A Reliable Topical Diversity Measure for Text Summarization

Prithvi Deep Chawla, Mohit Jain, Tushan Jain and Saksham Singhal

***Abstract*—We design novel methods to improve efficiency of the task of Text Summarization using a class of sub-modular functions. These functions each combine two terms, one which encourages the summary to be representative of the corpus, and the other which positively rewards diversity. Critically, our functions are monotone non-decreasing and sub-modular, which means that an efficient scalable greedy optimization scheme has a constant factor guarantee of optimality. Our approach extends on the previously existing methods and improve them both, mathematically and algorithmically. When evaluated on DUC 2004 corpora, we obtain atleast as good results as the existing state-of-art Text Summarization Systems in generic document summarization.**

Keywords*—*Text Summarization, Submodular Functions, Spectral Clustering, ROGUE, CLUTO.

I. Introduction

In this paper, we address the problem of generic extractive summarization from collections of related documents, a task commonly known as multi-document summarization. Text summarization has become an important and timely tool for assisting and interpreting text information in todays fast-growing information age. It is very difficult for human beings to manually summarize large documents of text. There is an abundance of text material available on the internet. However, usually the Internet provides more information than is needed. Therefore, a twofold problem is encountered: searching for relevant documents through an overwhelming number of documents available, and absorbing a large quantity of relevant information. The goal of automatic text summarization is condensing the source text into a shorter version preserving its information content and overall meaning.

Extractive text summarization process can be divided into two steps: (1) Pre Processing step and (2) Processing step.
Pre-Processing is structured representation of the original text. It usually includes: (a) Sentences boundary identification. In English, sentence boundary is identified with presence of dot at the end of sentence. (b) Stop-Word EliminationCommon words with no semantics and which do not aggregate relevant information to the task are eliminated. (c) Stemming-The purpose of stemming is to obtain the stem or radix of each word, which emphasize its semantics.
In *Processing step*, the coverage and diversity measures are calculated for each sentence using their *tf-idf* scores. Then a weight-age is appropriately given to these parameters and the final score for a sentence is calculated. The top ranking sentences are selected for the final summary.

This paper is organized as follows. First we discuss the tools used for Text Summarization, CLUTO and ROUGE.Next, we describe three different approaches to tackle the problem of Text Summarization. In the subsequent section, we display our results for each of the specified approaches and lastly, we conclude the paper.

II. Tools Used

A. *CLUTO*

CLUTO is a software package for clustering low and high dimensional datasets and for analyzing the characteristics of the various clusters. CLUTO provides three different classes of clustering algorithms that operate either directly in the objects feature space or in the objects similarity space. These algorithms are based on the partition, agglomerative and graph partitioning paradigms. A key feature in most of CLUTOs clustering algorithms is that they treat the clustering problem as an optimization process which seeks to maximize or minimize a particular clustering criterion function defined either globally or locally over the entire clustering solution space.

CLUTO provides access to its various clustering and analysis algorithms via the vcluster and scluster stand-alone programs. The key difference between these programs is that vcluster takes as input the actual multi-dimensional representation of the objects that need to be clustered whereas scluster takes as input the similarity matrix (or graph) between these objects. Besides this difference, both programs provide similar functionality. We have incorporated the vcluster method in our implementation.

Usage:
vcluster[optionalparameters]MatrixFileNClusters
scluster [optional parameters] GraphFile NClusters

MatrixFile, is the name of the file that stores the n objects to be clustered. In vcluster, each one of these objects is considered to be a vector in an m-dimensional space.
GraphFile, is the name of the file that stores the adjacency matrix of the similarity graph between the n objects to be clustered.
NClusters, is the number of clusters that is desired.

B. *ROUGE*

ROUGE, or Recall-Oriented Understudy for Gisting Evaluation, is a set of metrics and a software package used for evaluating automatic summarization and machine translation software in natural language processing. The metrics compare an automatically produced summary or translation against a reference or a set of references (human-produced) summary or translation. Rouge generates three scores (recall, precision and F-measure) for each evaluation. Precision and F-measure scores are useful when the target summary length is not enforced. ROUGE uses model average to compute the overall ROUGE scores when there are multiple references. The model average option is specified using "A" (for Average) and the best model option is specified using "B" (for the Best).
There is a specific format for the system generated file as: a)Each of the system generated sentences should be in single line. b) Output from the system should be preferably in form of plaint/text file.
Usage:
java -jar C_ROUGE.jar [System Generated FileName] [Folder Path Cointaining Reference Summary] [N] [C]
C_ROUGE.jar is the jar file which incorporate ROUGE-1.5.4.pl
[System Generated FileName] is file name of the document generated by our system
[Folder Path Cointaining Reference Summary] is folder path of Reference Summary
[N], N is N-N value of ROUGE-N to be calculated
[C] is A for Average Rouge Score or B for Best Rouge Score

III. Approaches

Submodular functions share many properties in common with convex functions, one of which is that they are closed under a number of common combination operations (summation, certain compositions, restrictions, and so on). These operations give us the tools necessary to design a powerful submodular objective for submodular document summarization that extends beyond any previous work.
We are given a set of objects V=$\{v_1, ... , v_n\}$ and a function F : $2^V \rightarrow$ R that returns a real value for any subset $S \subseteq$ V . We are interested in finding the subset of bounded size $|S| \leq$ k that maximizes the function, e.g, $argmax_{S \subseteq V}$ F(S). Sub-modular functions are those that satisfy the property of diminishing returns: for any $A \subseteq B \subseteq V \setminus v$, a sub-modular function F must satisfy F(A+v) - F(A) $\geq$ F(B + v) - F(B). That is, the incremental value of v decreases as the context in which v is considered grows from A to B. An equivalent definition, useful mathematically, is that for any $A, B \subseteq V$, we must have that $F(A) + F(B) \geq F(A \cup B) + F(A \cap B)$. If this is satisfied everywhere with equality, then the function F is called modular, and in such case $F(A) = c + \Sigma_{a \in A} \vec{f_a}$ for a sized $|V|$ vector f of real values and constant c. A set function F is monotone non-decreasing if $\forall A \subseteq B, F(A) \geq F(B)$. As shorthand, in this paper, monotone non-decreasing submodular functions will simply be referred to as monotone submodular.
Two properties of a good summary are relevance and non-redundancy. Objective functions for extractive summarization usually measure these two separately and then mix them together trading off encouraging relevance and penalizing redundancy. The redundancy penalty usually violates the monotonicity of the objective functions (Carbonell and Goldstein, 1998; Lin and Bilmes, 2010). We therefore propose to positively reward diversity instead of negatively penalizing redundancy. In particular, we model the summary quality as

$$F(S) = L(S) + \lambda R(S)$$

where L(S) measures the coverage, or fidelity, of summary set S to the document, R(S) rewards diversity in S, and $\lambda \geq 0$ is a trade-off coefficient. Note that the above is analogous to the objectives widely used in machine learning, where a loss function that measures the training set error (we measure the coverage of summary to a document), is combined with a regularization term encouraging certain desirable (e.g., sparsity) properties (in our case, we regularize the solution to be more diverse). In the following, we discuss how both L(S) and R(S) are naturally monotone submodular.

Coverage Measure :

L(S) can be interpreted either as a set function that measures the similarity of summary set S to the document to be summarized, or as a function representing some form of coverage of V by S. Most naturally, L(S) should be monotone, as coverage improves with a larger summary. L(S) should also be submodular: consider adding a new sentence into two summary sets, one a subset of the other. Intuitively, the increment when adding a new sentence to the small summary set should be larger than the increment when adding it to the larger set, as the information carried by the new sentence might have already been covered by those sentences that are in the larger summary but not in the smaller summary. This is exactly the property of diminishing returns. Indeed, Shannon entropy, as the measurement of information, is another well-known monotone submodular function.

$$L(S) = \sum_{i \in V} min\{C_i(S), \alpha C_i(V)\}$$

where $C_i : 2^V \rightarrow R$ is a monotone submodular function and $0 \leq \alpha \leq 1$ is a threshold co-efficient.

Basically, $C_i(S)$ measures how similar S is to element i, or how much of i is covered by S. Then C_i (V) is just the largest value that $C_i(S)$ can achieve. We call i saturated by S when $min\{C_i(S), \alpha C_i(V)\} = \alpha C_i(V)$. When i is already saturated in this way, any new sentence j can not further improve the coverage of i even if it is very similar to i (i.e., $C_i(S \cup \{j\}) - C_i(S)$ is large). This will give other sentences that are not yet saturated a higher chance of being better covered, and therefore the resulting summary tends to better cover the entire document. One simple way to define $C_i(S)$ is just to use

$$C_i(S) = \sum_{j \in S} w_{i,j}$$

where $w_{i,j} \geq 0$ measures the similarity between i and j.

Diversity Measure :

Instead of penalizing redundancy by subtracting from the objective, we propose to reward diversity by adding the following to the objective:

$$R(S) = \sum_{i=1}^{K} \sqrt{\sum_{j \in P_i \cap S} r_j}$$

where P_i , i = 1, ..., K is a partition of the ground set V (i.e.,$\cup_i P_i = V$ and the P_i s are disjoint) into separate clusters, and $r_i \geq 0$ indicates the singleton reward of i (i.e., the reward of adding i into the empty set). The value r_i estimates the importance of i to the summary.

The function R(S) rewards diversity in that there is usually more benefit to selecting a sentence from a cluster not yet having one of its elements already chosen. As soon as an element is selected from a cluster, other elements from the same cluster start having diminishing gain, thanks to the square root function.

A. K-means and Hierarchical Clustering

We used the DUC-2004 dataset which contains 50 TDT topics/events/timespan and a subset of the documents TDT annotators found for each topic/event/timespan. The documents were taken from the AP newswire and the New York Times newswire and subsets were formed with an average of 10 documents per subset. The dataset also contains very short summaries of each document ($\leq$ 75 bytes) and a short summary ($\leq$ 665 bytes) of each cluster. The dataset was fed into CLUTO to perform K-means and Hierarchical Clustering to obtain clusters referring to similar data.

We ran a Grid Search on the values of α and λ to get the best optimal value to maximize the sub modular function.

Algorithm:

- Summary $\rightarrow \phi$
- allowedClusters $\leftarrow$ allClusters
- *while size(Summary)* $\leq$ *665*:
  - *pick the cluster most similar to corpus from allowedClusters.* $\rightarrow$ chosenCluster
  - chosenSentence $\leftarrow$ *highest ranking sentence of chosenCluster based on coverage and diversity measure*
  - Summary $\leftarrow$ chosenSentence

B. Spectral Clustering

Spectral clustering techniques make use of the spectrum (eigenvalues) of the similarity matrix of the data to perform dimensionality reduction before clustering in fewer dimensions. The similarity matrix is provided as an input and consists of a quantitative assessment of the relative similarity of each pair of points in the dataset. Spectral clustering treats the data clustering as a graph partitioning problem without make any assumption on the form of the data clusters. The pre-processing required for the spectral clustering is that it construct graph and the similarity matrix representing the dataset. From dataset it forms the associated Laplacian matrix and then compute eigenvalues and eigenvectors of the Laplacian matrix. After this it maps each point to a lower-dimensional representation based on one or more eigenvectors.

Algorithm:

Given set of points S = $\{s_1,...,s_n\}$ in R^l that we want to cluster into k subsets:

- Form the affinity matrix $A \in R^{n*n}$ defined by $A_{ij} = exp(-(s_i - s_j)^2/2\sigma^2)$ if i$\neq$ j, and A_ii=0.
- Define D to be the diagonal matrix whose (i,i)-element is the sum of A's i-th row, and construct the matrix L = $D^{-1/2}\mathrm{A}D^{-1/2}$.
- Find $x_1, x_2, ...x_k$, the k largest eigenvectors of L (chosen to be orthogonal to each other in the case of repeated eigenvalues), and form the matrix X = $[x_1 x_2 ... x_k] \in R^{nk}$ by stacking the eigenvectors in columns.
- Form the matrix Y from X by renormalizing each of X's rows to have unit length (i.e. $Y_{ij} = X_{ij}/(\Sigma_j X_i^2{}_j)^{1/2}$)
- Treating each row of Y as a point in R^k, cluster them into k clusters via K-means or any other algorithm (that attempts to minimize distortion).
- Finally,assign the original point s_i to cluster j if and only if row i of the matrix Y was assigned to cluster j. Here, the scaling parameter σ^2 controls how rapidly the affinity A_{ij} falls off with the distance between s_i and s_j.

C. Our Revised Approach : K-means Clustering

We tried to improvise the algorithm discussed in part-A by choosing the clusters a bit more consciously and giving more weight-age to the coverage rather than the diversity. Once a sentence is chosen from a cluster, this cluster is not considered for the selection of the next sentence. In this way we incorporate diversity while giving coverage the pole position here.

Algorithm:

- Summary $\rightarrow \phi$
- allowedClusters $\leftarrow$ allClusters
- oldChosenCluster $\rightarrow \phi$
- *pick the cluster most similar to corpus from allowedClusters.* $\rightarrow$ chosenCluster
- chosenSentence $\leftarrow$ *highest ranking sentence of chosenCluster based on coverage*
- Summary $\leftarrow$ chosenSentence
- prevCluster $\leftarrow$ chosenCluster
- *while size(Summary)* $\leq$ *665*:
  - allowedClusters $\leftarrow$ allowedclusters - prevCluster + oldChosenCluster
  - *pick the cluster most similar to corpus from allowedClusters.* $\rightarrow$ chosenCluster

- chosenSentence ← *highest ranking sentence of chosenCluster based on coverage*
- Summary ← chosenSentence
- oldChosenCluster ← prevCluster

RESULTS AND CONCLUSION

The values for $\lambda and \alpha$ values for the diversity and coverage measures giving us the submodular function were calculated using a sweep-search for the best values of ROGUE scores. We recovered the best values as follows :

$$\alpha = 15$$
$$\lambda = 4$$

With the selected $\alpha \& \lambda$ values, the clusters were summarized and their ROGUE scores calculated to estimate the efficiency.

Approach	ROGUE-R	ROGUE-F
(A) K-means and Hierarchical	0.3843	0.3792
(B) Spectral	0.3724	0.3674
(C) Revised K-means	0.347	0.332

We can conclude by saying that K-means and Heirarchical clustering using a weighted consideration for both, the diversity and coverage, gives us the best scores in Text Summarization.

ACKNOWLEDGMENT

The success of this project couldn't have been possible without the support and guidance of our mentor, Litton J Kurisinkel. We would also like to thank our course instructor, Prof. Vasudev Verma, for giving us this opportunity to work on this challenging project and expand our knowledge in Information Retrieval and Extraction.

REFERENCES

[1] Vishal Gupta and Gurpreet Singh Lehal, *A Survey of Text Summarization Extractive Techniques*

[2] Hui Lin and Jeff Bilmes, *A Class of Submodular Functions for Document Summarization*

[3] Hui Lin and Jeff Bilmes, *Multi-document Summarization via Budgeted Maximization of Submodular Functions*

[4] Ying Zhao and George Karypis, *Criterion Functions for Document Clustering Experiments and Analysis*

[5] Chin-Yew LIN, *A Package for Automatic Evaluation of Summaries*

[6] DUC 2004, $http://www.nlpir.nist.gov/projects/duc/data/2004\_data.html$