DOCUMENT SUMMARIZATION IN KANNADA USING KEYWORD EXTRACTION

Jayashree.R[1], Srikanta Murthy.K[2] and Sunny.K[1]

[1]Department of Computer Science, PES Institute of Technology, Bangalore, India
jayashree@pes.edu, sunfinite@gmail.com
[2]Department of Computer Science, PES School of Engineering, Bangalore, India
srikantamurthy@pes.edu

ABSTRACT

The internet has caused a humongous growth in the amount of data available to the common man. Summaries of documents can help find the right information and are particularly effective when the document base is very large. Keywords are closely associated to a document as they reflect the document's content and act as indexes for the given document. In this work, we present a method to produce extractive summaries of documents in the Kannada language. The algorithm extracts key words from pre-categorized Kannada documents collected from online resources. We combine GSS (Galavotti, Sebastiani, Simi) coefficients and IDF (Inverse Document Frequency) methods along with TF (Term Frequency) for extracting key words and later use these for summarization. In the current implementation a document from a given category is selected from our database and depending on the number of sentences given by the user, a summary is generated.

KEYWORDS

Summary, Keywords, GSS coefficient, Term Frequency(TF), IDF(Inverse Document Frequency) and Recall

1. INTRODUCTION

With the growth of the internet, a large amount of data is available online. There is a demanding need to make effective use of data available in native languages. Information Retrieval [IR] is therefore becoming an important need in the Indian context. India is a multilingual country, any new method developed in IR in this context needs to address multilingual documents.

A very important task in Natural Language Processing is Text Summarization. Given a document or collection of documents, providing a quick and concise summary is very important. There are two main techniques for Text Document Summarization: extractive summary and abstractive summary. While extractive summary copies information that is very important to the summary, abstractive summary condenses the document more strongly than extractive summarization and require natural language generation techniques. In this paper, we present an extractive summarization algorithm which provides generic summaries. The algorithm uses sentences as the compression basis.

David Bracewell, et al. (Eds): AIAA 2011,CS & IT 03, pp. 121–127 , 2011.
 DOI : 10.5121/csit.2011.1311

There are around 50 million Kannada speakers and more than 10000 articles in Kannada Wikipedia. This warrants us to develop tools that can be used to explore digital information presented in Kannada and other native languages.

Keywords/phrases, which are a very important component of this work, are nothing but expressions; single words or phrases describing the most important aspects of a given document. The list of keywords/phrases aims to reflect the meaning of the document. Guided by the given keywords/phrases, we can provide a quick summary, which can help people easily understand what a document describes, saving a great amount of time and thus money. Consequently, automatic text document summarization is in high demand. Meanwhile, summarization is also fundamental to many other natural language processing and data mining applications such as information retrieval, text clustering and so on [11][2].

2. LITERATURE SURVEY

Previous work on keyphrase extraction by Letian Wang and Fang Li [3] has shown that it can be achieved using chunk based method. Keywords of document are used to select key phrases from candidates. Similarly, another approach by Mari-Sanna Paukkeri et al[2] selects words and phrases that best describe the meaning of the documents by comparing ranks of frequencies in the documents to the reference corpus. The SZETERGAK system by Gabor Berend[1] is a frame work that treats the reproduction of reader assigned keywords as a supervised learning task. In this work, a restricted set of token sequences was used as classification instances. One more method of You Ouyang[4] extracted the most essential words and then expanded the identified core words as the target key phrases by word expansion approach. A novel approach to key phrase extraction proposed by them consists of two stages: identifying core words and expanding core words to key phrases. The work of automatically producing key phrases for each scientific paper by Su Nam Kim et al[5] has compiled a set of 284 scientific articles with key phrases carefully chosen by both their authors and readers, the task was to automatically produce key phrases for each paper. Fumiyo Fukumoto[6] present a method for detecting key sentences from the documents that discuss the same event. To eliminate redundancy they use spectral clustering and classified each sentence into groups each of which consists of semantically related sentences. The work of Michael . J . Paul et al[7] use an unsupervised probabilistic approach to model and extract multiple viewpoints in text. The authors also use Lex rank, a novel random walk formulating to score sentences and pairs of sentences from opposite view points based on both representativeness of the collections as well as their contrast with each other. The word position information proves to play a significant role in document summarization. The work of You Ouyang [8] et al illustrates the use of word position information, the idea comes from assigning different importance to multiple words in a single document .Cross Language document summary is another upcoming trend that is growing in Natural Language Processing area. There was a proposal by Xiaojun Wan et al [9] to consider the translation from English to Chinese. First the translation quality of each English sentence in the document set is predicted with the SVM regression method and then the quality score of each sentence is incorporated into the summarization process; finally English sentences with high translation scores are translated to form the Chinese summary. There have been techniques which use A* algorithm to find the best extractive summary up to given length, which is both optimal and efficient to run. Search is typically performed using greedy technique which selects each sentence in the decreasing order of model score until the desired length summary is reached [10]. There are two approaches to document summarization, supervised and unsupervised methods. In supervised approach, a model is trained to determine if a candidate phrase is a key phrase. In unsupervised method graph based methods are state-of-the art. These methods first build a word graph according to word co occurrences within the document and then use random walk techniques to measure the importance of a word [12].

3. METHODOLOGY

The methodology adopted by us can be described as consisting of three major steps:

3.1. Crawling

The first step is creating the Kannada dataset. Wget , a Unix utility tool was used to crawl the data available on **http://kannada.webdunia.com**. Data was pre-categorized on this web site.

3.2. Indexing

Python was the language of choice. The indexing part consisted of removing HTML mark up; English words need not be indexed for our work. Beautiful Soup is a python HTML/XML parser which makes it very easy to scrape a screen. It is very tolerant with bad markup. We use Beautiful Soup to build a string out of the text on the page by recursively traversing the parse tree returned by Beautiful Soup. All HTML and XML entities (ಅ : ಅ , < : <) are then converted to their character equivalents. Normal indexing operations involve extracting words by splitting the document at non-alphanumeric characters, however this would not serve our purpose because dependent vowels (ಾ , ೂ etc.) are treated as non-alphanumeric, so splitting at non-alphanumeric characters would not have worked for tokenization. Hence a separate module was written for removing punctuations. Documents in five categories were fetched: sports, religion, entertainment, literature, astrology . The next step is to calculate GSS coefficients and the Inverse Document Frequency (IDF) scores for every word (in a given category in the former case). Every word in a given document has a Term Frequency(TF),which gives the number of occurrence of a term in a given document, defined by:

TF= frequency of a term in a document / number of terms in a given document.

IDF= Log_{10} (N / n)

where, N is the total number of documents indexed across all categories.

and n is the number of documents containing a particular term.

Hence TF and IDF are category independent. Also GSS coefficients which evaluate the importance of a particular term to a particular category are calculated. GSS(Galavotti- Sebastiani-Simi) co-efficient [13] is a feature selection technique used as the relevance measure in our case. Given a word w and category c it is defined as:

$$f(w , c) = p(w , c) * p(w' , c') - p(w ', c) * p(w , c')$$

where,

p(w, c) is the probability that a document contains word w and belongs to category c

p(w' , c') is the probability that a document does not contain w and does not belong to c

p(w' , c) is the probability that a document does not contain w and belongs to c

p(w , c') is the probability that a document contains w and does not belong to c

GSS coefficients give us words which are most relevant to the category to which the documents belong. IDF gives us words which are of importance to the given documents independently. Thus using these two parameters to determine relevant parts of the document provides a Wholesome summary**.**

3.3. Summarization

Given a document and a limit on the number of sentences, we have to provide a meaningful summary. We calculate the GSS coefficients and IDF of all the words in the given document (for stop words see below), if the document is already present in our database, GSS coefficients and IDF values are already calculated offline. These values are then multiplied by the TF of the individual words to determine their overall importance in the document. We then extract top n keywords from each of the lists (GSS coefficients and IDF). Then sentences are extracted from the given document by retrieving Kannada sentences ending with full stops. Due care is taken to see that full stops which do not mark the end of a sentence (ಡಾ. etc.) are not considered as split points. Each of these sentences is then evaluated for the number of keywords it contains from the list as follows:

$$\text{Rank of sentence} = \frac{\text{number of keywords contained by the sentence from both the lists}}{\text{total number of sentences in the document}}$$

The top m sentences are then returned, where m is the user specified limit .

4.TESTS

The following lists were obtained by running our algorithm with a sports article on cricket as input with n=20 :

GSS co-efficient list :

ಐಪಿಎಲ್, ಆಧ್ಯತೆ, ನನ್ನ, ಆಡುವುದು, ದೇಶಕ್ಕಾಗಿ, ಕ್ರಿಕೆಟ್, ಉತ್ತಪ್ಪ, ಸುದ್ದಿಗಳಿಗೆ, ಲೀಗ್, ಪ್ರತಿನಿಧಿದೇಸುವುದೇ, ಪ್ರೀಮಿಯರ್, ನಿರ್ಣಾಯಕ, ತಂಡದಲ್ಲಿ, ಇಂಡಿಯನ, ವಿಶ್ವಕಪ್, ಏಕದಿನ, ಕರ್ನಾಟಕ, ಭಾರತ, ನಲ್ಲಿ, ಸಂಭವನೀಯರ

IDF list :

ಐಪಿಎಲ್, ಕ್ರಿಕೆಟ್, ವಿಶ್ವಕಪ್, ನಲ್ಲಿ, ಮತ್ತಷ್ಟು , ಮೊದಲ, ಭಾರತ, ಕ್ರೀಡಾ, ಜಗತ್ತು , ಲೇಖನಗಳು, ಕ್ರಿಕೆಟಿಗರು , ಏಕದಿನ, ಅಂಕಿಅಂಶ, ಟಿ , ಟಿಕರ್, ಶೋಧಿಸು, ಸಹ, ಇದನ್ನು, ಮುಖ್ಯ, ಪುಟ

The following lists were obtained with a blog entry about another blog on films(category: literature) as input with n = 20 :

GSS co-efficient list:

ಈ, ಲೇಖನ, ವೆಬ್, ಕವನ, ಬ್ಲಾಗ್, ವಿವಿಧ, ಸಾಹಿತ್ಯ, ವಾರದ, ದುನಿಯಾ, ಅವರು, ನಮ್ಮ, ಮತ್ತು, ಎಂದು, ಕಥೆಗಳು , ಖ್ಯಾತ, ಪುಟ, ಸಾಹಿತಿಗಳು , ಅವರ, ನಾವು , ಎಂದು

IDF list:

ಕೃಷ್ಣ, ಪಿಚ್ಚರ್, ವೆಬ್, ಬ್ಲಾಗ್, ವಾರದ, ದುನಿಯಾ, ಇಲ್ಲಿ, ಲೇಖನ, ಸಿನಿಮಾ, ಅವರು, ಈ, ವಿಕಾಸ, ಮಚ್ಚು, ಕ್ರೌರ್ಯ, ಭೂಗತ, ಪತ್ರಕರ್ತ, ಬ್ಲಾಗಿನ, ಕಾಣುತ್ತಿದ್ದ, ಎತ್ತಿ, ಬ್ಲಾಗು

As evident from the lists , stop word (noise) removal is essential.

5. NOISE REMOVAL

Words considered as noise should not be evaluated as keywords. To remove stop words we have implemented an algorithm, which takes a stop word as input and finds structurally similar words and adds them to the stop word list.

Some of the words in our primary list of stop word which is created and maintained manually are

ನನ್ನ, ನಿನ್ನ, ಇದು, ಅದು, ಯಾಕೆ, ಹೇಗೆ, ಆದರೆ, ಮತ್ತು, ಹೋಗು, ನೀನು, ನೀವು , ಇತ್ತು , ಮಾಡು , ಈ , ಆ , ಅಲ್ಲಿ , ಎಲ್ಲಿ, ಹಾಗೂ, ಎಂಬ, ಅಥವಾ ,ನಲ್ಲಿ , ಇಲ್ಲ , ಬಾ , ಏನು, ಆಗದೆ , ತಾನು , ಇವರಿಗೆ , ಅಂದರೆ , ಈಗ , ಅಂಥ

5.1 Finding structurally similar words for words in the primary list :

Example: Consider the word 'ಯಾಕೆ '

When split into individual Unicode characters,
it becomes : ಯ (U+0CAF) +ಾ (U+0CBE) + ಕ (U+0C95) + ೆ(U+0CC6). The vowel sound at the end is not considered as an alphanumeric character. So our similarity module does the following in order:

1. Fuzzy search for words which contain the unmodified word at the beginning.
2. Strip all the non-alphanumeric characters at the end of the word and then fuzzy search for words which contain the modified word at the beginning.

A sample of stop words that were obtained by our algorithm: ಯಾಕೆಂದರೆ , ಯಾಕೋ, ಯಾಕಿಷ್ಟು , ಯಾಕ್ಸ್,ಯಾಕಾಗಬಾರದು , ಯಾಕೂಬ್ etc.

For the personal pronoun 'ನನ್ನ' , some of the words obtained are: ನನ್ನಿಂದ, ನನ್ನನ್ನ, ನನ್ನವಳು, ನನ್ನಜ್ಜನಿಗೊಂದಾನೆಯಿತ್ತು, ನನ್ನೊಂದಿಗೇ, ನನ್ನಾಸೆ etc.

As evident, though some words have semantic relationship to the primary stop word , a lot of words have no such relationship and further work needs to be done to find methods which will prevent such words from being penalized as stop words. Starting with a basic list of stop words, this program can be used to find structurally similar words and semantically unrelated words can be manually removed from the stop words list.

6. RESULTS

Table 1. Evaluation-I Results.

Category	Human 1 Score	Human 2 Score	Human 3 Score	Average
Literature	0.4	0.5	0.2	0.37
Entertainment	0.5	0.2	0.6	0.43
Astrology	0.8	0.7	0.7	0.73
Sports	0.4	0.4	0.4	0.40

Table 2. Evaluation-II Results.

Category	Score
Literature	0.7
Entertainment	0.8
Astrology	0.8
Sports	0.76

Evaluation-I gives the result of a manual evaluation of the summarizer with three different human summaries across various categories. Three different human summarizers were asked to create reference summaries for random documents in each category. The same documents were then fed to the program and the limit was kept at m=10.The number of sentences common between the two summaries gives the relevance score; the average of three scores is shown for each document.

Evaluation-II consisted of a single human reference summary for a data set consisting of 20 documents.

It is evident that evaluating summaries is a difficult task, as it is not deterministic. Different people may choose different sentences and also same people may choose different sentences at different times. Paraphrasing is another issue to be considered. However in our test, sentences were selected as a whole both by the machine and human summarizers. Sentence recall measure is used as the evaluation factor.

7. CONCLUSION

Though we are working only on pre categorized data; there are good classifiers which can provide the necessary classification. The work can be extended to first classify a given document and then create a summary. There is no standard stop word list for Kannada, or methods to do that. Hence a given procedure in this work can be used as a stop word removal method. The summarizer can be used as a tool in various organizations such as Kannada Development Authority, Kannada Sahitya Parishath etc.

REFERENCES

[1] Gabor Berend,Rich´ard FarkasSZTERGAK : Feature Engineering for Keyphrase Extraction ,Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010, pages 186–189, Uppsala, Sweden, 15-16 July 2010.

[2] Mari-Sanna Paukkeri and Timo Honkela,'Likey: Unsupervised Language-independent Keyphrase Extraction,Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010, pages 162–165,Uppsala, Sweden, 15-16 July 2010. 2010 Association for Computational Linguistics.

[3] Letian Wang, Fang Li, SJTULTLAB: Chunk Based Method for Keyphrase Extraction, Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010,pages 158–161,Uppsala, Sweden, 15-16 July 2010.Association for Computational Linguistics.

[4] You Ouyang Wenjie 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, pages 142–145, Uppsala, Sweden, 15-16 July 2010.Association for Computational Linguistics.

[5] Su Nam Kim,Ä Olena Medelyan,~ Min-Yen Kan} and Timothy BaldwinÄ,'SemEval-2010 Task 5: 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. Association for Computational Linguistics

[6] Fumiyo Fukumoto Akina Sakai Yoshimi Suzuki,'Eliminating Redundancy by Spectral Relaxation for Multi-Document Summarization',Proceedings of the 2010 Workshop on Graph- based Methods for Natural Language Processing, ACL 2010, pages 98–102,Uppsala, Sweden, 16 July 2010. Association for Computational Linguistics

[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. Association for Computational Linguistics

[8] You Ouyang ,Wenjie Li Qin Lu Renxian Zhang, 'A Study on Position Information in Document Summarization', Coling 2010: Poster Volume, pages 919–927, Beijing, August 2010

[9] Xiaojun Wan, Huiying Li and Jianguo Xiao,'Cross-Language Document Summarization Based on Machine Quality Prediction',Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 917–926,Uppsala, Sweden, 11-16 July 2010.Association for Computational Linguistics.

[10] Ahmet Aker Trevor Cohn,'Multi-document summarization using A* search and discriminative training',Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pages 482–491,MIT, Massachusetts, USA, 9-11 October 2010. Association for Computational Linguistics

[11] Hal Daum´e III*,Daniel Marcu*,'Induction ofWord and Phrase Alignments for Automatic Document Summarization',2006 Association for Computational Linguistics.

[12] Zhiyuan Liu,Wenyi Huang, Yabin Zheng and Maosong Sun,'Automatic Keyphrase Extraction via Topic Decomposition',Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pages 366–376,MIT, Massachusetts, USA, 9-11 October 2010.Association for Computational Linguistics

[13] L. Galavotti, F. Sebastiani, and M. Simi,'Experiments on the use of feature selection and negative evidence in automated text categorization' Proc.4th European Conf. Research and Advanced Technology for Digital Libraries, Springer Verlag, pp. 59-68, 2000.