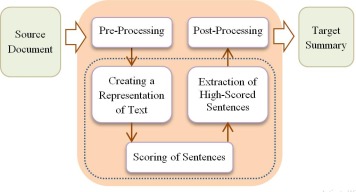
**TEXT SUMMARIZATION**

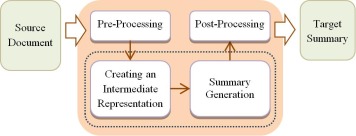
**Types of Text summarization:**

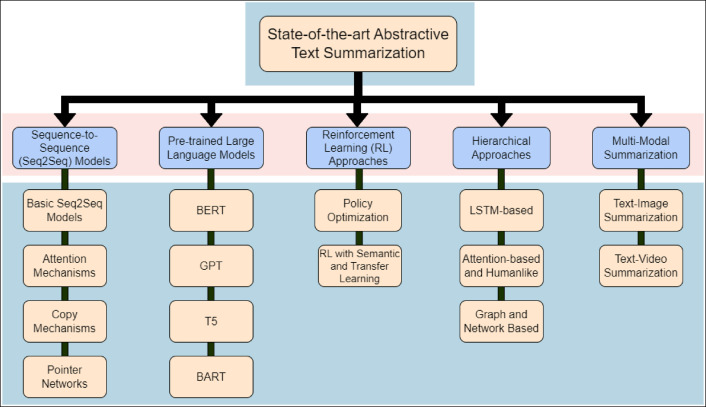
1. Extractive Summarization – Collects only important information from the text (gives us same text)
   1. Frequency-Based Methods
   2. Graph-Based Methods
   3. Latent Semantic Analysis (LSA)
   4. Cluster-Based Methods
2. Abstractive Summarization – Analyses the meaning of text and produces summarized sentence (reframes the text)
   1. Sequence-to-Sequence Models
   2. Transformers: GPT, BERT, T5
   3. BART (Bidirectional and Auto-Regressive Transformers)
   4. PEGASUS
   5. Seq2Seq, Encoder-Decoder Systems - feed-forward neural language model (FFNLM), CNN based Hierarchical Encoder-Decoder Models
3. Hybrid (Combining both Extractive and Abstractive)
4. General Mechanisms: Gensim, Sumy (LexRank, Luhn, LSA, TextRank), NLTK, T5, GPT-3

Extractive Summarization:

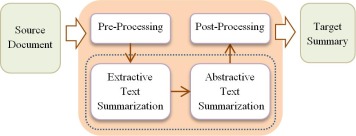


Abstractive Summarization:

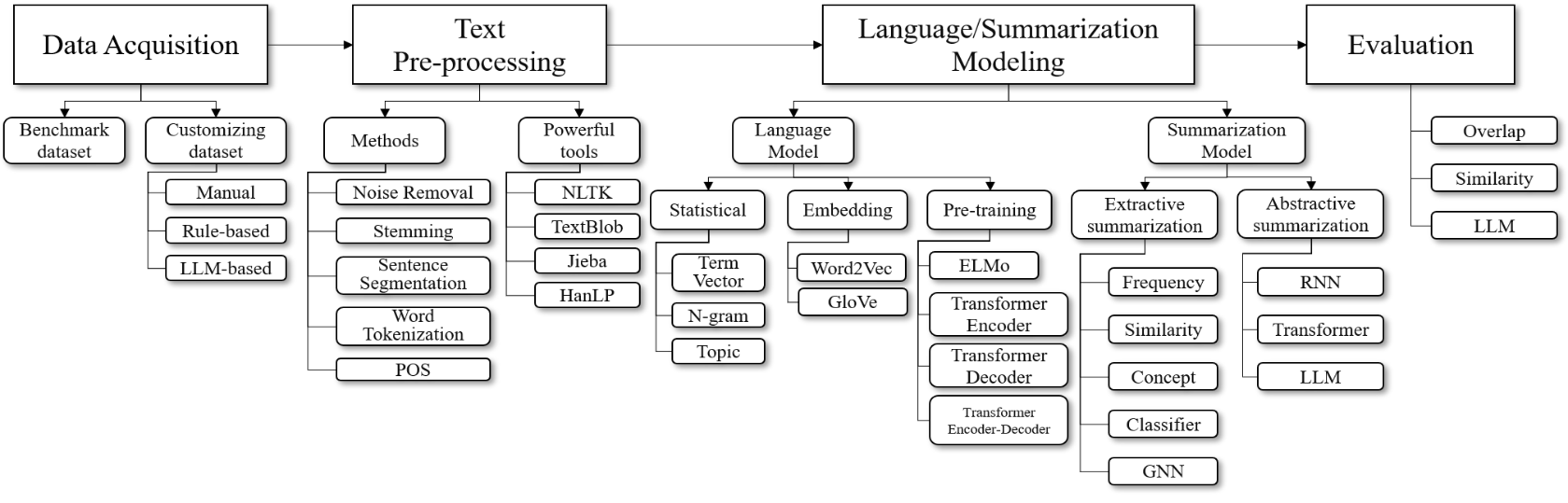


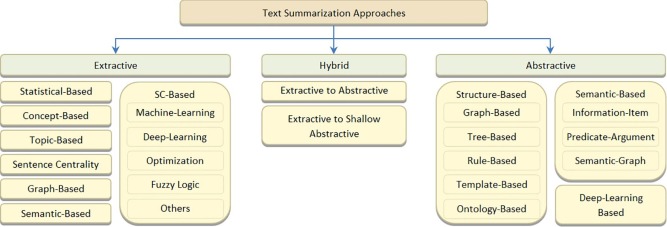


Hybrid:



Text Summarization Approaches:





**Predefined libraries in Python for Abstractive Summarization:**

1. Hugging Face Transformers
   1. BERTSUM
   2. GPT-2 / GPT-3
   3. T5
   4. BART
   5. PEGASUS
2. Fairseq
   1. BART
   2. PEGASUS
3. OpenNMT: Various transformer models can be configured for summarization.
4. AllenNLP: Includes implementations of sequence-to-sequence models that can be fine-tuned for summarization.
5. SpaCy: A library for advanced NLP, though primarily focused on traditional NLP tasks, While SpaCy does not have built-in abstractive summarization models, you can use it in conjunction with Hugging Face Transformers for more complex tasks.
6. FastAPI / Streamlit: Not specific to summarization but allows for easy deployment of models, you can build APIs or web applications to serve summarization models from libraries like Hugging Face.

**References:**

<https://arxiv.org/html/2403.02901v1>

<https://blog.paperspace.com/implement-seq2seq-for-text-summarization-keras/>

<https://ieeexplore.ieee.org/document/10170660>

<https://turbolab.in/types-of-text-summarization-extractive-and-abstractive-summarization-basics/>

<https://medium.com/nlplanet/two-minutes-nlp-four-different-approaches-to-text-summarization-5a0ce9c06c74>

<https://www.turing.com/kb/5-powerful-text-summarization-techniques-in-python>

<https://www.sciencedirect.com/science/article/pii/S0925231224010269?casa_token=5xudIxNwfdUAAAAA:D0t2eRd6-Y-lsn8rbTpsJi0FKgUR7drQj056g6_-je9k6dFS4tHlI3Nbqk_T7tQc66xOuu8Pkg>

<https://www.sciencedirect.com/science/article/pii/S0957417420305030?casa_token=2x2-X1yYO0IAAAAA:WYCXUWYxiOKxKXYVJCZ6S3NmsmEp3Ox-J5NdGL8AAoViTWcBdiKnLPLNmD4lm6ciV13m2hLvZA>

OpenAI: usage of openai in python for text summarization

<https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://www.youtube.com/watch%3Fv%3DSwBGozTFASs&ved=2ahUKEwjUqLjqjuSIAxXG4skDHasPOzUQwqsBegQIDRAF&usg=AOvVaw3RQNFikLjxYvigD0cxTsAd>

<https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://www.youtube.com/watch%3Fv%3DxTwy6RBkfr0&ved=2ahUKEwjUqLjqjuSIAxXG4skDHasPOzUQwqsBegQIDxAG&usg=AOvVaw1goP7Xw2hHhbvzWqHWn3xr>  
  
Some newer transformer models are designed to handle longer documents more efficiently (such as **Longformer**, **BigBird**, or **LED, reformer**). These models can handle larger token sequences compared to standard transformers like T5.