Information Retrieval

Summerify - Document Summarizer

Jui Ambikar
Sayali Deodikar

INTRODUCTION

- Document summarization is a practice of breaking down long documents 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.

TYPES OF SUMMARIZATION

EXTRACTIVE

The extractive text summarizing approach entails extracting essential words from a source material and combining them to create a summary.

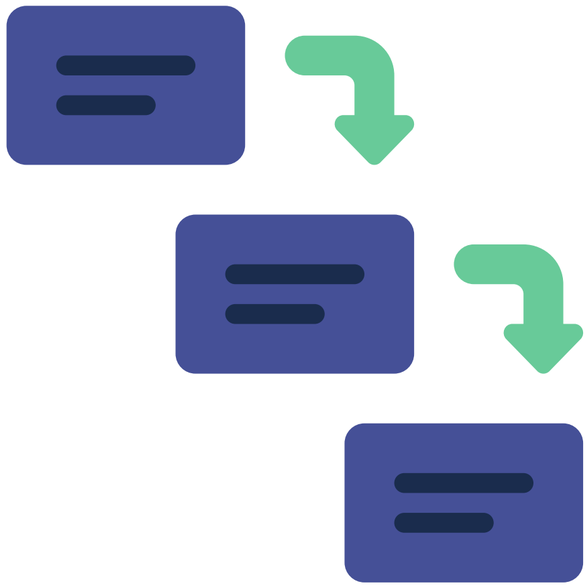
ABSTRACTIVE

An abstractive approach is more advanced. We create new sentences from the original content in this step.

OUR APPROACH

- Extractive summerization
- Can summarize wikipedia pages, web articles and pdf documents
- Based on Natural language processing and python
- Generates summary from URL of document

METHODOLOGY



Step 1 **Import necessary python libraries**

NLTK, requests, PyPDF2, Beautiful soup

Step 2 **For PDF - Extract text from pdf**

Created get_pdf_content function that returns text

Step 3 **For Articles - Web Scrapping using Beautiful Soup**

Created get_wiki_content function

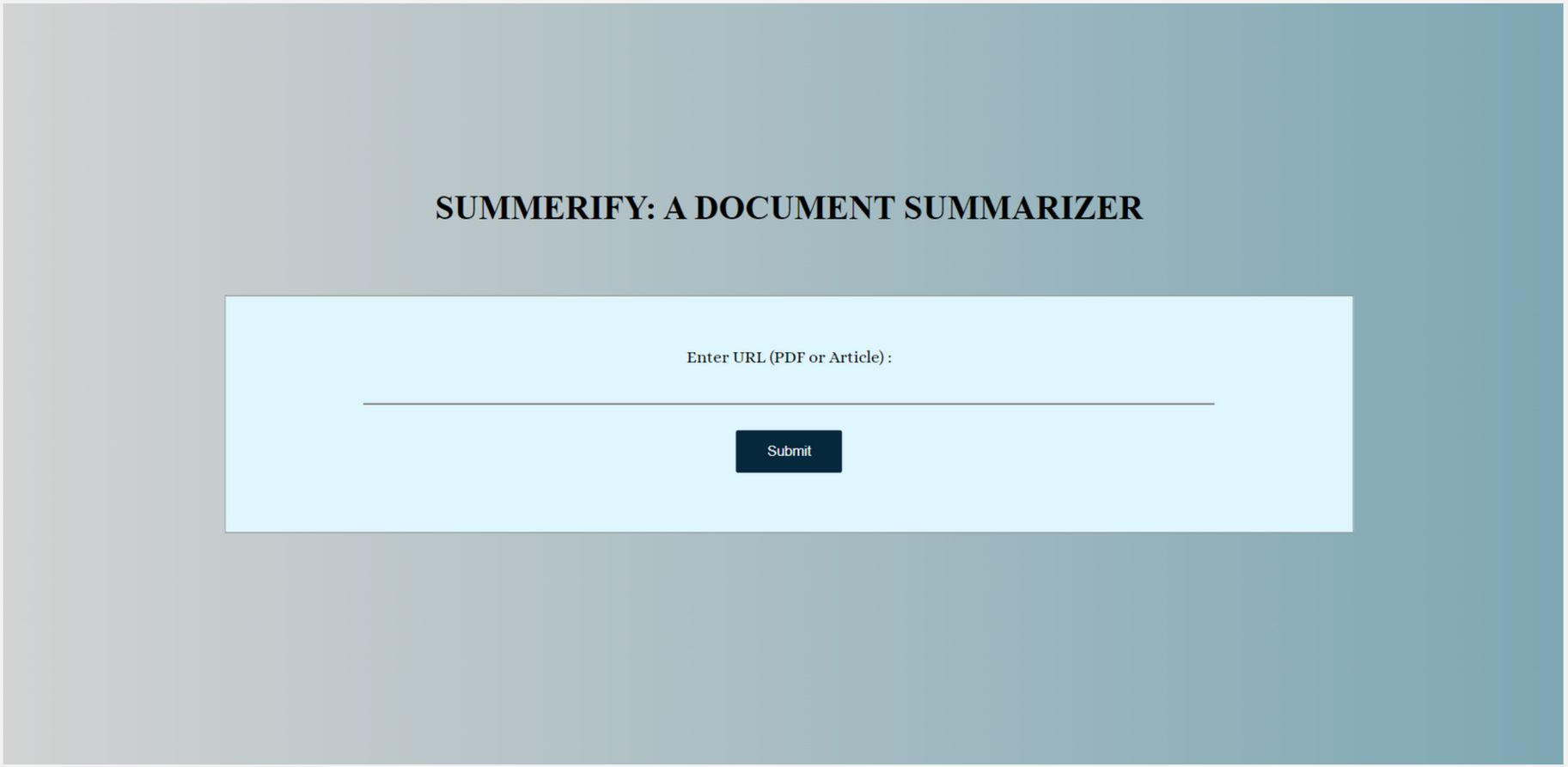
Step 4 **Preprocessing of text**

Tokenization, Stop-word removal

Step 5 **TF-IDF to get top 10 sentences**

Return top 10 sentences as summary

SCREENSHOTS



SUMMARY

ORIGINAL DOCUMENT TEXT

Level Up CodingApr 5, 2020SaveI'm writing my dissertation's literature review and I got to a point where I lost track of how many citations I have included so far.Since I'm still writing the first draft, I've been entering citations manually, in the form ("reference_title", p. page_number) or simply ("reference_title"). For example, ("Learning Analytics — Definitions, Processes and Potential", p. 2) or ("Wanted: A road map for understanding Integrated Learning Systems"), respectively.This is good enough if I want to look up where I cited specific documents, but to know how many citations are in the text or, even worse, which citations have been used, I have to agree this format I came up with is a mess.Thus, I came up with a short Python script to solve my problem. It loads the Word document, finds all instances of this "citation format" using Regular Expressions (Regex) and returns the citations found. The script uses the docx2txt package to load the Word file (a file with a .docx extension) into a single Python string.To show how the script works, we'll use the good old Lorem Ipsum text with some citations mixed in (the citations included at the beginning of the following code gist):The first part (until line 17) is pretty straightforward: import the docx2txt and Python's standard regular expressions libraries, load the text from the Word file as a single string (line 10) and replace any stylized quotes that may be in the text by regular double quotes (lines 16 and 17).The second part (lines 19–23) is where I defined the regex pattern to look up the citations (line 21) and actually try to find them in the text (line 23).In the above picture I divided the pattern in three parts by color-coding them:In other words, the above regex pattern matches

SUMMARY

The script uses the docx2txt package to load the Word file (a file with a .docx extension) into a single Python string.To show how the script works, we'll use the good old Lorem Ipsum text with some citations mixed in (the citations included at the beginning of the following code gist):The first part (until line 17) is pretty straightforward: import the docx2txt and Python's standard regular expressions libraries, load the text from the Word file as a single string (line 10) and replace any stylized quotes that may be in the text by regular double quotes (lines 16 and 17).The second part (lines 19–23) is where I defined the regex pattern to look up the citations (line 21) and actually try to find them in the text (line 23).In the above picture I divided the pattern in three parts by color-coding them:In other words, the above regex pattern matches any document title enclosed by quotes that is prefixed with an opening parenthesis. For example, ("Learning Analytics — Definitions, Processes and Potential", p. 2) or ("Wanted: A road map for understanding Integrated Learning Systems"), respectively.This is good enough if I want to look up where I cited specific documents, but to know how many citations are in the text or, even worse, which citations have been used, I have to agree this format I came up with is a mess.Thus, I came up with a short Python script to solve my problem. Level Up CodingApr 5, 2020SaveI'm writing my dissertation's literature review and I got to a point where I lost track of how many citations I have included so far.Since I'm still writing the first draft, I've been entering citations manually, in the form ("reference_title", p. page_number) or simply ("reference_title"). For instance, for the citation ("Learning Analytics — Definitions, Processes and

Thank you!