Query-Oriented Text Summarization using Density based Clustering

[1]Ms.Swati.B.Bhonde
[1]M.Tech.Student , Bharati vidyapeeth COE, Pune
Department of Computer Engineering,
Bharati vidyapeeth College of Engineering, Pune, India
swati.bhonde@gmail.com

[2]Prof.M.S.Bewoor and [3]Dr.S.H.Patil
[2]Asst. Professor, Department of Computer Engineering,
[3]Head, Computer Engineering Department,
Bharati Vidyapeeth College of Engineering, Pune, India
msbewoor@bvucoep.edu.

Abstract:

World Wide Web is the largest source of information. Huge amount of data is present on the Web. There has been a great amount of work on query-independent summarization of documents. However, due to the success of Web search engines *query-specific* document summarization has become an important problem. In this paper, we introduced a method to create query specific summaries by identifying the most query-relevant fragments and combining them using the semantic associations. While doing so we have also used new association algorithm to generate accurate distance matrix and then density based clustering methods can be used to from a cluster of related data. Here before clustering we are also handling text ambiguity to generate effective summary. Furthermore performance of the summary using different clustering techniques will be analyzed and the optimal approach will be suggested.

Keywords:
DBSCAN (Density Based SCANning algorithm),APRIORI

I. INTRODUCTION

1. Overview

Large amount of data is present on the Web. Users always need to search for the required information by using particular keywords. As the number of documents available on user's desktop and as the Internet access increases so does the need to provide high-quality summaries in order to allow the user to quickly locate the desired information. Summarization is the process of automatically creating a compressed version of a given text that provides useful information for the user. An application of document summarization is the snippets generated by web search engines for each query result. In particular, first we find relation between statements using a newly introduced association algorithm and then structure is added to the documents in the pre-processing stage and converts them to a document graph and then best summaries are calculated by calculating top spanning tress on the document graph[10].

Density based clustering is used before summarization to generate effective summary. The main aim of this research work is to combine both approaches of Document clustering and query dependent summarization.

This mainly includes applying different clustering algorithms on a text document. Create a weighted document graph of the resulting graph based on the keywords and obtain tree to get the summary of the document.

1.1 Motivation

The twenty first century should be called the century of Information Overload and Information explosion. Information, now the very lifeline of our lives, is increasing exponentially in the virtual space and archiving and managing it has become a challenge of sorts[9]. The effects of this explosion of information are all too evident[11]. A recent study put the number of web servers at over 70 million in the year 2005 alone. According to Technorati, the number of blogs doubles about every 6 months with a total of 35.3 million blogs as of April 2006. The enormous burden this places on the task of researchers in the field of natural language processing and information retrieval is breathtaking. And a lot of effort is being directed into finding newer and efficient ways to record, to archive and to retrieve such information. In this work, summarization is performed on the text document. Generally to use association before clustering helps to generate accurate distance matrix, which can form, clusters more precisely[12]. This motivated my research work towards developing a new algorithm combining properties of APRIORI, MS-APRIORI and FP-GROWTH algorithm. Density-based method owns the function that it can acquire clusters from spherical and unbalanced datasets.

Our present task aims at developing a Query dependant single-document summarizer using density based clustering approach. We hope it will add another dimension towards solving the seemingly complex task of document summarization and presents a good summary with cohesive sentences.

1.2 Need

In order to fully utilize these on-line documents effectively, it is crucial to be able to extract the relevant of these documents. Having a Text Summarization system would thus be immensely useful in serving this need. After looking at the present scenario of the electronic document we came to know that extracting of the relevant document is essential as well as if some powerful method of identifying relationship between statements is used then two or more statements will

be tightly coupled with each other[13]. While doing so, clustering is one approach that can be employed before summarization and if association is used before clustering then degree of relevancy of statements with each other will be more accurate. Thereafter results of the summary generated will be more clear & relative. [9]

II. SYSTEM DESCRIPTION

Several stages while generating summary are shown in the following figure:

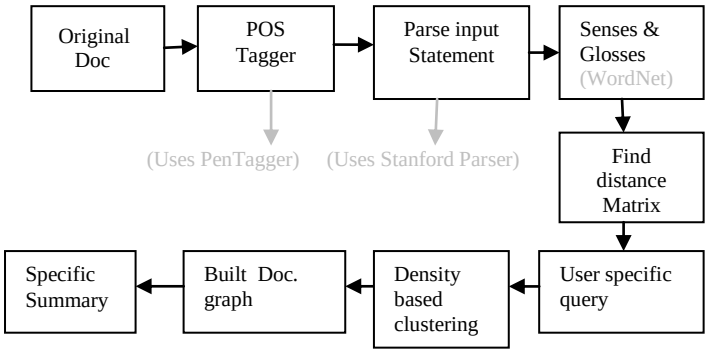

Figure 1: Several phases of text analysis in our system

Explanation of each of them is given in the following section:

2.1 Input:

First of all, original document containing set of statements is fed to the system. Generally document from which summary is to be retrieved should present relevant sentences probably from i.e. same context

E.g. Stress

V0) Stress is associated with migraines.
V1) Stress can lead to loss of magnesium.
V2) Calcium channel blockers prevent some migraines.
V3) Magnesium is a natural calcium channel blocker.
V4) Spreading Cortical Depression (SCD) is implicated in some migraines.
V5) High levels of magnesium inhibit SCD.
V6) Migraine patients have high platelet agreeability.
V7) Magnesium can suppress platelet agreeability.

This simply means that:

"Magne
sium deficiency may play a role in some kinds of migraine headache".

Given a document related to some context further it is given to second phase.

2.2 PoS Tagger:

We are also using Penn tagger for Part of Speech tagging. Penn Tagger has many standard tags used in Penn TreeBank :

CC- coordinating conjunction
CD- cardinal number
IS- existential there
DT- Determiner
NN- Noun
JJ- Adjective and many more.

- *Pre-processing of the text is needed as follows:*

a) Break the text into sentences:

Apply part of speech tagging to the words in the text. This is essential to pick the correct meaning of the word in WordNet. Hence if the word "pant" is used in the text as a verb, we will not associate it with a form of clothing.

b) Identify collocations in the text:

A collocation is a group of commonly co-occurring words, for example, "miles per hour". Identifying collocations helps in capturing the meaning of the text better than that of the individual words (just like any idiom).

c) Remove stop words:

like "the", "him" and "had" which do not contribute to understanding the main ideas present in the text. The sequence of the above operations is important since we must identify collocations before removing stop words as many stop words often form part of collocations.

2.3 Parse input statement:

Here stop words & determiners are eliminated from the sentence. Before this one need to find out whether elimination of stop words, determiners and prepositions doesn't affects meaning of the statement.

e.g. a) Sangamner is resided on the **bank** of Pravara river.
b) I have account in State **Bank** of India.

Observations that can be made from above statements are:

- *In both the sentences meaning of word bank is different.*

So **word ambiguity** can be there in more than one sentence.
Similarly one may face **sentence ambiguity** also.
For e.g. The king saw rabbit with his glasses.
This gives two meanings whether king was wearing a glasses or rabbit was wearing the glasses. And second is,

- If we are eliminating is, on, in, of from these statements they will not have any sense after tokenization i.e.
- Sangamner reside bank Pravara river.
- I account State Bank India

Therefore care must be taken that meaning of the sentence will not lost in any case[4].

Once the words are tokenized next PoS tagger is applied to them for knowing their grammatical semantics.

In this research work we are using Stanford parser is freely available on Internet. Stanford Parser can read various forms of plain text input and can output various analysis formats, including part-of-speech tagged text, phrase structure trees, and a grammatical relations (typed dependency) format.

For e.g.:

"The strongest rain ever recorded in India

The/DT strongest/JJS rain/NN ever/RB recorded/VBN in/IN India/NNP

(ROOT
(S
(S
(NP
NP: proper noun(the)
(NP (DT The) (JJS strongest) (NN rain))
DT: determiner
(VP
JJS: adjective, superlative (strongest)
RB: adverb (ever)
(ADVP (RB ever))
VBN: verb, past participle
(VBN **recorded**)
PP : Past participle
(PP (IN in)
NNP: proper noun, singular (India)
(NP (NNP India)))))
NN: Noun (rain)

2.4 Find senses & glosses:

Once grammatical aspect of word is clear we use WordNet to find out different senses of the word. WordNet (Fellbaum, 1998) is an online lexical reference system in which English nouns, verbs, adjectives and adverbs are organized into synonym sets or synsets, each representing one underlying lexical concept. Noun synsets are related to each other through hypernymy (generalization), hyponymy (specialization), holonymy (whole of) and meronymy (part of) relations. Of these, (hypernymy, hyponymy) and (meronymy,holonymy) are complementary pairs. The verb and adjective synsets are very sparsely con-nected with each other. No relation is available between noun and verb synsets. However, 4500 adjective synsets are related to noun synsets with pertainyms (pertaining to) and attra (attributed with) relations.

We use WordNet to understand the links between different parts of the document; subsequently extract the associations between two statements which are most relevant by introducing a new association algorithm that combines features of APRIORI & MS-APRIORI algorithm to combine their best features and contains the main ideas present in the document. The idea of first getting a global view of the whole document, even before beginning to rank sentences is what differentiates our approach from the rest and also makes it generic. Associativity of the sentences with each other can be stored in some data structure.

2.5 Built Distance matrix:

Lexical semantics begins with recognition that a word is a conventional association between a lexicalized concept and an utterance that plays a syntactic role. This definition of ‘‘word’’ raises at least three classes of problems for research. First, what kinds of utterances enter into these lexical associations? Second, what is the nature and organization of the lexicalized concepts that words can express? Third, what syntactic roles do different words play? Although it is possible to ignore any of these questions while considering only one, the emphasis here will be on the second class of problems, those dealing with the semantic structure of the English lexicon.[7]

Word forms are imagined to be listed as headings for the columns; word meanings as headings for the rows. An entry in a cell of the matrix implies that the form in that column can be used (in an appropriate context) to express the meaning in that row.

Table 1: Sample distance matrix

Word Meanings	Word Forms				
	F_1	F_2	F_3	. . .	F_n
M_1	$E_{1,1}$	$E_{1,2}$			
M_2		$E_{2,2}$			
M_3			$E_{3,3}$		
⋮				. . .	
M_m					$E_{m,n}$

Thus, entry E1,1 implies that word form F1 can be used to express word meaning M1. If there are two entries in the same column, the word form is polysemous; if there are two entries in the same row, the two word forms are synonyms. Mappings between forms and meanings are many: some forms have several different meanings, and some meanings can be expressed by several different forms.

2.6 User fires a query:

We have distance matrix as an input to the clustering algorithm. Once the distance matrix giving associativity of different sentences is ready, user can fire a query. There has been a great amount of work on query-independent summarization of documents. However, due to the success of Web search engines query-specific document summarization has become an important problem, which has received little attention.

For e.g. In a typical bibliography, we start with introduction of some personality, his birth place, schooling, college and so on. Next can be family background and then hobbies and then how his journey started and so on.

Now suppose one want to know only hobbies, user can fire query like hobbies of xyz then this is called as query specific information retrieval.

2.7 Density based clustering:

After accepting a query from user, we apply density based clustering on the distance matrix to group cohesive sentences with each other according to degree of association in distance matrix. For each point, DBSCAN determines the e-environment and checks, whether it contains more than MinPts data points DBSCAN. This algorithm uses index structures for determining the e-environment. And also it deals with arbitrary shape clusters.[8]

Major features of density based algorithms are:

- Discover clusters of arbitrary shape
- Handle noise
- One scan
- Need density parameters as termination condition

In this research work we are using:

- DBSCAN
- DENCLUE
- CLIQUE

By using these algorithms clusters will be formed according to the keywords in the user’s query and related clusters will be formed.

2.8 Built document graph:

Each cluster becomes a node in the document graph[1].

The *document graph G (V,E)* of a document *d* is defined as follows:

- *d* is split to a set of non-overlapping text fragments *t(v)*, each corresponding to a node *v*∈*V*.
- An edge *e (u,v)*∈*E* is added between nodes *u, v*∈*V* if there is an association between *t(u)* and *t(v)* in *d*.

Users typically desire concise and short summaries.[1] Adding weighted edge is the next step after generating document graph. Here for each pair of nodes u,v we compute the association degree between them, that is, the score (weight) EScore(e) of the edge e(u,v). If Score (e) ≥threshold, then e is added to E. The score of edge e(u,v) where nodes u, v have text fragments t(u), t(v) respectively is:

$$EScore(e) = \frac{\sum_{w \in (t(u) \cap t(v))} \left((tf(t(u), w) + tf(t(v), w)) \cdot idf(w) \right)}{size(t(u)) + size(t(v))}$$

Where tf(d,w) is the number of occurrences of w in d,

idf(w) is the inverse of the number of documents containing w, and size(d) is the size of the document (in words).That is, for every word w appearing in both text fragments we add a quantity equal to the tf.idf score of w. Notice that stop words are ignored.

Adding weight to nodes in document graph:

When a query Q arrives, the nodes in V are assigned query-dependent weights according to their relevance to Q. In particular, we assign to each node v corresponding to a text fragment t(v) node score NScore(v) defined by the Okapi formula as given below:

$$\sum_{t \in Q,d} \ln \frac{N - df + 0.5}{df + 0.5} \cdot \frac{(k_1 + 1)tf}{(k_1(1-b) + b\frac{dl}{avdl}) + tf} \cdot \frac{(k_3 + 1)qtf}{k_3 + qtf}$$

tf - is the term's frequency in document,
qtf is the term's frequency in query,
N- is the total number of documents in the collection,
Df- is the number of documents that contain the term,
dl -is the document length (in words),
avdl- is the average document length and
k1 (between 1.0–2.0), b (usually 0.75), and k3 (between 0–1000) are constants.[1]

2.9 Query specific summary generation:

Weights are assigned to each node of this sub-graph using a strategy similar to the Google Page ranking algorithm. And top five nodes are considered giving the best relevancy between statements according to user's query.

III. BENEFITS OF OUR SYSTEM

Following are the benefits of our system.

- Discover clusters of arbitrary shape
- Handle noise
- One scan algorithm
- We provide more accurate distance matrix using association Algorithms.
- Association algorithm used calculates good distance matrix, which is the most important requirement of any clustering algorithm using different parameters/attributes.
- It is observed that density based clustering is good clustering Algorithm than others.
- As association is used before clustering, efficient summary as per the users query.

IV. ASSUMPTIONS

While developing this project throughout it is assumed that input will be only in the text format which can be later on converted into structured format. Clustering of file with graphical data like tables, images are kept as a future enhancement part of this project.

V. CONCLUSION

In this work we presented a structure-based technique to create query-specific summaries for text documents. In particular, we first create the document graph of a document to represent the hidden semantic structure of the document and then perform keyword proximity search on this graph. We show with a user survey that our approach performs better than other state of the art approaches. Furthermore, we show the feasibility of our approach with a performance evaluation.

VI. FUTURE WORK

In the future, we plan to extend our work to account for links between documents of the dataset. For example, exploit hyperlinks in providing summarization on the Web. Furthermore, we are investigating how the document graph can be used to rank documents with respect to keyword queries. Finally, we plan to work on more elaborate techniques to split a document to text fragments and assign weights on the edge of the document graph.

VII. REFERENCES

[1] "A System for Query-Specific Document Summarization", Ramakrishna Varadarajan, Vagelis Hristidis
[2] E. Knorr and R. Ng. Algorithms for mining distance-based outliers in large datasets.
[3] Generic Text Summarization using WordNet, Kedar Bellare, Anish Das Sarma, Atish Das Sarma, Navneet Loiwal, Department of computer engg, IIT, Powai.
[4] Knowledge Discovery in Databases. AAAI/MIT Press, 1991. J. Han and M. Kamber. Data Mining: Concepts and Tec ques. Mogan Kaufmann, 2000.
[5].M. S. Chen, J. Han, and P. S. Yu., "Data mining: An overview from a database perspective", IEEE Trans. Knowledge and Data Engineering, 8:866-883, 1996.
[6] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, "Advances in Knowledge Discovery and Mining", AAAI/MIT Press, 19.
[7] L. Kaufman and P. J. Rousseeuw. Finding Groups in Data: an Introduction to Cluster Analysis. John Wiley & Sons,
[8] E. Knorr and R. Ng. Algorithms for mining distance-based outliers in large datasets.
[9] www.wikipedia.org
[10] www.google.com
[11] Text Mining, JISC, pp 1-2, March 2006.
[12] J. Nightingale, Digging for data that can change our world. The Guardian, 10 January, 2006.
[13] http://education.guardian.co.uk/elearning/story/0,,168249 6,00.html : Retrieved on January 10, 2008.