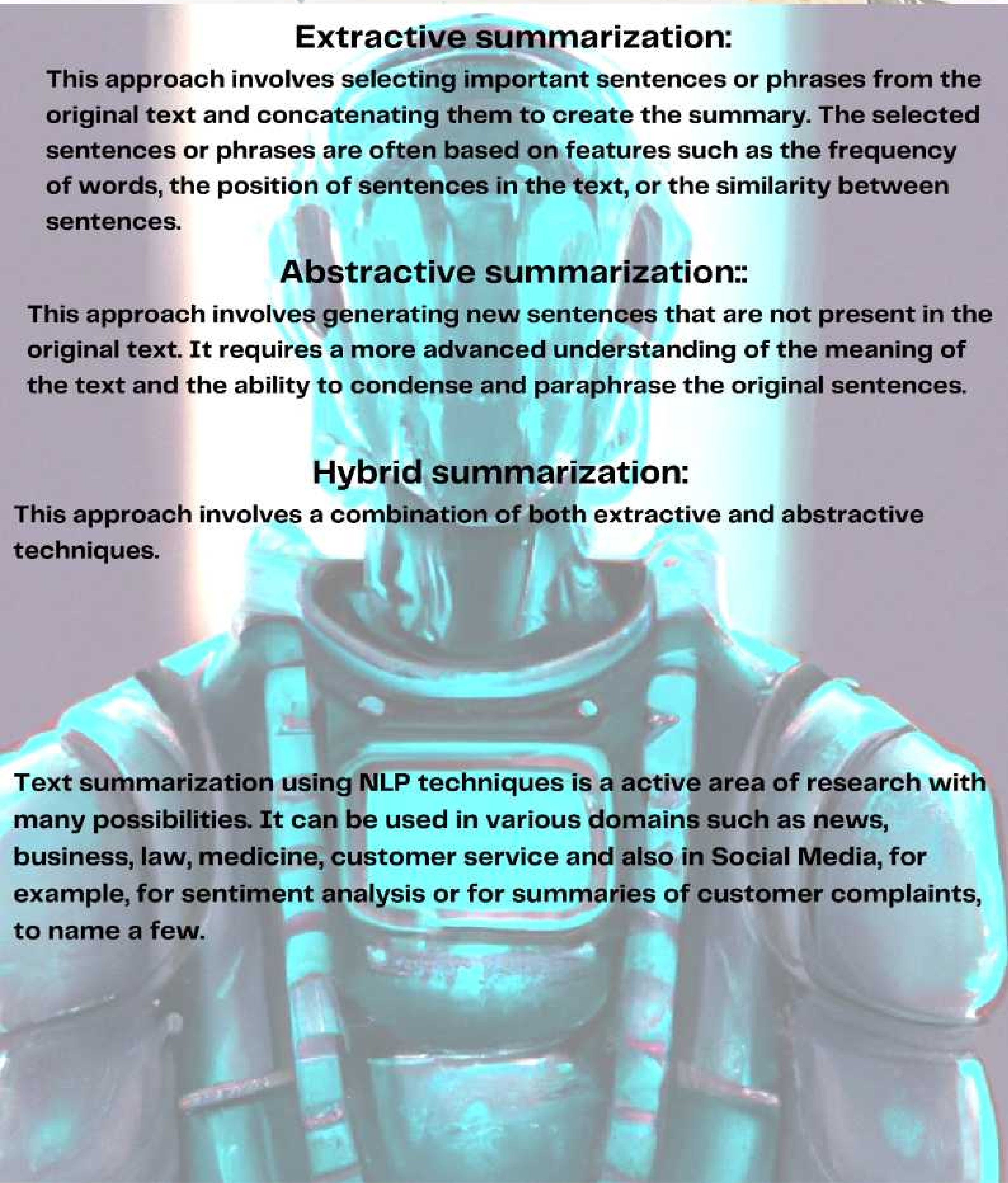
"AUTOMATIC TEXT SUMMARIZATION USING NATURAL LANGUAGE PROCESSING"

- **NLP stands for "Natural Language Processing."**
- **It is a field of artificial intelligence (AI) and computer science that focuses on the interaction between computers and humans using natural language.**
- **The goal of NLP is to develop algorithms and models that can understand, interpret, and generate human language in a way that is both meaningful and useful.**
- **NLP includes tasks such as:**
- **Language Translation,Sentiment Analysis,Text Summarization ,Question answering ,And many others.**

"Common Applications of Natural Language Processing (NLP) and its Reliance on Machine Learning"

- Machine translation: automatically translating one language to another
- Text summarization: automatically generating a shorter version of a text document
- Sentiment analysis: determining the emotional tone of a piece of text
- Named entity recognition: automatically identifying and extracting entities such as people, organizations, and locations from a piece of text
- Parts-of-speech tagging: identifying and classifying words in a sentence according to their grammatical function (such as noun, verb, adjective, etc.)
- NLP relies heavily on machine learning, particularly deep learning, which allows the model to learn from large amounts of data and to continuously improve over time.

"Exploring Text Summarization in Natural Language Processing: Techniques, Algorithms and Models"



Extractive summarization:

This approach involves selecting important sentences or phrases from the original text and concatenating them to create the summary. The selected sentences or phrases are often based on features such as the frequency of words, the position of sentences in the text, or the similarity between sentences.

Abstractive summarization::

This approach involves generating new sentences that are not present in the original text. It requires a more advanced understanding of the meaning of the text and the ability to condense and paraphrase the original sentences.

Hybrid summarization:

This approach involves a combination of both extractive and abstractive techniques.

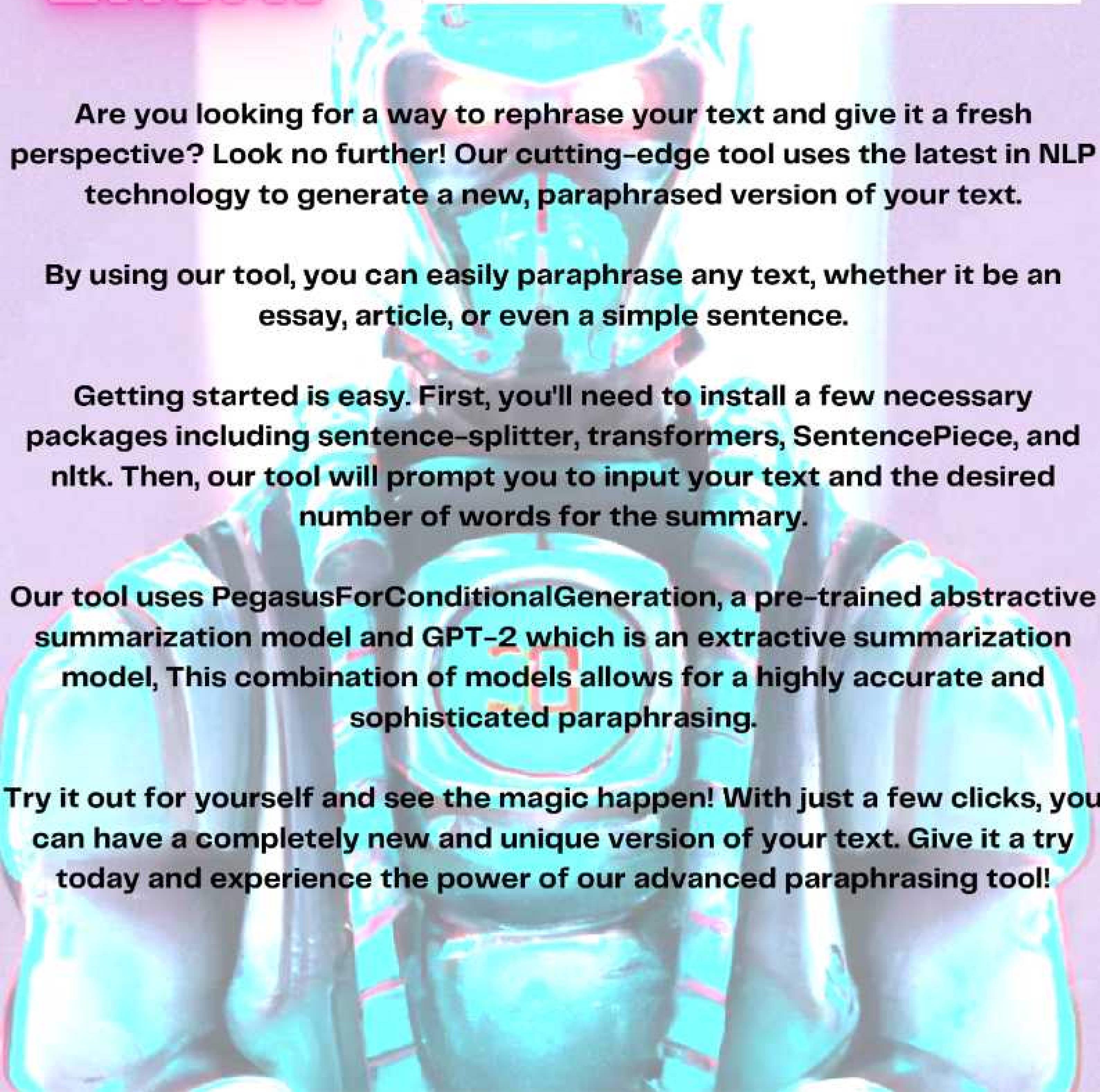
Text summarization using NLP techniques is a active area of research with many possibilities. It can be used in various domains such as news, business, law, medicine, customer service and also in Social Media, for example, for sentiment analysis or for summaries of customer complaints, to name a few.

A New Perspective on Text Summarization: My Hybrid Approach of Abstractive and Extractive Techniques with Pegasus and GPT-2

LINK:-

Google Colaboratory

google.com



Are you looking for a way to rephrase your text and give it a fresh perspective? Look no further! Our cutting-edge tool uses the latest in NLP technology to generate a new, paraphrased version of your text.

By using our tool, you can easily paraphrase any text, whether it be an essay, article, or even a simple sentence.

Getting started is easy. First, you'll need to install a few necessary packages including sentence-splitter, transformers, SentencePiece, and nltk. Then, our tool will prompt you to input your text and the desired number of words for the summary.

Our tool uses PegasusForConditionalGeneration, a pre-trained abstractive summarization model and GPT-2 which is an extractive summarization model, This combination of models allows for a highly accurate and sophisticated paraphrasing.

Try it out for yourself and see the magic happen! With just a few clicks, you can have a completely new and unique version of your text. Give it a try today and experience the power of our advanced paraphrasing tool!

To use this advanced text paraphrasing tool in a Jupyter notebook, you can follow these steps:

- Start by installing the necessary packages by running !pip install sentence-splitter, !pip install transformers, !pip install SentencePiece, and !pip install nltk in separate code cells.

- Then, in a new code cell, import the necessary libraries and modules such as torch, PegasusForConditionalGeneration, PegasusTokenizer, AutoModel, AutoTokenizer, and nltk.

- Next, define the model_name and torch_device variables and initialize the tokenizer and model using the appropriate modules and pre-trained models.

- Create a function called get_response(input_text) that takes in input_text as a parameter and uses nltk to tokenize the input text into sentences. The function then generates a paraphrased version of the input text using the pre-trained PegasusForConditionalGeneration model and the tokenizer.

- Create a TextGenerationPipeline from the transformers library for text generation using the GPT-2 model.

- In a new code cell, prompt the user to input the text they want to paraphrase and the desired number of words for the summary.

- **Pass the input text through the get_response function and store the result in a variable called paraphrased.**
- **Pass the paraphrased text to the TextGenerationPipeline and receive the summary.**

SENTENCE PIECE

SentencePiece is an unsupervised text tokenizer and detokenizer mainly for Neural Network-based text generation systems where the vocabulary size is predetermined prior to the neural model training. It is designed to be flexible, efficient, and highly adaptable to the specific needs of a given task.

SentencePiece allows you to build a model that can handle out-of-vocabulary (OOV) words by learning subword units, rather than using a fixed vocabulary. This can be especially useful in natural language processing tasks where the size of the vocabulary can be quite large, and where the text may contain rare or specialized words.

It also offers a number of different algorithm options for generating subword units, including unigram language modeling, and byte pair encoding (BPE) . SentencePiece also support several other features such as controlling the number of subword units to generate, encoding/decoding with pre-built models, and fine-tuning existing models.

It is widely used in NLP tasks such as Machine Translation, Text-to-Speech, and Text Summarization. And SentencePiece model is also supported by many pre-trained language model such as BERT, GPT, and T5 etc.

NLTK

The Natural Language Toolkit (nltk) is a library in Python that provides tools for working with human language data. It is designed to help researchers and developers who need to work with natural language data in their applications. The library provides a wide range of functionality, including the ability to:

- Tokenize text into words and sentences
- Tag parts of speech for words in a sentence
- Extract information from text, such as named entities and relationships
- Chunk and parse sentences to extract meaning and understand grammar
- Perform semantic analysis and perform operations on parse trees
- Perform Text classification and Tokenization
- It also contains a large number of pre-trained models for different languages, along with a variety of data sets and corpora (large bodies of text) that can be used to train and evaluate natural language processing models.

It is widely used in the field of Natural Language Processing (NLP) and Computational Linguistics. It can be used to build various NLP applications such as Text summarization, Text classification, Named Entity Recognition, Speech to text, and much more.

TRANFORMERS

In the context of machine learning, a transformer is a type of neural network architecture used for a variety of natural language processing tasks, such as language translation, text summarization, and question answering.

The transformer was introduced in a 2017 paper by researchers at Google, "Attention Is All You Need". The key innovation of the transformer is its use of self-attention mechanisms, which allows the model to weigh the importance of different words in the input when making predictions.

A transformer model is composed of an encoder and a decoder. The encoder takes the input text and produces a hidden representation of the text, which is then passed to the decoder to generate the output text.

The transformer uses a technique called self-attention, which allows the model to weigh the importance of different parts of the input when making predictions. This allows the model to focus

PIPELINE

Pipeline in Transformers refers to the process of transforming raw input data into a format that can be used in machine learning models. It involves a series of steps such as tokenization, numericalization, normalization, and feature engineering. The goal is to convert the raw text into a format that can be used by the model for training and prediction.

Pipelines in Transformers are used to allow for faster and more efficient data processing. They enable data to be processed in parallel, allowing for faster training and inference. Additionally, pipelines allow for a greater degree of scalability, allowing for larger datasets to be processed more efficiently.