SANSKRIT LANGUAGE TEXT SUMMARIZATION USING MODIFIED PAGE RANK ALGORITHM

Shalini Tomar
Department of Computer Science and Engineering
JPIET, MEERUT, U.P., INDIA
Email: Shalini.capricorn@gmail.com

Mr. Sandeep Rana
A.P., Department of Computer Science and Engineering
JPIET, MEERUT, U.P., INDIA
Email: sandeepmietcs@gmail.com

Mr. Baldivya Mitra
A.P., Department of Computer Science and Engineering
MIET, MEERUT, U.P., INDIA

Abstract- My paper proposed a methodology on Sanskrit text rundown. Content outline is one of the highlights that handling's applications which is utilized for decreasing the first content sum and recovering just the significant data from the first content. . The Sanskrit Language includes a high level morphological style that makes its uncommonly exhausting to dispose of things to be used as a feature of summation live. The suggested technique is a chart-based architecture that treats the record as a diagram, with the sentences as the vertices.
For each hub, a modified Page Rank algorithm is used, with an underlying score equal to the number of objects in this statement. More things in a sentence equals more data, hence things are used as the sentence's introduction position. The cycle of text rundown comprises of three significant stages: pre handling stage, highlights extraction, and diagram development stage, lastly applying the Modified Page Rank calculation.

Keywords- Sanskrit Text Summarization, Page Rank, Morphological Analyzer, Graph Based Approach

I INTRODUCTION

For the generation of meaningful textual content summaries, Text Summarization is a focal point of investigation beneath Computational Linguistics and Natural Language Processing (NLP). One of the early works came from IBM's Luhn (1958), who recommended creating summaries of abstracts of medical studies. Some Indian languages have also seen advances in text summarization. The number one strategies for summarising textual content are extractive textual content summarization (ETS) and abstractive textual content summary (ATS). IL net content, newspaper articles, studies papers, authentic documents, and so on (Talukder et al., 2019; Sankar et al., 2011; Embar et al., 2013; Lehal and Gupta, 2011; so on). Sanskrit is today widely studied in several contexts as a composing language, with millions of manuscripts of exceptional scholarly significance. The troubles of textual content availability, clarity and the want to get entry to the know-how in it have provided a large requirement for Text Summarization and associated studies for Sanskrit. Sanskrit's ability to infinitely condense a statement through the use of concatenating tactics such as euphonic combinations (sandhi), compounding (samasa), scrambling, verb elision for prosody, and so on makes it difficult to arrive at the structural or collocation meaning of the expression.

II EXISTING SYSTEM

1. Related works

So far, Sanskrit Text Summarization has looked at it from an extractive perspective.
Average Term Frequency-Inverse Sentence Frequency (tf-isf), Vector Space Model (VSM), and Graph-Based Approach are three TS methods used by Barve et al. (2015) to get text synopsis for Sanskrit based on a client's enquiry. They concluded that the VSM provided the most accurate summary, with an accuracy of 80%. As demonstrated by Barve et al., ETS is a good exposition methodology with a high recurrence of the query word (2015).
Significant Text Summarization approaches in Sanskrit language are:

2.1:- Abstractive Text Summarization: Scholars have various bases for getting sorted out the kinds of text outline. The greater part of them can go under at least one of these classifications:
1. Construction versus Semantic metho -dology (Sunitha et al, 2016)
2. AI (Machine Learning) based techniques (Traang and Anh, 2019; Talukder k et al., 2019)
3. Methodology based on Corpora (Haussler et. al., 2003)
2.1.1. Semantic approach vs. construction technique Sunitha et al. (2016) provide a summary of current ATS for ILs strategies.
The most important techniques to deal with ATS in ILs can be divided into two categories:
Semantics-based and structure-based
2.1.2. AI Approaches:
One alternate method of characterizing the TS types is the ML based methodology: managed and unaided techniques (Fizi-Derakashi and Mazid, 2015).
2.1.3. Methodology based on Corpus:
This clarifies Corpus using appropriate explanation plans such as POS, NER, and talk comment mechanisms such as the Rhetorical-Structure Theory (Thompson and Mann, 1988; Zahri et al., 2015; Jones, 1999), among others. 2.2:- Query Based Extracted Text Summarization: -
The strategies depend on normal term recurrence, the VSM (Vector Space Model) and a chart based method.
2.3:- Hybrid methodology for text outline: -
It is Syntactical information and Sanskrit Memamsa Principle based methodology.
It is a coordinated syntactic information and sentence combination for abstractive multi report outline framework.
3. Inspiration and issue of explanation:-
Many experts in the field of Sanskrit text outline utilise a chart based on Page Rank computation. This study employs a modified Page Rank calculation that incorporates the Weight of the edge as a requirement, as well as the quantity of items in the sentence as the underlying position of the sentence. In any case, in Sanskrit there may be no fundamental manner like English to test if a phrase is an issue in mild of the reality that there are not any capitalized or lowercase letters like in English; what is more, modern-day pupils do exclude diacritics of their composed content. As a result, morphological arrangement is required to extract things from text. In addition, in order to provide the optimum presentation, this exploration endeavours to use a change quantity of emphases with ModifiedPage Rank calculation to get the quality presentation.
4. Diagrams in textual content:
A graph G̓ (ⱴ, ̓E) is a numerical representation of a pair of logical connections between objects. The diagram has precept matters ⱴ: vertices and ̓E: edges. Edges address the idea of a relationship between two vertices, while vertices address the most important aspect of the discussed framework. In text outline there are numerous methodologies like Lex Rank, Text Rank or Page Rank calculation.
5. Page Rank calculation:
Page Rank calculation utilizes this plan to rank the pages that show up in the indexed lists. Page Rank doesn't consider every one of the inbound connections from the pages equivalent; the connection will take an additional significance relying upon the significance of the page that comes from it.

$$\mathbf{PR\ (P)} = (\mathbf{1} - \boldsymbol{d}) + \boldsymbol{d} * \sum_{\boldsymbol{i=0}}^{\boldsymbol{N}} \frac{\boldsymbol{PR\ (ui)}}{\boldsymbol{N}} \qquad \mathbf{(1)}$$

6. Modified Page Rank Calculation:
A Modified Page Rank depends primarily on the parts of Page Rank calculation with the accompanying contrasts:
(1) Pages are replaced with phrases from the record;
(2) The heaviness of the sides between hubs determined by the cosine resemblance, while within the first Page Rank there's no weight on the proposed graph's edge.
(3) The underlying position of every sente- nce is that the quantity of things during this dislike the primary Page Rank which provides the underlying position similarly to all or any hubs which approaches 1/N and here N is the quantity of sentence within the archive.

(4) The PageRank is altered as in MPR Formula. MPR Formula is employed to compute the new position of hub (h), Where PR (^{V}i) is that the current position of sentences and E (h, ^{V}i) is that the heaviness of edge interface sentences (h) and (^{V}i) which is likewise the cosine similitude between these two sentences, at long last the summation is separated by the number of the leftover sentences within the report N-1, which is that the quantity of sentences within the archive D with barring the present sentence, to urge the new position of sentence (h).

$$\mathbf{MPR\ (h)} = (\mathbf{1} - \boldsymbol{d})\ \boldsymbol{d} * \sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} \frac{\boldsymbol{PR\ (^{V}i) * E(h, ^{V}i)}}{\boldsymbol{N-1}} \qquad \mathbf{(2)}$$

III THE PROPOSED APPROACH

Three major stages are depicted in the suggested approach's flow diagram. Separating text from record is the first step, followed by preprocessing tasks such as standardisation, tokenization, stemming, stop word removal and morphological evaluation. The ideal highlights are deleted in the second stage, and the record is then displayed as a diagram.

Finally, in the third stage, the Page Rank calculation is altered, the synopsis is removed, and the presentation is evaluated.

The phases of the suggested strategy are as follows:

1. Stage First: Pre-handling:-

The archive is loaded and dissected at this point in order to prepare for highlight extraction.

1.1: Single-input archive: Extracting text from a single Sanskrit-language report and encoding it in utf-8 Itranslator 2003.

1.2: Normalization: In this step, numbers are removed from the sentence, but no letter set letters are removed from the sentence.

There is less emphasis on the letters 'I', 'II', and ',' in Sanskrit.For Ardhviram and Purnviram, the pictures 'I' and 'II' are used separately.

1.3: Tokenization: In this development the record is disconnected into regions, by then the segments into sentences, at long last the sentences into words.

1.4: Eliminating prevent phrases: discarding stop phrases bring down the substance material to extra cherished expressions. The calculation is executed as beneath given advances.

Stage 1: The text of the objective report is tokenized, and individual words are stored in exhibits.

Stage 2: From the stopword list, a single prevent word is chosen.

Stage 3: Using the consecutive inquiry approach, the stop word is compared to the target text in the form of a cluster.

Stage 4: If it matches, the word in the cluster is removed, and the correlation is delayed until the length of the exhibit is determined.

Stage 5: Following the complete expulsion of a stopword, a new stopword is selected from the stopword list, and the calculation is repeated as in stage 2. The process continues until all of the stopwords have been considered.

Stage 6: Stopword separation in the resultant content is shown, as well as other required insights such as the stopword eliminated, the number of stopwords eliminated from the target text, all out include of words in the target text, include of words in the resultant content, and singular stop word include found in the target text.

1.5: Stemming: To remove the base of each word in the phrase, Sanskrit Grammarian rendition 3.28 and INRIA Sanskrit Stemmer are used. This transaction is used to reduce the quantity of words; no doubt, throughout the entire existence of the set of experiences have re-get back to the great rich.

1.6: Morphological investigation: In this progression SanskritTagger is utilized it creates lexical and grammatical feature examinations of computerized Sanskrit messages utilizing a stochastic language model. SanskritTagger has been utilized to construct the explained text corpus from which the Digital Corpus of Sanskrit (DCS) has been removed. In this cycle, each word in the sentence is assigned a tag that corresponds to its POS location in the sentence.

The words' position could be a thing, an action word, a relational word, a stop word article, and so on. This interaction is used to determine the number of things in each phrase. 2. Stage Second: Extracting data and creating a diagram are included:-

In this stage the required highlights are removed, and afterward report is displayed as a diagram.

2.1: Extraction of features: At this stage, two types of highlights are extracted. The foundation of the word is equivalent to the term in this process.

• Cosine Two Sentences Have a Lot in Common

$$\textbf{Cosine\_Similarity} = \frac{\sum_{k=1}^{m} \mathbf{TF-IDF}(t_{ik}) * \mathbf{TF-IDF}(t_{jk})}{\sqrt{\sum_{k=1}^{m} \mathbf{TF-IDF}(t_{ik})^2} * \sqrt{\sum_{k=1}^{m} \mathbf{TF-IDF}(t_{jk})^2)} \qquad \textbf{(3)}$$

• Checking of things coming about because of the morphological examination step in each phrase.

In the morphological inquiry step, this element is used as an underlying position for each sentence that reveals a model.

2.2: Weighting and building chart

3. Stage Third: Using Modified Page Rank and extracting an outline:

At this point, the Modified PageRank calculation is performed, and the outline is then extracted.

3.1. Modified PageRank is used.

The Modified PageRank formula is used in this sequence, with the starting position for each sentence approaching its own tally of items. The PageRank is used with a variety of emphases: 10, 100, 1000, 10,000,

100,000, and a million. These numerous numbers of cycles are used to obtain the most emphases in order to arrive at the ideal show.

3.2. Extraction of the outline

Hubs are organised in this progression based on their previous location. Sentences are eliminated one at a time and added to the synopsis until the pressure proportion is met; if the overlaying between the selected sentence and some other sentence in the define is extraordinarily high at that point, this sentence is fail to stop repetition.

3.3. Excess is eliminated.

Repetitive sentences are removed from the rundown once the outline is divided in this step. Sentence covering is used to identify repeating sentences; when the overlaying between sentences is more noteworthy than 85% the closing sentence is eradicated from the outline.

3.4. Selecting and reviewing summary records that have already been prepared.

This stage involves looking over the corpus with a pre-made outline to assess the synopsis's presentation. The corpus contains five pre-generated rundowns. The following synopsis is compared to them.

The pseudo code for the proposed approach is shown below. It starts with analysing the record, then moves on to pre-preparation, highlight extraction, and diagram construction, PageRank, outline extraction, and reducing unnecessary phases.

Algorithm for the Proposed Approach:

Taking In: Sanskrit Text
Taking Out: Summarized Sanskrit Text
1 Arrange/Set the Max Sentences within the
Layout ←Add up to Sentences in Record
2 Sanskrit Sentences ← SansText.Sentences
3 Morphological Sanskrit Analyzer (Inflection
Morphological Analyzer for Sanskrit)
4 Sanskrit Normalization Function ()
5 Tokenization Function ()
6 StopsWords Removal Function ()
7 Stemming Function ()
8 Sanskrit Graph ← New SGraph ()
9 SansTF-IDF← numbering Sentences TF-IDF ()
10 Sanskrit Noun List ← Using the Morph logical Sanskrit Analyzer to Compile a List of Nouns ()
11 SansNode ← CreateGraphNode (Sanskrit _ TF-IDF, SanskritNounList)
12 SGraph.Add ← SansNode
13 EveryNode ← SGraph.Nodes
14 If (EveryNode <> SansNode)
15CosineSimilarity ←CosineSimilarity (Every Node, SansNode)
16 Sanskrit Nouns Measure ← CalcNoun (Node)
17 SGraph.CreateEdge (EveryNode, Sans Node, CosineSimilarity)
18 Set the number of nouns in each sentence as the first rank.
19 Apply Modified_Page_Rank Algo ()
1. Rundown ← Summary Extraction
2. Rundown ← Summary of Reduced Text
20 Yield ← Summaries

FLOW CHART: Proposed Approach

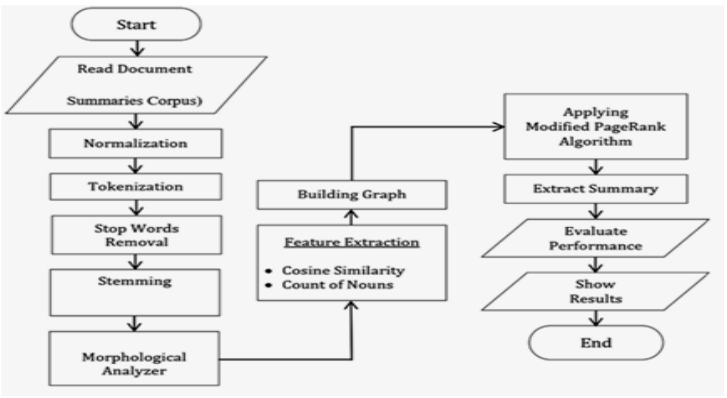

Fig 1 Flowchart

4. Sanskrit Text Summarization:

Table 1: Sanskrit POS tagger

S. No	Sanskrit Sentence	POS Tagged Detail
S01	कर्मण्येवाधिकारस्तेमा फलेषु कदाचन। मा कर्मफलहेतुर्भूर्माते सङ्गोस्त्वकर्मणि।।	कर्मण्येवाधिकारस्तेमा[_1.1] फलेषु[_7.3] कदाचन ।[PUN_RS] मा[_1.1] कर्मफलहेतुर्भूर्माते[_7.1] सङ्गोस्त्वकर्मणि ।[PUN_RS]
S02	योगस्थःकुरुकर्माणिसङ्गंत्यक्त्वाधनञ्जय। सिद्धासिद्धोःसमो भूत्वा समत्वं योग उच्यते।।	योगस्थःकुरुकर्माणिसङ्गंत्यक्त्वाधनञ्जय ।[PUN_RS]सिद्धोःसमो[सिद्धासिद्धोःसमो] भूत्वा[_3.1]समत्वं[_2.1]योग[योग]उच्यते।[PUN_RS]
S03	कर्मजंबुद्धियुक्ता हि फलं त्यक्त्वा मनीषिणः। जन्मबन्धविनिमुत्काःपदंगच्छन्त्यनामयम्।।	कर्मजंबुद्धियुक्ता[_1.1] हि[AV] फलं[_2.1] त्यक्त्वा[_3.1]मनीषिणः।[PUN_RS] जन्मबन्धविनिमुत्काःपदंगच्छन्त्यनामयम् ।[PUN_RS]
S04	न कर्मणामनारम्भान्नैष्कर्म्यपुरूषोऽशुरुते। न च संन्यसनादेवसिद्धंसमधिगच्छति।।	न[AV कर्मणामनारम्भान्नैष्कर्म्यपुरूषोऽशुरुते ।[PUN_RS] न[AV च[AVC] संन्यसनादेवसिद्धंसमधिगच्छति ।[PUN_RS]
S05	नियतं कुरु कर्म त्वं कर्म ज्यायो ह्यकर्मणः। शरीरयाञापि च ते न प्रसिद्धेः कर्मणः।।	नियतं[AVKV] कुरु[P_IoT_2.1] कर्म[कर्म] त्वं[_1.1] कर्म[कर्म] ज्यायो[ज्यायो]ह्यकर्मणः।[PUN_RS]शरीरयाञापि[शरीरयाञापि] च[AVC] ते[SND_m_1.3] न[AV प्रसिद्धेः[_5.1/6.1] कर्मणः ।[PUN_RS]

Fig: 2.1: Sanskrit Summarization Text

Summary generated by graph-based approach

कदलीवृक्षस्य शलाटुः हरितवर्णीयः भवति । कदलीपर्णं बृहदाकारकं पञ्चषप अदमितदीर्घम्, अधिक विशालम् च भवति। बौद्धाः कदली-वृक्षं पवित्रं मन्यन्ते । कदलीवृक्षस्य पुष्पं, शलाटुः, फलं च आहररुपेण उपयुज्यते । काभिः चित् भारतीयभाषाभिः कदली इति एव उच्यते । पक्वानि कदलीफलानि कदली-पुष्पम् यदा शलाटुः पक्वं भवति तदा वृक्षं कर्तयन्ति ।

Fig: 2.2: Sanskrit Summarization Text

१.कर्मण्येवाधिकारस्ते मा फलेषु कदाचन। मा कर्मफलहेतुर्भूर्मा ते सङ्गोस्त्वकर्मणि।।२-४७ २. योगस्थः कुरु कर्माणि सङ्गं त्यक्त्वा धनञ्जय। सिद्धयसिद्धयोः समो भूत्वा समत्वं योग उच्यते।।२-४८ ३. कर्मजं बुद्धियुक्ता हि फलं त्यक्त्वा मनीषिणः। जन्मबन्धविनिमुत्काः पदं गच्छन्त्यनामयम्।।२-५१ ४. न कर्मणामनारम्भान्नैष्कर्म्यं पुरूषोऽशुरुते। न च संन्यसनादेव सिद्धं समधिगच्छति।।३-४ ५. नियतं कुरु कर्म त्वं कर्म ज्यायो ह्यकर्मणः। शरीरयाञापि च ते न प्रसिद्धेः कर्मणः।।३-८

Summarized Text

१.कर्मण्येवाधिकारस्ते मा फलेषु कदाचन। मा कर्मफलहेतुर्भूर्मा ते सङ्गोस्त्वकर्मणि।।२-४७ २.

नियतं कुरु कर्म त्वं कर्म ज्यायो ह्यकर्मणः। शरीरयाञापि च ते न प्रसिद्धेः कर्मणः।।३-८

योगस्थः कुरु कर्माणि सङ्गं त्यक्त्वा धनञ्जय। सिद्धयसिद्धयोः समो भूत्वा समत्वं योग उच्यते।।२-४८ ३.

कर्मजं बुद्धियुक्ता हि फलं त्यक्त्वा मनीषिणः। जन्मबन्धविनिमुत्काः पदं गच्छन्त्यनामयम्।।२-५१ ४.

Fig: 2.3: Sanskrit Summarization Text

V Experimentation and results

5. 1. Dataset (Corpus)

The Digital Corpus of Indo-Aryan (DCS) could be a Sandhi-split corpus of Indo-Aryan texts with full morphological and lexical analysis.Words is retrieved from the wordbook through an easy question or a wordbook page. for every lexical unit contained within the corpus, DCS provides the entire set of occurrences and a applied math analysis supported historical principles.

The text interface shows all contained texts alongside their textual matter lexical and morphological analysis.

5.2. Analysis Metrics

Exactness, recall, and F-measure are used to calculate the analysis. Formula (4), Formula (5), and Formula (6) will be used to calculate the value of exactness recall and F-measure.

Precision: Measuring the correct text size provided by the system.

$$\textbf{Precision}=\frac{Extracted\ Summary \cap Provided\ Summary}{Extracted\ Summary} \quad (4)$$

Recal: This reminder system contains data that reflects the quantitative relationships of the extracted related passage.

$$\textbf{Recal} = \frac{Extracted\ Summary \cap Provided\ Summary}{Provided\ \ Summary} \quad (5)$$

F-measure: Establishes a balanced relationship between the recall and exactness metrics.

$$\textbf{F-measure} = \frac{2*Precision*Recall}{Precision+Recall} \quad (6)$$

5.3. Setup for the experiment

This section provides an example of how PageRank can be changed. Figure Three shows the record sculptural as a graph, every node with inside the diagram takes its initial rank. Table four displays associate degree information. There are numbered and listed examples of Indo-Aryan POS in the table. Fig. 4 indicates the format as soon as applying modified PageRank components with ten, zero iterations; every node within the diagram has its new text weight. Fig. 4 shows the closing ranks of format nodes, in line with the figure node that represent sentence Se01, Se03, Se09 come returning the nice rank for that reason it's enclosed within the outline.

5.4. Discussion and analysis of the findings

This subcategory examines the system's outcomes and compares them to the outcomes of other techniques. Table 5 displays the formula outcomes after numerous iterations.

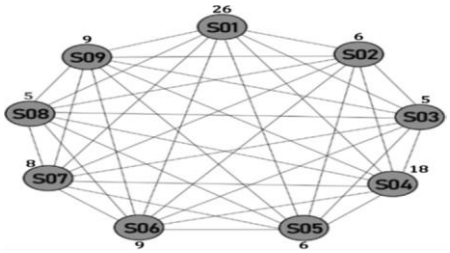

Fig. 3 Graph showing the starting position

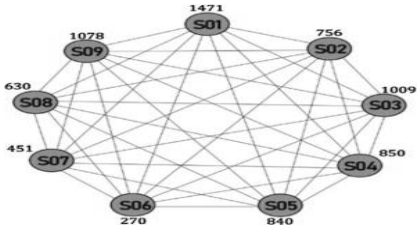

Fig. 4 after using the MPR method, the graph Position

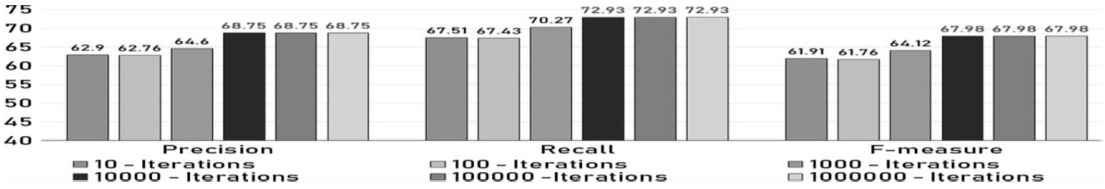

Fig. 5 Evaluation of performance in comparison to other research

when the wide variety of iterations is multiplied to 1,00,00 iterations, the findings in the table show that the performance of the algorithm will improve, and after 1,00,00 iterations or more, the performance will stabilize. Figure 5 and Table 2 shows the performance indicators for different iteration times. Table 3 and Figure 6 compare the findings of the present survey to the results of other surveys.

Table 2

With a large number of iterations, the algorithm's findings are evaluated.

Num of Iteration	Precisions	Recalls	F-Measures
1,0	62.897	67.581	61.921
1,00	62.765	67.423	61.756
1,000	64.598	70.270	64.123
1,00,00	68.748	72.932	67.981
1,00,00,0	68.751	72.931	67.983
1,00,00,00	68.754	72.934	67.985

Table 3

compared to other works

Method	Precisions	Recalls	F-Measures
Lex Rank	51.031	56.514	50.601
Text Rank	50.887	56.223	49.810
Statistical & Semantic Analysis	57.621	58.801	58.200
PageRank Algorithm	054	047	051
Proposed Method	68.753	72.945	67.991

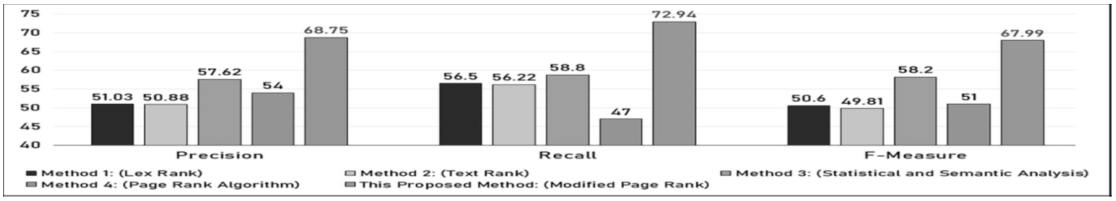

Fig. 6 Evaluation of performance in comparison to other research

The comparison reveals that the Modified PageRank algorithm produces superior results than Lex Rank, Page Rank, and Text Rank algorithms.

VI Conclusion

This method uses the Modified Page Rank Algorithm with a handful of iteration ranges to try to improve the overall performance of the generated summaries. This set of rules utilized through making the preliminary rank of the sentence because the vary of nouns it has, and the load of the edges. The technique of summarising method cuts down on study time. It generates filtered and relevant data using the Page Rank Algorithm. Summarization summaries in textual content make choosing a method easier and improve indexing efficacy.

REFERENCES:

[1] Extractive Text Summarization of Marathi News Articles, Yogeshwari V. Rathod, International Research Journal of Engineering and Technology (IRJET), Volume: 05 Issue: 07 | July 2018.
[2] Egyptian Informatics Journal 21 (2020) 73-81, Reda Elbarougy and Gamal Behery, Extractive Arabic Text Summarization Using Modified Page Rank Algorithm.
[3] Tahseen Ahmed Jilani, International Journal of Computer Science, June 2015, a Survey and Comparative Study of Page Rank Algorithms.
[4] Raulji Jaideepsinh k., Stop-Word Removal Algorithm and Sanskrit Language Implementation, International Journal of Computer Science, Volume 150-No. 2, September 2016.
[5] Sanskrit as a PL and NLP, Raghav Agrawal and Shashank Saxena, Global Journal of Business Studies and management. Volume 3, pp. 1135-1142
[6] Farzad Kiyani, Oguzhan Tas, A Survey based on Automatic Text Outlines, Press Academia, p.204-213.
[7] Sunitha K C et. al. (2016)
[8] Fizi-Derakashi and Majid, 2015.
[9] Embar et al., 2013; Shankar et al., 2011 Talukder et al., 2019;
[10] Jones, 1999; Zahri et al., 2015, Mann and Thompson, 1988)