International Journal of Soft Computing and Engineering (IJSCE)
ISSN: 2231-2307, Volume-1, Issue-3, July 2011

Corpus based Automatic Text Summarization System with HMM Tagger

M.Suneetha, S. Sameen Fatima

Abstract—**The rapid growth of the data in the Internet has overloaded the user with enormous amounts of information which is more difficult to access huge volumes of documents. Automatic text summarization technique is an important activity in the analysis of high volume text documents. Text Summarization is condensing the source text into a shorter version preserving its information content and overall meaning. In this paper a frequent term based text summarization technique with HMM tagger is designed and implemented in java. The proposed system generates a summary for a given input document based on identification and extraction of important sentences in the document. The model consists of four stages. In first stage, the system decomposes the given text into its constituent sentences, assigning the POS (tag) for each word in the text and stores the result in a table. The second stage removes the stop words, stemming the text and applying lemmatization. Feature term identification is done in third stage. Finally each sentence is ranked depending on feature terms. This stage reduced the amount of the sentences in the summary in order to produce a qualitative summary.**

***Index Terms*— Text Summarization, HMM Tagger, Brown Corpus, POS tagging.**

I. INTRODUCTION

Internet has made a profound change in the lives of many enthusiastic innovators and researchers. The information available on the web has knocked the doors of Knowledge Discovery leading to a new Information era. Automatic summarization is the distillation of important information from a source into an abridged form for a particular user or task. Automatic text summarization has been an active research area for many years. Evaluation of summarization is a quite hard problem [9]. Even though automatic text summarization dates back to Luhn's work in the 1950's, several researchers continued investigating various approaches to the summarization problem up to nowadays [1]. Automatic text summarization can be classified into two categories: extraction and abstraction [17].Extraction summary is a selection of sentences or phrases from the original text with the highest score and put it together to a new shorter text without changing the source text. Abstraction summary method uses linguistic methods to examine and interpret the text [14]. Most of the current automated text summarization system use extraction method to produce summary.

Automatic part of speech tagging, is a well known problem that has been addressed by several researchers during the last twenty years. Part-of-Speech (POS) tagging is a technique for automatic annotation of lexical categories. Part-of –Speech tagging assigns an appropriate part of speech tag for each word in a sentence of a language. POS tagging is widely used for linguistic text analysis. Part-of-speech tagging is an essential task for all the natural language processing activities [11]. A POS tagger takes a sentence as input and assigns a unique part of speech tag to each lexical item of the sentence.

It is a firm belief that when it comes to keyword extraction, the nouns of the text carry most of the sentence meaning. In a sense, extracted nouns should lead to better semantic representation of the text. Noun extraction, a subtask of POS tagging, is the process of identifying every noun (either proper or common) in a document. In many languages, nouns are used as the most important terms (features) that express a document's meaning in Natural Language Processing applications such as information retrieval, document categorization, text summarization, information extraction, etc.

POS tagging is used as an early stage of linguistic text analysis in many applications including subcategory acquisition; text to speech synthesis; and alignment of parallel corpora. There are a variety of techniques for POS tagging [19]. Two approaches to POS tagging are Supervised POS Tagging and Unsupervised POS Tagging.

Supervised tagging technique requires a pre tagged corpora where as unsupervised tagging technique do not require a pre tagged corpora. Both supervised and unsupervised tagging can be of two types, Rule based and stochastic. Rule based system needs context rule for POS tagging [12]. Typical rule based approaches use contextual information to assign tags to unknown or ambiguous words.

Stochastic tagging technique makes use of a corpus. The most common stochastic tagging technique uses a Hidden Markov Model (HMM) [15]. The states usually denote the POS tags. The probabilities are estimated from a tagged training corpus or an untagged corpus in order to compute the most likely POS tags for the word of an input sentence. Stochastic tagging techniques can be of two types depending on the training data. Supervised stochastic tagging techniques use only tagged data. However the supervised method requires large amount of tagged data so that high level of accuracy can be achieved. Unsupervised stochastic techniques, on the other hand, are those which do not require a pre-tagged corpus but instead use sophisticated

Manuscript submitted on July 1, 2011.

(**Manne Suneetha**, Assistant Professor, Department of Information Technology , VR Siddhartha Engineering college, Vijayawada, Andhra Pradesh, India, e-mail: suneethamanne74@gmail.com)

Dr. S. Sameen Fatima, Professor and HOD, Department of Computer Science engineering, Osmania University, Hyderabad, Andhra Pradesh, India e-mail: sameenf@gmail.com).

computational methods to automatically induce word groupings (i.e. tag sets), and based on these automatic groupings, they calculate the probabilistic values needed by stochastic taggers.

Our approach is an extractive method to solve the problem with the idea of sentence ranking. One of the main contributions of the proposed is the document summarization, where the goal is to select whole sentences to create a short paragraph summary

In this paper, we propose text summarization extractive method based on POS tagging with Brown corpus to extract important sentences as a summary [13]. The rest of this paper is organized as follows. Section II presents related work. Section III describes preprocessing with Brown corpus and the important features. and proposed system, followed by results in section IV. Finally, we conclude the work that can be carried out in Section V.

II. RELATED WORK

The study of text summarization [3] proposed an automatic summarization method combining conventional sentence extraction and trainable classifier based on Support Vector Machine. The study [3] introduces a sentence segmentation process method to make the extraction unit smaller than the original sentence extraction. The evaluation results show that the system achieves closer to the human constructed summaries (upper bound) at 20% summary rate. Still the system needs to improve accuracy of its summary output.

In [2] presents a sentence reduction system for automatically removing extraneous phrases from sentences that are extracted from a document for summarization purpose. The system uses multiple sources of knowledge to decide which phrases in an extracted sentence can be removed, including syntactic knowledge, context information, and statistics computed from a corpus which consists of examples written by human professionals. Reduction can significantly improve the conciseness of automatic summaries.

The most relevant corpus-based POS tagging approaches are based on Hidden Markov Models [4], transformation-based learning [6][7], memory-based learning [8], decision trees [20], maximum entropy principle [10], etc. Moreover, some works combine the output of different taggers by means of some "voting" methods in order to improve their performance [5].

The comparison among these different approaches is difficult due to the multiple factors that must be considered: the language, the number and type of the tags, the size of the vocabulary, the ambiguity ratio, the difficulty of the test set, the size of the training and the test sets, etc. For English, most of the taggers have been evaluated on the Wall Street Journal (WSJ) corpus [18]. The results reported on the WSJ achieved a precision ratio of between 96% and 97%. Some comparison experiments have been conducted in order to rigorously contrast the different approaches [6].

Although some of these works report that HMM-based taggers achieved results which are lower than the results obtained for taggers based on other paradigms, it has recently been shown that HMMs perform better than or similar to other single taggers[11]. Only combined methods slightly outperform the singles approaches [8].

III. PROPOSED SYSTEM

The following Fig.1 represents the proposed system block diagram. The model consists of the following stages.

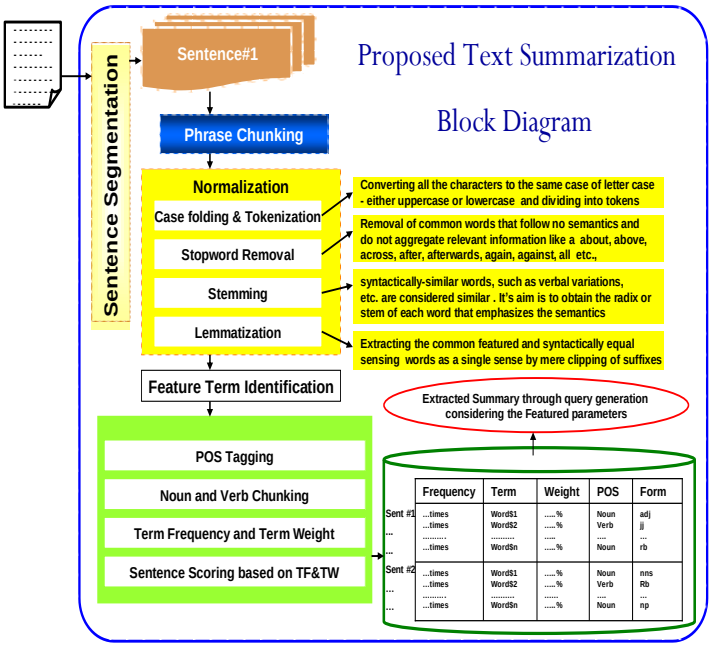

Fig. 1. Proposed System Architecture

A. Text Processing

The pre- processing is a primary step to load the text into the proposed system, and decompose the given text into its constituent sentences.

B. Normalization

This involves the process of converting words into normalized form. The following are the processes that come under Normalization techniques

Case Folding: Converting entire words in the sentences into lower case so as to avoid repetition of same word in different cases like sentence case, capital case, title case, Uppercase etc., This improves the accuracy of the system to distinguish similar words.

Tokenization: splitting of the sentence into words using StringTokenizer class in the java.util package

Stop word Removal: Removing the words, numbers, special symbols etc., which have less significance during the retrieval of intended results.

Lemmatization: Extracting the commonly featured, same meaning tokenized words so as to avoid repetition (e.g. problems-problem, risks-risk, etc.),

Stemming: Mechanically removing or changing the suffixes of some nouns or verbs (e.g. Economies-Economy, professes-profess, etc.).

1) Stop Word Removal:

Some words are extremely common and occur in a large majority of documents. For example, articles such as “a”, “an”, “the”, “by” appear almost in every text but do not include much semantic information. Since categorization is based on the featured terms not on commas, full stops, colons, semicolons etc., we remove them from document to list in tokens so that these words will not be stored in the signature file. List of some stop words considered while building Model are presented in Fig 2.

a	become	done	Further
about	becomes	don’t	Get
above	becoming	down	Give
across	Been	due	Go
after	Before	during	Had
afterwards	beforehand	each	hadn’t
again	Behind	eg	Has
against	Being	eight	Hasn’t
all	Below	either	hasn’t
almost	Beside	eleven	Have
alone	besides	else	haven’t
along	between	elsewhere	Having
already	beyond	empty	He
also	Bill	enough	Hence
although	Both	etc	Her
always	Bottom	even	Here

Fig. 2. Stop words considered in the proposed model

2) Stemming:

Stemming refers to identifying the root of a certain word in the document [5,6]. Any text document, in general contain repetition of same word but with variations in the grammar such as word appearing to be in past , or in present tense and sometimes containing gerund (“ing” suffixed at the end) Stemming is of two types.

1) Derivational Stemming

2) Inflectional Stemming

Derivational stemming aims at creating a new word from an existing word, most often by changing the grammatical category.

e.g.: Ration*alize*- Ration*al*, Use*ful*-Us*e*

Music*al* – Mus*ic*, Fin*alize*-Fin*al*

Inflectional Stemming aims at confining normalized words to regular grammatical variants such as singular or plural or past or present.

e.g.: Classific*ation*- Classific, Manag*ement*-Manag, Pay*ment*-Pay *etc.*

The stemming words considered in the proposed model are shown in Fig 3. The two main advantages of stemming algorithms [7] are space efficiency and retrieval generality. The size of the inverted file can be reduced dramatically because many different words are indexed under the same stem and require only a single entry in the inverted file. The stemming algorithm is shown in Fig.4.

Penultimate of the Word (n-1) th word	*Ends with (suffix)*	*Replace with*
n	ant	“ “
	ement	
	ment	
	ent	
l i a	able	
	ible	
	ic	
	al	
c n e o s	ance	
	ence	
	ition	
	ication	
	er	
	ion	
	ou	
	ism	

Suffix of the Word	*Ends with (suffix)*	*Replace with*
e	icate	“ic”
	ative	“ ”
	alize	“al”
i	iciti	”ic”
l	ical	“ic”
s	ful	“ ”
c	Ic	“ ”
n	ition	“y”
	ication	“y”
t	iest	“y”

Fig. 3. Stemming words

```
Stemming (String word)
{
  String StopwordStem (word)
  {
   if word ends with any of (. : , ;? ‘ “ ) } ] )
   return word.replace(word.trim(), word.substring(0,
word.length() - 1));
  else if word start with any of ( { [ ( ‘ “ . , : ; )
  return word.replace (word.trim(), word.substring(1));
  }
 String Stemm (word)
  {
  word = ReplaceStem (word);
  if the word ends with any of( second column of table1,2 )
  replace with the respective terms
  return word.replace(word, word.substring(0,
word.length() – suffix length removed));
  }
```

Fig. 4. Stemming Algorithm

C. Feature Term Identification

The tokenized terms after applying normalized techniques are now considered as Feature Terms. Now the preliminary step is identification of parts of speech to each feature term. This process is known as parts of speech tagging (POS tagging).

1) POS Tagging:

POS tagging is the process of assigning Parts of Speech like (noun, verb, and pronoun, Etc.) to each word in a sentence to give word class. The input to a tagging algorithm is a set of words in a natural language and specified tag to each. The first step in any tagging process is to look for the token in a lookup dictionary. The dictionary that created in the proposed system consists of I million

words in order to assign words to its right tag. The dictionary had partitioned into tables for each tag type (class) such as table for (noun, verb, Etc.) based on each P.O.S category [16]. The system searches the tag of the word in the tables and selects the correct tag depending on the tags of the previous and next words in the sentence. We used Brown corpus and the Brown Corpus of Standard American English is considered to be the first general English corpus that could be used in computational linguistic processing tasks [1]. The corpus consists of one million words of American English texts printed in 1961. For the corpus to represent as general a sample of the English language as possible, 15 different genres were sampled such as Fiction, News and Religious text. Subsequently, a POS-tagged version of the corpus was also created with substantial manual effort. Various approaches of Pos tagging are shown in Fig.5.

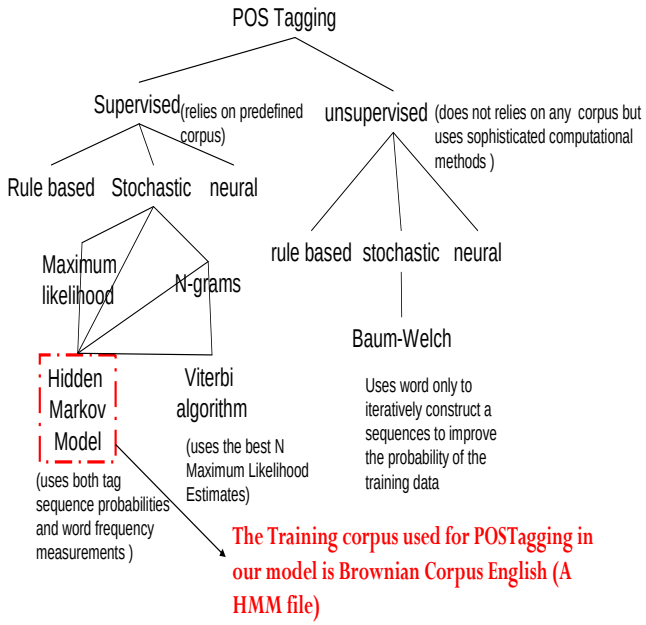

Fig. 5. POS tagging approaches

POS tagging is implemented as follows

Tagging: using a predefined model (simply say the feature term) to assign part of speech tags to text.

Training: A model file which is manually tagged is used to tag the predefined model.

Here we used training corpus used is pos-en-general-brown.HMM

A Hidden Markov Model can be considered a generalization of a mixture model where the hidden variables, which control the mixture component to be selected for each observation, are related through a Markov process rather than independent of each other. Few Tag set conventions for Brownian Corpus are presented in Fig. 6.

Tag	Description
ABN	determiner/pronoun, pre-qualifier
ABL	determiner/pronoun, pre-quantifier
ABX	determiner/pronoun, double conjunction or pre-quantifier
AP	determiner/pronoun, post-determiner
AT	article
CC	conjunction, coordinating
CS	conjunction, subordinating
IN	preposition
JJ	adjective
NN	noun, singular, common
NNS	noun, plural, common
NP	noun, singular, proper
NPS	noun, plural, proper
PN	pronoun, nominal
RB	adverb
UH	interjection
VB	verb
VBN	verb, past participle
VBZ	verb, present tense, 3rd person singular

Fig. 6. Some of Tag sets in Brown Corpus

2) Noun and Verb Chunking:

Extracting of high level structures like phrases can be possible by using Noun and Verb Chunking. Nouns may start with determiners, adjectives, common nouns or pronouns and they continued with any category that may start a noun, or adverbs or punctuation. Verbs may start with verbs, auxiliaries, or adverbs and may be continued with any of the tags, or with punctuation. These sets are defined statically by using a set of Determiner tags to a Noun or Verb. The n-best output for taggers could be used to define chunks. Rather than running over just the first-best output, we can use n-best output.

Example of Pos chunking:

Prime Minister Manmohan Singh made the announcement today that India will help Africa in rooting off the Terrorism

Tagged Sentence:

Prime/jj Minister/nn Manmohan/np Singh/np made/vbd the/at announcement/nn today/nr that/cs India/np will/vbj help/vb Africa/np in/in rooting/vb off/rb the/at terrorism/nn

3) Term Frequency and Term weight

The term frequency tf (t, d) of term t in document d is defined as the number of times that t occurs in d. Relevance does not increase proportionally with term frequency. A document with 10 occurrences of the term is more relevant than a document with 1 occurrence of the term. But not 10 times more relevant

Term Frequency (TFi) = no. of times a term repeated

Term Weight (TWi) = [TFi *1000] /total no. of terms

4) Sentence scoring

Sentence Length: Too short sentences are not expected to belong to the summary.

Normalized sentence length=

$$\frac{\text{Number of words occurring in the sentence}}{\text{The number of words occurring in the longest sentence of the document}}$$

Sentence Position: Number of sentences in a paragraph be n, Then n/2 top sentences are considered top priority than that the next n/2 sentences. Paragraph can be recognized using ends with "//s//s//s//s" (sentence ended with four or more spaces)

The proposed system computes the weight for each sentence and counts the number of words present in each sentence. The sentence weight age is calculated using the following formula.

Sentence weight (SW)=

$$\frac{\text{Number of featured terms within the sentence}*1000}{\text{Total number of terms in a paragraph}}$$

IV. RESULTS

The system decomposed the given text into its constituent sentences, assigning the POS tag for each word in the text and stores the results in a table I. After segmentation, stop words elimination, stemming and lemmatization techniques are applied. Other hand unique words are generated, using these words the summary will be presented.Further the feature terms are identified. Finally each sentence is ranked depending on feature terms.

Ranked sentences have been selected and the summary will be generated based on weight age of the sentences whose value has 50% and count has 250 for a given sample document. The system will work dynamically and was implemented in JAVA.

The Table I shows the word frequency, word weight age and POS forms. Table II gives the paragraph sentence weightage.

Table I. Word frequency, count and POS tags

frequency	term	weight	pos	form
1	Baye	12	noun	np
1	CV	12	noun	nn
1	Categorization	12	noun	nn
1	Curriculum	12	noun	nn
1	Deci	12	noun	nn
1	HR	12	noun	nn
1	Human	12	noun	np
1	IR	12	noun	nn
1	Multinational	12	noun	jj
1	Resource	12	noun	nns
1	Retrieval	12	noun	nn
1	Vitae	12	noun	np
1	address	12	noun	nns
1	amount	12	noun	nns
1	analyze	12	verb	vb
---	----	--	--	--

Table II. Paragraph sentence weight age

Paragraph	count	pweight
The process of identifying interesting knowledgeable information from large amounts of databases, data warehouses, or any other information repositories is known as Data Mining.	253	96
Where as Information Retrieval (IR) mainly concerned with the organization and retrieval of Information from a large number of text-based documents.	149	56
Some common information retrieval problems are in general not encountered in traditional database systems, such as unstructured documents, approximate search based on keywords, text mining and the notion of relevance.	218	82
Text categorization, text routing and text filtering systems are all concerned with	113	42
Text categorization labels the web document automatically based on a set of predefined categories.It is observed that people who are involved in research study need to analyze the available research papers, e-books and other resources and recognize their	305	115
The same is the situation where a doctor finds difficulty in comparing the symptoms of a cancer patient to the already available categories to recognize the stage he/she is suffering now.	188	71
----	-----	----

Best query need to be generated considering the parameters:

1. The sentence with best sentence score
2. Sentences containing words with high TF and TW values
3. Sentence which have optimal sentence position within the
4. paragraph
5. Sentence build on best POS Tagging criteria
6. Relevance of the (n-1) sentences with the first sentence i.e., the title of the document

V. CONCLUSIONS

In this work, we proposed an extractive automatic text summarization approach by sentence

extraction using a supervised POS tagging. A frequent term based text summarization technique with HMM tagger is designed and implemented in java. Ranked sentences are collected by identifying the feature terms and text summary is obtained. This gave the advantage of finding the most related sentences to be added to the summary text. The system produced the most compressed summary with high quality and good results in comparison to manual summarization extraction. The work will be extended for multi documents.

REFERENCES

[1] Vishal Gupta , Gurpreet Singh LehalKuceral., A Survey of Text Summarization Extractive, Journal of Emerging Technologies in Web Intelligence, vol. 2, no. 3, august 2010

[2] TechniquesComputational Analysis of Present-Day American English. Brown University Press, Providence, RI

[3] Hongyan Jing, Sentence Reduction for Automatic Text Summarization, Proceedings *of the sixth conference on Applied natural language processing*, Seattle, Washington, pp.310 – 315, 2000.

[4] Kai ISHIKAWA et. al.; "Trainable Automatic Text Summarization Using Segmentation of Sentence"; Multimedia Research Laboratories, NEC Corporation 4-1-1 Miyazaki Miyamae-kuKawasaki-shi Kanagawa 216-8555, 2003.

[5] Church, K.W., A Stochastic parts program and noun phrase parser for unrestricted text. Proceedings 1st Conference on Applied Natural Language Processing, ANLP, pp. 136–143. ACL, 1988.

[6] Brill, E. A Simple Rule-Based Part-of-speech Tagger. Proceedings 3rd Conference on Applied Natural Language Processing, ANLP, pp. 152–155. ACL, 1992.

[7] Brill, E. Automatic Grammar Induction and Parsing Free Text: A Transformation based Approach. Proceedings 31st Annual Meeting of the Association for Computational Linguistics, 1993

[8] Brill, E. Transformation–based error–driven learning and natural language processing: A case study in art-of-speech tagging. Computational Linguistics **21**(4): 543–565. 1995a

[9] Daelemans, W., Zavrel, J., Berck, P. and Gillis, S.MBT: A memory-based part-ofspeech tagger generator. Proceedings 4th Workshop on Very Large Corpora, pp. 14–27. Copenhagen, Denmark, 1996.

[10] Goldstein, J., et al.: Summarizing Text Documents: Sentence Selection and Evaluation Metrics. In: Proceedings of ACM SIGIR Conference 1999.

[11] Ratnaparkhi, A. A maximum entropy part-of-speech tagger. Proceedings 1st Conference on Empirical Methods in Natural Language Processing, EMNLP, 1996.

[12] L. Bahl and R. L. Mercer, *Part-Of-Speech assignment by a statistical decision algorithm*, IEEE International Symposium on Information Theory, pages: 88 - 89, 1976.

[13] D. Cutting, J. Kupiec, J. Pederson and P. Sibun, *A practical Part-Of-Speech Tagger*, In proceedings of the Third Conference on Applied Natural Language Processing, pages: 133 - 140, ACL, Trento, Italy, 1992.

[14] Brown Tagset, available online at: http://www.scs.leeds.ac.uk/amalgam/tagsets/brown. html

[15] Edmundson, H.P. New Methods in Automatic Extraction. *Journal of the ACM* 16(2), 264–285, 1968.

[16] Ferranpla and Antoniomol i n a, Improving part-of-speech tagging using lexicalized HMMs, Cambridge University Press, Natural Language Engineering 10 (2): 167–189, 2004

[17] Rafeeq Al-Hashemi, Text Summarization Extraction System (TSES) Using Extracted Keywords, International Arab Journal of e-Technology, Vol. 1, No. 4, June 2010 pp 164-168

[18] I. Mani and M. Maybury. Advances in Automatic Text Summarization. MIT Press, ISBN 0-262-13359-8, 1999.

[19] Marcus, M. P., Marcinkiewicz, M. A. and Santorini, B. Building a large annotated corpus of English: The Penn Treebank. Computational Linguistics **19**(2), 1993.

[20] Cutting, D., Kupiec, J., Pederson, J. and Sibun, P. A practical part-of-speech tagger. Proceedings 3rd Conference on Applied Natural Language Processing, ANLP, pp. 133–140. ACL, 1992.