International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 10 (2015) pp. 25511-25519

http://www.ripublication.com

Text Summarization using Clustering Technique and SVM Technique

ShivaKumar KM and Soumya R

Dept.of Computer Science, Amrita Vishwavidyapeetham, Mysore campus
MCA Student, Amrita Vishwavidyapeetham, Mysore campus
3Sushma K Prasad, Amrita Vishwavidyapeetham, Mysore campus

Abstract

The Text Summarization is one of the problem under Natural Language Processing.This system which gives a single summarized document from multiple related documents. The summarizer provides an accurate result to the input query in the form of a precise text document by analyzing the text from various text document clusters. There are two methodologies- Clustering and Support Vector Machine (SVM) are used to solve this NLP problem.The present text summarizer system uses either SVM or Clustering technique. In this work we propose a Hybrid approach to serve our purpose by cascading both techniques to get an improved summary of data on related documents. We pre process the documents to get tokens obtained after stemming and stop word removal. The hybrid approach helps in summarizing the text documents efficiently by avoiding redundancy among the words in the document and ensures highest relevance to the input query.The guiding factors of our results are the ratio of input to output sentences after summarization.

Keywords: NLP, Summarization, Sentence Score, Word count,cluster,SVM, tokens, stemming, Frequency.

I. Introduction-

Text summarization has become very significant from many years. In the early days storage for large data files was expensive. Hence if we store only summarized documents we can overcome from this disadvantage. To generate a summarized document we need a reader and identifier to choose between redundant and important words/sentences in the document cluster to generate summary. A summary is a content produced by collecting similar information files and extracting only important points to be added in summary. When the user searches for information by hitting a

query, the internet willprovide with large number of files which matches the score of related content in query, user will waste his time in searching for the relevantcontent. But it is impossible for the user to decide on required file. This problem grows exponentially as information flow in to web increases.

Text Summarization is a method of Information Retrieval from multiple documents, in which the output will be a generic processed text document with the required accurate content as queried by the user.Depending on the nature of text representation in the documents, summary can be categorized as an abstract andan extract. An extract is a summary consisting of anumber of important text units selected from the input. Anabstract is a summary, which represents the subject matterof the article with the text units, which are generated byreformulating the important units selected from the input. Anabstract may contain some text units, which are notpresent in to the input text.Although sentence extraction method is not the usualway that humans follow while creating summaries fordocuments, some sentences in the documents representsome aspects of their contents to some extent. Moreover,speed will be an important factor while incorporating thesummarization facility on the web. So, extraction basedsummarization is still useful on the web. The extractivemulti-document summarization can be conciselyformulated as extracting important textual units frommultiple related documents, removing redundancies andreordering the units to produce the effective summary.

An alternative approach to ensure good coverage andavoid redundancy is the clustering based approach thatgroups the similar textual units (paragraphs, sentences)into multiple clusters to identify themes of commoninformation and selects text units one by one from clustersin to the final summary. Each cluster consists of a group of similar text units representing a subtopic (theme). Domain independency and languageindependency are the key features of the clustering basedapproaches to multi-document text summarization. In this paper, we present a multi-document text summarization system, which clusters sentences using a similarity based sentence-clustering algorithm to identifymultiple sub-topics (themes) from the input set of relateddocuments and selects the representative sentences from the appropriate clusters to form the summary.

II. Literature Review

The Text summarization system proposed in [1] uses linguistic methods to examine and interpret the text and then to find the new concepts andexpressions to best describe it by generating a summarytext that conveys the most important information from theoriginal text document.

In“Multi-document summarization”, [2] presents an approach to cluster multiple documents by using document clustering approach and to produce cluster wise summary based on feature profile oriented sentence extraction strategy.

The clustering algorithm feature profile[3] is used to extract most important sentences from multiple documents, In clustering based multidocument summarization[4] performance heavily depends on three important factors like

a)clustering sentences, b)cluster ordering, c) selection of representative sentences from the clusters.

The work proposed in [5] uses Vector Space Model for finding similar sentences to the query and Sum and Focus to find word frequency, which achieves good accuracy rate.

In Paper[6] Important text features like,sentence position, positive keywords in sentence, negative keywords insentence centrality , sentence resemblance tothe title sentence inclusion of name entity , sentenceinclusion of numerical data, sentence relative length ,bushy path of the node , summation of similarities for eachnode , and latent semantic feature .

The system proposed in [7] givestwo kinds of summaries. The first one gives the similarities of each cluster of documents retrieved. The second one shows the particularities of each document with respect to the common topic in the cluster. The document multitopic structure has been used in order to determine similarities and differences of topics in the cluster of documents.From the work proposed in [8] We understood the concept of Open NLP tool for natural language processing of text for word matching in order to extract meaningful and query dependent information from large set of offline documents.

In "Cosine similarity", [9] Similarity function which is used to derive the distance between positive vectors. Usually used information retrieval and text mining.

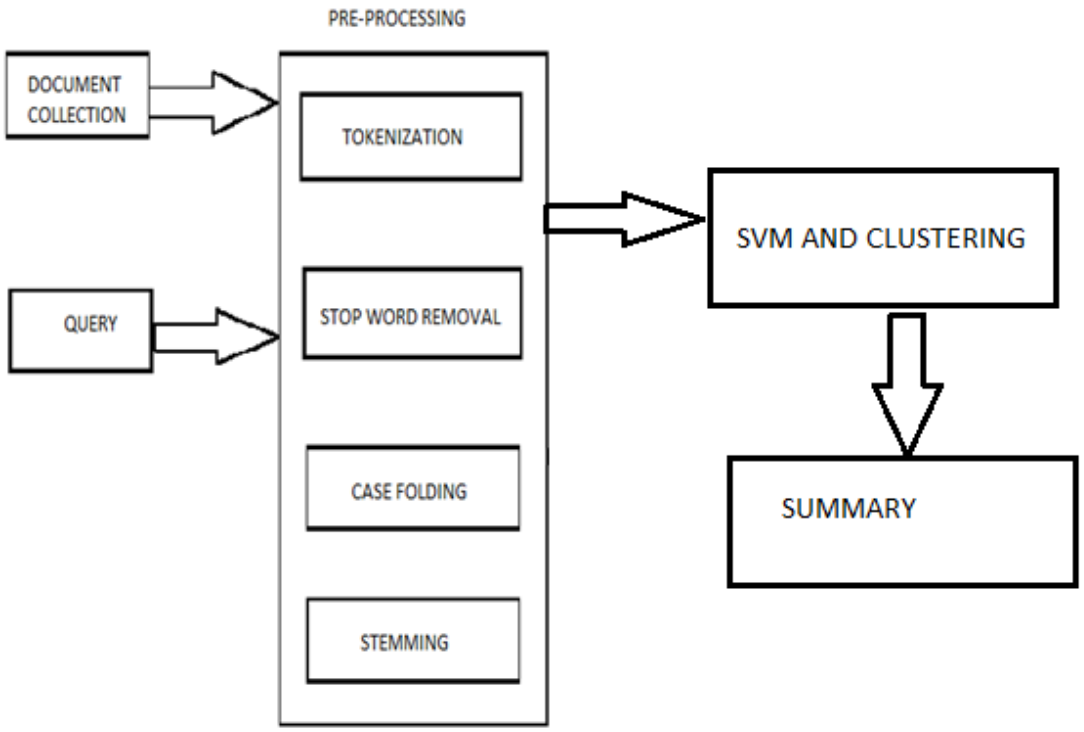
Thepaper[10] proposes an algorithm that learns orderings from a set of human ordered texts. This model consists of a set of ordering experts,eachexpert gives its precedence preference between two sentences.

The usage of XDOCTOOL[11] The highlighted terms, indicating terms that documents in a cluster have in common, and terms documents have in common with the topic description, were helpful for quickly scanning the summaries and documents.

Given a group of sentences to be organized into a summary, each sentence was mapped to a theme in source documents by a semi-supervised classification method, and adjacency of pairs of sentences is learned from source documents based on adjacency of clusters they belong to, Then the ordering of the summary sentences is derived logically,[12]

The use of automatic syntactic simplification for improving contentselection in multi-document summarization.[13]deals with, simplifying parentheticals by removing relative clauses and appositives results in improved sentence clustering, which is based on clustering central information.

II.i . System Overview



III.Algorithm

1. The collection of documents & query is the input to the summarizer. Open database connection is set ,and query as query <- file path from file category

1(a) Split a query into tokens &find the synonym for each token. We will get the synonym from list of maps if the token or synonym exists in a document collection & append the most frequent synonym of the query term to query. [The most frequently occurred words from data set are selected & those words are appended to the query. So the query is strengthened].

2. Pre- processing steps:

2.1 TOKENIZATION:

Splits every words as tokens using delimiters

```
char[] delimiters = new char[] { '\r', '\n', ' ', ',', '.' };
char[] paradelim = new char[] { '\r', '\n' };

result=content.Split(delimiters,StringSplitOptions.RemoveEmptyEntries);
paragraphs=content.Split(paradelim,StringSplitOptions.RemoveEmptyEnts);
for (int i = 0; i <paragraphs.Length; i++)
totalpara.Add(paragraphs[i]);
```

2.2 STOP WORD REMOVAL

Every word in documents is compared with the below stopwords list and replaces with blank space.

stopwordslist = new string[] { "a", "about", "above", "after", "again", "against", "all", "am", "an", "and", "any", "are", "aren't", "as", "at", "be", "because", "been", "before", "being", "below", "between", "both", "but", "by", "can't", "cannot", "could", "couldn't", "did", "didn't", "do", "does", "doesn't", "doing", "don't", "down", "during", "each", "few", "you've", "your", "yours", "yourself", "yourselves"……. };

```
for (int i = 0; i <result.Length; i++)
{for (int j = 0; j <stopwordslist.Length; j++)
{If(result[i].Equals(stopwordslist[j],StringComparison.OrdinalIgnoreCase))
{
result[i] = " ";
}}
```

2.3 CASE FOLDING

Coverts all words to lower case

```
for (int i = 0; i <result.Length; i++)
{ result[i] = result[i].ToLower();  }
```

2.4 STEMMING

Gets rid of plurals and -ed or -ing. e.g.

caresses -> caress ponies ->poni ties ->ti
caress -> caress cats -> cat matting -> mat
mating -> matemeeting -> meet milling -> mill

3. CLUSTERING

Prepares cluster center using n-dimensional vector space Document similarity is measured using cosine similarity. Prepares k initial centroid and assign one object randomly to each centroid.

```
foreach(intpos in uniqRand)
{
c = new Centroid();
c.GroupedDocument = new List<DocumentVector>();
c.GroupedDocument.Add(documentCollection[pos]);
centroidCollection.Add(c);
}
```

4. SIMILARITY MEASURE:

The documents are clustered by using, cosine similarity as a similarity measure to generate the appropriate document clusters.

```
public static float DotProduct(float[] vecA, float[] vecB)
{
floatdotProduct = 0;
for (var i = 0; i <vecA.Length; i++)
{dotProduct += (vecA[i] * vecB[i]); }
returndotProduct;       }
```

Magnitude of the vector is the square root of the dot product of the vector with itself.

```
public static float Magnitude(float[] vector)
{
return (float)Math.Sqrt(DotProduct(vector, vector));
}
```

5. Find UniqueTokens

This step is used to Find out the total no of distinct terms in the whole data set so that it will be easy to represent the document in the vector space. The dimension of the vector space will be equal to the total no of distinct terms.

```
foreach (string documentContent in collection.DocumentList)
{
foreach (string term in r.Split(documentContent))
{
if (!StopWordsHandler.IsStotpWord(term))
distinctTerms.Add(term);
else
continue;
} }
```

6. Calculate the score of each group (sentence cluster).
7. Sort sentence clusters, in reverse order of group score.
8. Pick the best scored sentences from each sentence cluster and add it to the summary.

```
returns index of closest cluster centroid
private                                                    static
intFindClosestClusterCenter(List<Centroid>clusterCenter,DocumentVectorobj)
{
float[] similarityMeasure = new float[clusterCenter.Count()];
for (int i = 0; i <clusterCenter.Count(); i++)
```

{similarityMeasure[i]=
SimilarityMatrics.FindCosineSimilarity(clusterCenter[i].GroupedDocument[0].Vector Space, obj.VectorSpace); }

9. From every document cluster, sentences are clustered based on their similarity values.
10. We have decided the number of sentences to be selected depending on sentence clusters size.

IV. EXPERIMENTAL RESULTS & EVALUATION

The summarization system is measured with no. of input words in the source document, number of words in the output summary file and its reduced words percentage is given below.

Table1. Text Summarizer Results

Trials	# input words	# output words	Reduced %
I	1027	846	17%
II	6650	2390	64%
III	908	206	77%
IV	8259	1130	86%

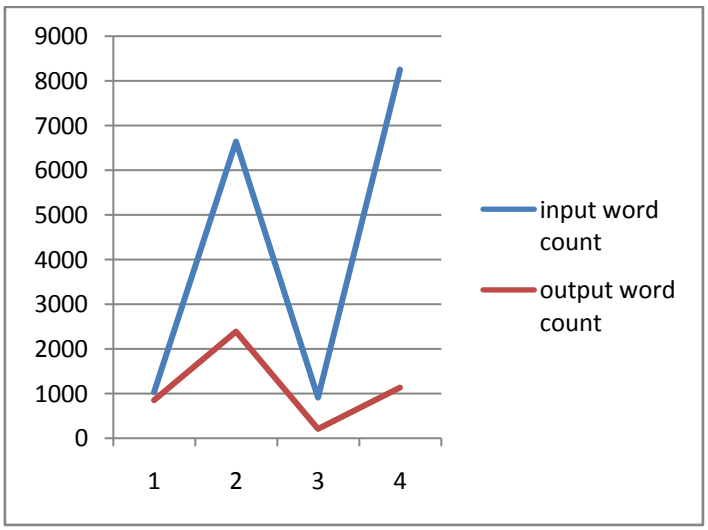


Fig. 1. Graph showing the count of input and output words in summarizer.

V. CONCLUSION

This method concentrates on extractive summarization technique as we compared the results with the conventional systems by using correctness measure an precession measure. As per results our method improves sentence simplification andreduces redundancy.

ACKNOWLEDGMENT

We would like to give deep gratitude for the blessings of Amma and all those who have supported us to complete the implementations we have presented in this paper. Very special thanks tothe Management, Principal and Head of department ,Amrita School of Arts and Sciences, Amrita VishwaVidyapeetham, Mysore campus, for all their support & giving us the resources required completing the development work.

REFERENCES

[1] Vishal Gupta, Gurpreet Singh Lehal, "A Survey of Text Summarization Extractive Techniques", Journal of emerging technologies in web intelligence, vol. 2, no. 3, Aug. 2010.

[2] "Multi-document summarization", *Wikipedia, the free encyclopedia*, 2015.

[3] A. Kogilavani, Dr.P.Balasubramani, "CLUSTERING AND FEATURE SPECIFIC SENTENCE EXTRACTION BASED SUMMARIZATION OF MULTIPLE DOCUMENTS", *International journal of computerscience & information Technology*, vol.2, no.4, Aug. 2010.

[4] Kamal Sarkar, "Sentence Clustering-based Summarization of Multiple Text Documents", *TECHNIA – International Journal of ComputingScience and Communication Technologies*, vol. 2, no. 1, Jul. 2009.

[5] A. P. Siva kumar, Dr. P. Premchand and Dr. A. Govardhan, "Query Based Summarizer Based on Similarity of Sentences and Word Frequency", *International Journal of Data Mining & KnowledgeManagement Process*, vol.1, no.3, May 2011.

[6] Aristoteles, YeniHerdiyeni, Ahmad Ridha and Julio Adisantoso, "Text Feature Weighting for Summarization of Document in Bahasa Indonesia Using Genetic Algorithm", *International Journal ofComputer Science Issues*, vol. 9, no. 3, May 2012.

[7] Multidocument Summarization: An AddedValue to Clustering in Interactive RetrievalMANUEL J. MAN˜ A-LO´ PEZUniversidad de VigoandMANUEL DE BUENAGA and JOSE´ M. GO´ MEZ-HIDALGOUniversidad Europea de Madrid.

[8] Harshal J. Jain, M. S. Bewoor and S. H. Patil, "Context Sensitive TextSummarization Using K Means Clustering Algorithm", *InternationalJournal of Soft Computing and Engineering* ,volume-2, no.2, May2012.

[9] "Cosine similarity", *Wikipedia, the free encyclopedia*, 2015.

[10] A Machine Learning Approach to Sentence Ordering for MultidocumentSummarization andits EvaluationDanushkaBollegala, Naoaki Okazaki, Mitsuru IshizukaUniversity of Tokyo, Japan.

[11] Multi-Document Summarization: Methodologies andEvaluationsGees C. Stein, AmitBagga and G. Bowden WiseGeneral Electric, Corporate R&D, One Research Circle, Niskayuna NY 12309, USA*Conférence TALN 2000, Lausanne, 16-18 octobre 2000.*

[12] Sentence Ordering based on Cluster Adjacency in Multi-Document SummarizationJiDonghong, Nie Yu Institute for InfocommResearchSingapore, 119613.

[13] Syntactic Simpliflication for Improving Content Selection in Multi-Document Summarization AdvaithSiddharthan, AniNenkovaand Kathleen McKeown Columbia University Computer Science Department.