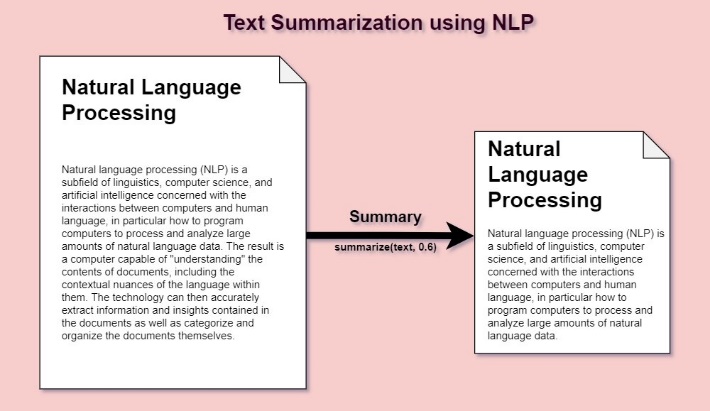
**Definition of Text Summarization**

Text summarization is the creation of a short, accurate, and fluent summary of a longer text document. It 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.



**Need for text summarization**

Today, our world is parachuted by the gathering and dissemination of huge amounts of data. With such a big amount of data circulating in the digital space, there is need to develop machine learning algorithms that can automatically shorten longer texts and deliver accurate summaries that can fluently pass the intended messages. Furthermore, applying text summarization reduces reading time, accelerates the process of researching for information, and increases the amount of information that can fit in an area.

* Understanding Context was Tr

**Limitations of other methods**

* icky.

**Limited Context Understanding: Previous models often had limited context understanding due to fixed-length representations, such as bag-of-words or fixed-size hidden states in RNNs. Transformers, with their self-attention mechanism, can capture relationships and dependencies across the entire input sequence, resulting in a more comprehensive understanding of the text.**

**Difficulty with Ambiguity: Earlier models struggled with handling ambiguous language or multiple interpretations in a text. Transformers can learn to disambiguate and generate more contextually appropriate summaries by leveraging the global context.**

**Lack of Paraphrasing Abilities: Traditional methods and older models often failed to produce summaries that effectively paraphrased the original content. Transformers are better at generating diverse and paraphrased summaries by learning from a vast amount of text data during pre-training.**

**Fixed-Length Output: Some earlier models generated summaries with fixed lengths or used template-based approaches, limiting their ability to produce concise and informative summaries. Transformers can dynamically adjust the length of the output summary based on the input content, resulting in more flexible and contextually relevant summaries.**

**Poor Handling of Named Entities: Transformers, with their contextual understanding and named entity recognition capabilities, can better handle named entities, ensuring they are preserved accurately in the summaries.**

**Language Generation Quality: Earlier abstractive summarization models often struggled with fluency, coherence, and grammaticality in their generated summaries. Transformers have improved the quality of generated language, making summaries more readable and human-like.**

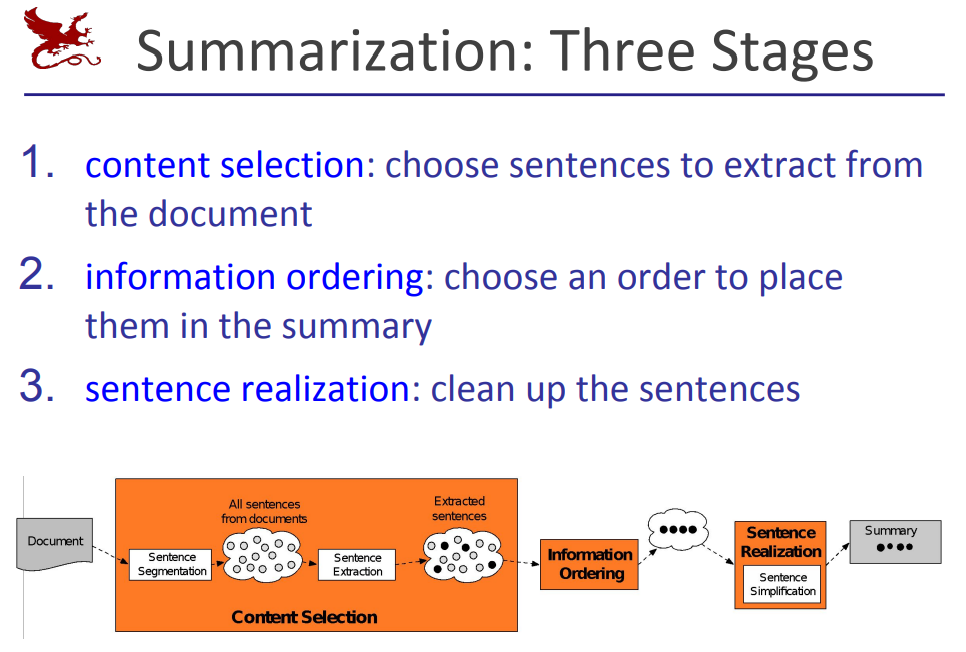
**Transfer Learning: Transformers are designed for transfer learning, enabling fine-tuning on specific summarization tasks with limited labeled data. This transfer learning capability allows models to adapt to different domains and languages more effectively.**

**Multimodal Summarization: Transformers can be extended to handle both text and other modalities, such as images and videos, which was challenging for many previous models.**

**Handling Long Documents: Transformers can effectively handle long documents, whereas earlier models often struggled with document length, leading to incomplete or fragmented summaries.**

**Global Coherence: Transformers can maintain global coherence in summaries, ensuring that the summary as a whole is coherent and contextually relevant to the entire document, which was challenging for some earlier models.**

**Steps in summarization**



**Extraction based summary**

Extractive summarization selects a subset of sentences from the text to form a summary

**How Extraction-based Summarization Works:**

**Sentence Selection:** In this approach, sentences or phrases from the original text are chosen for inclusion in the summary based on their relevance and importance. These sentences are typically selected based on various criteria, such as the presence of keywords, sentence importance scores, or other heuristics.

**Scoring and Ranking:** Each sentence in the source document is assigned a score or rank based on its content, relevance, and importance. Common methods for scoring sentences include TF-IDF (Term Frequency-Inverse Document Frequency), cosine similarity, and more advanced techniques like TextRank or PageRank.

**Thresholding:** Sentences with scores above a certain threshold are selected for the summary. The threshold can be determined based on the desired length of the summary or other criteria.

**Sentence Ordering:** The selected sentences are usually arranged in a logical order to create a coherent summary. Sometimes, chronological order or another predefined structure is used.

**Abstraction based summary**

abstractive summarization reorganizes the language in the text and adds novel words/phrases into the summary if necessary.

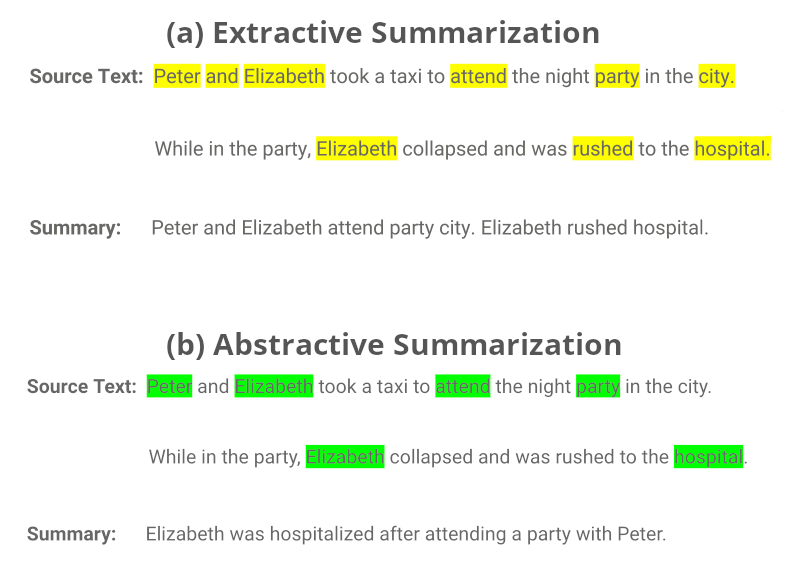
**How Abstraction-based Summarization Works:**

**Understanding Content:** The system analyzes the source text to understand its key concepts, ideas, and important details. This typically involves natural language processing (NLP) techniques such as part-of-speech tagging, syntactic parsing, named entity recognition, and semantic analysis.

**Paraphrasing and Rewriting:** Instead of directly copying sentences from the source, the system generates new sentences that convey the same information in a condensed and coherent form. This often involves rephrasing and paraphrasing the source content to produce more human-readable summaries.

**Sentence Generation:** The summarization system generates sentences that capture the essence of the source text. It may use synonyms, change sentence structures, and combine information from multiple sentences to create a coherent summary.

**Quality Assessment:** The generated sentences are evaluated for coherence, fluency, and informativeness. Some summarization systems use reinforcement learning or other machine learning techniques to improve the quality of the generated summaries.



**Applications**

* [1. Media monitoring](https://www.frase.io/blog/20-applications-of-automatic-summarization-in-the-enterprise/#1_Media_monitoring)
* [2. Newsletters](https://www.frase.io/blog/20-applications-of-automatic-summarization-in-the-enterprise/#2_Newsletters)
* [3. Search marketing and SEO](https://www.frase.io/blog/20-applications-of-automatic-summarization-in-the-enterprise/#3_Search_marketing_and_SEO)
* [4. Internal document workflow](https://www.frase.io/blog/20-applications-of-automatic-summarization-in-the-enterprise/#4_Internal_document_workflow)
* [5. Financial research](https://www.frase.io/blog/20-applications-of-automatic-summarization-in-the-enterprise/#5_Financial_research)
* [6. Legal contract analysis](https://www.frase.io/blog/20-applications-of-automatic-summarization-in-the-enterprise/#6_Legal_contract_analysis)
* [7. Social media marketing](https://www.frase.io/blog/20-applications-of-automatic-summarization-in-the-enterprise/#7_Social_media_marketing)
* [8. Question answering and bots](https://www.frase.io/blog/20-applications-of-automatic-summarization-in-the-enterprise/#8_Question_answering_and_bots)
* [9. Video scripting](https://www.frase.io/blog/20-applications-of-automatic-summarization-in-the-enterprise/#9_Video_scripting)
* [10. Medical cases](https://www.frase.io/blog/20-applications-of-automatic-summarization-in-the-enterprise/#10_Medical_cases)
* [11. Books and literature](https://www.frase.io/blog/20-applications-of-automatic-summarization-in-the-enterprise/#11_Books_and_literature)
* [12. Email overload](https://www.frase.io/blog/20-applications-of-automatic-summarization-in-the-enterprise/#12_Email_overload)
* [13. E-learning and class assignments](https://www.frase.io/blog/20-applications-of-automatic-summarization-in-the-enterprise/#13_E-learning_and_class_assignments)
* [14. Science and R&D](https://www.frase.io/blog/20-applications-of-automatic-summarization-in-the-enterprise/#14_Science_and_R_D)
* [15. Patent research](https://www.frase.io/blog/20-applications-of-automatic-summarization-in-the-enterprise/#15_Patent_research)
* [16. Meetings and video-conferencing](https://www.frase.io/blog/20-applications-of-automatic-summarization-in-the-enterprise/#16_Meetings_and_video-conferencing)
* [17. Help desk and customer support](https://www.frase.io/blog/20-applications-of-automatic-summarization-in-the-enterprise/#17_Help_desk_and_customer_support)
* [18. Helping disabled people](https://www.frase.io/blog/20-applications-of-automatic-summarization-in-the-enterprise/#18_Helping_disabled_people)
* [19. Programming languages](https://www.frase.io/blog/20-applications-of-automatic-summarization-in-the-enterprise/#19_Programming_languages)
* [20. Automated content creation](https://www.frase.io/blog/20-applications-of-automatic-summarization-in-the-enterprise/#20_Automated_content_creation)

## ****What is Tokenization?****

Tokenisation is the process of breaking up a given text into units called tokens. Tokens can be individual words, phrases or even whole sentences. In the process of tokenization, some characters like punctuation marks may be discarded. The tokens usually become the input for the processes like parsing and text mining. These tokens are very useful for finding such patterns as well as is considered as a base step for stemming and lemmatization.

Input Text: "Tokenization is an important NLP technique."

Tokenized Output: ["Tokenization", "is", "an", "important", "NLP", "technique", "."]

**Libraries used**

**Streamlit:** Streamlit is an open-source Python library that simplifies the creation of web applications for data science and machine learning, allowing users to build interactive and data-driven apps with minimal code.

**OpenAI:** OpenAI is an artificial intelligence research organization that develops cutting-edge AI models and technologies. It's known for pioneering language models like GPT-3 and GPT-2, which have applications in natural language understanding and generation.

**Transformers:** Transformers is a popular Python library by Hugging Face that provides pre-trained models and tools for natural language processing (NLP), including state-of-the-art transformer-based models for tasks like text classification, translation, and summarization.

**T5Tokenizer, TFT5ForConditionalGeneration:** T5Tokenizer is a tokenizer specific to the T5 (Text-to-Text Transfer Transformer) model, while TFT5ForConditionalGeneration is an implementation of T5 for conditional text generation tasks. T5 is a versatile NLP model that converts various text-based tasks into a text-to-text format.

**GPT2Tokenizer, TFGPT2LMHeadModel:** GPT2Tokenizer is a tokenizer for the GPT-2 model, and TFGPT2LMHeadModel is an implementation of GPT-2 for language modeling. GPT-2 is a generative language model known for its ability to generate coherent and contextually relevant text.

**PegasusTokenizer, TFPegasusForConditionalGeneration:** PegasusTokenizer is a tokenizer designed for the Pegasus model, while TFPegasusForConditionalGeneration is an implementation of Pegasus for conditional text generation tasks. Pegasus is a model known for its strong performance in abstractive text summarization.

**BartTokenizer, TFBartForConditionalGeneration:** BartTokenizer is a tokenizer tailored for the Bart model, and TFBartForConditionalGeneration is an implementation of Bart for conditional text generation tasks. Bart is a model suitable for tasks like text summarization, translation, and text generation.

**Models used**

Sure, here's a brief definition of each of the models used in the code:

1. \*\*T5 (Text-to-Text Transfer Transformer):\*\*

- T5 is a transformer-based model designed for various natural language processing tasks. It converts input text into output text, making it highly versatile for tasks like translation, summarization, and question answering.

2. \*\*GPT-3 (Generative Pre-trained Transformer 3):\*\*

- GPT-3 is a state-of-the-art language model developed by OpenAI. It is known for its ability to generate coherent and contextually relevant text. It's often used for text generation tasks without fine-tuning.

3. \*\*PEGASUS:\*\*

- PEGASUS is a transformer-based model designed for abstractive text summarization. It can generate concise and coherent summaries of input text, making it valuable for summarizing long documents and articles.

4. \*\*BART (Bidirectional and Auto-Regressive Transformers):\*\*

- BART is another transformer-based model used for text generation and summarization. It excels in tasks like text summarization and can generate human-readable and informative summaries from input text.