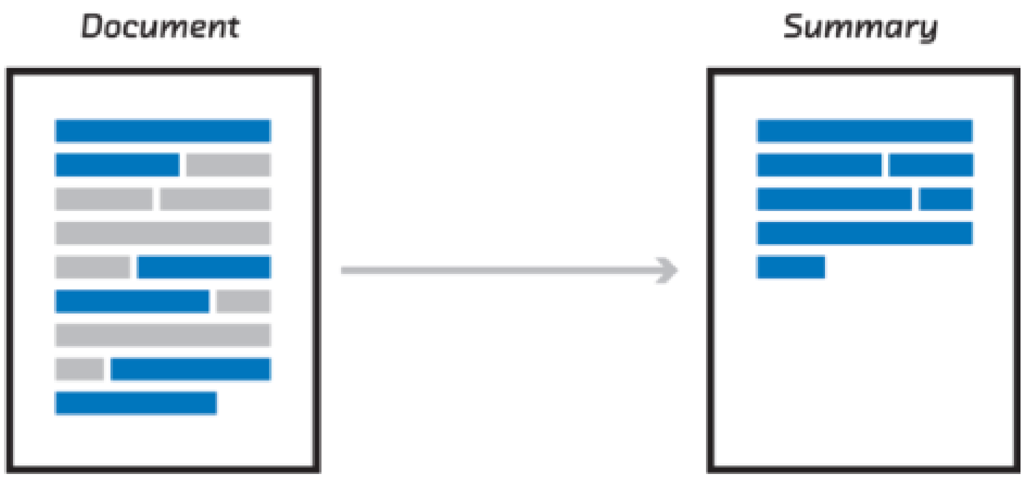
DEEP LEARNING PROJECT

Text Summarization Application

1) Amr Ahmed Abdelrehem
2) Esraa Alaa El-din El-far
3) Noran Ali El-Saied
4) Basmala Gamal Yehya



CONTENTS

I. Introduction

Text Summarization and Natural Language Processing (NLP) Application

This application is designed to provide a comprehensive set of natural language processing (NLP) tools, aimed at simplifying complex text processing tasks. Built with **Streamlit**, the application offers a user-friendly interface for various text manipulation and analysis functions. From summarizing lengthy documents to correcting grammatical errors, this tool integrates advanced NLP techniques in an accessible way for both novice and experienced users.

The core functionality revolves around **text summarization**, where users can input or upload text, select from multiple summarization algorithms (LSA, LUHN, LexRank, and TextRank), and generate concise summaries. Beyond summarization, the application offers additional features to enrich the text analysis experience:

1. **Text Uploading**: Users can upload `.txt` files, allowing them to quickly summarize or analyze text from external documents.
2. **Auto-Correction**: Using `Text Blob`, the application corrects grammatical errors and spelling mistakes in the input text, ensuring clarity and correctness.
3. **Named Entity Recognition (NER)**: Leveraging `spacy`, the application identifies and extracts key entities (such as people, organizations, and locations) from the text, providing users with enhanced insights into the content.
4. **Language Detection**: The app can automatically detect the language of the input text and adjust the summarization process, accordingly, supporting dynamic language handling.

With these features, this tool is ideal for anyone who needs to summarize large blocks of text, correct language errors, or extract important entities from documents. Whether you're working with research papers, business reports, or web articles, this application streamlines the entire text analysis process.

2. Features

1. Text Summarization (ALL)

At the core of the application is the **text summarization** feature, which allows users to condense large blocks of text into concise summaries. This is particularly useful for processing lengthy articles, reports, or documents quickly without losing the main points. The application provides multiple summarization algorithms, each with its own approach to selecting important sentences:

- **LSA (Latent Semantic Analysis)**: This method uses statistical techniques to identify key themes and compress text based on semantic structures.
- **LUHN**: Based on word frequency, LUHN summarizes the text by identifying the most relevant sentences with the most significant words.
- **LexRank**: A graph-based algorithm, LexRank selects sentences for the summary based on how central they are to the overall text.
- **TextRank**: Like LexRank, this algorithm identifies key sentences using a ranking system based on importance.

Users can control the length of the summary by adjusting the **number of sentences** they want to include, making the feature highly flexible for different needs.

2. Text Uploading (Basmala’s Part)

In addition to manually entering text, the application allows users to upload `.txt` files. This feature is particularly helpful for those who need to work with existing documents, such as business reports, academic papers, or articles. Once uploaded, the content is automatically displayed, and the user can apply the available NLP features (summarization, correction, NER) directly to the file's content.

This eliminates the need to copy and paste text, streamlining the workflow for users dealing with large volumes of text.

3. Auto-Correction (Noran’s Part)

The **Auto-Correction** feature helps users improve the quality of their text by correcting grammatical errors and spelling mistakes. Powered by `TextBlob`, this function reads the input text and automatically provides corrected suggestions. It's particularly useful when working with rough drafts, content written quickly, or text that might contain typos or other errors.

Users can simply click the "Auto-Correct Text" button to see the corrected version of their text, ensuring clarity and improving the overall quality of the content.

4. Named Entity Recognition (Esraa's Part)

The **Named Entity Recognition (NER)** feature, powered by `spacy`, identifies key entities within the text, such as people, organizations, locations, dates, and more. This feature is crucial for extracting important information from text without having to read through the entire content.

For instance, when working with news articles, research papers, or legal documents, users can instantly pull-out names of relevant people, places, or organizations. The extracted entities are displayed with their corresponding categories, providing a clear overview of important details present in the text.

5. Language Detection (Amr's Part)

The **Language Detection** feature automatically identifies the language of the input text. This is particularly useful in multilingual environments where users may be working with documents written in different languages. Using the `langdetect` library, the app will analyze the input text and detect the language, which helps in setting the appropriate tokenizer and improving the accuracy of the summarization process.

If the language detection fails or cannot be determined, the app falls back to English, ensuring smooth operation even when language recognition encounters challenges.

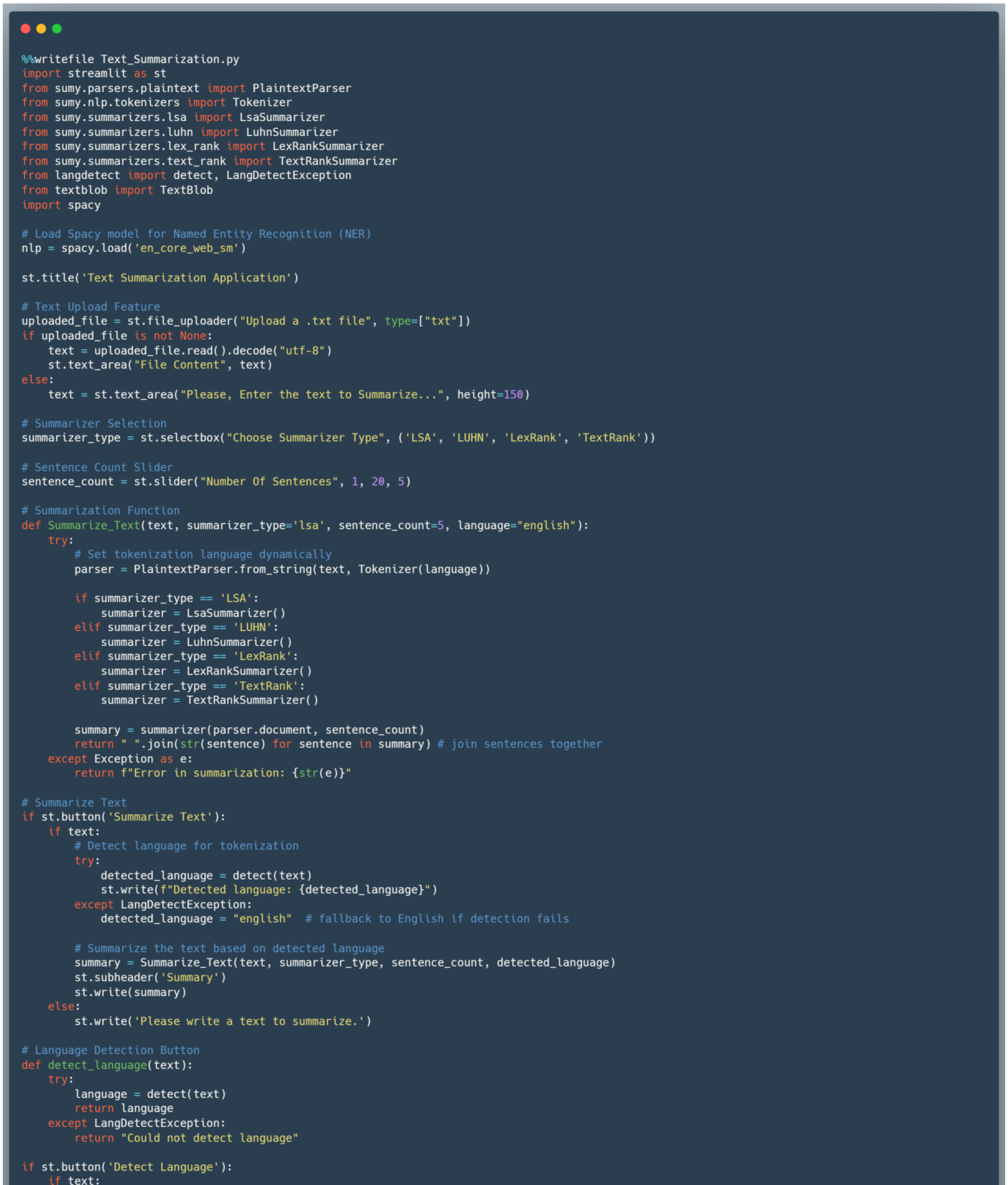
3. Code

```
!pip install pyarabic

!pip install langdetect

!pip install streamlit sumy langdetect textblob spacy

!python -m spacy download en_core_web_sm

!pip install textblob

!pip install streamlit textblob sumy PyPDF2 python-docx
```

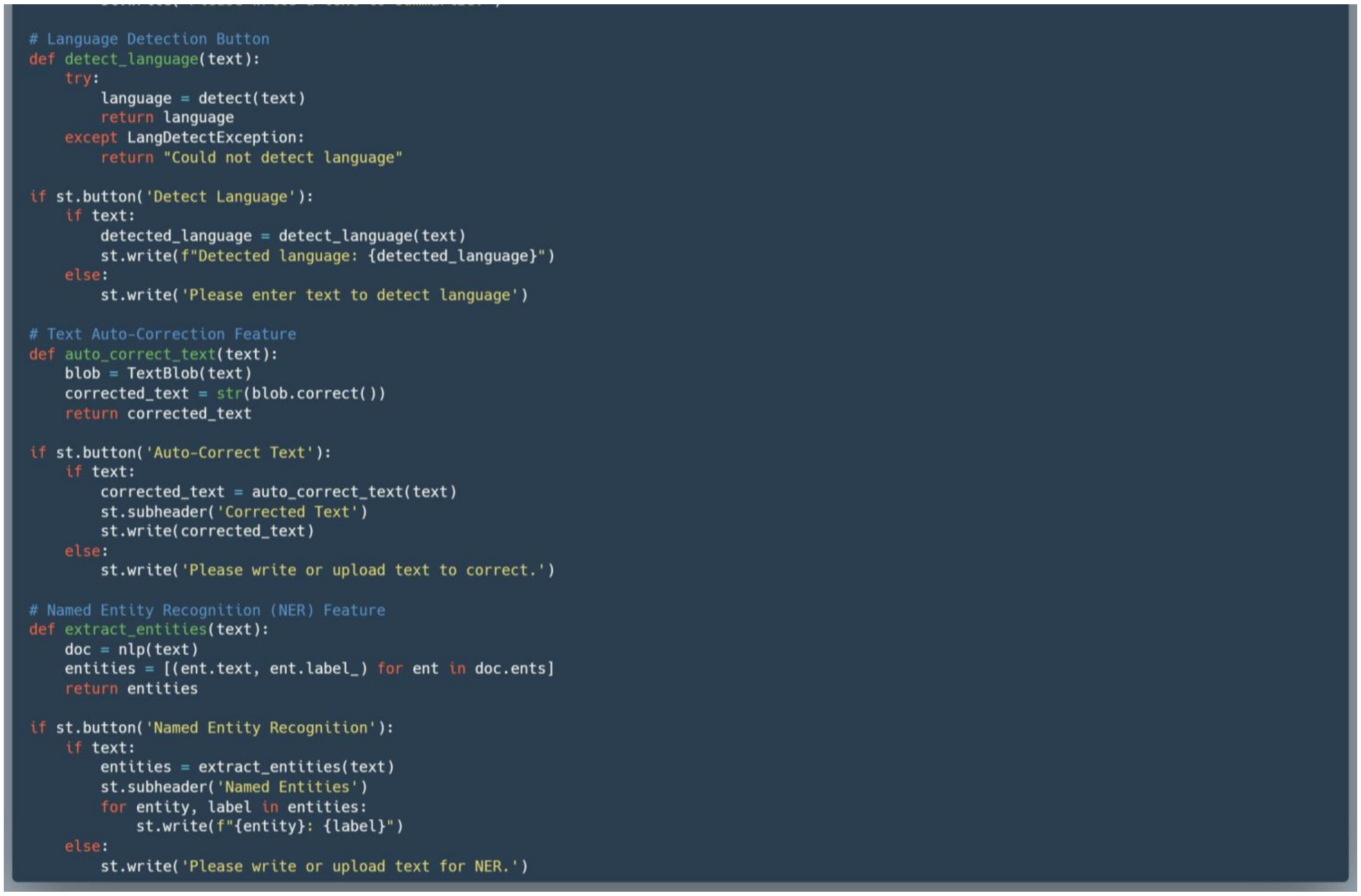
```
from langdetect import detect

from textblob import TextBlob

%%writefile Text_Summarization.py

import streamlit as st

from sumy.parsers.plaintext import PlaintextParser

from sumy.nlp.tokenizers import Tokenizer

from sumy.summarizers.lsa import LsaSummarizer

from sumy.summarizers.luhn import LuhnSummarizer

from sumy.summarizers.lex_rank import LexRankSummarizer

from sumy.summarizers.text_rank import TextRankSummarizer

from langdetect import detect, LangDetectException

from textblob import TextBlob

import spacy


# Load Spacy model for Named Entity Recognition (NER)

nlp = spacy.load('en_core_web_sm')
```

```
st.title('Text Summarization Application')


# Text Upload Feature

uploaded_file = st.file_uploader("Upload a .txt file", type=["txt"])

if uploaded_file is not None:

    text = uploaded_file.read().decode("utf-8")

    st.text_area("File Content", text)

else:

    text = st.text_area("Please, Enter the text to Summarize...", height=150)


# Summarizer Selection

summarizer_type = st.selectbox("Choose Summarizer Type", ('LSA', 'LUHN',
'LexRank', 'TextRank'))


# Sentence Count Slider

sentence_count = st.slider("Number Of Sentences", 1, 20, 5)


# Summarization Function

def Summarize_Text(text, summarizer_type='lsa', sentence_count=5,
language="english"):

    try:

        # Set tokenization language dynamically

        parser = PlaintextParser.from_string(text, Tokenizer(language))
        if summarizer_type == 'LSA':
            summarizer = LsaSummarizer()
        elif summarizer_type == 'LUHN':
            summarizer = LuhnSummarizer()
        elif summarizer_type == 'LexRank':
            summarizer = LexRankSummarizer()
        elif summarizer_type == 'TextRank':
            summarizer = TextRankSummarizer()

        summary = summarizer(parser.document, sentence_count)
        return " ".join(str(sentence) for sentence in summary) # join sentences together
    except Exception as e:
        return f"Error in summarization: {str(e)}"

# Summarize Text
if st.button('Summarize Text'):
    if text:
        # Detect language for tokenization
        try:
            detected_language = detect(text)
            st.write(f"Detected language: {detected_language}")
        except LangDetectException:
            detected_language = "english"  # fallback to English if detection fails

        # Summarize the text based on detected language
        summary = Summarize_Text(text, summarizer_type, sentence_count, 
detected_language)
        st.subheader('Summary')
        st.write(summary)
    else:
        st.write('Please write a text to summarize.')

# Language Detection Button
def detect_language(text):
    try:
        language = detect(text)
        return language
    except LangDetectException:
        return "Could not detect language"

if st.button('Detect Language'):
    if text:
        detected_language = detect_language(text)
        st.write(f"Detected language: {detected_language}")
    else:
        st.write('Please enter text to detect language')

# Text Auto-Correction Feature
def auto_correct_text(text):
    blob = TextBlob(text)
    corrected_text = str(blob.correct())
    return corrected_text

if st.button('Auto-Correct Text'):
    if text:
        corrected_text = auto_correct_text(text)
        st.subheader('Corrected Text')
        st.write(corrected_text)
    else:
        st.write('Please write or upload text to correct.')

# Named Entity Recognition (NER) Feature
def extract_entities(text):
    doc = nlp(text)
    entities = [(ent.text, ent.label_) for ent in doc.ents]
    return entities

if st.button('Named Entity Recognition'):
```

```
if text:
    entities = extract_entities(text)
    st.subheader('Named Entities')
    for entity, label in entities:
        st.write(f"{entity}: {label}")
else:
    st.write('Please write or upload text for NER.')
```

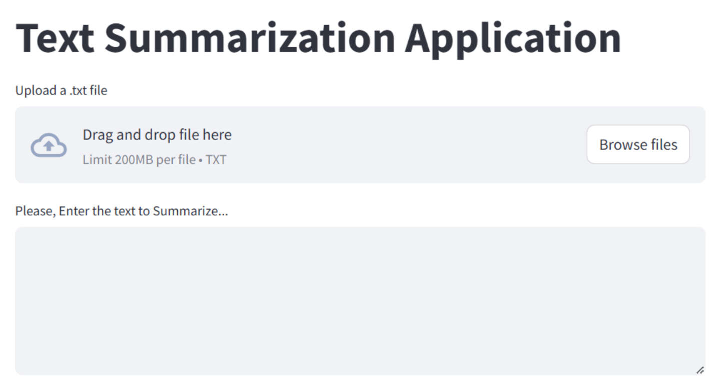


```
%%writefile Text_Summarization.py
import streamlit as st
from sumy.parsers.plaintext import PlaintextParser
from sumy.nlp.tokenizers import Tokenizer
from sumy.summarizers.lsa import LsaSummarizer
from sumy.summarizers.luhn import LuhnSummarizer
from sumy.summarizers.lex_rank import LexRankSummarizer
from sumy.summarizers.text_rank import TextRankSummarizer
from langdetect import detect, LangDetectException
from textblob import TextBlob
import spacy

# Load Spacy model for Named Entity Recognition (NER)
nlp = spacy.load('en_core_web_sm')

st.title('Text Summarization Application')

# Text Upload Feature
uploaded_file = st.file_uploader("Upload a .txt file", type=["txt"])
if uploaded_file is not None:
    text = uploaded_file.read().decode("utf-8")
    st.text_area("File Content", text)
else:
    text = st.text_area("Please, Enter the text to Summarize...", height=150)

# Summarizer Selection
summarizer_type = st.selectbox("Choose Summarizer Type", ('LSA', 'LUHN', 'LexRank', 'TextRank'))

# Sentence Count Slider
sentence_count = st.slider("Number Of Sentences", 1, 20, 5)

# Summarization Function
def Summarize_Text(text, summarizer_type='lsa', sentence_count=5, language="english"):
    try:
        # Set tokenization language dynamically
        parser = PlaintextParser.from_string(text, Tokenizer(language))

        if summarizer_type == 'LSA':
            summarizer = LsaSummarizer()
        elif summarizer_type == 'LUHN':
            summarizer = LuhnSummarizer()
        elif summarizer_type == 'LexRank':
            summarizer = LexRankSummarizer()
        elif summarizer_type == 'TextRank':
            summarizer = TextRankSummarizer()

        summary = summarizer(parser.document, sentence_count)
        return " ".join(str(sentence) for sentence in summary) # join sentences together
    except Exception as e:
        return f"Error in summarization: {str(e)}"

# Summarize Text
if st.button('Summarize Text'):
    if text:
        # Detect language for tokenization
        try:
            detected_language = detect(text)
            st.write(f"Detected language: {detected_language}")
        except LangDetectException:
            detected_language = "english"  # fallback to English if detection fails

        # Summarize the text based on detected language
        summary = Summarize_Text(text, summarizer_type, sentence_count, detected_language)
        st.subheader('Summary')
        st.write(summary)
    else:
        st.write('Please write a text to summarize.')

# Language Detection Button
def detect_language(text):
    try:
        language = detect(text)
        return language
    except LangDetectException:
        return "Could not detect language"

if st.button('Detect Language'):
    if text:
```



```
# Language Detection Button
def detect_language(text):
    try:
        language = detect(text)
        return language
    except LangDetectException:
        return "Could not detect language"

if st.button('Detect Language'):
    if text:
        detected_language = detect_language(text)
        st.write(f"Detected language: {detected_language}")
    else:
        st.write('Please enter text to detect language')

# Text Auto-Correction Feature
def auto_correct_text(text):
    blob = TextBlob(text)
    corrected_text = str(blob.correct())
    return corrected_text

if st.button('Auto-Correct Text'):
    if text:
        corrected_text = auto_correct_text(text)
        st.subheader('Corrected Text')
        st.write(corrected_text)
    else:
        st.write('Please write or upload text to correct.')

# Named Entity Recognition (NER) Feature
def extract_entities(text):
    doc = nlp(text)
    entities = [(ent.text, ent.label_) for ent in doc.ents]
    return entities

if st.button('Named Entity Recognition'):
    if text:
        entities = extract_entities(text)
        st.subheader('Named Entities')
        for entity, label in entities:
            st.write(f"{entity}: {label}")
    else:
        st.write('Please write or upload text for NER.')
```

4. Streamlit Test



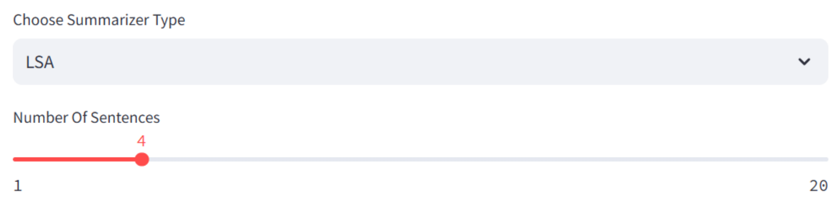
Text Summarization Application

Upload a .txt file

Drag and drop file here
Limit 200MB per file • TXT

Please, Enter the text to Summarize...

Figure1. Upload Text and Text Bar



Choose Summarizer Type: LSA

Number Of Sentences: 4 (1–20)

Figure2. Summarizer Type and Number of Sentences

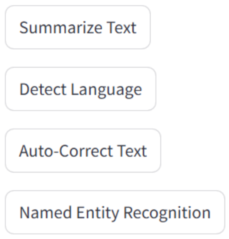
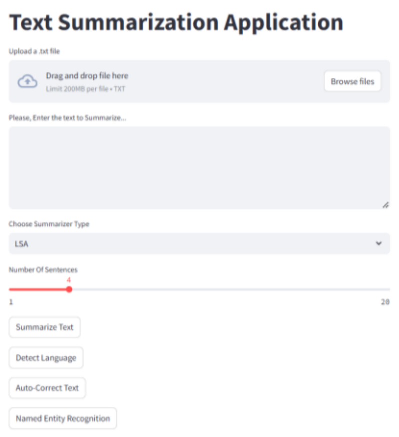


Figure3. The Four Features


Text Summarization Application

Upload a .txt file

Drag and drop file here

Limit 200MB per file • TXT

Please, Enter the text to Summarize...

Choose Summarizer Type

LSA

Number Of Sentences

4

1

20

Figure4. The Application on Streamlit

- Testing Video:

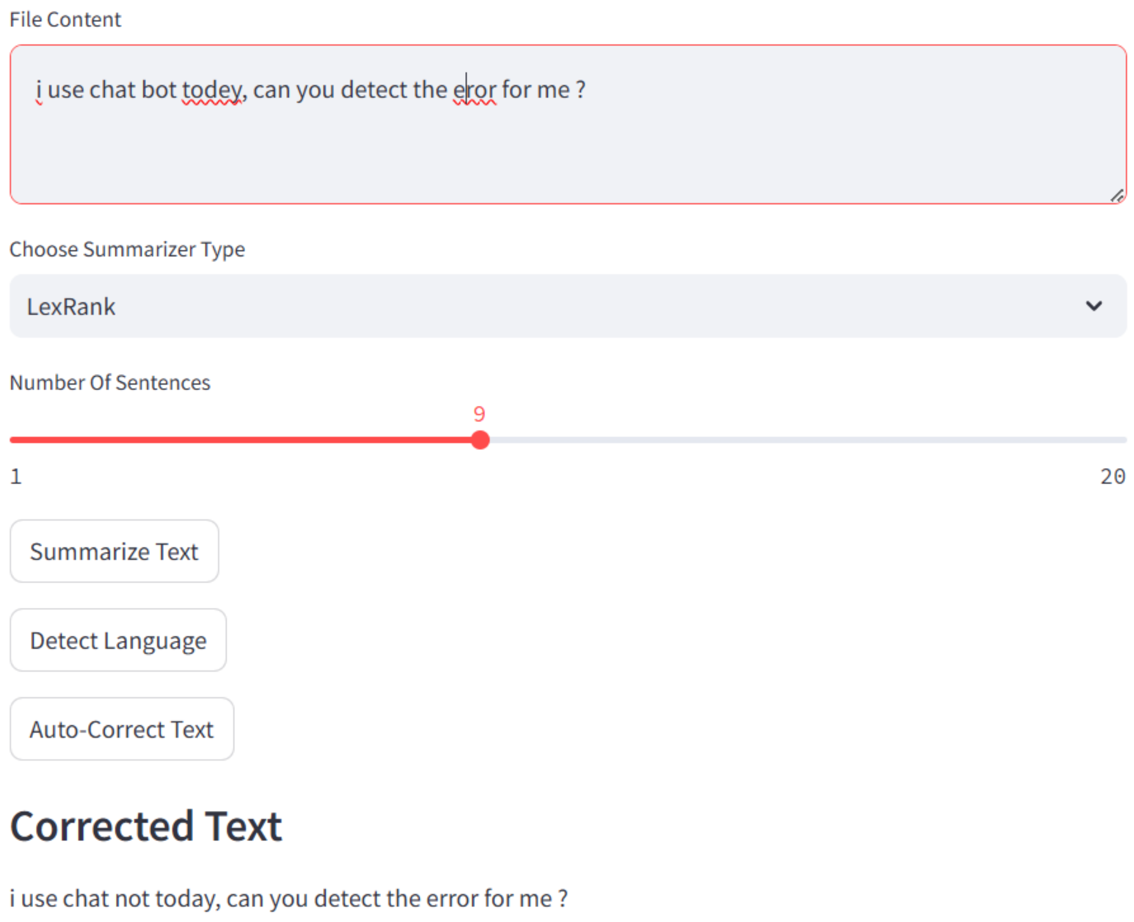
- ***English Text***:

https://drive.google.com/file/d/1K1VqaR-1ngw9JEef8tvPfmPsTjc4SSgO/view?usp=sharing

- ***Arabic Test***:

https://drive.google.com/file/d/19R9dVoQF6YOoVoqdhnhhpgQJ7EawG65L/view?usp=sharing

Since all of text are right with no mistakes, I test the auto correction function with other one:



File Content

i use chat bot todey, can you detect the eror for me ?

Choose Summarizer Type

LexRank

Number Of Sentences

9

1 20

Corrected Text

i use chat not today, can you detect the error for me ?