CS769: Course Project

Optimization in Machine Learning

May 4, 2023

Instructor: Prof Ganesh Ramakrishnan

GROUP MEMBERS
Alok Panigrahi, 200100019
Annie D'souza, 20d070028

Contents

1 Introduction

Document summarization is the act of rewriting a document in a shorter form while still retaining the important content. The task of document summarization is primarily divided into two classes namely extractive and abstractive. Extractive document summarization involves generating summaries of the text by using sentences and phrases only present in the text. Abstractive Text Summarization is the task of generating a short and concise summary that potentially contain new phrases and sentences that may not appear in the source text.

Multi-Document Summarization is a process of representing a set of documents with a single summary by capturing the relevant information and filtering out the redundant information. There are several approaches to multi document summarization including - pretrained encoders, unsupervised multitask learners, CNNs, fine-tuning LLMs, using submodularity, etc.

2 Problem Statement

The task being performed in this project is multi-document summarization. Herein, this task has been performed by modelling it as a monotone submodular function maximization task. The summaries obtained by this method are guaranteed to be better than 63.2% of the optimal summary. The objective is to find a low-cost subset (of sentences) of the document under the constraint that the summary covers all or sufficient amount of information across all documents. ROGUE(reference!!!) is used to evaluate the performance of the summaries obtained.

3 Methodology

- The dataset used for this project is the DUC2004 dataset. (refrence)
- The various functions used in this task are as follows:

3.1 Objective Function

$$F(S) = L(S) + \lambda R(S) \tag{1}$$

Where,
F(S) - Objective function that we aim to maximize
L(S) - Coverage function
R(S) - Diversity reward function
λ - trade-off coefficient

3.2 Coverage Function

L(S) is the coverage function that measures the fidelity of the summary to the documents. It is a set function that measures the similarity of the summary to the documents being summarized.
It is given by the following formula:

$$L(S) = \Sigma_{i \in V} min\{\Sigma_{j \in S} w_{i,j}, \alpha \Sigma_{k \in V} w_{i,k}\} \tag{2}$$

Where,
α - can be determined using grid search
$w_{i,j}$ is defined as follows:

$$w_{i,j} = \frac{\Sigma_{w \in s_i} tf_{w,i} \times tf_{w,j} \times idf_w^2}{\sqrt{\Sigma_{w \in s_i} tf_{w,i}^2 idf_w^2} \sqrt{\Sigma_{w \in s_j} tf_{w,j}^2 idf_w^2}} \tag{3}$$

Where$tf_{w,i}$ and $tf_{w,j}$ are the numbers of times that w appears in s_i and sentence s_j respectively, and idf_w is the inverse document frequency (IDF) of term w (up to bigram), which was calculated as the logarithm of the ratio of the number of articles that w appears over the total number of all articles in the document cluster.

3.3 Diversity reward function

R(S) is the diversity reward function. It is modelled such that there is more benefit to selecting a sentence from a cluster not yet having one of its elements already chosen.
It is given by the following formula:

$$R(S) = \Sigma_{k=1}^{K} \sqrt{\Sigma_{j \in S \cap P_k} \frac{1}{N} \Sigma_{i \in V} w_{i,j}} \tag{4}$$

Where P_k: k = 1,...,K are the clusters/ partitions.

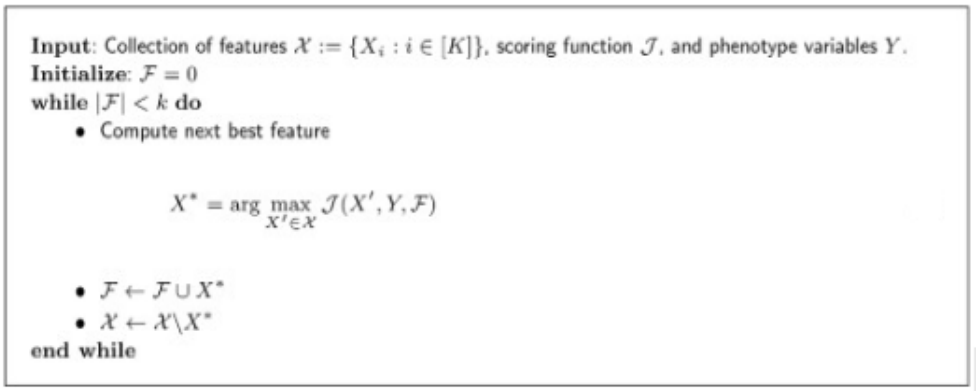

Figure 1: Greedy algorithm for generating summaries

- The partitions were created using sklearn k-means clustering package.
- ROUGE scores are used for checking performance.

4 Code

Link to the github repository of the project:
`https://github.com/Annie2603/CS769`

5 Results

The ROGUE scores obtained on summarizing the documents were as follows:

Lambda	F score	Recall
0.5	38.42	37.9
0.7	33.73	34.05
0.8	49.04	47.06

Figure 2: ROGUE scores obtained for summarizing two documents from the DUC2004 dataset for different values of λ.

Number of partitions - 0.2 $\times$ Number of sentences
$\alpha = 0.8$

6 Challenges faced and Solutions

- **Challenge:** To solve this submodular function we are using a greedy algorithm that has a time complexity of O(n^2) where n is the number of sentences. Thus, running this algorithm on all 10 documents was taking a lot of time and gave runtime errors on colab.
  Solution: Instead of summarizing all documents at once, we summarized documents in pair of two, got their individual ROGUE score and did this iteratively across all documents.
- **Challenge:** The hyperlinks referring to the clustering algorithm mentioned in the CLUTO paper are no longer functioning.
  Solution: We used sklearn k-means clustering package for making the partition function.

- **Challenge:**Reference summary in DUC dataset was not available for all the tasks in the required format.
  Solution: We generated our own custom extractive summary using gensim packages.

7 Analysis and Future Works

- Calculation of the cosine similarity between pairs of sentences is very slow.
- Summaries of individual documents can be generated and multi-document summarization can be applied on these generated summaries to obtain a final summary.
  - Pros: Speedup
  - Cons:Frequency of a sentence across documents may not be captured (which is being done using the idf term).

  In case of a large number of documents, summaries can be generated in batches (randomly chosen) and combined (would capture the inverse document frequency to some extent).
- Instead of using traditional greedy algorithm we can use lazy algorithm for generating the summary.
- The quality of the summaries can be improved using ensemble learning.

8 References

Hui Lin and Jeff Bilmes. 2011. A Class of Submodular Functions for Document Summarization. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pages 510–520, Portland, Oregon, USA. Association for Computational Linguistics.