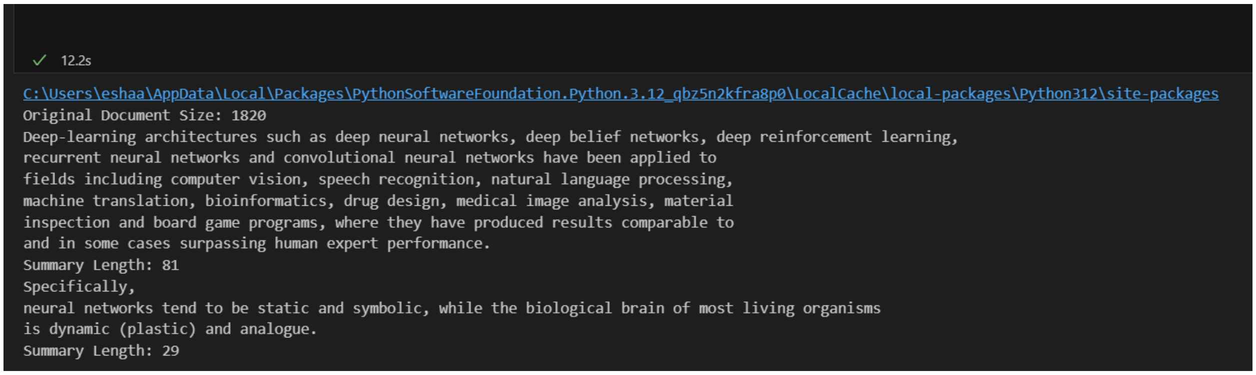
SUPERIOR UNIVERSITY

**Name: Esha asif**
**Roll no:034**
**Section:BSAI 4A**
**Subject:Programming for AI**
**LAB TASK 9**

## Output ss:

1.

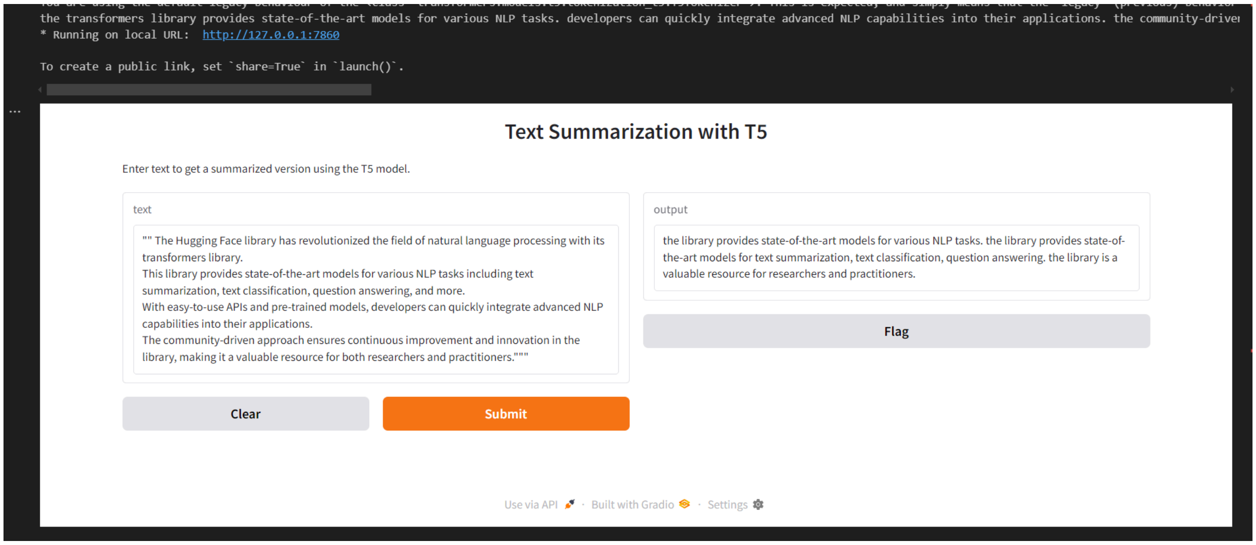


```
✓ 12.2s
C:\Users\eshaa\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.12_qbz5n2kfra8p0\LocalCache\local-packages\Python312\site-packages
Original Document Size: 1820
Deep-learning architectures such as deep neural networks, deep belief networks, deep reinforcement learning,
recurrent neural networks and convolutional neural networks have been applied to
fields including computer vision, speech recognition, natural language processing,
machine translation, bioinformatics, drug design, medical image analysis, material
inspection and board game programs, where they have produced results comparable to
and in some cases surpassing human expert performance.
Summary Length: 81
Specifically,
neural networks tend to be static and symbolic, while the biological brain of most living organisms
is dynamic (plastic) and analogue.
Summary Length: 29
```

How and why:

This code uses spacy along with the pytextrank package to generate a summary of a long text. It first loads a pre-trained language model and adds the pytextrank component to help extract key phrases and sentences from the text. The input text is about deep learning, explaining various neural networks and their applications. The texrank algorithm processes the text, selects the most important sentences, and generates a shorter summary. The code limits the summary to just two sentences and two key phrases, providing a more concise version of the original document. Finally, the code prints the size of the original text and the length of each summarized sentence.

2.



```
You are using the default legacy behaviour of the <class 'transformers.models.t5.tokenization_t5.T5Tokenizer'>. This is expected, and simply means that the `legacy` (previous) behavior
the transformers library provides state-of-the-art models for various NLP tasks. developers can quickly integrate advanced NLP capabilities into their applications. the community-drivei
* Running on local URL:  http://127.0.0.1:7860

To create a public link, set `share=True` in `launch()`.
```

Text Summarization with T5

Enter text to get a summarized version using the T5 model.

text

"" The Hugging Face library has revolutionized the field of natural language processing with its transformers library.
This library provides state-of-the-art models for various NLP tasks including text summarization, text classification, question answering, and more.
With easy-to-use APIs and pre-trained models, developers can quickly integrate advanced NLP capabilities into their applications.
The community-driven approach ensures continuous improvement and innovation in the library, making it a valuable resource for both researchers and practitioners."""

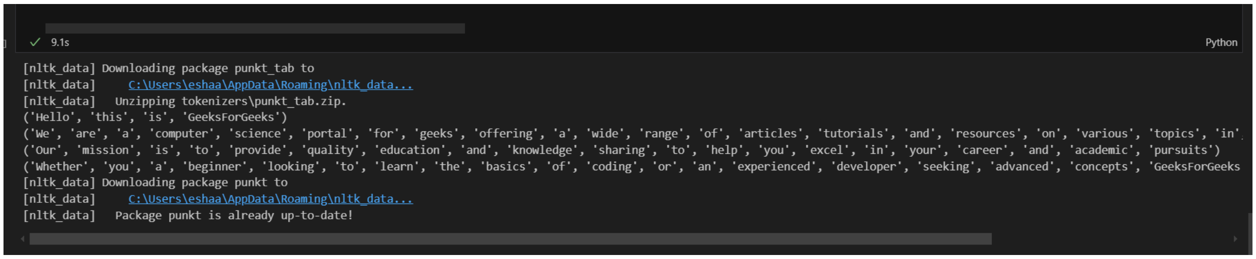
output

the library provides state-of-the-art models for various NLP tasks. the library provides state-of-the-art models for text summarization, text classification, question answering. the library is a valuable resource for researchers and practitioners.

## How and why:

This code uses the Gradio library along with the T5 model from Hugging Face to create a text summarization tool. It first loads the pre-trained T5 model and tokenizer, which are designed for tasks like summarization. The summarize function takes an input text, processes it through the tokenizer, and generates a summary using the model. The summary is limited to a maximum length of 150 characters and a minimum of 40, with additional constraints to ensure the summary is meaningful. The text about the hugging fave library is provided as an example, and the summary of this text is printed. Afterward, a Gradio interface is set up, allowing users to input their own text, get a summarized version, and interact with the tool through a simple web-based interface. The Gradio app is launched, letting users enter any text and see the generated summary in real-time.

3.



```
[nltk_data] Downloading package punkt_tab to
[nltk_data]     C:\Users\eshaa\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping tokenizers\punkt_tab.zip.
('Hello', 'this', 'is', 'GeeksForGeeks')
('We', 'are', 'a', 'computer', 'science', 'portal', 'for', 'geeks', 'offering', 'a', 'wide', 'range', 'of', 'articles', 'tutorials', 'and', 'resources', 'on', 'various', 'topics', 'in',
('Our', 'mission', 'is', 'to', 'provide', 'quality', 'education', 'and', 'knowledge', 'sharing', 'to', 'help', 'you', 'excel', 'in', 'your', 'career', 'and', 'academic', 'pursuits')
('Whether', 'you', 'a', 'beginner', 'looking', 'to', 'learn', 'the', 'basics', 'of', 'coding', 'or', 'an', 'experienced', 'developer', 'seeking', 'advanced', 'concepts', 'GeeksForGeeks
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\eshaa\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

## How and why:

This code uses NLTK and the Sumy library to break down a text into sentences and words. First, it downloads necessary models for sentence tokenization. Then, it takes a paragraph and splits it into individual sentences. After that, it breaks each sentence into words, printing them one by one. For example, the sentence "Hello, this is GeeksForGeeks!" would be split into the words hello, this, is, and GeeksForGeeks. The goal is to process text into manageable parts like sentences and words for further analysis.