging the IndicBART model for training, they achieved a ROUGE-1 F-score of 0.544 on the testing dataset, demonstrating the model’s effectiveness in generating high-quality summaries. Singh et al. [7] employed a sequence-to-sequence attention model based on recurrent neural networks (RNNs) for English in ILSUM-2022, which has shown promising results for abstractive text summarization. Specifically, we used article text descriptions as input data in Bidirectional Long Short-Term Memory (Bi-LSTM) networks in the encoding layer, and generated a simplified summary of the article description as output using LSTMs in the decoding layer. Singh et al. [8] extracted features from each sentence in the document using ten statistical features, and then summed these features to score sentences for both Hindi and English languages to generate the summary. Kumari et al. [9] introduced an extractive text summarization method employing K-means clustering for the ILSUM-2022 dataset. The technique comprises text tokenization, Word2Vec-based word and sentence vectorization, and dimensionality reduction using autoencoders and K-means clustering. This process facilitates the identification and extraction of key sentences and phrases, generating a coherent and informative summary. Chakraborty et al. [10] experimented pre-trained BART model, GPT model, and T5 model for English Languae in ILSUM-2022 at FIRE-2022. TeamMT-NLP-IIITH [11, 2] achieved the best performance across all three summarization tasks. The authors fine-tuned various transformer models, treating text summarization as a bottleneck task. Specifically: For Hindi and Gujarati, they fine-tuned MT5, MBart, and IndicBART for five epochs with a learning rate of 5e-5 and a maximum input length of 512. MT5 emerged as the best-performing model for Hindi, while MBart performed best for Gujarati. For English, they fine-tuned PEGASUS, BART, T5, and ProphetNet using similar hyperparameters, and PEGASUS outperformed the other models on text data. Satapara et al. [11] offers a comprehensive overview of the first edition of the ILSUM shared task, organized as part of the 14th FIRE-2022 conference. They covered the task’s goals, approach, participant submissions, and evaluation outcomes, providing a valuable snapshot of the current research landscape in Indian language summarization. TeamBITSPilani [12] fine-tunedmT5 (mT5-multilingual-XLSum) model on the ILSUM dataset for all four languages. TeamNITK-AI [12] outperformed other teams where they fine-tuned T5-base on ILSUM English dataset. TeamIrlab-IITBHU [12] utilized name entity-aware text summarization, NER emerges as important factor to extract in-depth information and prioritising key entities for the summary by utilizing a pre-trained Muril-based Hindi NER model and fine-tuning MBART-50 for Hindi language. This literature is primarily based on the FIRE-2023 (ILSUM) shared task. The NITK-AI (SCALAR) team [13] utilized the T5-Base model for Indian English, achieving scores of 0.3321, 0.1731, 0.121, and 0.282 for ROUGE-1 F1, ROUGE-2 F1, ROUGE-4 F1, and ROUGE-L F1, respectively. The authors [14] employed mT5-base along with a fine-tuned T5-base to generate more accurate summaries, resulting in scores of 0.3022, 0.1111, 0.2504, and 0.8616 for English, and 0.2701, 0.1214, 0.2237, and 0.6782 for Hindi across the same metrics. The Irlab-IITBHU team [12] fine-tuned the MBART-50 pre-trained model, achieving ROUGE scores of 0.5625, 0.471, 0.4032, and 0.5373 for Hindi. Meanwhile, the BITS Pilani [12] team fine-tuned the mT5 (mT5-multilingual-XLSum) model, with results of 0.174, 0.0747, 0.0333, and 0.1655 for Gujarati, and 0.12, 0.0567, 0.0254, and 0.1087 for Bengali [12], respectively. Among all the teams, NITK-AI [12] performed the best, fine-tuning the T5-base model on the ILSUM English dataset and achieving the scores mentioned. Gupta et al. [15] employed an approach called Named Entity-Aware Abstractive Text Summarization (NEA-ATS) for the Hindi language. Their method distinctively combines Named Entity Recognition with advanced pretrained language models, emphasizing key entities like people, places, and organizations.

3. Proposed Model

In this section, we discussed the methodology and datasets. We proposed an extractive text summarization framework for varioues Indian languagess. We will explain each step in detail in the following section, and the overall architecture is shown in Figure 1. So, the proposed model generates multi-sentence summaries.

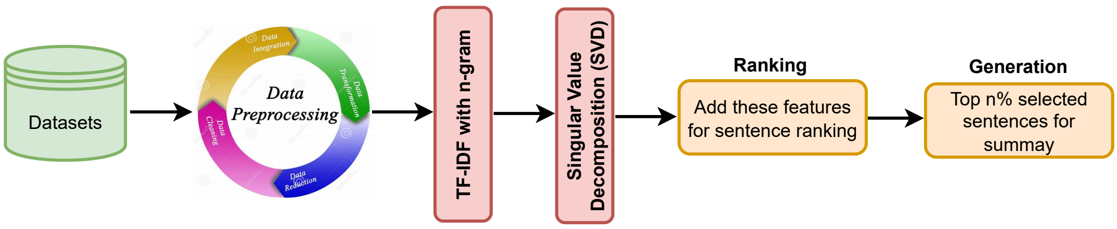

Figure 1: Proposed architecture diagram for extractive text summarization across multiple languages

3.1. Data Collection

To evaluate the our model, we utilized the ILSUM-2024 datasets provided by FIRE-2024 [3]. By developing reusable corpora for different Indian languages summarization, they hope to fill the current gap through this joint effort. The third edition of ILSUM adds three Dravidian languages—Kannada, Tamil, and Telugu—in addition to Hindi, Gujarati, Bengali, and Indian English from the previous [16] edition. The dataset for this task is built using articles and headline pairs from several leading newspapers of the country. They provide over 15,000 news articles for each language (except Tamil). The dataset description is shown in Table 1. The objective is to generate a concise, fixed-length summary for each article, which can be either extractive or abstractive in nature.

3.2. Data Preprocessing

We preprocessed the dataset by removing the missing and duplicates values from the "Article" or "Summary" columns. We then tokenized the article into sentences and removed punctuation and empty strings to prepare the text for further processing.

Table 1
Datasets description for multiple Indian language

Datasets	Train data	Validation data	Test data
English	9376	1500	2500
Hindi	10427	1500	3000
Gujarati	33630	-	1457
Telugu	9583	1065	4564
Kannada	10694	1188	5093
Tamil	4104	456	1955
Bengali	12356	-	2206

3.3. TF-IDF with n-gram

In this section, we used TF-IDF technique to respresent the sentence in the vector form of the article. Here, the TF-IDF vectorizer is configured to extract features from the text data using a range of n-grams, including single words (unigrams), two-word combinations (bigrams), three-word combinations (trigrams), and four-word combinations (4-grams). By selecting the top 2000 most frequent terms across all documents, the TF-IDF matrix is truncated, reducing its dimensionality and focusing on the most important terms. This helps conserve memory and boost computational efficiency. It combines two measures:

1. **Term Frequency (TF)**: Term Frequency (TF) is a numerical measure that represents how frequently a term appears in a given document. Here, t represents a word (or n-gram, which could be a single word or a sequence of n words), and d represents a specific document. This score increases with the frequency of a word in a document but doesn't consider whether the word is common across other documents in the corpus. The TF score indicates the relative importance of a term within a document by measuring its occurrence. For a specific n-gram g in a document d, the TF is calculated as:
$$\text{TF}(t, d) = \frac{f_{t,d}}{\sum_{t' \in G} f_{t',d}} \tag{1}$$
where
    - $f_{t,d}$ is the frequency of n-gram t in the document d.
    - G is the set of all n-grams in the document d.
    - $\sum_{t' \in G} f_{t',d}$ is the total count of all n-grams in the document.

2. **Inverse Document Frequency (IDF)**: Measures how common or rare a word is across all documents in the corpus. A word that appears in many documents will have a lower IDF score. If a term appears in almost every document, its IDF score will be close to zero, meaning it has less unique significance. The inverse document frequency of n-gram t is:
$$\text{IDF}(t) = \log\left(\frac{N}{1 + n_t}\right) \tag{2}$$
where, N is the total number of documents in the corpus, n_t is the number of documents containing the n-gram t.
3. **TF-IDF Calculation:** The TF-IDF score for a term in a document is the product of its TF and IDF scores:
$$TF\text{-}IDF(t, d) = \text{TF}(t, d) \times \text{IDF}(t) \tag{3}$$

Words with high TF-IDF scores are considered important or unique to that document compared to other documents in the corpus.

3.4. Singular value decomposition (SVD):

SVD is a mathematical technique used in linear algebra for decomposing a matrix into three other matrices. SVD is used in Latent Semantic Analysis (LSA) to uncover relationships between terms and documents by reducing dimensionality in text data. SVD can reduce the dimensionality of feature vectors for sentences in the article, produced by the TF-IDF technique, while preserving the essential features and relationships in the article.

$$A = U \sum V^T \tag{4}$$

- A is a matrix of dimension M× N.
- U: $M \times M$ matrix of the orthonormal eigenvectors of AA^T.
- $\sum$: diagonal matrix with r elements equal to the root of the positive eigenvalues of AA^T or A^TA.
- V^T: transpose of a $N \times N$ matrix containing the orthonormal eigenvectors of A^TA.

The sums the rows of V^T, giving a score for each sentence based on its importance.

3.5. Ranking

In this step, we prioritize the sentences in the article, ranking them in descending order of importance. This produces a list of ranked sentences, with the most important sentences at the top and the less significant ones at the bottom.

3.6. Generation

To generate the summary, we first calculate the sentence count by taking n=15% of the total sentences. We then combine the top-ranked sentences up to this count to create the summary.

4. Evaluation Metric and Results

In this section, we present the evaluation metrics used to assess the performance of our proposed approach, followed by a detailed discussion of the results obtained.

4.1. Evaluation Metric

In this study, we utilized the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [17] and BERTScore (B) metrics to assess the performance of our model. ROUGE (R) measures the quality of the generated summaries by counting the number of overlapping lexical units between the generated and reference summaries. Unlike R, which relies on exact word or phrase matches, B leverages pre-trained contextual embeddings from BERT (Bidirectional Encoder Representations from Transformers) to calculate the semantic similarity between the generated and reference summaries. ROUGE and BERTScore includes precision (Pre), recall (Rec), and F1 score as part of its evaluation metrics. Here, we employed R-N (with N=1,2,4) and R-L (Longest Common Subsequence) to compute the R-1, R-2, and R-L scores based on Pre, Rec and F1 for both the training and validation datasets. For the testing dataset, R-N (with N=1, 2, and 4) and R-L were evaluated based on the F1-score, while B was evaluated using Pre, Rec, and F1-score.

4.2. Results

In this section, we discussed about the results obtained on Training, validation and test datasets provided by ILSUM-2024 [18] [11]. Table 2 shows the results on the training dataset for all different languages for Task 1. Table 3 shows the results on the validation dataset for five different languages for Task 1. In the Bengali and Gujarati languages category, we secured the 3^{rd} position and present the corresponding results in Table 4 and Table 6, respectively, for Task 1. In the Hindi and Telugu languages category, we secured the 4^{th} position and present the corresponding results in Table 5 and Table 8, respectively, for Task 1. Similarly, for the Tamil language, we achieved the 5^{th} position and show the results in Table 7. In this study, TF-IDF was employed to represent text data by assigning weights to terms based on their importance within the corpus. This method proved effective in reducing the influence of high-frequency, low-relevance terms, resulting in a more meaningful feature space. Singular value decomposition further enhanced the feature set by capturing latent semantic relationships and reducing dimensionality, thereby improving computational efficiency and model generalization.

Table 2
Results on the training dataset for various Indian languages using the proposed approach

	R-1			**R-2**			**R-L**		
Methods	**Pre**	**Rec**	**F1**	**Pre**	**Rec**	**F1**	**Pre**	**Rec**	**F1**
Telugu	17.667	23.465	18.951	10.356	13.699	10.959	17.347	22.919	18.604
Tamil	12.704	17.583	13.527	6.912	9.619	7.270	12.229	16.929	13.014
Kannada	17.274	24.937	19.252	9.933	14.676	11.027	16.823	24.608	18.738
Hindi	29.55	36.000	30.153	13.559	16.57	13.564	27.138	32.970	27.641
English	17.255	36.006	21.508	7.343	16.083	9.123	15.167	31.758	18.915
Gujarati	20.357	27.029	21.696	10.279	13.846	10.873	18.775	24.924	19.999
Bengali	13.742	17.115	13.869	8.123	10.210	8.071	12.339	15.286	12.396

Table 3
Results on the validation dataset for various Indian languages using the proposed approach

	R-1			**R-2**			**R-L**		
Methods	**Pre**	**Rec**	**F1**	**Pre**	**Rec**	**F1**	**Pre**	**Rec**	**F1**
Telugu	17.024	23.540	18.637	9.870	13.680	10.682	16.370	22.614	17.915
Tamil	14.207	19.314	15.076	8.065	11.665	8.747	13.664	18.594	14.505
Kannada	17.627	25.587	19.638	10.130	15.283	11.323	16.752	24.304	18.650
Hindi	28.510	35.739	29.441	12.061	15.883	12.487	24.503	31.043	25.392
English	17.5976	36.607	21.925	7.619	16.926	9.510	15.339	32.095	19.133

Table 4
Results on the testing dataset for Bengali language using the proposed approach

Rank	**Team_Name**	**R-1**	**R-2**	**R-4**	**R-L**	**B-Pre**	**B-Rec**	**B-F1**
1	Data Lovers	0.2471	0.1658	0.1187	0.2297	0.7372	0.7316	0.7338
2	Curious Coders	0.2411	0.161	0.1133	0.2233	0.7384	0.7251	0.731
3	Sangita_NIT_Patna	0.2096	0.1409	0.1057	0.1847	0.6996	0.7233	0.7099
4	SynopSizers	0.1957	0.1224	0.0938	0.1693	0.6755	0.732	0.7014
5	SCaLAR	0.0397	0.0103	0.001	0.0379	0.4178	0.6081	0.4948

Table 5
Results on the testing dataset for Hindi language using the proposed approach

Rank	**Team_Name**	**R-1**	**R-2**	**R-4**	**R-L**	**B-Pre**	**B-Rec**	**B-F1**
1	Data Lovers	0.3659	0.1975	0.1233	0.3388	0.7196	0.7621	0.7396
2	CSSG	0.3421	0.1713	0.102	0.312	0.741	0.7343	0.7371
3	Curious Coders	0.331	0.163	0.0978	0.2981	0.7204	0.7449	0.7318
4	Sangita_NIT_Patna	0.3015	0.1375	0.0827	0.2729	0.7129	0.7308	0.7207

Table 6
Results on the testing dataset for Gujarati language using the proposed approach

Rank	Team_Name	R-1	R-2	R-4	R-L	B-Pre	B-Rec	B-F1
1	Data Lovers	0.2792	0.1496	0.0942	0.2669	0.7506	0.7301	0.7398
2	Curious Coders	0.2723	0.1467	0.086	0.2607	0.7485	0.7303	0.7388
3	Sangita_NIT_Patna	0.2526	0.1188	0.0703	0.2415	0.7242	0.732	0.7274
4	SynopSizers	0.2109	0.0835	0.0437	0.1958	0.7578	0.6844	0.7186
5	Squad	0.181	0.0811	0.0347	0.1748	0.6929	0.7301	0.7105
6	SCaLAR	0.0819	0.0244	0.0045	0.0802	0.5796	0.6116	0.5942
7	Trojan Horses	0.0516	0.0101	0.001	0.0484	0.4712	0.6497	0.5458

Table 7
Results on the testing dataset for Tamil language using the proposed approach

Rank	Team_Name	R-1	R-2	R-4	R-L	B-Pre	B-Rec	B-F1
1	Data Lovers	0.2376	0.1507	0.1018	0.2284	0.7226	0.7496	0.7354
2	INITIATORS	0.218	0.1336	0.0905	0.2091	0.7301	0.7292	0.729
3	Curious Coders	0.1962	0.1175	0.0801	0.1872	0.7123	0.7283	0.7197
4	SynopSizers	0.1547	0.0877	0.0561	0.1468	0.6606	0.7292	0.6925
5	Sangita_NIT_Patna	0.1392	0.075	0.0487	0.1334	0.6867	0.7051	0.6948
6	Squad	0.0121	0.0007	0.0001	0.012	0.6192	0.5747	0.5951
7	SCaLAR	0.0097	0.0019	0.0001	0.0096	0.3928	0.5839	0.4693

Table 8
Results on the testing dataset for Telugu language using the proposed approach

Rank	Team_Name	R-1	R-2	R-4	R-L	B-Pre	B-Rec	B-F1
1	INITIATORS	0.3146	0.2318	0.1802	0.3079	0.7488	0.7637	0.7555
2	Data Lovers	0.3022	0.2149	0.1606	0.2963	0.7544	0.7527	0.7532
3	Curious Coders	0.1973	0.1161	0.0755	0.1916	0.7175	0.7164	0.7166
4	Sangita_NIT_Patna	0.1926	0.1139	0.0769	0.1876	0.7006	0.7236	0.7112
5	Squad	0.1498	0.0695	0.0226	0.1434	0.7392	0.6765	0.7058

5. Conclusion and Future work

In this work, we applied TF-IDF and SVD techniques for extractive text summarization in various Indian languages. These findings offer valuable insights for higher quality summaries, reduction of redundancy, more coherent summaries, when used together, TF-IDF and SVD create summaries that better capture both the key terms (from TF-IDF) and latent concepts (from SVD), producing summaries that are both relevant and coherent. While these traditional methods lack the contextual embeddings provided by deep learning techniques, their simplicity and interpretability make them valuable tools, particularly for resource-constrained applications. Singular value decomposition helps reduce redundancy by identifying overlapping information within sentences. It also captures semantic relationships between words and sentences. However, TF-IDF doesn't consider sentence structure or semantics, so the resulting summary may miss out on coherence, as it focuses purely on the frequency and rarity of terms. Singular value decomposition can be computationally expensive, especially for large documents. Also, it may struggle with small texts where latent structures are harder to detect.

Several areas can be explored to further enhance the extractive text summarization process for Indian languages. Future work could explore hybrid approaches combining TF-IDF and SVD with contextual word embeddings for improved performance. One promising direction is the integration of more advanced techniques, such as transformer-based architectures (e.g., BERT or GPT), which can capture deeper semantic understanding and sentence structure beyond what TF-IDF and SVD offer.

Acknowledgments

This first author would want to acknowledge the Ministry of Education (MOE), Government of India for financial support during the research work through the Rajiv Gandhi fellowship Ph.D scheme (UGC) for computer science & engineering.

Declaration on Generative AI

The authors confirm that no generative AI tools were used in the writing, editing, or analysis processes of this manuscript. All content was created and reviewed by the authors.

References

[1] W. S. El-Kassas, C. R. Salama, A. A. Rafea, H. K. Mohamed, Automatic text summarization: A comprehensive survey, Expert Systems with Applications 165 (2021) 113679.

[2] S. Satapara, B. Modha, S. Modha, P. Mehta, FIRE 2022 ILSUM track: Indian language summarization, in: D. Ganguly, S. Gangopadhyay, M. Mitra, P. Majumder (Eds.), Proceedings of the 14th Annual Meeting of the Forum for Information Retrieval Evaluation, FIRE 2022, Kolkata, India, December 9-13, 2022, ACM, 2022, pp. 8–11. URL: https://doi.org/10.1145/3574318.3574328. doi:10.1145/3574318.3574328.

[3] S. Satapara, P. Mehta, S. Modha, A. Hegde, S. HL, D. Ganguly, Overview of the third shared task on indian language summarization (ilsum 2024), in: K. Ghosh, T. Mandl, P. Majumder, D. Ganguly (Eds.), Working Notes of FIRE 2024 - Forum for Information Retrieval Evaluation, Gandhinagar, India. December 12-15, 2024, CEUR Workshop Proceedings, CEUR-WS.org, 2024.

[4] D. V. P. Kumar, S. S. Raj, P. Verma, S. Pal, Extractive text summarization using meta-heuristic approach., in: FIRE (Working Notes), 2022, pp. 464–474.

[5] S. Singh, J. P. Singh, A. Deepak, Supervised weight learning-based pso framework for single document extractive summarization, Applied Soft Computing 161 (2024) 111678.

[6] A. Agarwal, S. Naik, S. S. Sonawane, Abstractive text summarization for hindi language using indicbart., in: FIRE (Working Notes), 2022, pp. 409–417.

[7] S. Singh, J. P. Singh, A. Deepak, Deep learning based abstractive summarization for english language., in: FIRE (Working Notes), 2022, pp. 383–392.

[8] S. Singh, J. P. Singh, A. Deepak, Statistical and linguistic features based extractive text summarization for english and hindi languages, in: 2024 First International Conference on Pioneering Developments in Computer Science & Digital Technologies (IC2SDT), IEEE, 2024, pp. 222–227.

[9] K. Kumari, R. Kumari, An extractive approach for automated summarization of indian languages using clustering techniques., in: FIRE (Working Notes), 2022, pp. 418–423.

[10] S. Chakraborty, D. Kaushik, S. R. Laskar, P. Pakray, Exploring text summarization models for indian languages., in: FIRE (Working Notes), 2022, pp. 443–448.

[11] S. Satapara, B. Modha, S. Modha, P. Mehta, Findings of the first shared task on indian language summarization (ILSUM): approaches challenges and the path ahead, in: K. Ghosh, T. Mandl, P. Majumder, M. Mitra (Eds.), Working Notes of FIRE 2022 - Forum for Information Retrieval Evaluation, Kolkata, India, December 9-13, 2022, volume 3395 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 369–382. URL: https://ceur-ws.org/Vol-3395/T6-1.pdf.

[12] S. Satapara, P. Mehta, S. Modha, D. Ganguly, Key takeaways from the second shared task on indian language summarization (ILSUM 2023), in: K. Ghosh, T. Mandl, P. Majumder, M. Mitra (Eds.), Working Notes of FIRE 2023 - Forum for Information Retrieval Evaluation (FIRE-WN 2023), Goa, India, December 15-18, 2023, volume 3681 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023, pp. 724–733. URL: https://ceur-ws.org/Vol-3681/T8-1.pdf.

[13] S. Gowhar, B. Sharma, A. K. Gupta, A. K. Madasamy, Advancing human-like summarization: Approaches to text summarization., in: FIRE (Working Notes), 2023, pp. 747–754.

[14] V. Ilanchezhiyan, R. Darshan, E. M. Dhitshithaa, B. Bharathi, Text summarization for indian languages: Finetuned transformer model application., in: FIRE (Working Notes), 2023, pp. 766–774.
[15] S. Gupta, S. Pal, Named entity-aware abstractive text summarization for hindi language., in: FIRE (Working Notes), 2023, pp. 755–765.
[16] S. Satapara, P. Mehta, S. Modha, D. Ganguly, Indian language summarization at FIRE 2023, in: D. Ganguly, S. Majumdar, B. Mitra, P. Gupta, S. Gangopadhyay, P. Majumder (Eds.), Proceedings of the 15th Annual Meeting of the Forum for Information Retrieval Evaluation, FIRE 2023, Panjim, India, December 15-18, 2023, ACM, 2023, pp. 27–29. URL: https://doi.org/10.1145/3632754.3634662. doi:10.1145/3632754.3634662.
[17] F. Liu, Y. Liu, Exploring correlation between rouge and human evaluation on meeting summaries, IEEE Transactions on Audio, Speech, and Language Processing 18 (2009) 187–196.
[18] S. Satapara, P. Mehta, S. Modha, A. Hegde, S. HL, D. Ganguly, Key insights from the third ilsum track at fire 2024, in: Proceedings of the 16th Annual Meeting of the Forum for Information Retrieval Evaluation, FIRE 2024, Gandhiinagar, India. December 12-15, 2024, ACM, 2024.