Document Summarization using Clustering and Submodular Functions

Aishwary Gupta
Computer Science
and Engineering
201302216
aishwary.gupta@research.iiit.ac.in

Prabhakar Bikkaneti
Computer Science
and Engineering
201505618
b.prabhakar@students.iiit.ac.in

Sahil Swami
Computer Science
and Engineering
201302071
sahil.swami@research.iiit.ac.in

***Abstract*—We design novel methods to do the task of Text Summarization using a class of sub-modular functions and clustering techniques. The sub-modular functions each combine two terms, one which encourages the summary to be representative of the corpus, and the other which positively rewards diversity. Our clustering techniques cluster similar types of sentences optimally. Critically, our functions are monotone non-decreasing and submodular, 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, ROUGE.**

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 Elimination. Common 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. Next, we describe three different clustering 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. *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

B. *NetworkX*

NetworkX is a Python library for studying graphs and networks. NetworkX is free software released under the BSD-new license.

Features:

- Classes for graphs and digraphs.
- Conversion of graphs to and from several formats.
- Ability to construct random graphs or construct them incrementally.
- Ability to find subgraphs, cliques, k-cores.
- Explore adjacency, degree, diameter, radius, center, betweenness, etc.
- Draw networks in 2D and 3D.

NetworkX is suitable for operation on large real-world graphs: e.g., graphs in excess of 10 million nodes and 100 million edges. Due to its dependence on a pure-Python "dictionary of dictionary" data structure, NetworkX is a reasonably efficient, very scalable, highly portable framework for network and social network analysis.

III. Datasets

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 (75 bytes) and a short summary (665 bytes) of each cluster.

IV. 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 $A \subseteq B \subseteq V$ 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 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

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

where $\omega_{i,j} \geq 0$ 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 Clustering*

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. *Hierarchical Clustering*

Hierarchical clustering is a general family of clustering algorithms that build nested clusters by merging or splitting them successively. This hierarchy of clusters is represented as a tree (or dendrogram). The root of the tree is the unique cluster that gathers all the samples, the leaves being the clusters with only one sample.
Each observation(here sentence) starts in its own cluster, and clusters are successively merged together. The linkage criteria determines the metric used for the merge strategy. In computing the clusters, we used "Ward" criteria which minimizes the sum of squared distances within all clusters.

C. *Chinese Whispers Graph-Clustering*

The algorithm works in the following way in an undirected unweighted graph:

1) All nodes are assigned to a random class. The number of initial classes equals the number of nodes.
2) Then all of the network nodes are selected one by one in a random order. Every node moves to the class which the given node connects with the most links. In the case of equality the cluster is randomly chosen from the equally linked classes.
3) Step two repeats itself until a predetermined number of iteration or until the process converges. In the end the emerging classes represent the clusters of the network.

The predetermined threshold for the number of the iterations is needed because it is possible, that process does not converge. On the other hand in a network with approximately 10000 nodes the clusters does not change significantly after 40-50 iterations even if there is no convergence.
Here we used similarity between nodes to be 0.05 atleast, only then we connect them with a weighted edge. And we set the number of iterations to 10.

V. Results

The and 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 α and λ values, the clusters were summarized and their ROGUE scores calculated to estimate the efficiency.

Approach	ROUGE-1R	ROUGE-1F
K-means	0.3858	0.3785
Hierarchical	0.3777	0.3712
Chinese WHispers	0.3470	0.3432

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

Acknowledgment

The success of this project couldnt 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] Hui Lin and Jeff Bilmes, A Class of Submodular Functions for Document Summarization.

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

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

[4] DUC 2004, http : //www.nlpir.nist.gov/projects/duc/data/2004 data.html