SubModular Functions & Text Summarizations

CS 691

R&D Project Report

by

Jayaprakash S

Roll No: 123050045

under the guidance of

Prof. Dr. Pushpak Battacharya

Department of Computer Science and Engineering

Indian Institute of Technology, Bombay

Mumbai

Acknowledgement

I would like to thank my guide Prof. Dr. Pushpak Bhattacharyya for his constant encouragement and guidance during the entire course of Project. Further I would like to thank the whole CLIR group for giving me support while doing the project.

Abstract

The aim of this work is to experiment with designing a class of *submodular functions* for extractive document summarization tasks. Monotone submodular functions tends to give near optimal solutions for subset selection (summary). Proper designing of function should measure the representativeness and diversity of summary sentences. We have experimented different coverage and diversity functions and discussed the results.

Contents

Chapter 1

Introduction

...a wealth of information creates a poverty of attention...

Herbert A. Simon

1.1 Problem Statement

In this project, we are trying to design a submodular functions for extractive text summarization tasks.

1.2 Motivation

1.3 Submodular Functions

We are given a set of objects $V = \{v_1, ..., v_n\}$ and a function $F : 2^V \mapsto R$ that returns a real value for any subset $S \subseteq V$.

Subset Function (F)

$F : 2^V \mapsto R$

In text summarization perspective, we are interested in finding subset of bounded size $|S| \leqslant K$ that maximizes the function F. Here the subset is set of sentences chosen as summary for the document.

Maximize the Subset Function

$S = argmax_{S \subseteq V} F(S)$

Subject to $|S| \leqslant K$.

Finding a subset that maximizes this function is hopelessly intractable. The submodular functions have wide applications in various domain including NLP such text summarization and word alignment.

Example

F might correspond to the value or coverage of a set of sensor locations in an environment, and the goal is to find the best locations for a fixed number of sensors.

if the function is monotone submodular, still the objective is NP-Complete. But there exist a greedy algorithm which will give the near optimal solution.

1.3.1 Submodular Functions

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,

1. $F(A+v) - F(A) \geqslant F(B+v) - F(B)$
2. $F(A) + F(B) \geqslant F(A \cup B) + F(A \cap B)$

1.3.2 Monotone Submodular Functions

A set function F is monotone nondecreasing if $\forall A \subseteq B, F(A) \leqslant F(B)$. Monotone nondecreasing submodular functions are referred to as monotone submodular functions.

1.4 Greedy Algorithm

Algorithm 1 : Greedy Algorithm

$G \leftarrow \emptyset$

$U \leftarrow V$

while $U \neq \emptyset$ do

$k \leftarrow argmax_{l \in U} \frac{f(G \cup \{l\}) - f(G)}{(c_l)^r}$

$G \leftarrow G \cup \{k\}$ if $\sum_{i in G} c_i + c_k \leqslant B$ and $f(G \cup \{k\}) - f(G) \geqslant 0$

$U = U \setminus \{k\}$

end while

$v^* \leftarrow argmax_{v in V, c_v \leqslant B} f(v)$

return $G_f = argmax_{S \in \{\{v^*\}, G\}} f(S)$

1.5 Proof of Near Optimal Solution

$f(.) : 2^V \mapsto \mathbb{R}$ is a monotone submodular function. and $P_k(S)$ is the gain of adding k to S, i.e., $f(S \cup \{k\}) - f(S)$.

Lemma 1

$$\forall X, Y \subseteq V, f(X) \leqslant F(Y) + \sum_{k in X \setminus Y} P_k(Y) \tag{1.1}$$

Lemma 2

For i = 1,..., $|G|$, when $0 \leqslant r \leqslant 1$,

$$f(S^*) - f(G_{i-1}) \leqslant \frac{B^r |S^*|^{1-r}}{c^r_{v_i}} (f(G_i) - f(G_{i-1})) \tag{1.2}$$

and when $r \geqslant 1$,

$$f(S^*) - f(G_{i-1}) \leqslant \frac{B^r}{c^r_{v_i}} (f(G_i) - f(G_{i-1})) \tag{1.3}$$

Theorem

Case 1: $\exists v \in V$ such that $f(\{v\}) > \frac{1}{2} f(S^*)$. Then it is guaranteed that $f(G_f) \geqslant f(v) \geqslant \frac{1}{2} f(S^*)$ from Algorithm 1.

Case 2: $\forall v \in V$ such that $f(\{v\}) \leqslant \frac{1}{2} f(S^*)$ then,

Case 2.1: if $\sum_{v in G} c_v \leqslant \frac{1}{2} B$, then $\forall v \notin G, c_v > \frac{1}{2} B$. Submodularity of f(.) gives us:

$f(S^* \setminus G) + f(S * \cap G) \geqslant f(S^*)$,

which implies $f(S * \cap G) \geqslant \frac{1}{2} f(S^*)$. So we have

$f(G_f) \geqslant f(G) \geqslant f(S^* \cap G) \geqslant \frac{1}{2} f(S^*)$,

where the second in-equality is from monotonicity property.

Case 2.2: if $\sum_{v in G} c_v > \frac{1}{2} B$, then

$f(G) \geqslant (1 - \prod_{k=1}^{|G|} 1 - \frac{c^r_{v_k}}{B^r |S*|^{1-r}}) f(S^*)$

$f(G) \geqslant (1 - \prod_{k=1}^{|G|} 1 - \frac{c^r_{v_k} |S*|^{r-1}}{2^r (\sum_{k=1}^{|G|} c_v k^r)^r}) f(S^*)$

$f(G) \geqslant (1 - \prod_{k=1}^{|G|} 1 - \frac{c^r_{v_k} |S*|^{r-1}}{2^r |G|^r}) f(S^*)$

$f(G) \geqslant (1 - e^{-\frac{1}{2} (\frac{|S*|}{2|G|})^{r-1}}) f(S^*)$

In all cases, we have

$f(G_f) \geqslant min\{\frac{1}{2}, 1 - e^{-\frac{1}{2} (\frac{|S*|}{2|G|})^{r-1}}\} f(S^*)$

When r=1, we obtain constant approaximation factor, (i.e),

$$\boxed{f(G_f) \geqslant (1 - e^{\frac{-1}{2}}) f(S^*)}$$

1.6 Contributions

Contibutions towards this R&D project includes,

1. implementation submodular functions for text summarization using similarity score as TFxIDF.
2. experiments with semantic similarity measures instead of TFxIDF.
3. Using different clustering methods K-means and Single link to improve the diversity of summary sentences.

Chapter 2

Experiments

Good summary of text document expected to have good coverage and non-redundancy (novelty).

Objective functions for extracive summarization usually measure these two components separately and combine them together with tradeoff between encouraging the relavency and penalizing for redundancy.

Objective

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

F(S) measures the coverage

R(S) rewards diversity

$\lambda > 0$ is a trade-off coefficent.

2.1 Coverage Functions

Coverage functions, measures the representativeness of summary sentences to the whole document,

The following functions measures coverage of each sentence in a text document by summary sentences.

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

One simple way to define $C_i(S)$ is just to use,

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

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

Coverage Function

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

2.1.1 TFxIDF

Here $w_{i,j}$ is cosine similarity between sentence i and sentecen j.

Documents were pre-processed by segmenting sentences and stemming words using the Porter Stemmer. Each sentence was represented using a bag-of-terms vector, where we used context terms up to bi-grams. Similarity between sentence i and sentence j, was computed using cosine similarity,

Sentence Similarity (TFxIDF)

$$w_{i,j} = \frac{\sum_{w in s_i} tf_{w,i} x tf_{w,j} x idf_w^2}{\sqrt{\sum_{w in s_i} tf_{w,i}^2 idf_i^2}\sqrt{\sum_{w in s_j} tf_{w,j}^2 idf_j^2}}$$

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, 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.

2.1.2 Semantic Measures

2.2 Diversity or Reward Functions

Instead of penalizing the redundancy by subtracting from the objective, rewarding diversity is used here.

Diversity Function

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

P_i is the set of sentences in Cluster i

r_j is the reward for the sentence j

where P_i is a partition of the ground set V into separate clusters. 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. After a sentence is selected from a cluster, other sentences from the same cluster start having diminishing gain, because of the square root function.

2.2.1 Clustering

1. K-means Clustering

2. Single Link Clustering

2.3 Results

2.4 Summary

In this chapter, we described the experiments done in unsupevised and supervised settings. First unsupervised approach is based on *number of occurences and proximity*. Second unsupervised approach is based on *modified textrank which includes co-referencing for constructing a text graph*.

In the next chapter, we conclude this report with future plans.

Chapter 3

Conclusion and Future work

3.0.1 Conclusion

The problem of summarizing a text document is difficult and well studied. In this work, we have discussed and experimented with different coverage and diversity measures for sub-modular functions.

3.0.2 Features

Similarity methods for measuring representatives of summary sentences for a given text document can further utilize,

1. concepts in the sentences
2. co-references
3. other semantic similarities

References

Erkan, G. and Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Intell. Res.(JAIR)*, 22(1):457–479.

Lin, H. and Bilmes, J. (2010). Multi-document summarization via budgeted maximization of submodular functions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 912–920. Association for Computational Linguistics.

Lin, H. and Bilmes, J. (2011). A class of submodular functions for document summarization. In *ACL*, pages 510–520.