A SIMPLE AND EFFICIENT TEXT SUMMARIZATION MODEL FOR ODIA TEXT DOCUMENTS

Sagarika Pattnaik
Dept. of CSE, ITER, S'O'A Deemed to be University, Bhubaneswar, Pin: 751030, India
sagarika.pari@gmail.com

Ajit Kumar Nayak
Dept. of CS&IT, ITER, S'O'A Deemed to be University, Bhubaneswar, Pin: 751030, India
ajitnayak2000@gmail.com

Abstract - The present scenario witnesses an exponential growth in the research field pertaining to text mining and automatic text summarization is a relevant topic. Varied methods in this regard have been developed for English and other European languages, but Odia language maintains a nascent status. The complexity and the highly inflective property of the language restrict the existing models for being directly applied on it. This paper proposes effective extractive single document automatic summarizer for Odia text document. Both statistical and clustering methods are applied and their evaluation metric F scores are compared. The present work is the need of the existing scenario. The experimental documents belong to news domain. The proposed algorithms meticulously deal with the complexity of the language and solve the problem of summarization to an appreciable extent.

***Keywords*:** Text summarization; extractive; TF-IDF, clustering, F score.

1. Introduction

Automatic text summarization, a thriving topic abridges the text document while preserving the original information content [Siddiqui and Tiwari (2008)]. From the two major categories of text summarization abstractive and extractive, the former is more human like but harder to implement [Moratanch and Chitrakala (2016)]. On the contrary extractive text summarization is simpler and fulfills the objective of producing an informative summary [Al-Sabahi et al. (2018)]. Overall a good summarizer tries to achieve the requisite features in the output summary like content coverage, coherency among sentences and less redundancy [Qaroush et al. (2019)]. To achieve higher performance researchers try to bring variations in the methodology. Diversified approaches [Munot and Govilkar (2014)] like statistical, linguistic, graph theory based, clustering and machine learning based are being implemented and are successful in enhancing the performance of a summarizer and make it useful in different domains like news [Modaresi et al. (2017)], web pages [Hu et al. (2017)], medical documents [Rouane et al. (2019)] and creating patent summaries [Tseng et al. (2007)]. But considering the developed state of art of summarization models with respect to Odia language, it is in a primitive state. The morphological complexity [Sethi (2014)] and lack of computational resources like availability of machine readable text corpus and automated NLP (Natural Language Processing) tools such as lexicons, POS taggers and named entity recognizers has isolated it from the computational scenario. Also Odia language does not have benchmark training and testing corpus for comparison and evaluation. As it is the official language of Odisha, a land rich in cultural heritage and spoken by 45 million people including Odisha and some parts of neighboring states auto text summarization has become an essential requirement of the present scenario.

The proposed techniques are competent attempts to bring Odia, a computationally impoverished language into computational field. The building of auto summarizer model has to start almost from the scratch. So, this paper deploys a combination of both statistical and linguistic approach to break the morphological complexity and solve the issue of automatic text summarizer for Odia language. It uses a variance of Term frequency-Inverse Document Frequency (TF-IDF) i.e. a mapped out sentence position value has been appended to classical TF-IDF. The output summary generated is of extractive type. The hierarchical clustering method has been experimented on the summarizer model with an aim to remove redundancy.

The rest of the paper is organized as follows: section 2 deals with the literature related to the proposed model. Section 3 analyses the data set used in the experiment. Section 4 describes the proposed model. Section 5 evaluates the work with a final conclusion and scope laid down for future work in section 6.

2. Related Works

This section cites some of the notable works related to the proposed model. The pioneering work in the field of text summarization is cited to Luhn [Luhn (1958)] further enhanced by Edmundson [Edmundson (1969)]. Experiments based on statistical methods were further carried out considering various other parameters like TF-IDF in combination with linguistic features. Qaroush Aziz, et al. [Qaroush et al. (2019)] in their proposed work demonstrated the efficiency of statistical methods on Arabic documents. A simple application of TF-IDF for getting an informative summary of a single document with 67% accuracy is seen in the works of Hans Christian et al. [Christian et al. (2016)]. Similarly considering applicability of clustering one of the method used by the proposed model, has always been effective in removing redundancy [Oufaida et al. (2014)]. Shetty and Jagadish [Shetty and Kallimani (2017)] adopted K means clustering for extracting informative summary sentences from text documents of CNN data set. Fejer et al. [Fejer and Omar (2014)] used hierarchical and K means clustering methods to extract sentences for summary from Arabic text achieving a result of 80% and 62% using Rouge (Recall Oriented Understudy for Gisting Evaluation).

Summarization model based on machine learning techniques with an aim to get an optimized result is also reported. NetSum a summarizer [Svore et al. (2007)] uses artificial neural network to extract the sentences for summary. Support Vector Machine (SVM) [Desai and Shah (2016)], Hidden Markov [Conroy and O'leary (2001)], Neural Network [Kaikhah (2004), Nallapati et al. (2016)] and Naïve Bayes [Neto et al. (2002)] are some of the successfully used machine learning approaches for auto text summarization. The disadvantage is that these methods require a voluminous training corpus raising difficulties for resource poor (computational resources) Indian languages. Few notable works considering various Indic languages [Abujar et al. (2017)] like Hindi text document using SVM [Desai and Shah (2016)] and TF-IDF [Vijay et al. (2017)] claims of achieving appreciable result. Similarly summarization work considering Bengali multi and single document is reported [Abujar et al. (2017), Akter et al. (2017)]. Kumar et al. [Kumar et al. (2011)] has proposed a summarizer for Tamil text document that uses graph theoretic scoring technique getting a score of 0.4723 under Rouge.

Considering Odia language, limited research on the discussed topic has been reported. A summarizer [Balabantaray et al. (2012)] based on normalized frequency count of the content words (exclusive of stop words) for determining significant sentences has claimed an average accuracy of 61%. The first sentence of the document has been mandatorily included in the summary giving importance to sentence position in the input document and compression ratio is kept at 50%. Similarly summarization work doing a performance analysis considering individually the features word frequency, positional criteria, cue phrase and title overlap is proposed [Biswas et al. (2015)]. It has claimed of achieving both precision and recall for word frequency based method 83% and for rest of the methods 100%. Thus a reinforced summarizer of Odia text is required that can cover up the prevailing deficits like obtaining a satisfactory output with proper result analysis and transparency of the data set experimented.

The literature survey concludes that limited amount of work are done in Indian languages concentrating mainly on Hindi and Bengali and for Odia language more effort with varied methods have to be experimented to fill up the loop holes and get an optimization path.

3. Dataset Considered

Getting data sets for Odia text summarization is a cumbersome task. The language lacks availability of suitable normalized dataset. Researchers have to prepare standard datasets for achieving their objectivity. The present research has gathered documents related to cricket news from popular Odia news papers Samaja [thesamaja.in], Dharitri [dharitri.com] and Sambada [sambad.in] for experimentation. Total 200 documents are considered for the experiment. The length of the documents varies from a minimum of 16 to a maximum of 35 sentences. The resultant peer summaries are compared with 5 gold standard summaries prepared by linguists who are acknowledged in the acknowledgement section of the manuscript. The input document used is purely textual and monolingual.

4. Proposed Model

The proposed automatic extractive text summarizing system employs both statistical and linguistic methods. Computation on the Odia data set is broadly categorized under the steps preprocessing, word analysis, sentence analysis and sentence extraction. To attain the objective, the model adopts two simple principles, one based on TF-IDF of words, giving emphasis on nouns and verbs. Feature like sentence position value with a modification in its formulation is added to the analysis to mark the performance change. Another method based on clustering (agglomerative hierarchical clustering) is also deployed for the experiment. As document on news data contain maximum number of informative sentences, compression ratio is kept at 50% to avoid the risk of information loss. Finally the performances of all the methods are analyzed and compared.

4.1. *Preprocessing*

The preprocessing step done in this experiment comprises of normalization of the input document i.e. mechanism of transforming text into a single canonical form, tokenization, grammatical tagging of the words, stop word removal and stemming. Some of the normalization rules [Mahapatra (2007)] adopted is given in table 1. It rectifies the prevailing errors like omission and deletion of character, addition or repetition of character, substitution of character and displacement of character.

Table 1. Odia normalization rules

Rule	Example			
	Incorrect	Correct	Transliterated form	Translated form
Space is kept between Noun and Verb/ adjective and Verb	ମନାକରିବା ଲାଜଲାଗିବା	ମନା କରିବା ଲାଜ ଲାଗିବା	manā karibā lāja lāgibā	to deny to feel shy
ନ is kept separate from a word.	ନଗଲେ	ନ ଗଲେ	na gale	not going
ହିଁ and ବି are kept separate from other words.	ତୁମେ ହିଁଯିବ ସେ ବିଖାଇବ	ତୁମେ ହିଁ ଯିବ ସେ ବି ଖାଇବ	tume hĩ jiba se bi khāiba	You will go he/she will also eat
Space should be there between numerical and the word following it.	୧୪ଜଣ	୧୪ ଜଣ	14 jaṇa	14 persons

Filtering (Stop word removal) removes the insignificant words that are frequently used in a sentence formation but are less informative like pronouns, postpositions, conjunctions, question words and punctuations. Examples of such Odia words are shown in table 2.

Table 2. Category of filtered out words

	Example	Transliterated form	Gloss
Pronouns	ସେ, ତୁ	se, tu	he/she, you
Conjunction	ଓ, ବା, କିନ୍ତୁ	o, bā, kintu	and, or, but
Question word	କଣ, କେମିତି, କାହାକୁ	kaṇa, kemiti, kāhāku	what, how, whom
Postposition	ଉପରୁ, ତଳୁ, କୁ	uparu, taḷu, ku	above, below, to
Punctuation	?, !, “,’,:, ;etc		

Stemming is a prerequisite step in this experiment to get accuracy in term frequency. So, after filtration the remaining words are sorted lexicographically. Then each Odia character, including the matras (◌ା, ◌ି, ◌େ, ◌୍ etc.) is analyzed. After analysis we have drawn out a statistic for grouping similar words that differ only in their suffix but have same meaning. To proclaim two words as similar, minimum percentage of similarity is kept at 67% on a dry run basis. The process adopted is a simple technique achieving the objective of a stemmer with an appreciable accuracy. Though there are advanced and commonly used stemmers like Porter stemmer developed for different languages and some Indian languages, but is not applicable for Odia language. The main reason behind it is the unavailability of sufficient electronic Odia corpus.

Dry run:

Input: ଘର, ଘରେ, ଘରୁ,ପିଲା,ପିଲାମାନଙ୍କ

Transliterated form: ghara, ghare, gharu, pilā, pilāmānaṅka

Output: (ଘର:3) (ghara:3); (ପିଲା:2)(pilā:2)

The words “ghara, ghare, gharu” will be considered as similar words and will show frequency count of 3. Similarly the words “pilā” and “pilāmānaṅka” will be considered as similar words and will have a frequency count of 2.

4.2. *Summarization methodology*

4.2.1. Text summarization based on TF- IDF

Word scoring technique TF-IDF a probabilistic method is used for determining the significance of a sentence. The TF-IDF [Christian et al. (2016)] is directly proportional to the frequency of a word in the document (TF) and is offset by the number of documents in the corpus that contain the word. The IDF (Inverse Document Frequency) value intimates with the fact that some words appear more frequent in general and should be filtered out. It is calculated considering a corpus (number of documents relating to a particular domain) relating to the document to be summarized. The proposed model is tested on document related to news on cricket. For calculation of IDF (inverse document frequency) of a word, a corpus of 100 documents of related subject matter is taken. Preprocessing is carried out before applying the summarization model to the text document. Algorithm 1 gives a systematic process of TF-IDF calculation for a word.

Algorithm 1: TF-IDF Process

1. Input text document D_{in}.
2. Preprocessing
   (i)Tokenization
   (ii)POS tagging
   (iii)Remove the stop words.
3. Arrange the remaining words w_i in an alphabetical order.
4. Stem each word w_i.
5. Calculate the TF-IDF score 'sc' of nouns and verbs present in the input document or word weight (sc).

$$sc = TFw_i * IDFw_i \quad (1)$$

$$TF = \frac{frequency\ of\ a\ word\ or\ a\ term\ in\ a\ document}{Total\ words\ in\ a\ document} \quad (2)$$

$$IDF = \log \frac{total\ number\ of\ documents}{1 + number\ of\ documents\ containing\ the\ term} \quad (3)$$

6. The score of each sentence is calculated by taking the summation of TF-IDF values of the nouns and verbs present in that sentence.

$$Sentence\ score(ssc) = \sum sc \quad (4)$$

7. Depending on the compression ratio, required numbers of high scoring sentences are selected.
8. The selected sentences are sorted in accordance with the original summary.

4.2.1.1. Adding sentence position value to the sentence score

Sentence location feature [Baxendale (1958)] is an important criterion for consideration during auto summarization process. The position of a sentence in a document indicates its significance. It is assumed that sentences occurring towards the beginning of the document are positively relevant to the topic [Edmundson (1969)]. Same is the case with news data. Sentences towards the beginning of the document have greater significance unlike other documents like research articles and judgment articles. In research articles the conclusion may contain the important information.

Sentence position value of each sentence is calculated according to the formula $Spv = \frac{1}{\sqrt{sp}}$

[Akter et al. (2017)]

$$Sentence\ score = SSC + Spv \quad (5)$$

The sentence position value of each individual sentence is added to its sentence score 'SSC' and a new sentence score is generated according to Eq. (5).

The output summary is drawn out by selecting the sentences having the higher order significance value, keeping into consideration the compression ratio.

The graph in Fig. 1 gives a sketch of magnitude of change in sentence score on addition of sentence position value.

It has been observed that the result obtained is a biased one suppressing the momentousness of word weights (sc) in a sentence. The summary gets polarized towards the upper portion of the document and has an adverse effect on the performance as far as our documents are concerned.

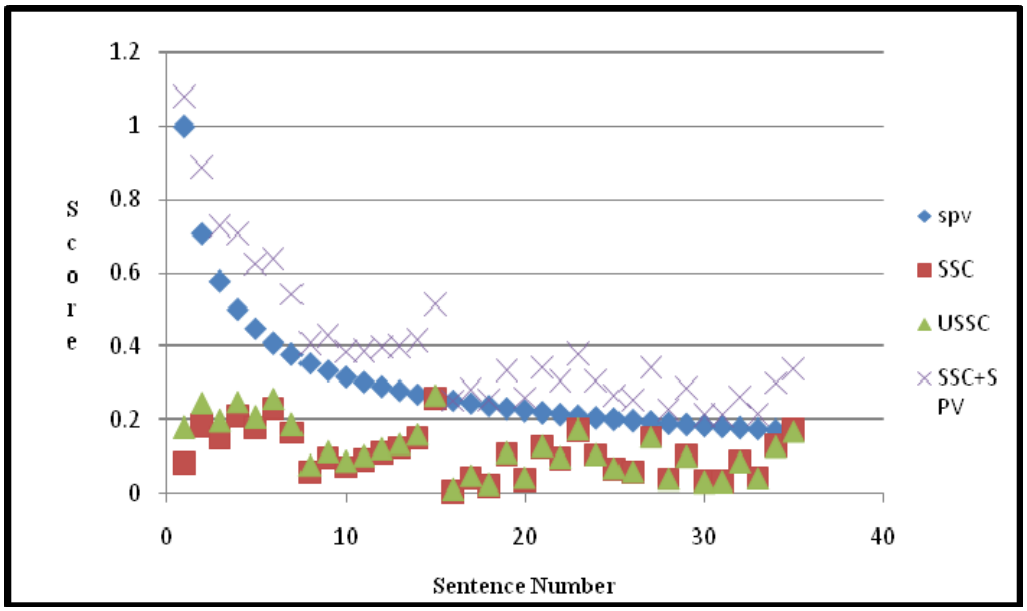

Fig. 1. Comparison of Sentence Scores

So a modification has been proposed in the computation of sentence position value as described in the following section.

4.2.1.2. Modified sentence position value

Our proposed modified model maps sentence position value (Spv) to values ranging from average of sentence score value and 0. The reason behind it is to give justification both to word weight (sc) i.e ultimately sentence score (ssc) and sentence position. The mapping function sets a relationship between average sentence score (x) and sentence position in the input document. After analysis relationship has been established between sentence score and sentence position value and has been mapped accordingly. Updated sentence score (USSC) is now a combination of sentence scores (ssc) and mapped sentence position value (Spvm). The difference in the performance after modification is noticeable from Fig. 1. The change that is visible is not a biased one.

$$spvm = \frac{(k-a)(y-x)}{b-a} + x \tag{6}$$

x: Upper limit of mapping range i.e. mean value of sentence score ssc.

y: Lower limit of mapping range i.e 0

a: Starting sentence position value(spv)

b: Ending sentence position value(spv)

k ϵ [a … b]

$$USSC = ssc + spvm \tag{7}$$

USSC: Updated sentence score value

Spvm: Modified sentence position value

Other commonly used features

Sentence length

Generally sentences which are very short do not add to the informative value of a summary. The TF-IDF value of these sentences is generally low and does not get selected in the summarization process. For our document the parameter sentence length has least effect in the computation of sentence significance.

Presence of Nouns and Verbs

Generally presence of nouns and verbs in a sentence increases its importance. In the computation process TF-IDF of words tagged as nouns and verbs are taken into consideration for determining the significance or the sentence score of a sentence. After analysis by linguists it has been concluded that for news data not only proper nouns but in general nouns and verbs in a sentence increase its significance.

4.2.2. *Text Summarization based on clustering*

Our proposed model employs an agglomerative hierarchical clustering [towardsdatascience.com] with a purpose of getting non-redundant summarized text. The structure adopted by it is more informative than the unstructured set of flat clusters in K means [Shetty and Kallimani (2017)]. The other reason is it is easier to determine the number of clusters from the resultant dendrogram. Our proposed model uses cosine similarity as it can be used in high dimensional positive space and computes information between the objects [wikipedia.org]. Similar clusters are linked through average linkage. The data set has been experimented with different linkages like single, ward, weighted, complete and average linkage. It has been found that the cophenet function for average linkage yields a better result and is distinct from Fig. 2. The experimentation adopting the aforementioned method is novel as far as Odia text data set is concerned. As the data set belongs to news domain and most of the sentences are informative, the compression ratio for the auto summarizer is kept at 50%.

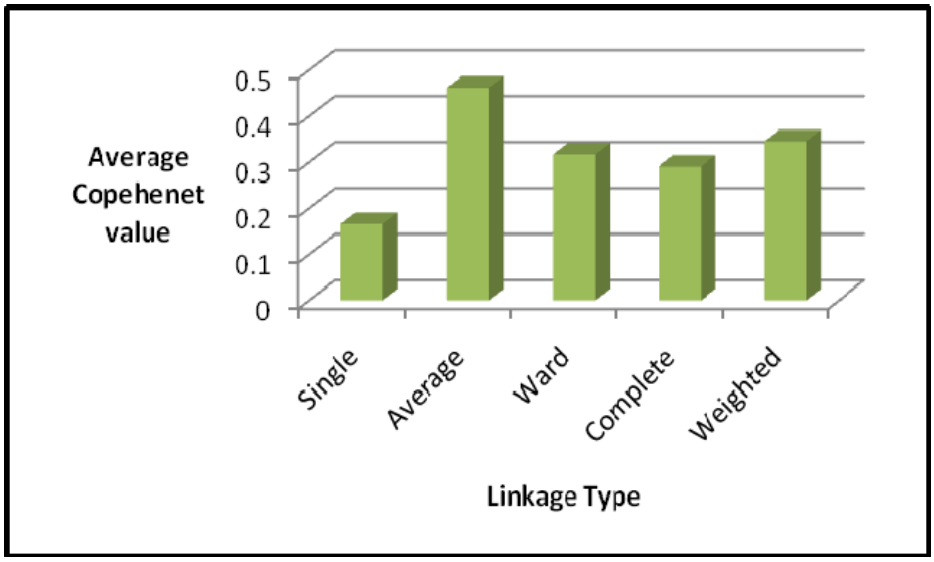

Fig. 2. Comparative analysis of types of Linkages applied on test documents

The Cosine similarity matrix between pair of sentences is calculated using Eq. (8). The output gives the similarity matrix between each pair of sentences.

$$Cos\theta = \frac{s_{ik} \cdot s_{jk}}{\|s_{ik}\| \|s_{jk}\|} = \frac{\sum_{k=1}^{n} s_{ik} \cdot s_{jk}}{\sqrt{\sum_{k=1}^{n} s_{ik}^2} \sqrt{\sum_{k=1}^{n} s_{jk}^2}} \quad (8)$$

Where, S_i and S_j are two sentences.

The cosine similarity matrix is an input for generating clusters.

The overall steps followed by the model after generation of similarity matrix are pictorially depicted in Fig. 3 to give a clear view of the process.

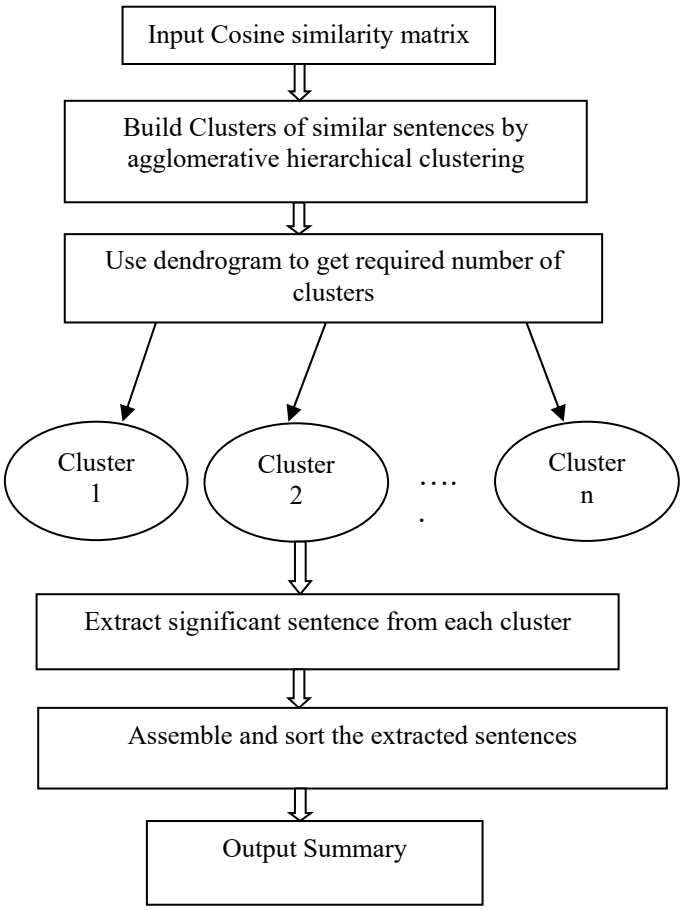

Fig. 3. Extracting non-redundant summary through agglomerative hierarchical clustering

5. Evaluation

The experimental phase is succeeded by an evaluation phase that tests the validity of the process.

The system generated summaries are compared with the gold standard summaries. The data set consisting of 200 documents is divided into 100 test documents and 100 documents for training (required for computing IDF in TF-IDF). The evaluation metric F score that is a combination of precision (p) and recall (r) is considered for the performance check of the summarization model.

$$p = \frac{Number\ of\ sentences\ similar\ in\ system\ generated\ summary\ and\ ideal\ summary}{Number\ of\ sentences\ in\ system\ generated\ summary} \quad (9)$$

$$r = \frac{Number\ of\ sentences\ in\ system\ generated\ summary\ and\ ideal\ summary}{Number\ of\ sentences\ in\ ideal\ summary} \quad (10)$$

$$F\ score = \frac{2.p.r}{p+r} \quad (11)$$

A comparative performance trend of the models deployed on individual documents counted up to 100 can be visualized from Fig. 4. The model considering the modified sentence position value USSC (Ssc+Spvm) shows a better performance trend.

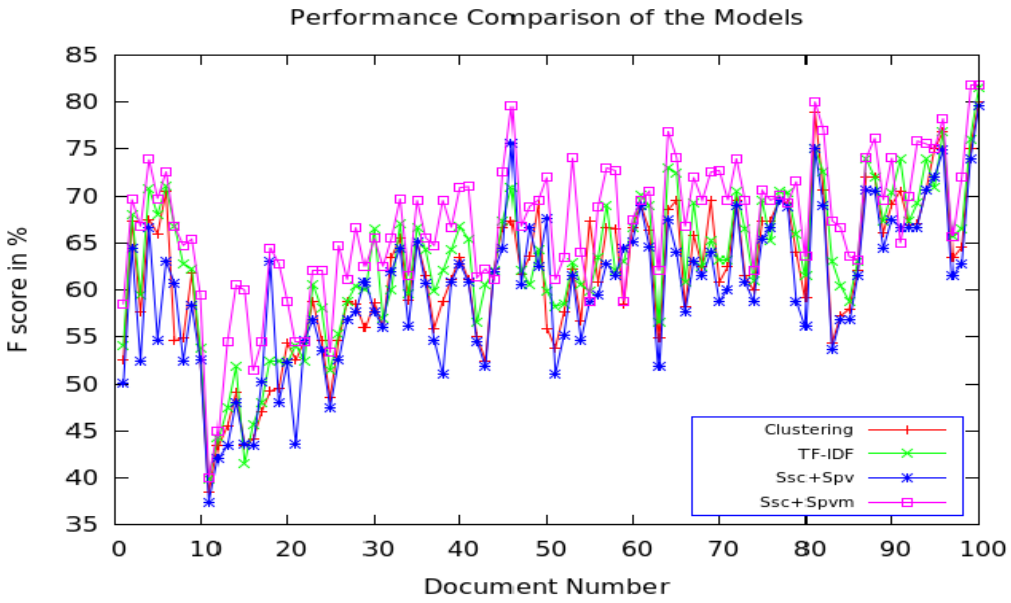

Fig. 4. Performances of the proposed models tested on 100 documents

Fig. 5 shows the pictorial representation of the proposed models performances considering their average F scores. It is clear that the method based on TF-IDF with a modification made to the computation of sentence position value (Spvm) giving an updated sentence score USSC (ssc + Spvm) shows a better performance. The output through clustering though has a lesser F score value than the TF-IDF with modified sentence position value (TF-IDF + Spvm), is successful in reducing redundancy.

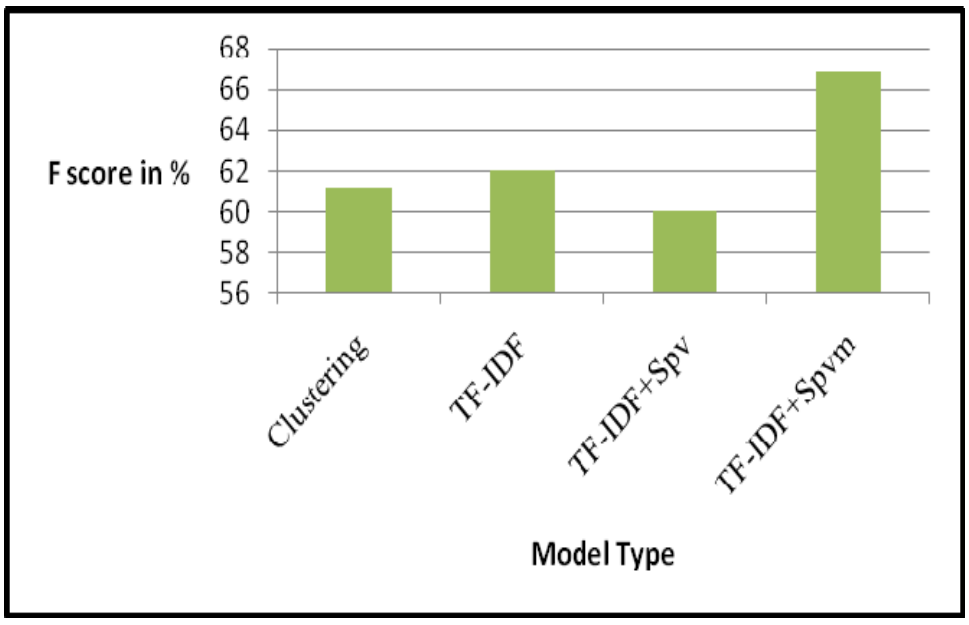

Fig. 5. Performance Comparison of the Models

6. Conclusion and Future Work

In the present state with numerous works done in European languages and Odia lacking behind in the race of computation and automation, the proposed work is a small contribution in the progress of the language. This auto summarization process saves users time and help to carry out the work faster. The proposed technique have efficiently analyzed the language and used statistical methods considering some linguistic features to develop the system. TF-IDF in combination with a modified sentence position feature value has got a leading F score of 66.858%. Similarly the technique based on clustering is efficient as far as redundancy control is concerned, though its F score value is less than the modified TF-IDF model. The morphological richness of the language and lack of NLP tools has placed some impediments in the processing path. But, still the system is successful in giving a satisfactory result. The methods discussed are efficient in their own aspect, but for Indian languages which are morphologically rich need a hybrid approach. There are issues like anaphora resolution, handling structured data and coherency. Our future work will lead towards handling the mentioned issues and development of rich corpus and NLP tools for application of machine learning methods.

Acknowledgments

As the domain of the work includes both linguistics and computer science, so we have taken help from linguists, grammarians and organizations working in Odia language. In this regard we acknowledge the support extended by team Srujanika, Bhubaneswar and Dr Hare Krishna Patra of Kedarnath Gabesana Pratisthana, Bhubaneswar.

References

[1] Abujar, S.; Hasan, M.; Shahin, M. S. I.; Hossain, S. A. (2017, July): A heuristic approach of text summarization for Bengali documentation. In 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), IEEE, pp. 1-8.

[2] Akter, S.; Asa, A. S.; Uddin, M. P.; Hossain, M. D.; Roy, S. K.; Afjal, M. I. (2017, February). An extractive text summarization technique for Bengali document (s) using K-means clustering algorithm. In 2017 IEEE International Conference on Imaging, Vision & Pattern Recognition (icIVPR), IEEE, pp. 1-6.

[3] Al-Sabahi, K.; Zuping, Z.; Nadher, M. (2018): A hierarchical structured self-attentive model for extractive document summarization (HSSAS). IEEE Access, 6, 24205-24212.

[4] Balabantaray, R. C.; Sahoo, B.; Sahoo, D. K.; Swain, M. (2012): Odia text summarization using stemmer. Int. J. Appl. Inf. Syst, 1(3), pp. 2249-0868.

[5] Baxendale, P. B. (1958). Machine-made index for technical literature—an experiment. IBM Journal of research and development, 2(4), pp. 354-361.

[6] Biswas, S.; Acharya, S.; Dash, S. (2015): Automatic text summarization for Oriya language. International Journal of Computer Applications, pp. 975, 8887.

[7] Christian, H.; Agus, M. P.; Suhartono, D. (2016): Single document automatic text summarization using term frequency-inverse document frequency (TF-IDF). ComTech: Computer, Mathematics and Engineering Applications, 7(4), pp. 285-294.

[8] Conroy, J. M.; O'leary, D. P. (2001, September): Text summarization via hidden Markov models. Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, pp. 406-407.

[9] Desai, N. P.; Shah, P. (2016): Automatic text summarization using supervised machine learning technique for Hindi language. International Journal of Research in Engineering & Technology, pp. 361-367.

[10] Edmundson, H. P. (1969): New methods in automatic extracting, Journal of the ACM (JACM) 16.2, pp. 264-285.

[11] Fejer, H. N.; Omar, N. (2014, November): Automatic Arabic text summarization using clustering and keyphrase extraction. In Proceedings of the 6th International Conference on Information Technology and Multimedia, IEEE, pp. 293-298.

[12] Gupta, V. (2013): A survey of text summarizers for Indian Languages and comparison of their performance. Journal of emerging technologies in web intelligence, 5(4), pp. 361-366.

[13] Hu, Y. H.; Chen, Y. L.; Chou, H. L. (2017): Opinion mining from online hotel reviews–a text summarization approach. Information Processing & Management, 53(2), pp.436-449.

[14] Kaikhah, K. (2004, June): Automatic text summarization with neural networks. In 2004 2nd International IEEE Conference on Intelligent Systems. Proceedings (IEEE Cat. No. 04EX791) Vol. 1, IEEE, pp. 40-44.

[15] Kumar, S.; Ram, V. S.; Devi, S. L. (2011): Text extraction for an agglutinative language. Proceedings of Journal: Language in India, pp. 56-59.

[16] Luhn, H. P. (1958): The Automatic Creation of Literature Abstracts, IBM Journal of Research and Development, 2(2), pp.159-165.

[17] Mahapatra, B.P. (2007, March): Prachalita, Odia Bhasara Eka Byakarana, published by Pitambar Mishra, Vidyapuri, Cuttack, First Edition.

[18] Modaresi, P.; Gross, P.; Sefidrodi, S.; Eckhof, M.; Conrad, S. (2017): On (commercial) benefits of automatic text summarization systems in the news domain: a case of media monitoring and media response analysis. arXiv preprint arXiv:1701.00728.

[19] Moratanch, N.; Chitrakala, S. (2016): A survey on abstractive text summarization. 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT), IEEE.

[20] Munot, N.; Govilkar, S. S. (2014): Comparative study of text summarization methods. International Journal of Computer Applications, 102(12).

[21] Nallapati, R.; Zhai, F.; Zhou, B. (2016): Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. arXiv preprint arXiv:1611.04230.

[22] Neto, J. L.; Freitas, A. A.; Kaestner, C. A. (2002, November): Automatic text summarization using a machine learning approach. In Brazilian symposium on artificial intelligence . Springer, Berlin, Heidelberg, pp. 205-215.

[23] Oufaida, H.; Nouali, O.; Blache, P. (2014): Minimum redundancy and maximum relevance for single and multi-document Arabic text summarization. Journal of King Saud University-Computer and Information Sciences, 26(4), pp. 450-461.

[24] Pradhan, K.C.; Hota, B.K.; Pradhan, B. (2006). Saraswat Byabaharika Odia Byakarana, Styanarayan Book Store, Fifth Edition.

[25] Qaroush, A.; Farha, I. A.; Ghanem, W.; Washaha, M.; Maali, E. (2019): An efficient single document Arabic text summarization using a combination of statistical and semantic features. Journal of King Saud University-Computer and Information Sciences.

[26] Rouane, O.; Belhadef, H.; Bouakkaz, M. (2019): Combine clustering and frequent item sets mining to enhance biomedical text summarization. Expert Systems with Applications, 135, pp. 362-373.

[27] Sethi, D. P. (2014): A survey on Odia computational morphology. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) 3.3.
[28] Shetty, K.; Kallimani, J. S. (2017, December): Automatic extractive text summarization using K-Means clustering. In 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT). IEEE, (pp. 1-9).
[29] Siddiqui, T.; Tiwari , U.S. (2008): Natural Language Processing and Information Retrieval, Oxford University Press 343-358.
[30] Svore, K.; Vanderwende, L.; Burges, C. (2007, June). Enhancing single-document summarization by combining RankNet and third-party sources. In Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL), pp. 448-457.
[31] Tseng, Y. H.; Wang, Y. M.; Lin, Y. I.; Lin, C. J.; Juang, D. W. (2007): Patent surrogate extraction and evaluation in the context of patent mapping. Journal of Information Science, 33(6), pp.718-736.
[32] Vijay, S.; Rai, V.; Gupta, S.; Vijayvargia, A.; Sharma, D. M. (2017, December): Extractive text summarisation in Hindi. In 2017 International Conference on Asian Language Processing (IALP), IEEE, pp. 318-321.
[33] https://thesamaja.in, Accessed 1st September, 2020.
[34] https://www.dharitri.com, Accessed 1st September, 2020
[35] https://sambad.in, Accessed 3rd September, 2020.
[36] https://towardsdatascience.com/understanding-the-concept-of-hierarchical-clustering-technique, Accessed 1st October, 2020.
[37] wikipedia.org/wiki/Cosine_similarity, Accessed 4th October, 2020.