Extractive Text Summarizer and Paraphrase Generator

Project Proposal: Natural Language Processing Fall 2021

Background

Through this project, we plan to address two of the challenging tasks in computational linguistics namely text summarization and paraphrasing, together. The task of identifying the most important parts of a long piece of text which preserves the key ideas and overall meaning, and rephrasing these texts for better understanding has been a difficult task not just for computers but also for humans. The main motivation behind this project is to come up with a computationally less intensive methodology using statistical and linguistic techniques to produce the best-rephrased summary of the text for applications with less accessibility to heavy computation.

Objective

The main goals of this project are to create an **extractive text summarizer** that will condense a given text by extracting important sentences that can be represented as a summary for the input text, as well as a **paraphrase generator** that will rewrite the summarized sentences to produce a unique summary of the input text.

Dataset

For this project, we will be using **CNN-Daily Mail dataset**. It consists of 300,000 unique news articles from CNN and the Daily Mail in the English language, along with the summaries for every article. The dataset is divided into 287,226 training pairs, 13,368 validation pairs and 11,490 test pairs. This is a well-known dataset that is used to build both extractive and abstractive text summarization models.

Project Workflow and Methodology

The proposed system primarily consists of two sections:

I. **Text Summarization**

II. **Paraphrase Generation**

After basic pre-processing and cleaning of the input text, the most important sentences from the input text will be extracted by scoring sentences and ranking or classifying sentences that can be used for representing the summary of the input text. This approach of text summarization is called extractive text summarization. In this method, no new sentences are generated, only existing sentences are retrieved from the original text.

First and foremost task in creating an extractive text summarizer is constructing meaningful representation of the input text known as intermediate representation. In this process, sentences are converted to a mathematical representation to make them easier to score and compare against each other. This representation will be carried out using techniques like **Term Frequency - Inverse Document Frequency (TF-IDF)**, **Latent Semantic Analysis (LSA)** and **Latent Dirichlet Allocation (LDA)**.

After creating representations, the relationship between words and topic will be acquired. Each sentence in the input text will then be scored using this relationship. Once every sentence is scored, the highest scored sentences will be selected. These sentences will represent the extracted summary of the paragraph.

This extracted summary will then be paraphrased to create a unique looking summary. The paraphrase generator will rephrase the summarized sentences using computational linguistics and NLP techniques like **POS tagging, Name entity recognition, Collocation extraction,** and finally **replacing synonyms using WordNet**.

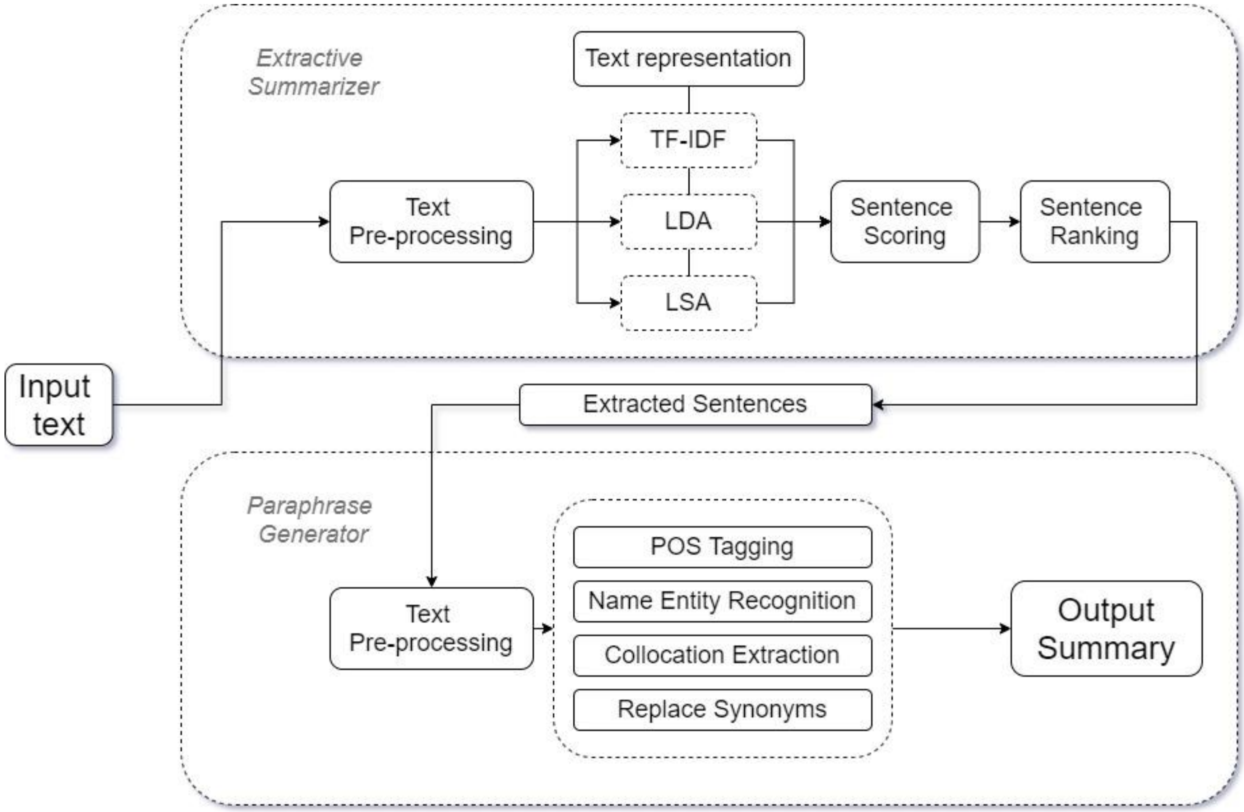

Fig: Project Workflow

Evaluation Metric

We will be using 3 different variants of **ROGUE** (Recall-Oriented Understudy for Gisty Evaluation), namely: **ROGUE-n**, **ROGUE-L** and **ROGUE-SU**, for evaluating summarized sentences where-in we compare n-grams for evaluation process**.** ROGUE-n will be used to evaluate the number of common terms in our summary as compared to the dataset's summary. ROGUE-L will be used to check if any pattern of the same sequence is present in our paraphrased summary as compared with the dataset's summary. We will also be using ROGUE-SU as it is said to give the most correlated measure in accordance with human judgements.

Expected timeline

We have distributed the overall workflow into 3 weeks to work upon:

Week 1: Start building the text summarizer, carrying out gathering data, pre-processing text and constructing text representations using TF-IDF, LSA, LDA methods.

Week 2: Finishing up building text summarizer and further start creating the paraphrase generator.

Week 3: Wrapping up and finalizing the overall project. Write a report for the project.

Team members and Roles

1. **Sanika Katekar:**
    a. Implementing LSA technique for building text summarizer
    b. Working on paraphrase generator collectively with other members
2. **Praveen Kumar Reddy:**
    a. Implementing TF-IDF technique for building text summarizer
    b. Working on paraphrase generator collectively with other members
3. **Soham Sajekar:**
    a. Implementing LDA technique for building text summarizer
    b. Working on paraphrase generator collectively with other members