INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN 2278 – 0211 (Online)

Federated Document Summarization Using Probabilistic Approach for Kannada Language

Ranganatha S.
Assistant Professor, Govt. Engineering College, Hassan, Karnataka, India
Vinay S. K.
Student, PES Institute of Technology, Bangalore, Karnataka, India
Bhargava H. S.
Student, Govt. Engineering College, Hassan, Karnataka, India

Abstract:
The number of documents and the amount of information available online is being overloaded. From the last one decade information is getting doubled in size leading to the concept of big data; at the same time, it is being saved in unstructured manner. People used to collect huge amount of information related to many issues and areas, whether it is useful or not at that moment, and when it is required to get the needed information out of the collected information, summarization of that particular document can be made. Summaries of large documents will help to find the correct information. In this work, we present a method to produce extractive summaries of documents in Kannada language, limited to the number of sentences mentioned by user. This paper proposes a federated approach to summarization combining Text Rank algorithm and Naïve Bayesian approach. Text Rank uses keyword extraction to rank the sentences with Jaccard's similarity score. The sentences with higher ranks are expected to be a part of summary. Since Text Rank is unsupervised, the proposed work uses Naïve Bayesian to incorporate supervised learning aspects. Training sets are prepared for certain category of Kannada documents, followed by training the system.

Key words: *big data, information, summary, federated, Text Rank, Naïve Bayesian, similarity, supervised, unsupervised*

1. Introduction

The amount of information and the number of documents available online are getting doubled day by day. One of the general causes of information overload is the lack of an effective method for comparing and processing different kinds of incoming information through various sources like telephone, e-mail, instant messaging etc. E-mail remains a major source of information overload with billions of emails sent each day across the globe. In addition to e-mail, the World Wide Web has provided access to billions of pages of information. Though the search engines provide quick access to the relevant information, the information being accessed would be unstructured. There are about 61 million Kannada speakers and around 11000 articles in Kannada Wikipedia. This suggests that tools must be developed to explore, compare and process digital information available in Kannada and other Indian languages. Text document summarization is very important for the native Indian languages. The proposed system implements federated summarization using a probabilistic approach for Kannada language.

The two main methods for text document summarization are keyword extraction and keyword abstraction. Keyword extraction works by copying the information that is very relevant to the summary, abstractive summary reduces the document in volume more effectively than extractive summarization. To include a sentence in the final extractive summary some of the following features may be considered [8] [9]: keywords are determined using Term Frequency (TF) and Inverse Document Frequency (IDF). Words that appear in the title closely reflect the documents' theme and hence, more chances for inclusion of such sentences. Very long and very short sentences are usually not included in the final extractive summary. Text document's first and last sentences of the first and last paragraphs are very important and have greater chances for inclusion in the final summary. Sentences containing proper nouns like person name, place etc., have greater chances for inclusion. Some of the important extractive summarization methods [10][11][12] are Term Frequency-Inverse Document Frequency method (TF-IDF), cluster based method, graph theoretic approach, machine learning approach etc. For abstractive summarization one needs to understand the original text and re-tell it in fewer words. New concepts and themes that best describe the original document are obtained after examining and interpreting the text using linguistic methods.

In this paper, we present an extractive summarization method using federated approach, combining Text Rank and Naïve Bayesian algorithms. The existing summarization systems use single algorithm to mark importance of the sentence and to generate the summary. Attempts are also being made to combine multiple algorithms which work in similar manner (either supervised or unsupervised) and takes into consideration several factors such as position, frequency, parts of speech etc. The proposed method utilizes both the approaches to make the summary effective. There exists more number of unsupervised algorithms compared to supervised approaches because supervised algorithms depend more on the intuition of author and are not expected to work globally. Since the proposed work is implemented for Kannada language, supervised aspects are expected to work locally in an efficient manner. Further, intelligence of unsupervised algorithms which considers different related factors to improve summarization is also incorporated. Combination (federation) of both the approaches (supervised and unsupervised) proves effective and useful for any local language such as Kannada.

The proposed work federates Text Rank algorithm which is unsupervised and Naïve Bayesian algorithm which is supervised to make summary effective. Text Rank uses keywords as the basis for ranking the sentences using jaccard's similarity algorithm. Using similarity scores, Text Rank algorithm assigns a new score called Sentence Rank which is expressed as percentage score. Higher percentage of a sentence means, the sentence is more related to other sentences in a document than the remaining, which makes it to be a part of summary. Naïve Bayesian algorithm which is supervised makes use of efficient training sets to calculate probability of a sentence appearing in a summary. Again, this calculation is keyword based as Text Rank. High probability of a sentence mean, the sentence is expected to be a part of summary as guided by the training sets which involves human intervention to prepare training sets, thus making summary more similar to human summary. The final summary generated by the proposed work combines both the summaries to make it more effective and relative to human summary. The probability value calculated by Naïve Bayesian algorithm is expressed as percentage by multiplying with 100. After that, ranks from Text Rank and probability from Naïve Bayesian are sorted together in non-increasing order. Since the summary generated by proposed system is sentence limited (as mentioned by user), the sentences present in both the summaries (Text Rank and Naïve Bayesian) are extracted first and are removed from the sorted list. If the count of extracted sentences matches with the number given by user, summary is considered more efficient and the process ends there. If the count of extracted sentences does not match with the number given by user, top number – (minus) count sentences are chosen from the sorted list.

2. Literature Survey

Previous work on keyword based Kannada document summarization by Jayashree R, Srikanta Murthy K and Sunny K [1] suggested an extractive summarization algorithm which provides generic summaries. The algorithm uses sentences as the compression basis. Guided by a list of keywords it provided a quick summary of the document; keywords reflected the meaning of the document effectively. Categorized document summarization by the same team [2] was produced by considering the documents from five different categories. With a limit on the number of sentences in a given document meaningful summary was produced. Similarly, another approach by Letian Wang and Fang Li [3] extracted the key phrases using chunk based method. Key phrases from the candidates were selected based on the keywords of the documents. You Ouyang [4] present a method for extracting the most important words and then expanding the core words as the target key phrases by word expansion. The work of Su Nam Kim [5] automatically produces the key phrases for each scientific paper. They compiled a set of 284 scientific articles with key phrases effectively chosen by both authors and readers. Extractive summaries [6] based on statistical analysis of individual or mixed surface level features like word or phrase frequency are formulated by extracting key text segments either sentences or passages from the text. The content is either treated as "most frequent" or "most favourably positioned". The approach avoids efforts in understanding the text. Michael. J . Paul [7] presents an unsupervised probabilistic approach to model and extract multiple viewpoints in text. The information of the word position plays a significant role in document summarization.

One more method by Mari-SannaPaukkeri [13] selects words and phrases describing the meaning of the documents, where it compares the ranks of frequencies in the documents with the corpus considered as reference corpus. The work of Gabor Berend [14] is a frame work called "SZETERGAK system" which treats the reproduction of reader assigned keywords as supervised learning task. In this approach token sequences were used as classification instances. The two approaches for document summarization are supervised and unsupervised methods. In case of supervised approach, a model is trained in order to determine whether a candidate phrase is a key phrase or not. In case of unsupervised approach graph based methods first build a word graph as per word co- occurrences within the document and then random walk techniques are used to measure the importance of a word [15]. Given the extractive summary of a training document, summarization process can be modelled as classification problem. Classification of the sentences based on the features they possess are summary sentences and non-summary sentences. Given the training data, using the Bayes' rule classification, probabilities are learnt statistically [16].

3. Methodology

The proposed system uses 2 different algorithms to generate summary. The summaries generated by the algorithms are combined in a particular way to produce efficient summary. The methodology is explained in three sections, one for each Text Rank, Naïve Bayesian and Federated summary generation technique. As explained earlier in the introduction part, Text Rank is unsupervised and Naïve Bayesian is probabilistic.

3.1. Algorithm: Text Rank

This algorithm takes input file in the text format and produces sentence limited summary.

Input: File; number of sentences required in the output – 'm'
Output: Summary containing required number of sentences.
Logic:

Prepare the input file
Read from file and build array of sentences
For each sentence from the file:
Extract words into array
For each word in the array:
Apply stemming and store back the root word
For each sentence having root words
Eliminate stop words

For i=1 to n:
For j=1 to n:
Similarity (i, j) = common words in sentence (i,j) / Math.log (total words in sentence i) + Math.log (total words in sentence j)

For i= 1 to n:
Rank(i) = $\sum_{j=1}^{n}$ similarity (i,j)
Sort the Rank vector in descending order
Get the position of top 'm' sentences
Sort the position vector in ascending order
Extract all the sentences in position vector from the original input file

In this algorithm, it creates a similarity matrix for each sentence which contains similarity rank by comparing a sentence to all other sentences.
Once the similarity matrix is computed, the row sum needs to be calculated to assign ranks for each sentence. Similarity score will be calculated by using the below formula:
Similarity (i, j) = common words in sentence (i,j) /
Math.log (total words in sentence i) + Math.log (total words in sentence j)
The row sum will be calculated as:

$$Rank(i) = \sum_{j=1}^{n} similarity\ (i,j)$$

3.2. Algorithm: Naive Bayesian
This algorithm takes input file in the text format and produces sentence limited summary.
Input: File; number of sentences required in the output – 'm'
Output: Summary containing required number of sentences.
Logic: The logic part is implemented in two parts, one for training and one for testing phase.

- **Training:** In the training phase, proposed system is prepared with a set of documents which have manual summary. All manual summaries for a category are aggregated into a single file to apply stemming. Stemmed file is stored back for further reference. Unique words appearing in the generated file is stored in a structure called wordbank.

For each category chosen:
For each document within category:
Aggregate manual summaries into a single file
Stem the aggregated file for root words
Store the root words in 'wordbank'
Store the stemmed sentences of manual summary

- **Testing.** This phase calculates the score for each sentence of given file using Naïve Bayesian rule. Since Naïve Bayesian works on the principle of conditional probability, three components are needed. The following algorithm assumes one component to be 1 always and remaining components are calculated as 'p' and 'q'.

```
Prepare the input file
Read from file and build array of sentences
For each sentence 's' from the file:
  Stem the sentence 's' for root words
  Eliminate stop words

  For i=1 to length(aggregated_manual_summary):
    sim(i) = Similarity('s', manual summary(i))

  p = max(sim)
 q = length(Intersection(words('s'), 'wordbank'))

  Rank = (1 * p ) /q

Sort the Rank vector in descending order
Get the position of top 'm' sentences
Sort the position vector in ascending order
Extract all the sentences in position vector from the original
input file
```

The below formula shows Naïve Bayesian approach to ranking sentences

$$P(s \in S \mid F1, F2\ldots.Fn) = P(F1, F2...FN \mid s \in S) * P(s \in S) \; / \; P(F1, F2\ldots\ldots.Fn)$$

The Naïve Bayesian approach contains three components:

- **P (F1, F2… Fn | s ϵ S):** Probability of the features of the sentence given that sentence's' has appeared in summary 'S'. Since the generation of summary is based on words, this component is assumed to be true always. Therefore, this component is replaced with the value 1 in the algorithm.
- **P (s ϵ S):** Probability of the sentence's' to appear in summary 'S'. It is calculated using the similarity with manual summaries. Similarity score is calculated between the sentence's' and all of the sentences of manual summary. The maximum of these scores is taken for this component.
- **P (F1, F2… Fn):** Probability of the features in the sentence's'. It is calculated as an intersection between the 'wordbank' and the words in the sentence's'.

3.3. Algorithm: Federated Summary Generation

This algorithm takes input file in the text format and produces sentence limited summary.
Input: File; number of sentences required in the output – 'm'
Output: summary containing required number of sentences.
Logic:

```
Prepare the input file
Generate summary by Text Rank algorithm
Store ranks for  sentences
Generate Summary by Naive Bayesian algorithm
Store ranks for sentences
Common = Select the common sentences from the summaries generated by Text Rank and Naive
Bayesian algorithm
Count = no. of sentences selected = length(Common)
Eliminate the selected sentences from Text Rank and Naïve Bayesian summaries.
Merge the remaining sentences with score from both Text Rank and Naïve Bayesian
Sort the sentences according to score
Remaining = Choose top m-(minus)count sentences
Combine Common and Remaining sentences chosen
Summary = Common + Remaining
```

4. Results

The proposed algorithms are executed against documents of five different categories chosen from web dunia and other Kannada portals. The five categories chosen are:

- Cricket (Sport)
- Astrology
- Karnataka Darshan (Tourism)
- Literature
- Religious journey (Philosophy)

The notations A1, A2...A10 denote article numbers. The numbers shown in the table indicate the efficiency of the algorithm with respect to manual summary. Figure 1 shown at the end of this section illustrates the comparison of efficiencies of all the algorithms.

Category /Article	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	Avg
Cricket	45	53	47	40	38	60	58	48	42	40	**47.89**
Astrology	50	62	75	50	50	53	27	40	50	50	**50.78**
Kamataka Darshan	66	40	66	25	80	57	42	50	57	60	**53.67**
Literature	57	54	50	30	38	50	66	53	64	53	**51.33**
Religious Journey	57	50	66	14	50	28	45	50	50	66	**45.56**

Table 1: Result of Naïve Bayesian summary

Category /Article	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	Avg
Cricket	62	60	41	30	33	65	70	60	57	46	**52.4**
Astrology	60	62	75	66	42	40	38	50	50	41	**52.4**
Kamataka Darshan	66	40	83	33	80	57	28	50	42	80	**55.9**
Literature	52	54	58	40	27	66	58	38	71	61	**52.5**
Religious Journey	57	25	66	28	50	14	45	40	50	66	**44.1**

Table 2: Result of Text Rank summary

Category /Article	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	Avg
Cricket	74	67	55	51	64	71	77	66	62	56	**64.3**
Astrology	65	69	84	73	58	61	59	58	67	54	**64.8**
Kamataka Darshan	76	47	88	43	100	71	52	61	68	87	**69.3**
Literature	67	64	72	65	58	79	80	74	81	77	**71.7**
Religious Journey	71	65	77	52	69	51	64	71	75	89	**68.4**

Table 3: Result of federated summary

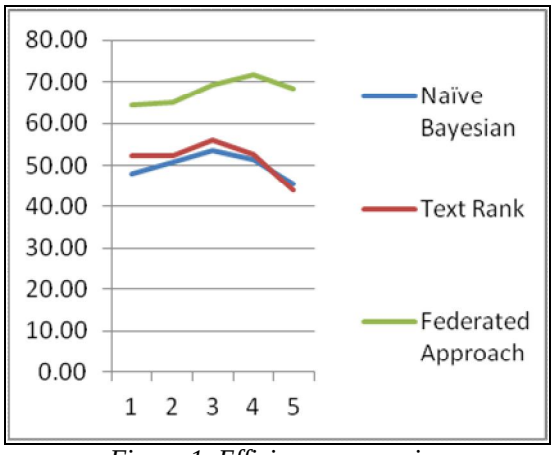

Figure 1: Efficiency comparison

5. Conclusion

The current implementation attempted to combine (federate) one algorithm from supervised learning category (Naïve Bayesian) and one from unsupervised learning category (Text Rank) to produce efficient summary. From the experiment done on 50 documents from five different categories of Kannada language, demonstrates that the proposed Federated algorithm's results are more efficient when compared to experimenting with individual Naïve Bayesian and Text Rank algorithm. The results can be improved even more by selecting other efficient algorithms from supervised and unsupervised learning category.

For supervised learning algorithms, availability of good classifiers and NLP support for language affects the results. For kannada language there are no standard stemmers, stop-words and classifiers. Unsupervised algorithms are heavily based on NLP support available for that language and the efficiency of the algorithm itself. The proposed algorithm tries to utilize available support from both the categories and produces efficient summary.

6. References

1. Jayashree. R, Srikanta Murthy.K and Sunny.K , "Document Summarization in Kannada using keyword Extraction", David Bracewell, AIAA 2011,CS & IT 03, pp. 121–127 , 2011.
2. Jayashree. R, Srikanta Murthy.K and Sunny.K, "Keyword Extraction based Summarization of Categorized Kannada Text Documents", International journal on soft computing (IJSC), Vol.2, N0.4, November 2011.
3. Letian Wang, Fang Li, SJTULTLAB: " Chunk Based Method for Keyphrase Extraction", Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010,pp 158– 161,Uppsala, Sweden,15-16 July 2010.
4. You OuyangWenjie Li Renxian Zhang,'273. Task 5. "Keyphrase Extraction Based on Core Word Identification and Word Expansion", Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010, pp 142–145, Uppsala, Sweden, 15-16 July 2010.
5. Su Nam Kim,Ä Olena Medelyan,~ Min-Yen Kan and Timothy BaldwinÄ, "Automatic Keyphrase Extraction from Scientific Articles' ", Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010, pages 21–26,Uppsala, Sweden, 15-16 July 2010.
6. Farshad Kyoomarsi, Hamid Khosravi, Esfandiar Eslami and Pooya Khosravyan Dehkordy, "Optimizing Text Summarization Based on Fuzzy Logic", Proceedings of Seventh IEEE/ACIS International Conference on Computer and Information Science, IEEE, University of Shahid Bahonar Kerman, UK, 347-352, 2008.
7. Michael J. Paul,ChengXiang Zhai,Roxana Girju, "Summarizing Contrastive Viewpoints in Opinionated Text", Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pages 66–76,MIT, Massachusetts, USA, 9-11 October 2010.
8. Fang Chen, Kesong Han and Guilin Chen, "An Approach to sentence selection based text summarization", Proceedings of IEEE TENCON02, 489-493, 2002.
9. Rasim M. Alguliev and Ramiz M. Aliguliyev, "Effective Summarization Method of Text Documents", Proceedings of IEEE/WIC/ACM international conference on Web Intelligence (WI'05), 1-8, 2005.
10. Madhavi K. Ganapathiraju, "Overview of summarization methods", 11-742: Self-paced lab in Information Retrieval, November 26, 2002.
11. Klaus Zechner, "A Literature Survey on Information Extraction and Text Summarization", Computational Linguistics Program, Carnegie Mellon University, April 14, 1997.
12. Berry Michael W., "Automatic Discovery of Similar Classification and Retrieval", Springer Verlag, New York, LLC, 24-43, 2004.
13. Mari-SannaPaukkeri and TimoHonkela, "Likey: Unsupervised Language-independent Key phrase Extraction", Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010, pp 162–165,Uppsala, Sweden, 15-16 July 2010.
14. Gabor Berend,Rich´ardFarkasSZTERGAK : " Feature Engineering for Key phrase extraction", Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010, pp 186–189,Uppsala,Sweden, 15-16 July 2010.
15. ZhiyuanLiu,Wenyi Huang, YabinZheng and MaosongSun, "Automatic Keyphrase Extraction via Topic Decomposition", Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pp 366–376,MIT, Massachusetts, USA, 9-11 October 2010.
16. Joel larocca Neto, Alex A. Freitas and Celso A.A.Kaestner, "Automatic Text Summarization using a Machine Learning Approach", Book: Advances in Artificial Intelligence: Lecture Notes in computer science, Springer Berlin / Heidelberg, Vol 2507/2002, 205-215, 2002