CLUSTERING BASED DOCUMENT SUMMARIZATION

Simran kaur¹, wg.cdr Anil Chopra²

¹Assistant Professor, lingayas university Faridabad
² Manav Rachna International University Fbd

Abstract: Document summarization involves summarizing document as the information is continuously increasing with such a huge amount. Users do not have much time to spend reading thousands of lines. Today users want maximum information which describes everything and occupies minimum space. This paper discusses an improved approach for document summarization by using clustering. Summarization is process of producing single summaries from a document. The three major problems that were introduced in single document summarization were coped in k means clustering summarization i.e. coping with redundancy, coherency in summary, identifying difference in sentences. To identify similarity in documents various similarity measures are used i.e. similarity between the sentences of documents and then grouping them in clusters based on their tfidf values of the words.

KEYWORDS: Document Summarization, Sentence Pre-processing, Natural Language processing (nlp), Clustering, Word Net.

1. INTRODUCTION

Text summarization is the process of extracting important information from a given text. Based on the how this important information is presented to the user, two types of text summarization systems are defined. They are 1) Extractive summarization system 2) abstractive summarization system. In Extractive in Extractive Summarization system important text segments of the original text are identified and presented as they are. In abstractive summarization original text is interpreted and is written in a condensed form so that the resulting summary contains the essence of the original text. The summary in extractive summarization contains the words and sentences of the original text. This may not happen in abstractive summarization system. Stop lists play an important role in building search engines and text summarization systems. They help in filtering useful information from the original text. Traditional Stop lists are those which are specific to a natural language and are primarily developed for use in a search engine. Since Text summarization is a complex task involving natural language processing, it uses natural language processing tools like Dictionaries, Thesaurus, Word net, and POS Tagger etc... A Parts-Of-Speech Tagger (POS Tagger) is a piece of software that reads text in some language and assigns parts of speech to each word (and other token), such as noun, verb, adjective, etc., although generally computational applications use more fine-grained POS tags like ‘noun-plural’. Word net is an on-line lexical reference system whose design is inspired by current psycholinguistic theories of human lexical memory. Second one is a Stanford Log-linear Part-Of-Speech Tagger. Term frequency is a statistical measure used in calculating relevance of a document. It tells something about the document as a whole with respect to a user query. Many document summarization methods are based on conventional term weighting approach for picking a set of frequencies and term weights based on the number of occurrences of the words is calculated. Summarization methods based on semantic analysis also use term weights for final sentence selection. The term weights generally used are not directly derived based on any mathematical model of term distribution or relevancy. In our approach, we use a term frequency model to mathematically characterize the relevance of terms in a document. This model is then used to extract important sentences from the documents. Another major issue to be handled in our study is to generate a “user-friendly” summary at the end.

2. RELATED WORK ON SENTENCE PRE-PROCESSING

The developments in storage devices and computer networks have given the scope for the world to become a paperless community, for example Digital newspaper systems and digital library systems. A paperless community is heavily dependent on information retrieval systems. Text summarization is an area that supports the cause of information retrieval systems by helping the users to get their needed information. This paper discusses on the relevance of using traditional stop lists for text summarization and the use of Statistical analysis for sentence scoring. A new methodology is proposed for implementing the stop list concept and statistical analysis concept based on parts of speech tagging. A sentence scoring mechanism has been developed by combining the above methodologies with semantic analysis. This sentence scoring method has given good results when applied to find out the relation between natural language queries and the sentences in a document. Abstract-Much existing research on blogs focused on posts only, ignoring their comments. Our user study conducted on summarizing blog posts, however, showed that reading comments does change one’s understanding about blog
In information retrieval it is assumed that documents that are similar to each other are likely to be relevant for the same query, and therefore having the document collection organized in clusters can provide improved document access. Different clustering techniques exists the simplest one being the one-pass.

3.1 Clustering Algorithm

We have implemented an agglomerative clustering algorithm which is relatively simple, has reasonable complexity, and as it will be shown gave us good results. Our algorithm operates in an exclusive way, meaning that a document belongs to one and only one cluster— while this is our working hypothesis, it might not be valid in some cases. The input to the algorithm is a set of document representations implemented as vectors of terms and weights (term_1 D weight_1; term_n D weight_n).

Initially, there are as many clusters as input documents; as the algorithm proceeds clusters are merged until a certain termination condition is reached. The algorithm computes the similarity between vector representations in order to decide whether or not to merge two clusters. The pseudo code of the algorithm is shown in Algorithm.

Algorithm 1: Clustering Algorithm

Given: LDOCS: a list of vector representations; THR: a similarity threshold

Begin for all vectori 2 LDOCS do

1. clusteri vectori
2. end for
3. compute similarity matrix for every pair of vectors (simD.vectori ; vectorj )
4. max_sim 0; change true;
5. while change do
6. for all active.clusteri / do
7. for all active.clusterej (i 6= j) do
8. current simC .clusteri ; clusterej /
9. if max_sim _ current then
10. max_sim current
11. cluster1 clusteri
12. cluster2 clusterej
13. end if
14. end for
15. end for
16. if max_sim > THR then
17. /* creates a new cluster with the vectors from cluster1 and cluster2 and marks cluster1 and cluster2 as inactive */
18. merge.clusteri; cluster2/ 
19. else
20. change false
21. end if
22. end while
23. return clusters
24. end

The similarity metric we use is the cosine of the angle between two vectors. This metric gives value one for identical vectors and zero for vectors which are orthogonal.
(no related). Various options have been implemented in order to measure how close two clusters are, but for the experiments reported here we have used the following approach: the similarity between two clusters (simC) is equivalent to the similarity (simD) between the two more similar documents in the two clusters—this is known as single linkage in the clustering literature; we take simD to be the cosine metric computed as follows:

\[ \text{Cosine}(d1, d2) = \frac{w1d1 \cdot w2d2}{\sqrt{w1d1 \cdot w1d1} \cdot \sqrt{w2d2 \cdot w2d2}} \]

Here \( d1 \) and \( d2 \) are document vectors and \( w_i; d_k \) is the weight of term \( i \) in document \( d_k \). If this similarity between two clusters is greater than a threshold—experimentally obtained—they are merged together. Each iteration in the algorithm the most similar pair of clusters is merged. If this similarity is less than a certain threshold the algorithm stops. Merging two clusters consist of a simple step of set union, so there is no re-computation involved—such as computing a cluster centroid.

3.2 Proposed Methodology

So we presented our approach towards ‘k means clustering Automated Text Summarization’. Our approach attempts to generate a text summary from the article of newspapers, while avoiding the repetition of identical or similar information and presenting the information in such a way that makes sense to the reader. The proposed algorithm work as follows in fig6:

1. \( S= (D1...Dn) \) here’s is a collection of all documents which are related to same topic.
2. We use vector of tokens to represent each documents \( T= (t1...tn) \).
3. These tokens are the words appearing frequently in document except some stop words and the threshold value is determined for each document for the words to be taken.
4. After getting tokens or words from each documents we get tf*idf values or weights of the words using word net dictionary.
5. After we get weighted measures of all the words which is denoted by \( V(d1)= (w1...wi...xn) \) where \( \text{win} = \text{TF(di, tj)} \cdot \log(N/DF(tj)) \)
6. Here TF stands for term frequency which tells how many times a term occurs in documents divided by total number of terms in documents.
7. IDF stands for inverse document frequency which is \( \log_e (\text{total no of docs/no of documents with term t in it}) \).
8. After getting tf*idf values of all the tokens we group them in cluster by k means clustering algorithm. Here we initially take three tokens randomly as initial centroids.
9. For \( k=\text{no of desired clusters i.e. 3} \) we take three values \( r1,r2,r3 \) where each of them have a particular tf.idf value

10. Then we compare the Euclidean distance \( r1f1=r11, r2f1=r21, r3f1=r31 \).
11. If \( d11<d21 \& d11<d31 \) then \( f \) belongs to first cluster i.e. \( c1 \) or
12. \( D21<d11 \& d21<d31 \) then \( f1 \) belongs to \( c2 \)
13. These iterations continue till the epoch value which is no of iterations given i.e. or mean value of all the \( d1...dn \) values is nearly equal to mean of all \( k \) means which is the centroid for all documents.
14. After we get a stable clusters for all the tokens then they are grouped according to their frequencies in the respective clusters \( c1,c2,c3 \)
15. After we get clusters and all tokens arranged in clusters the sentences having the words are selected in clusters from documents \( d1...dn \).
16. After applying summarizer on all the three clustered documents we get summaries of the cluster on the basis of word scoring.

K MEANS ALGORITHM

1. Input:
   \( D = d1, d2....... dn \) // set of n data items or tokens
   \( k // Number of desired clusters \)
2. Output:
   A set of \( k \) initial centroids.
3. Steps:
   A. Calculate the average score of each data point;
      \( d1=x1,x2,x3....xn \)
   B. \( d1(avg)=(w1*x1+w2*x2+w3*x3+......+wm*xm)/m \) // where \( x = \text{the attributes value, m = number of attributes and } w = \text{weight to multiply to ensure fair distribution of cluster Sort the data based on Euclidean distance from the k centroids; } \)
      a. Calculate Euclidean distance of the data points from the centroid which is \( d^2(xn,xk) \)
      b. Divide the data into clusters;
      c. Calculate the mean value of the each cluster;
      d. Take the nearest possible data point of the mean as the initial centroid for each data subsets.
      e. The clustering continues until it converges to its stable clusters.

3.3 Results

1. Calculate distance.
   For \( k=3 \)
   Random function () \( R1\ R2\ R3 \) \( R1=250\ R2=350\ R3=543 \) \( C0=R1\ NUM=250 \)
2. \( D1=\sqrt{\text{sqrt}(C0)^2-\text{sqrt}(\text{num})^2} \) \( 62500-62500=0 \)
   \( C0 \text{ is a random function that is changing } \)
   \( C0=350 \)
3. \( D2=\sqrt{\text{sqrt}(350)^2-\text{sqrt}(250)^2} \) \( D2=122500-62500=244.94 \text{ Similarly d3=482 } \)
4. \( D1 \text{ is smallest so it belongs to c0 For round2 or epoch2 D1=244.94 } \)
D2=0
D3=415.14
1. For round 3
D1=482
D2=415
D3=0
2. For round 4
D1=206.09
D2=132.37
D3=435.74
Mean of all distances = sum of all distances/count
3. Mean1 = mean of all distances
4. Mean2 = mean of random numbers(R1,R2,R3)
5. If mean1==mean2 then clusters are appropriate
6. else
7. Repeat iterations again by taking different random functions. After getting the clusters for all words we get the sentences belonging to these words and get highly occurring sentences from each document in three clusters. After getting the sentences in each cluster highly occurring sentence are in ne cluster similarly. Then we apply summarizer to each cluster of sentences and get the summaries of all three clusters to get summaries based on their tf.idf values. Our proposed approach mainly summarizes the clustered documents and generating summaries of clusters instead of creating summaries of the entire document into single summary. Clustering of tokens is done by comparing the Euclidean distance of their frequencies to randomly generated tokens. The stability of the cluster is seen by comparing mean values of their distances to mean value of the random number.

3.4 Evaluation
1. In order to evaluate the quality of of our summarized documents we calculate intra cluster similarity and inter cluster similarity.
2. Intracluster similarity: For each cluster j we measure the intra cluster similarity using:
   IntraSim (j) = \sum \text{IntraSim} (j) = \sum \text{cosine (ci, d)/d} 
   order to compute the average intra cluster similarity, we compute an average over all the clusters.
3. So cosine similarity is the cosine of two vectors can be derived by using the Euclidean dot product formula:
   a.b=|a||b|cos (θ)
4. Given two vectors of attributes, A and B, the cosine similarity, cos(θ), is represented using a dot product and magnitude as The resulting similarity ranges from −1 meaning exactly opposite, to 1 meaning exactly the same, with 0 indicating orthogonality (decorrelation), and in-between values indicating intermediate similarity or dissimilarity.

Figure 1 summarization flowchart
5. So for k=20 the similarity is = 5.6sqrt5^2+sqrt6^2=1 so two clusters are highly similar
6. Inter Cluster Similarity: We compute the pair wise similarity between each pair of cluster by computing the similarity between their centroids. Average inter-cluster similarity is measured by computing an average over all the distinct pair wise similarity between centroids.

\[ \text{Sim} (c_0, c_1) = \sqrt{5(x_1^2-x_0^2)} = 3/20 \]

3.5 Observations

1. The time taken for doing clustering increases as the number of clusters increase.
2. We also observe that K-Means is extremely sensitive to initial seeds and outliers. We tried multiple runs for K-Means and each time the clusters we got were significantly different. Sometimes due to selection of a document which had no similar documents
3. As the number of clusters increased, the intra cluster similarity increases and the inter cluster similarity decreases. The intra-cluster similarity always remained greater than inter-cluster similarity.
4. The above figure shows the effect of increase in N on the quality of the clusters obtained by K-Means for K=6. We see that as N increases the quality of cluster degrade as the intra cluster similarity decreases and the inter cluster similarity increases. We speculate this is due to large number of documents being present in cluster which are barely similar to each other. However, if we increase the value of K with the increase in the value of N, the cluster quality will not degrade.

References

Number citations consecutively in square brackets [1]. The sentence punctuation follows the brackets [2]. Multiple references [2], [3] are each numbered with separate brackets [1]–[3]. Please note that the references at the end of this document are in the preferred referencing style. Please ensure that the provided references are complete with all the details and also cited inside the manuscript (example: page numbers, year of publication, publisher’s name etc.).

Equations

The similarity metric we use is the cosine of the angle between two vectors. This metric gives value one for identical vectors and zero for vectors which are orthogonal (no related). Various options have been implemented in order to measure how close two clusters are, but for the experiments reported here we have used the following approach: the similarity between two clusters (simC) is equivalent to the similarity (simD) between the two more similar documents in the two clusters—this is known as single linkage in the clustering literature; we take simD to be the cosine metric computed as follows:

\[ \text{Cosine } (d_1, d_2) = \frac{w_1d_1^T w_2d_2}{\sqrt{w_1d_1^T w_1d_1} \sqrt{w_2d_2^T w_2d_2}} \]

Here \( d_1 \) and \( d_2 \) are document vectors and \( w_i \; d_k \) is the weight of term i in document dk. If this similarity between two clusters is greater than a threshold—experimentally obtained—they are merged together. Each iteration in the algorithm the most similar pair of clusters is merged. If this similarity is less than a certain threshold the algorithm stops. Merging two clusters consist of a simple step of set union, so there is no re-computation involved—such as computing a cluster centroid.

References


[8] Quantifying The Limits And Success Of Extractive Summarization Systems Across Domains


[10] A Novel Automatic Text Summarization Study Based On Term Co-Occurrence

[12] Improvement of K-means Clustering algorithm with better initial centroids based on weighted average

[13] Collaborative Clustering of XML Documents
Sergio Greco, Francesco Gullo, Giovanni Ponti, Andrea Tagarelli.


[17] A New Centroid-Based Classifier for Text Categorization

[18] Extracting main content of a topic on online social network by multi-document summarization

[19] Query-focused Multi-document Summarization Using Keyword Extraction Liang Ma1, 2 Tingting He1, 2 Fang Li1, 2 Zhuomin Gui1, 2 Jinguang Chen1, 2 (1Department of Computer Science, Huazhong Normal University, Wuhan, China, 430079 2Network Media Branch, National Language Resources Monitoring and Research Center, Wuhan, China, 430079) E-mail: maliang1897@yahoo.com.cn; tthe@mail.ccnu.edu.cn.