IOSR Journal of Engineering (IOSRJEN)
e-ISSN: 2250-3021, **p**-ISSN: 2278-8719
Vol. 3, Issue 7 (July. 2013), ||V5|| PP 32-37

Text Summarization Using Demographic Algorithm with Clustering Method

Priyanka katlana, Shweta Pandey
Computer Science and Engineering Department Shri vaishnav Institute of Technology RGPV University Bhopal
Computer Science and Engineering Department Shri vaishnav Institute of Technology RGPV University Bhopal

Abstract: - Text summarization is a data mining process of extracting the summary or zest from one or more documents. A summary is nothing but the actual theme of the document or set of documents. Most commonly document summery is considered to be the sentences or words from set of documents or a single document that appear more number of times in the document with corresponding to the other words. But a report on solar power may emphasis on several aspects of solar energy and may not actually have the term solar power repeated many a times. Therefore sophisticated algorithms are needed to extract the summary from the documents.

There have been several algorithms on Text and Document summarizations, utilization various aspects of similarity measures, clustering, lexical rules and distance measures. It is understood from the literature that no single technique can give best interpretation or desired result in the summarization process. Therefore in this work we propose a summarization technique with document clustering and to improve its efficiency we will include demographic algorithm in this technique.

Keywords: - *Demographic Algorithm, Condorcet, Sentence clustering, Similarity Measures, summarization, content-based searching, Entropy etc.*

I. INTRODUCTION

Demographic clustering is distribution-based. It provides fast and natural clustering of very large databases. Clusters are characterized by the value distributions of their members. It automatically determines the number of clusters to be generated. Typically, demographic data contains many categorical variables. The mining function works well with data sets that consist of this type of variables. In addition to single document summarization, which has been first studied in this field for years, researchers have started to work on multi-document summarization whose goal is to generate a summary from multiple documents that cover similar information. In this paper, we focus on generic single-document sentence extraction which forms the basis for other summarization tasks and is still a hot research topic.

The process can be decomposed into three phases: analysis, transformation and synthesis. The analysis phase analyzes the input document and selects a few salient features. The transformation phase transforms the results of analysis into a summary corresponding to users' needs with the help of new method added to it. In the overall process, compression rate, which is defined as the ratio between the length of the summary and that of the original, is an important factor that influences the quality of the summary.

Text summarization is a complex task which ideally would involve deep natural language processing capacities. In order to simplify the issue, current research is focused on extractive-summary generation. Sentence based extractive summarization techniques are commonly used in automatic text summarization to produce extractive summaries. This project proposes a sentence similarity computing method based on the three features of the sentences, on the base of analyzing of the word form feature, the word order feature and the semantic feature, using the weight to describe the contribution of each feature of the sentence, describes the sentence similarity more preciously using the demographic algorithm. Determinates the number of the clusters, uses the K-means method to cluster the sentences of the document, the similarity of each record with each of the currently existing clusters is calculated. If the biggest calculated similarity is above a given threshold, the record is added to the relevant cluster. This cluster's characteristics change accordingly. and extracts the topic sentences to generate the extractive summary for the document.

II. OBJECTIVE OF THE STUDY

The main objective of the work is to find the sentences or the words in a document that can describe the document with utmost clarity. A document contains several sentences and words. The weight of the sentences and the words differ based on the size of the document, type of the sentences, their occurrences, and their uniqueness, the similarity of the sentences or the words with other sentences and words. Though there have been several document summarization methods, there is still an immense scope of improvement in this field. No technique proposed in summarization is yet being accepted as the best technique. Hence the objective is to

develop a fast and efficient technique for document summarization based on unsupervised classification. Further the summarized document is cross verified with Google desktop search to find the significance of the generated summery.

III. PRESENT SYSTEM

In the past, extractive summarizers have been mostly based on scoring sentences in the source document based on a set of predefined features [4]. These features include linguistic features and statistical features, such as location, rhetorical structure, Entropy, presence or absence of certain syntactic features, Lexical features like presence or precedence of Nouns, Adjectives, Verbs, presence of proper names, statistical measures of term prominence, similarity between sentences, Free Text Similarity and measures of prominence of certain semantic concepts and relationships.

Two kinds of approaches have been designed to leverage the above features, supervised and unsupervised. In most supervised approaches, summarization is seen as a two class classification problem and the sentences are treated individually. However, we observe that the individual treatment of the sentences cannot take full advantage of the relationship between the sentences. For example, intuitively, two neighboring sentences with similar contents should not be put into a summary together, but when treated individually, this information is lost. Sequential learning systems such as Hidden Markov Models have also been applied, but they cannot fully exploit the rich linguistic features mentioned above since they have to assume independence among the features for tractability.

On the other hand, unsupervised approaches rely on heuristic rules that are difficult to generalize. What is ideal for us is to develop a machine learning method based on a training corpus of documents, which can take full advantage of the inter-sentence relationship and rich features which may be dependent.

The basic that have been followed in most methods can be summarized as bellow.

- ✓ Scan the Sentences and the words in the document from start till end
- ✓ Find the Probability of Occurrence of a Sentence with respect to the other Sentences.
- ✓ Find the Log Likelihood of a sentence with respect to other sentences.
- ✓ Order the Sentences into High Probability to Low probability and select the higher probability Sentences.
- ✓ In supervised Classification, The entire data set is divided into training and test set and a Neural Network or KNN classifier is build. These classifiers are fed with the features like what kind of sentence is suitable for summery with respect to particular document. Then while new document is given as input, the system can classify and extract the sentences.

IV. PROPOSED SYSTEM

First Remove the Stop words like ‘the’, ’an’, ’or’, ’am’, ’are’, ’and’ etc from the document Extract all the unique words or terms from the document and construct a Matrix T= {t1, t2, tn} where ti is ith term.
Extract all the sentences minus the stop words from the documents. S={s1,s2,…sn}
Extract the Frequency Sentence matrix by calculating if a sentence has a particular term.
S’={S11,S12,S13…Snm}, where Snm is the nth sentences frequency for mth word or term
Now create a Graph (V, E) with each sentence S’ at the vertex and if two sentences are similar, they are connected with an edge with Weight of the Edge.
The Weight is nothing but the cosine similarity between two sentences.
Thus the sentence actually “recommends” sentences which like itself under this weight calculating mechanism.
A long sentence that is similar with most of the sentences will obtain high rank
If a sentence is started with Discourse word like ‘because’, ‘hence’,’therefor’ etc, then it’s weight is decreased.
Once a Graph is built, weight of each vertices are calculated as

$$r(v_i) = d\sum_{j=1}^{n} r(v_j)\tilde{w_{ji}} + (1-d)$$

Where

$$\tilde{w_{ij}} = \begin{cases} \frac{w_{ij}}{\sum_{k=1}^{n} w_{ik}} & \text{if } \sum_{k=1}^{n} w_{ik} \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

Once the vertices are ranked, top ranked vertices are selected which is the summery of the document.

While multiple documents are considered, the top rank sentences from these documents are extracted and the sparseness of each document with respect to the sentence is calculated. A document with highest similarity with a particular sentence is kept in the cluster defined by that sentence.

If the calculated similarity is not above the threshold, or if there is no cluster (which is initially the case) a new cluster is created that contains the record alone. You can specify the maximum number of clusters, as well as the similarity threshold.

Demographic Clustering uses the statistical Condorcet criterion to manage the assignment of records to clusters and the creation of new clusters. The Condorcet criterion evaluates how homogeneous each discovered cluster is (in that the records it contains are similar) and how heterogeneous the discovered clusters are among each other.

For Sentence level similarity, Cosine similarity measure is considered and for word level summarization Entropy measure Combined with TF-IDF similarity measure is considered. Therefore the work is named as multi parameter document summarization

V. METHODOLOGY

Text summarization is a product of electronic document explosion, and can be seen as the condensation of the document collection. The use of text summarization allows a user to get a sense of the content of a full-text, or to know its information content, without reading all sentences within the full-text. Demographic correlations between two users or items, uii and uij , are de¯ned by the similarity of the vectors which represent the speci¯c users or items. That similarity is calculated by the dot-product of the two vectors.Data reduction increases scale by allowing users to find relevant full-text sources more quickly, and assimilating only essential information from many texts with reduced effort. Generic summarization system is divided into three modules: text preprocessing, summarization algorithm, and post processing, in Figure 1.

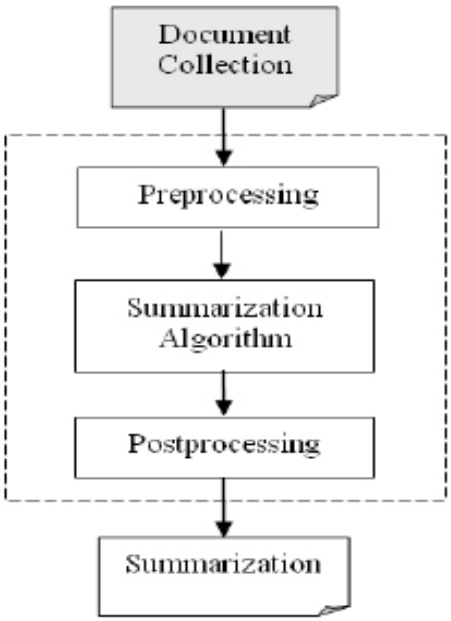

Fig 1: Summarization System Architecture

A. *Implementation steps:*

The approach consists of four steps: a) Similarity measure between sentences b) Estimating the number of clusters c) Enhancing with Demographic algorithm d) Sentences Clustering. e) Topic Sentences Extraction

1. Similarity measure between sentences

1.1. Word Form Similarity:

The word form similarity is mainly used to describe the form similarity between two sentences, is measured by the number of same words in two sentences. It should be getting rid of the stop words in the computation. If S1 and S2 are two sentences, the word form similarity is calculated by the formula (1).

Sim1 (S1, S2) =2*(Same Word (S1, S2)/ (Len (S1) +Len (S2))) (1)

Here, Same Word (S1, S2) is the number of the same words in two sentences; Len(S) is the word number in the sentence S.

1.2 Word Order Similarity:

The word order similarity is mainly used to describe the sequence similarity between two sentences. Chinese sentence can be presented by many kinds of style, the different sequence of the words stand for different meanings. Here we describe the sentence as three vectors as follows:
V1= {d11, d12… d1n1}
V2= {d21, d22… d2n2}
V3= {d31, d32… d3n3}
Here the weight d1i in vector V1 is the tf-idf value of the words; the weight d2i in vector V2 is the bi-gram whether occur in the sentence (0 stands for no-occurring, 1 stands for occurring); the weight d3i in vector V3 is the tri-gram whether occur in the sentence. The word order similarity between S1 and S2 is:

$$Sim2(S1,S2)=\lambda1*Cos(V11,V21)+\lambda2*Cos(V12,V22)+\lambda3*Cos(V13,V23) \quad (2)$$

Here $\lambda1+\lambda2+\lambda3=1$. λi stands for the ratio of each part.

1.3. Word Semantic Similarity:

The word semantic similarity is mainly used to describe the semantic similarity between two sentences. Here the word semantic similarity computing (Jiang Min, 2008) is based on the HowNet [28]. Based on semantic similarity among words, we define word-Sentence Similarity (WSSim) to be the maximum similarity between the word w and words within the sentence S. Therefore, we estimate WSSim (w, S) with the following formula:

$$WSSim(w, S) = \max \{Sim(w, i) \mid Wi \in S, \text{ where w and Wi are words}\} \quad (3)$$

Here the Sim (w, Wi) is the word similarity between w and Wi. With WSSim (w, S), we define the sentence similarity as follows:

$$Sim_3(S_1,S_2)=\frac{\sum_{w_i\in S_1} WSSim(w_i,S_2)+\sum_{w_j\in S_2} WSSim(w_j,S_1)}{|S_1|+|S_2|} \quad (4)$$

Here S1, S2 are sentences; |S| is the number in the sentence S.

1.4. Sentence Similarity:

The sentence similarity usually described as a number between zero and one, zero stands for non-similar, one stands for total similar. The larger the number is, the more the sentences similar. The sentence similarity between S1 and S2 is defined as follows:

$$Sim(S1,S2)=\lambda1*Sim1(S1,S2)+\lambda2*Sim2(S1,S2)+\lambda3*Sim3(S1,S2) \quad (5)$$

Here $\lambda1$, $\lambda2$, $\lambda3$ is the constant, and satisfied the equation:
$\lambda1+\lambda2+\lambda3=1$. In this paper, $\lambda1=0.2$, $\lambda2=0.1$, $\lambda3=0.7$.

2. Estimating the number of clusters

Determination of the optimal number of sentence clusters in a text document is a difficult issue and depends on the compression ratio of summary and chosen similarity measure, as well as on the document topics. For clustering of sentences, customers can't predict the latent topic number in the document, so it's impossible to offer k effectively. The strategy that we used to determine the optimal number of clusters (the number of topics in a document) is based on the distribution of words in the sentences:

$$k = n\frac{|D|}{\sum_{i=1}^{n}|S_i|} = n\frac{\left|\bigcup_{i=1}^{n} S_i\right|}{\sum_{i=1}^{n}|S_i|} \quad (6)$$

Where |D| is the number of terms in the document D, |Si| is the number of terms in the sentence Si, n is the number of sentences in document D. Here we analyze the property of this estimation by two extreme cases, please references the(Ramiz M. Aliguliyev, 2008) if you want to learn more detailed process of proof.(1) The document is constituted by n sentences which have the same set of terms. Therefore, the set of terms of the document coincides with the set of terms of each sentence: D= (t1, t2… tm) =Si=S. From the definition (6) follows that

$$k = n\frac{\left|\bigcup_{i=1}^{n} S_i\right|}{\sum_{i=1}^{n}|S_i|} = n\frac{\left|\bigcup_{i=1}^{n} S\right|}{\sum_{i=1}^{n}|S|} = n\frac{|S|}{\sum_{i=1}^{n}|S|} = 1$$

(2) The document is constituted by n sentence which do not have any term in common, that is, Si∩Sj=Φ for i≠j. This means that each term belonging to $D = \bigcup_{i=1}^{n} S_i$ belongs only to one of the sentences Si, therefore $|D| = \left|\bigcup_{i=1}^{n} S_i\right| = \sum_{i=1}^{n} |S_i|$ from which follows that k=n.

3. Enhancing with Demographic algorithm

The correlations between neighbourhood members and active users or items are re-evaluated, this time by also taking into account existing demographic correlations. Demographic correlations between two users or items, uii and uij , are defined by the similarity of the vectors which
represent the specific users or items. That similarity is calculated by the dot-product of the two vectors. Once the demographic vectors were constructed for all users, we could now proceed and calculate the proximity between the active user, ua and the users ui, for i = 1; 2; :::; l, belonging to his neighbourhood, as it was de¯ned by their registered demographic data. Their demographic correlation, dem corai, was calculated by applying the vector similarity formula on the corresponding demographic vectors. The final step in the recommendation procedure was Prediction Generation. For that purpose, the formula used in plain User-based Filtering [5] was modi¯ed as follows:

$$udem\_pr_{aj} = \bar{r_a} + \frac{\sum_{i=1}^{l}(r_{ij} - \bar{r_i}) * enh\_cor_{ai}}{\sum_{i=1}^{l} |enh\_cor_{ai}|}$$

the only difference from prediction generation, as executed in plain User-based Collaborative Filtering, being the enhanced correlation factor.

4. Sentences Clustering

Once determinates the number of sentences clusters, we can use the K-means method to cluster the sentences of the document. K-means algorithm can be described as follows:
Input: n sentences
K: the number of clusters
Output: the sentences clusters
Algorithm to be used:
Step1: Random select K sentences into K clusters respectively, these sentences represent the initial cluster central sentences.
Step2: Assign each sentence to the cluster that has the closest central sentence.
Step3: When all sentences have been assigned recalculate the central sentence of each cluster. The central sentence is the one which own the lowest accumulative similarity.
Step4: Repeat Steps 2 and 3 until the central sentence no longer move. This produces a separation of the sentences into K clusters from which the metric to be minimized can be calculated.

5. Topic Sentences Extraction

Based on the result of section C, assume the sentences cluster is: D = {C1, C2… Ck}. First, determinates the central sentence μi of each cluster based on the accumulative similarity between the sentence Si and other sentences, then calculates the similarity between the sentence Si and the central sentence μi. Assume that the similarity of central sentence μi as 1, sorts the sentences based on its similarity weight, and chooses the high weight sentences as the topic sentences. At the same time, considering the recall rate of the text summarization, the text summary should include every cluster sentences according to the principle of priority extract clusters in the process of extracting sentences

VI. CONCLUSION

Document Summarization is the process of extracting the excerpt from the document which could be either a single sentence, set of sentences from the document, set of words or sentences that are synthesized from the abstract which are not part of the documents. A sentence cluster is a tree structure where a sentence common to all document is at the top of the tree followed by those sentences which are similar to next degree and so on.
In this work we have presented a unique ¯ltering approach that draws ideas from existing algorithms and combines them with demographic information available in recommender systems data sets. Tagging the documents with such clusters has the advantage of easy indexing and searching. Several techniques are proposed in the past for efficient document summarization. Further there are several thresholding and similarity schemes available for document summarization. Each of the features has its pros and cons.

REFERENCE

[1] M. Balabanovic and Y. Shoham. Fab: Content-based, collaborative recommendation. Communications of the ACM, 40, 1997

[2] M.J. Pazzani. A framework for collaborative, content-based and demographic filtering. Artificial Intelligence Review, 13:393{408, 1999.

[3] Paul Resnick, Neophytos Iacovou, Mitesh Sushak, Peter Bergstrom, and John Riedl. Grouplens: An open rchitecture for collaborative ¯ltering of netnews. In ACM 1994 conference on Computer Supported Cooperative Work, pages 175{ 186, New York, NY, 1994.

[4] S. Brin and L. Page. "The anatomy of a large-scale hyper textual Web search engine".

[5] Computer Networks and ISDN System. 30(1–7): 107–117. 1998.

[6] J. Carbonell and J. Goldstein. "The use of MMR, diversity-based reranking for reordering Documents and producing summaries". SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 24-28 August. Melbourne, Australia, 335-336. 1998

[7] G. Erkan and D. R. Radev. "LexRank: Graph-based Lexical Centrality as Salience in Text Summarization". Journal of Artificial Intelligence Research (JAIR), 22, 457-479. AI Access Foundation. 2004.

[8] K Filippova, M. Mieskes, V. Nastase, S. P. Ponzetto and M. Strube. "Cascaded Filtering for Topic-Driven Multi-Document Summarization". Proceedings of the Document Understanding Conference. 26-27 April. Rochester, N.Y., 30-35. 2007.

[9] M. K. Ganapathiraju. "Relevance of Cluster size in MMR based Summarizer: A Report 11-742: Self-paced lab in Information Retrieval". November 26, 2002.

[10] "The Document Understanding Conference (DUC)". http://duc.nist.gov.

[11] A. Jain, M. N. Murty and P. J. Flynn. "Data Clustering: A Review". ACM Computing Surveys. (3), 264-323, 1999.

[12] C. Jaruskulchai and C. Kruengkrai. "Generic Text Summarization Using Local and Global Properties". Proceedings of the IEEE/WIC international Conference on Web Intelligence. 13- 17 October. Halifax, Canada: IEEE Computer Society, 201-206, 2003.

[13] A. Kiani –B and M. R. Akbarzadeh –T. "Automatic Text Summarization Using: Hybrid Fuzzy GA-GP". IEEE International Conference on Fuzzy Systems. 16-21 July. Vancouver, BC, Canada, 977 -983, 2006.