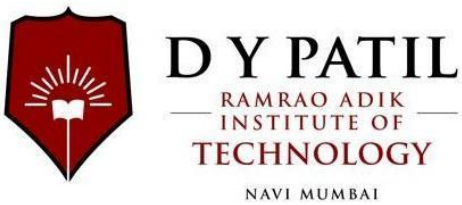


Ramrao Adik Institute of Technology (Affiliated to the University of Mumbai) Dr D. Y. Patil Vidyanagar, Sector 7, Nerul, Navi Mumbai 400706.

A Mini Project Report

On

"Text Summarization"

Submitted by

Shubham Koli	19CE8003
Sujit Bhoir	19CE7023
Nirav Lengade	19CE7042

Signature of Faculty

Contents:

1. Introduction:

We all interact with applications which uses text summarization. Many of those applications are for the platform which publishes articles on daily news, entertainment, sports. With our busy schedule, we prefer to read the summary of those article before we decide to jump in for reading entire article. Reading a summary help us to identify the interest area, gives a brief context of the story. Summarization can be defined as a task of producing a concise and fluent summary while preserving key information and overall meaning.

Summarization systems often have additional evidence they can utilize in order to specify the most important topics of document(s). For example, when summarizing blogs, there are discussions or comments coming after the blog post that are good sources of information to determine which parts of the blog are critical and interesting.

In scientific paper summarization, there is a considerable amount of information such as cited papers and conference information which can be leveraged to identify important sentences in the original paper.

2. Project Description:

Text summarization is the problem of reducing the number of sentences and words of a document without changing its meaning. There are different techniques to extract information from raw text data and use it for a summarization model, overall they can be categorized as Extractive and Abstractive. Extractive methods select the most important sentences within a text (without necessarily understanding the meaning), therefore the result summary is just a subset of the full text. On the contrary, Abstractive models use advanced NLP (i.e. word embeddings) to understand the semantics of the text and generate a meaningful summary. Consequently, Abstractive techniques are much harder to train from scratch as they need a lot of parameters and data.

In this approach we build algorithms or programs which will reduce the text size and create a summary of our text data. This is called automatic text summarization in machine learning.

Text summarization is the process of creating shorter text without removing the semantic structure of text.

Text summarization is the practice of breaking down long publications into manageable paragraphs or sentences. The procedure extracts important information while also ensuring that the paragraph's sense is preserved. This shortens the time it takes to comprehend long materials like research articles while without omitting critical information.

The process of constructing a concise, cohesive, and fluent summary of a lengthier text document, which includes highlighting the text's important points, is known as text summarization.

3. Implementation details:

Abstractive Summarization: Abstractive methods select words based on semantic understanding, even those words did not appear in the source documents. It aims at producing important material in a new way. They interpret and examine the text using advanced natural language techniques in order to generate a new shorter text that conveys the most critical information from the original text.

It can be correlated to the way human reads a text article or blog post and then summarizes in their own word.

Input document → understand context → semantics → create own summary.

2. Extractive Summarization: Extractive methods attempt to summarize articles by selecting a subset of words that retain the most important points.

This approach weights the important part of sentences and uses the same to form the summary. Different algorithm and techniques are used to define weights for the sentences and further rank them based on importance and similarity among each other.

Input document → sentences similarity → weight sentences → select sentences with higher rank.

The limited study is available for abstractive summarization as it requires a deeper understanding of the text as compared to the extractive approach.

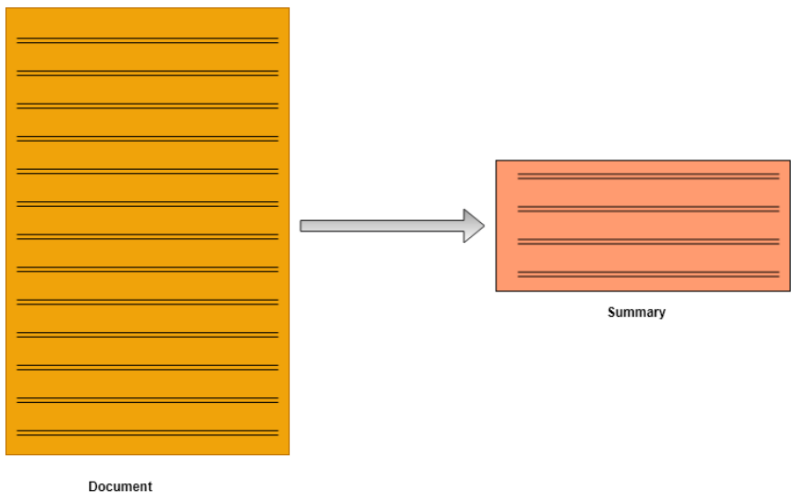
Purely extractive summaries often times give better results compared to automatic abstractive summaries. This is because of the fact that abstractive summarization methods cope with problems such as semantic representation,

inference and natural language generation which is relatively harder than data-driven approaches such as sentence extraction.

There are many techniques available to generate extractive summarization. To keep it simple, we will be using an **unsupervised learning** approach to find the sentences similarity and rank them. One benefit of this will be, you don't need to train and build a model prior start using it for your project.

It's good to understand **Cosine similarity** to make the best use of code you are going to see. **Cosine similarity** is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them. Since we will be representing our sentences as the bunch of vectors, we can use it to find the similarity among sentences. Its measures cosine of the angle between vectors. Angle will be **0** if sentences are similar.

System Design:



Code:

```
import nltk
from nltk.corpus import stopwords
from nltk.cluster.util import cosine_distance
import numpy as np
import networkx as nx
nltk.download('stopwords')

def read_article(file_name):
    file = open(file_name, "r")
    filedata = file.readlines()
    article = filedata[0].split(". ")
    sentences = []

    for sentence in article:
        print(sentence)
        sentences.append(sentence.replace("[^a-zA-Z]", " ").split(" "))
    sentences.pop()

    return sentences

def sentence_similarity(sent1, sent2, stopwords=None):
    if stopwords is None:
        stopwords = []

    sent1 = [w.lower() for w in sent1]
    sent2 = [w.lower() for w in sent2]

    all_words = list(set(sent1 + sent2))

    vector1 = [0] * len(all_words)
    vector2 = [0] * len(all_words)

    # build the vector for the first sentence
    for w in sent1:
        if w in stopwords:
```

```
            continue
        vector1[all_words.index(w)] += 1

    # build the vector for the second sentence
    for w in sent2:
        if w in stopwords:
            continue
        vector2[all_words.index(w)] += 1

    return 1 - cosine_distance(vector1, vector2)

def build_similarity_matrix(sentences, stop_words):
    # Create an empty similarity matrix
    similarity_matrix = np.zeros((len(sentences), len(sentences)))

    for idx1 in range(len(sentences)):
        for idx2 in range(len(sentences)):
            if idx1 == idx2: #ignore if both are same sentences
                continue
            similarity_matrix[idx1][idx2] = sentence_similarity(sentences[idx1], sentences[idx2], stop_words)

    return similarity_matrix


def generate_summary(file_name, top_n=5):
    stop_words = stopwords.words('english')
    summarize_text = []

    # Step 1 - Read text anc split it
    sentences =  read_article(file_name)

    # Step 2 - Generate Similary Martix across sentences
    sentence_similarity_martix = build_similarity_matrix(sentences, stop_words)

    # Step 3 - Rank sentences in similarity martix
    sentence_similarity_graph = nx.from_numpy_array(sentence_similarity_martix)
    scores = nx.pagerank(sentence_similarity_graph)

    # Step 4 - Sort the rank and pick top sentences
```

```
    ranked_sentence = sorted(((scores[i],s) for i,s in enumerate(sentences)), reverse=True)
    print("Indexes of top ranked_sentence order are ", ranked_sentence)

    for i in range(top_n):
      summarize_text.append(" ".join(ranked_sentence[i][1]))

    # Step 5 - Offcourse, output the summarize texr
    print("Summarize Text: \n", ". ".join(summarize_text))

# let's begin
generate_summary( "msft.txt", 2)
```

4. Result and Analysis:

Input:

In an attempt to build an AI-ready workforce, Microsoft announced Intelligent Cloud Hub which has been launched to empower the next generation of students with AI-ready skills
Envisioned as a three-year collaborative program, Intelligent Cloud Hub will support around 100 institutions with AI infrastructure, course content and curriculum, developer support, development tools and give students access to cloud and AI services
As part of the program, the Redmond giant which wants to expand its reach and is planning to build a strong developer ecosystem in India with the program will set up the core AI infrastructure and IoT Hub for the selected campuses
The company will provide AI development tools and Azure AI services such as Microsoft Cognitive Services, Bot Services and Azure Machine Learning.According to Manish Prakash, Country General Manager-PS, Health and Education, Microsoft India, said, "With AI being the defining technology of our time, it is transforming lives and industry and the jobs of tomorrow will require a different skillset
This will require more collaborations and training and working with AI
Thatâ€™s why it has become more critical than ever for educational institutions to integrate new cloud and AI technologies
The program is an attempt to ramp up the institutional set-up and build capabilities among the educators to educate the workforce of tomorrow." The program aims to build up the cognitive skills and in-depth understanding of developing intelligent cloud connected solutions for applications across industry
Earlier in April this year, the company announced Microsoft Professional Program In AI as a learning track open to the public
The program was developed to provide job ready skills to programmers who wanted to hone their skills in AI and data science with a series of online courses which featured hands-on labs and expert instructors as well
This program also included developer-focused AI school that provided a bunch of assets to help build AI skills.

Process:

Indexes of top ranked_sentence order are [(0.15083257041122708, ['Envisioned', 'as', 'a', 'three-year', 'collaborative', 'program,', 'Intelligent', 'Cloud', 'Hub', 'will', 'support', 'around', '100', 'institutions', 'with', 'AI', 'infrastructure,', 'course', 'content', 'and', 'curriculum,', 'developer', 'support,', 'development', 'tools', 'and', 'give', 'students', 'access', 'to', 'cloud', 'and', 'AI', 'services']), (0.13161201335715553, ['The', 'company', 'will', 'provide', 'AI', 'development', 'tools', 'and', 'Azure', 'AI', 'services', 'such', 'as', 'Microsoft', 'Cognitive', 'Services,', 'Bot', 'Services', 'and', 'Azure', 'Machine', 'Learning.According', 'to', 'Manish', 'Prakash,', 'Country', 'General', 'Manager-PS,', 'Health', 'and', 'Education,', 'Microsoft', 'India,', 'said,', '"With', 'AI', 'being', 'the', 'defining', 'technology', 'of', 'our', 'time,', 'it', 'is', 'transforming', 'lives', 'and', 'industry', 'and', 'the', 'jobs', 'of', 'tomorrow', 'will', 'require', 'a', 'different', 'skillset']), (0.11403047674961148, ['Earlier', 'in', 'April', 'this', 'year,', 'the', 'company', 'announced', 'Microsoft', 'Professional', 'Program', 'In', 'AI', 'as', 'a', 'learning', 'track', 'open', 'to', 'the', 'public']), (0.10721749759953525, ['In', 'an', 'attempt', 'to', 'build', 'an', 'AI-ready', 'workforce,', 'Microsoft', 'announced', 'Intelligent', 'Cloud', 'Hub', 'which', 'has', 'been', 'launched', 'to', 'empower', 'the', 'next', 'generation', 'of', 'students', 'with', 'AI-ready', 'skills']), (0.10404298514456578, ['As', 'part', 'of', 'the', 'program,', 'the', 'Redmond', 'giant', 'which', 'wants', 'to', 'expand', 'its', 'reach', 'and', 'is', 'planning', 'to', 'build', 'a', 'strong', 'developer', 'ecosystem', 'in', 'India', 'with', 'the', 'program', 'will', 'set', 'up', 'the', 'core', 'AI', 'infrastructure', 'and', 'IoT', 'Hub', 'for', 'the', 'selected', 'campuses']), (0.10031366655994461, ['Thatâ€™s', 'why', 'it', 'has', 'become', 'more', 'critical', 'than', 'ever', 'for', 'educational', 'institutions', 'to', 'integrate', 'new', 'cloud', 'and', 'AI', 'technologies']), (0.10001137283486655, ['The', 'program', 'is', 'an', 'attempt', 'to', 'ramp', 'up', 'the', 'institutional', 'set-up', 'and', 'build', 'capabilities', 'among', 'the', 'educators', 'to', 'educate', 'the', 'workforce', 'of', 'tomorrow."', 'The', 'program', 'aims', 'to', 'build', 'up', 'the', 'cognitive', 'skills', 'and', 'in-depth', 'understanding', 'of', 'developing', 'intelligent', 'cloud', 'connected', 'solutions', 'for', 'applications', 'across', 'industry']), (0.09916750119894319, ['This', 'will', 'require', 'more', 'collaborations', 'and', 'training', 'and', 'working', 'with', 'AI']), (0.09277191614415067, ['The', 'program', 'was', 'developed', 'to', 'provide', 'job', 'ready', 'skills', 'to', 'programmers', 'who', 'wanted', 'to', 'hone', 'their', 'skills', 'in', 'AI', 'and', 'data', 'science', 'with', 'a', 'series', 'of', 'online', 'courses', 'which', 'featured', 'hands-on', 'labs', 'and', 'expert', 'instructors', 'as', 'well'])]

Output:

Summarize Text:

Envisioned as a three-year collaborative program, Intelligent Cloud Hub will support around 100 institutions with AI infrastru cture, course content and curriculum, developer support, development tools and give students access to cloud and AI services. T he company will provide AI development tools and Azure AI services such as Microsoft Cognitive Services, Bot Services and Azure Machine Learning.According to Manish Prakash, Country General Manager-PS, Health and Education, Microsoft India, said, "With AI being the defining technology of our time, it is transforming lives and industry and the jobs of tomorrow will require a differ ent skillset

6. Conclusion:

Automatic text summarization is an old challenge but the current research direction diverts towards emerging trends in biomedicine, product review, education domains, emails and blogs. This is due to the fact that there is information overload in these areas, especially on the World Wide Web.Automated summarization is an important area in NLP (Natural Language Processing) research. It consists of automatically creating a summary of one or more texts. The purpose of extractive document summarization is to automatically select a number of indicative sentences, passages, or paragraphs from the original document .Text summarization approaches based on Neural Network, Graph Theoretic, Fuzzy and Cluster have, to an extent, succeeded in making an effective summary of a document.Both extractive and abstractive methods have been researched. Most summarization techniques are based on extractive methods. Abstractive method is similar to summaries made by humans. Abstractive summarization as of now requires heavy machinery for language generation and is difficult to replicate into the domain specific areas