# Lab 6 - Text Summarization

Instructor
Parth Mehta (parth_mehta@daiict.ac.in)

Teaching Assistants
Adarsh Gupta (202411083@daiict.ac.in),
Bhavesh Baraiya (202101241@daiict.ac.in)

September 2024

## Lab Manual

**Task**: Text Summarization

You are given a dataset consisting of 50 clusters of news articles. Each cluster has around 10 news articles, all of which are appended in a single file. The files are in plain text format. You are also provided human written summaries for the same set of articles.

The task is to implement simple extractive summarization systems including but not limited to the ones mentioned below:

- **Frequency based summarization**: Compute a tf dictionary for a given file. Each word can now be scored based on their tf values. Score of a sentence is then equal to the sum of scores of their words. Be careful to apply all the preprocessing steps!
- **Centroid based summarization**: Compute tf-idf vectors for each sentence in a given document cluster. Centroid of the cluster is then defined as the mean of all sentence vectors. Sentences very similar to centroid tend to have a good overlap with the content of the entire cluster and make good candidates for extractive summaries. But how will you calculate idf if you have only one document?
- **Graph based summarization**: Create a sentence graph from the tf-idf scores computed above. Each sentence is a node and each edge is the similarity between two sentences. Such a sentence graph can be written as a markov transition matrix, which may or may not be acyclic and irreducible. However, a small modification can ensure that it is. Of course, such a acyclic and irreducible matrix is guaranteed to reach a steady state, which can be easily computed using the power method. If we assume equal importance for each sentence initially, then the mean of the final state matrix gives relative importance of each sentence.
- **KLD based greedy summarization**. Normalized TF values within a cluster. These values form the term distribution in the original document. Compute term distributions for each sentence and find the best matching sentence. Remove that sentence from the candidates, find the next sentence such that the distribution of both sentences combined is closest to the document term distribution. Repeat.

**Note:** The use of LLMs is allowed for understanding some of the concepts above. However, directly using code generated by LLMs is not allowed. Any LLM generated code will make you ineligible for the challenge.

General Steps to be followed:

1. Data loading

2. Data Preprocessing
3. Case normalization
4. Word and sentence tokenization (Use nltk)
5. Stemming (use nltk porter stemmer)
6. Stop word removal (use nltk stopwords)
7. Find summaries
    - Find sentences (nltk sent tokenizer)
    - For each sentence find score
    - Use highest ranked sentences as extractive summary. Add one sentence at a time.
    - Optionally exclude redundant sentences even when they have hgh scores.
    - Limit summary size to 100 words. Stop adding sentences when you exceed 100 words in the summary (including stopwords)
8. Evalaute using rouge toolkit. Generate Rouge-1, Rouge-2, Rouge-4 and Rouge-L scores. Steps to use rouge toolkit mentioned below.

# Evaluation

We will use a python implementation of ROUGE toolkit available here: `https://pypi.org/project/rouge-score/`

**Steps to install**:

```
pip install rouge-score
```

**Steps to use**:

```
from rouge_score import rouge_scorer
scorer = rouge_scorer.RougeScorer(['rouge1','rouge2','rouge4', 'rougeL'], use_stemmer=True)
scores = scorer.score(generated_summary, original_summary)
```

# Challenge

- Teams of two compete for best ROUGE-1 score.
- No restriction on technique used. As long as the code is written on your own
- Top-5 Rouge-1 F-scores at the end of the lab get 5 Marks each towards the lab exam after TAs evaluate their codes.

**How to participate:**

- Report to the TAs when you have a new rouge score.
- Top 5 scores will be displayed publicly (like in a leaderboard)
- At the end of the lab submit your working code and generated summaries in google classroom along with the ROUGE scores.
- If code is found to be plagiarised, or fail to produce rouge scores as claimed by the team, there will be a penalty.