**Classical Approaches**

Classical Approaches to Natural Language Processing

* Text Preprocessing
  + Regular Expressions
    - [How to write Regular Expressions?](https://www.geeksforgeeks.org/write-regular-expressions/)
    - [Properties of Regular expressions](https://www.geeksforgeeks.org/properties-of-regular-expressions/)
    - Text Preprocessing using RE
    - [Regular Expression](https://www.geeksforgeeks.org/regular-expression-python-examples-set-1/)
    - [Email Extraction using RE](https://www.geeksforgeeks.org/extracting-email-addresses-using-regular-expressions-python/)
  + [Tokenization](https://www.geeksforgeeks.org/nlp-how-tokenizing-text-sentence-words-works/)
    - [White Space Tokenization](https://www.geeksforgeeks.org/python-nltk-nltk-whitespacetokenizer/)
    - Dictionary Based Tokenization
    - Rule-Based Tokenization
    - [Regular Expression Tokenizer](https://www.geeksforgeeks.org/python-nltk-tokenize-regexp/)
    - Penn Treebank Tokenization
    - [Spacy Tokenizer](https://www.geeksforgeeks.org/tokenization-using-spacy-library/)
    - Subword Tokenization
    - [Tokenization with Textblob](https://www.geeksforgeeks.org/python-tokenize-text-using-textblob/)
  + [Tokenize text using NLTK in python](https://www.geeksforgeeks.org/tokenize-text-using-nltk-python/)
  + [How tokenizing text, sentences, and words works](https://www.geeksforgeeks.org/nlp-how-tokenizing-text-sentence-words-works)
  + [Lemmatization](https://www.geeksforgeeks.org/python-lemmatization-approaches-with-examples/)
  + [Stemming](https://www.geeksforgeeks.org/introduction-to-stemming/)
    - Types
      * [Porter Stemmer](https://www.geeksforgeeks.org/introduction-to-stemming/)
      * [Lovins Stemmer](https://www.geeksforgeeks.org/introduction-to-stemming/)
      * [Dawson Stemmer](https://www.geeksforgeeks.org/introduction-to-stemming/)
      * [Krovetz Stemmer](https://www.geeksforgeeks.org/introduction-to-stemming/)
      * [Xerox Stemmer](https://www.geeksforgeeks.org/introduction-to-stemming/)
  + Stopwords removal
    - [Removing stop words with NLTK in Python](https://www.geeksforgeeks.org/removing-stop-words-nltk-python/)
  + Parts of Speech (POS)
    - [Part of Speech – Default Tagging](https://www.geeksforgeeks.org/nlp-part-of-speech-default-tagging/)
    - [Part of speech tagging – word corpus](https://www.geeksforgeeks.org/nlp-part-of-speech-tagged-word-corpus/)
    - [Part of Speech Tagging with Stop words using NLTK in python](https://www.geeksforgeeks.org/part-speech-tagging-stop-words-using-nltk-python/)
    - [Part of Speech Tagging using TextBlob](https://www.geeksforgeeks.org/python-part-of-speech-tagging-using-textblob/)
  + [Text Normalization](https://www.geeksforgeeks.org/normalizing-textual-data-with-python/)
* **Text Vectorization or Encoding:**
  + [vector space model (VSM)](https://www.geeksforgeeks.org/web-information-retrieval-vector-space-model/)
  + Words and vectors
  + [Cosine similarity](https://www.geeksforgeeks.org/cosine-similarity/)
  + Basic Text Vectorization approach:
    - One-Hot Encoding
    - Byte-Pair Encoding (BPE)
    - [Bag of words (BOW)](https://www.geeksforgeeks.org/bag-of-words-bow-model-in-nlp/)
    - N-Grams
    - [Term frequency Inverse Document Frequency (TFIDF)](https://www.geeksforgeeks.org/understanding-tf-idf-term-frequency-inverse-document-frequency/)
    - [N-Gram Language Modelling with NLTK](https://www.geeksforgeeks.org/n-gram-language-modelling-with-nltk/)
  + Distributed Representations:
    - Word Embeddings
    - Pre-Trained Word Embeddings
      * [Word Embedding using Word2Vec](https://www.geeksforgeeks.org/python-word-embedding-using-word2vec/)
      * [Finding the Word Analogy from given words using Word2Vec embeddings](https://www.geeksforgeeks.org/finding-the-word-analogy-from-given-words-using-word2vec-embeddings/)
      * [GloVe](https://www.geeksforgeeks.org/pre-trained-word-embedding-using-glove-in-nlp-models/)
      * [fasttext](https://www.geeksforgeeks.org/fasttext-working-and-implementation/)
    - [Train Own Word Embeddings](https://www.geeksforgeeks.org/word-embeddings-in-nlp/)
      * Continuous bag of words (CBOW)
      * [SkipGram](https://www.geeksforgeeks.org/implement-your-own-word2vecskip-gram-model-in-python/)
    - Doc2Vec
  + Universal Text Representations
    - [Embeddings from Language Models (ELMo)](https://www.geeksforgeeks.org/overview-of-word-embedding-using-embeddings-from-language-models-elmo/)
    - [Bidirectional Encoder Representations from Transformers (BERT)](https://www.geeksforgeeks.org/explanation-of-bert-model-nlp/)
  + Embeddings Visualizations
    - t-sne (t-distributed Stochastic Neighbouring Embedding)
    - TextEvaluator
  + Embeddings semantic properties
* **Semantic Analysis**
  + [What is Sentiment Analysis?](https://www.geeksforgeeks.org/what-is-sentiment-analysis/)
  + [Understanding Semantic Analysis](https://www.geeksforgeeks.org/understanding-semantic-analysis-nlp/)
  + Sentiment classification:
    - [Naive Bayes Classifiers](https://www.geeksforgeeks.org/naive-bayes-classifiers/)
    - Logistic Regression
    - [Sentiment Classification Using BERT](https://www.geeksforgeeks.org/sentiment-classification-using-bert/)
    - [Twitter Sentiment Analysis using textblob](https://www.geeksforgeeks.org/twitter-sentiment-analysis-using-python/)
* **Parts of Speech tagging and Named Entity Recognizations:**
  + [Parts of Speech tagging with NLTK](https://www.geeksforgeeks.org/nlp-part-of-speech-default-tagging/)
  + [Parts of Speech tagging with spacy](https://www.geeksforgeeks.org/python-pos-tagging-and-lemmatization-using-spacy/)
  + Hidden Markov Model for POS tagging
    - Markov Chains
    - Hidden Markov Model
    - Viterbi Algorithm
  + Conditional Random Fields (CRFs)
    - Conditional Random Fields (CRFs)  for POS tagging
  + Named Entity Recognition
    - Rule Based Approach
    - [Named Entity Recognizations](https://www.geeksforgeeks.org/named-entity-recognition/)
* **Neural Network for NLP:**
  + Feedforwards networks for NLP
  + [Recurrent Neural Networks](https://www.geeksforgeeks.org/recurrent-neural-networks-explanation/)
  + RNN for Text Classifications
  + RNN for Sequence Labeling
  + Stacked RNNs
  + Bidirectional RNNs
  + [Long Short-Term Memory (LSTM)](https://www.geeksforgeeks.org/long-short-term-memory-networks-explanation/)
  + [LSTM with Tensorflow](https://www.geeksforgeeks.org/long-short-term-memory-lstm-rnn-in-tensorflow/)
  + Bidirectional LSTM
  + [Gated Recurrent Unit (GRU)](https://www.geeksforgeeks.org/gated-recurrent-unit-networks/)
  + [Sentiment Analysis with RNN,LSTM, GRU](https://www.geeksforgeeks.org/sentiment-analysis-with-an-recurrent-neural-networks-rnn/)
  + [Emotion Detection using Bidirectional LSTM & GRU](https://www.geeksforgeeks.org/emotion-detection-using-bidirectional-lstm/)
  + [Transformers for NLP](https://www.geeksforgeeks.org/getting-started-with-transformers/)
* **Transfer Learning for NLP:**
  + [Bidirectional Encoder Representations from Transformers](https://www.geeksforgeeks.org/explanation-of-bert-model-nlp/)
  + [RoBERTa](https://www.geeksforgeeks.org/overview-of-roberta-model/)
  + [SpanBERT](https://www.geeksforgeeks.org/intuition-of-spanbert/)
  + Transfer Learning with Fine-tuning
* **Informations Extractions**
  + Keyphrase Extraction
  + Named Entity Recognition
  + Relationship Extraction
* **Information Retrieval**
* **Text Generations**
  + [Text Generations introductions](https://www.geeksforgeeks.org/text-generation-using-knowledge-distillation-and-gan/)
* **Text summarization**
  + [**Extractive Text Summarization using Gensim**](https://www.geeksforgeeks.org/natural-language-processing-nlp-tutorial/Extractive%20Text%20Summarization%20using%20Gensim)
* **Questions – Answering**
* **Chatbot & Dialogue Systems:**
  + [Simple Chat Bot using ChatterBot](https://www.geeksforgeeks.org/chat-bot-in-python-with-chatterbot-module/)
  + [GUI chat application using Tkinter](https://www.geeksforgeeks.org/gui-chat-application-using-tkinter-in-python/)
* **Machine translation**
  + [Machine translation Introductions](https://www.geeksforgeeks.org/machine-translation-of-languages-in-artificial-intelligence/)
  + [Statistical Machine Translation Introduction](https://www.geeksforgeeks.org/statistical-machine-translation-of-languages-in-artificial-intelligence/)
* **Phonetics**
  + [Implement Phonetic Search in Python with Soundex Algorithm](https://www.geeksforgeeks.org/implement-phonetic-search-in-python-with-soundex-algorithm/)
  + [Convert English text into the Phonetics](https://www.geeksforgeeks.org/convert-english-text-into-the-phonetics-using-python/)
* **Speech Recognition and Text-to-Speech**
  + [Convert Text to Speech](https://www.geeksforgeeks.org/convert-text-speech-python/)
  + [Convert Speech to text and text to Speech](https://www.geeksforgeeks.org/python-convert-speech-to-text-and-text-to-speech/)
  + [Speech Recognition using Google Speech API](https://www.geeksforgeeks.org/speech-recognition-in-python-using-google-speech-api/)

**Empirical and Statistical Approaches**

* Treebank Annotation
* Fundamental Statistical Techniques for NLP
* Part-of-Speech Tagging
* Rules-based system
* Statistical Parsing
* Multiword Expressions
* Normalized Web Distance and Word Similarity
* Word Sense Disambiguation