MS9007 Practical Natural Language Processing

Welcome to this Course! 😎

This module will immerse you in the rapidly evolving field of natural language processing (NLP), which focuses on enabling computers to learn and understand human languages. In this module, you will learn to leverage on text-based data to perform various analyses such as **sentiment analysis, text classification, text summarization, question answering** systems and learn to generate **text from image** inputs. You will have the opportunity to work on datasets that are based on real-world scenarios and apply what they have learnt to solve practical business problems in a capstone project.

Chapter 1: Introduction to Natural Language Processing

By the end of the lesson, students should be able to:

(1) explain what NLP is
(2) appreciate why NLP is a complicated task
(3) describe core tasks in NLP
(4) use transformers to perform basic NLP applications

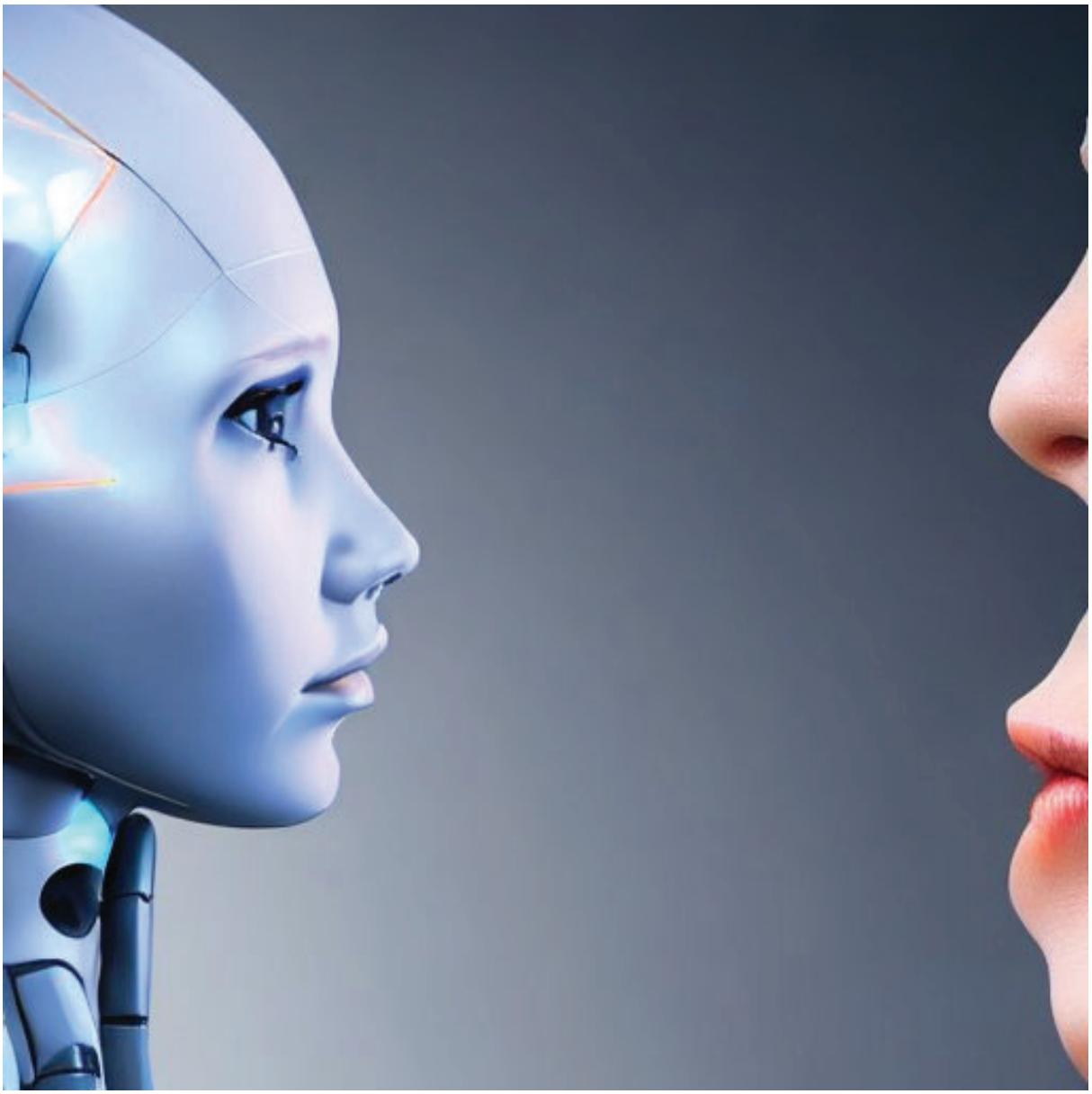
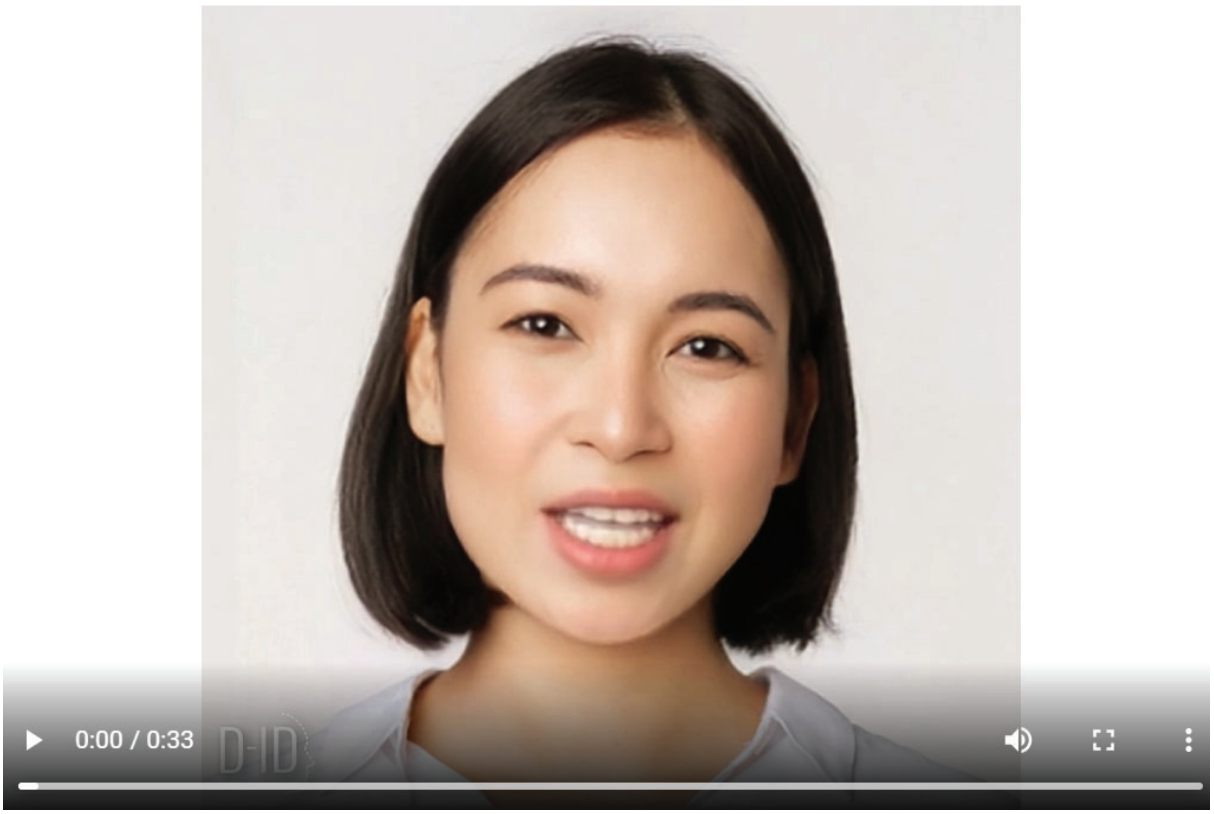


Image created by AI with the text prompt "*artificial intelligence talking to human*"
(Note: A different image would be generated each time.)

What is NLP?

NLP stands for Natural Language Processing. Let's break this up! *Natural Language* refers to human language. Thus, English, Chinese, Malay, Tamil, Spanish, Japanese, Korean are all examples of human language. However, computer programming languages like Python, Java Script are not. Processing refers to how a computer carries out instructions.

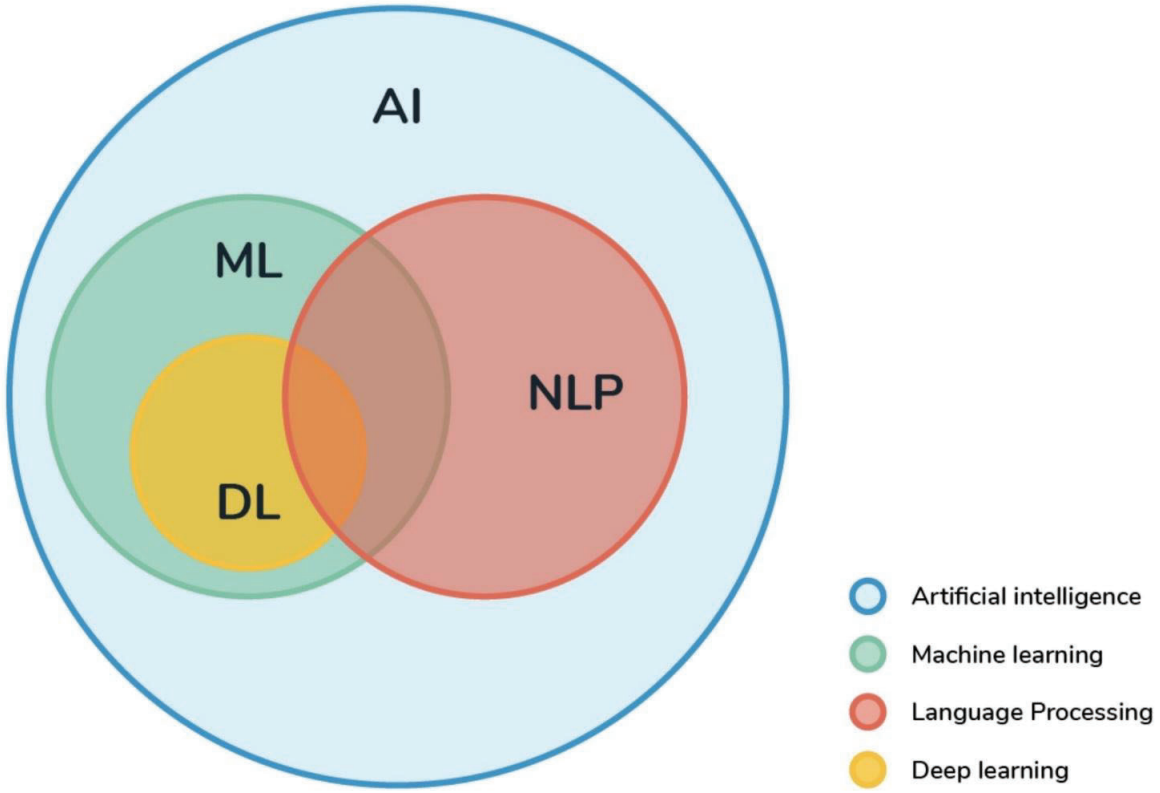


Click on this link to watch another video on the introduction of NLP created using AI.
(Does it sound natural to you? It was created by adding a script only and choosing an AI voice.)

Natural Language Processing is a subfield of **artificial intelligence** that is concerned with the interactions between computers and human (natural) language, in particular how to program computers to process and analyze natural language data.

Rule-based methods, techniques of ML, Machine Learning and DL, Deep Learning are utilised in NLP. Machine Learning is concerned about giving computers the ability to learn from data.

Whereas, Deep Learning is a subset of machine learning that uses deep neural networks to learn from data. You can read more on AI, ML and DL
here or via this link to read up.



Usefulness of NLP

How is NLP useful to humans? Natural Language Processing gives the machines the ability to read, understand and derive meaning from human languages. This would facilitate the analysis and interpretation of massive amounts of unstructured text data that would otherwise be challenging to process. This can assist businesses in making **wiser decisions**, **enhancing the customer experience**, and **automating several operations** that would otherwise demand a lot of manual labor.

Applications of NLP

There are many practical applications of natural language processing (NLP). Here are some applications which we will be looking at in this Practical NLP course, namely Sentiment

Analysis, Text Classification, Text Summarization, Question & Answering as well as Vision and Language Applications such as Image-to-Text.

Sentiment analysis: NLP algorithms can be used to automatically analyze the sentiment (positive, neutral, or negative) of text, Some use cases would be to classify news articles and social media posts by sentiment, enabling businesses and individuals to quickly understand the overall sentiment towards a particular topic or brand.

Text classification: NLP algorithms can be used to automatically classify text into different categories or topic classification (politics, sports, etc.). Some use cases include automatically categorizing emails (e.g. spam or not), inquiries, and customer feedback into relevant topics, enabling businesses to prioritize and respond to them more efficiently.

Text summarization: NLP algorithms can be used to automatically summarize long texts, such as articles or reports, into shorter, more concise summaries. Some use cases would be using it to automatically generate summaries of news articles, scientific papers, and legal documents, enabling users to quickly understand the key points without having to read the entire text. Additionally, NLP algorithms can be used to summarize large volumes of customer feedback or social media posts, allowing businesses to quickly understand customer sentiment and identify key issues or trends.

Question & answering: NLP algorithms can be used for question answering. Use cases include automatically answering frequently asked questions, such as those found on a website's FAQ page or in a customer support knowledge base, using natural language processing to understand the user's query and provide a relevant answer. Additionally, NLP algorithms can be used for complex question answering, such as in a trivia game or as part of a virtual assistant, by interpreting the user's question and providing a correct and relevant response.

Image to text: NLP algorithms can be used for image to text conversion which is automatically extracting text from images of documents, such as receipts or forms, making it easier to store and process large volumes of data. Additionally, NLP algorithms can be used to convert text within images to other languages (translation), making it easier for people to communicate across language barriers.

Overall, the practical applications of NLP are vast and can be applied in a wide range of industries, including customer service, marketing, finance, and healthcare.

Let’s use the transformer library to get a glimpse of how some NLP applications work. (If you want to find out more about transformers, read more here.)

[Lab] MS9007 Lesson 1 A tour of NLP applications.ipynb

Concepts in NLP

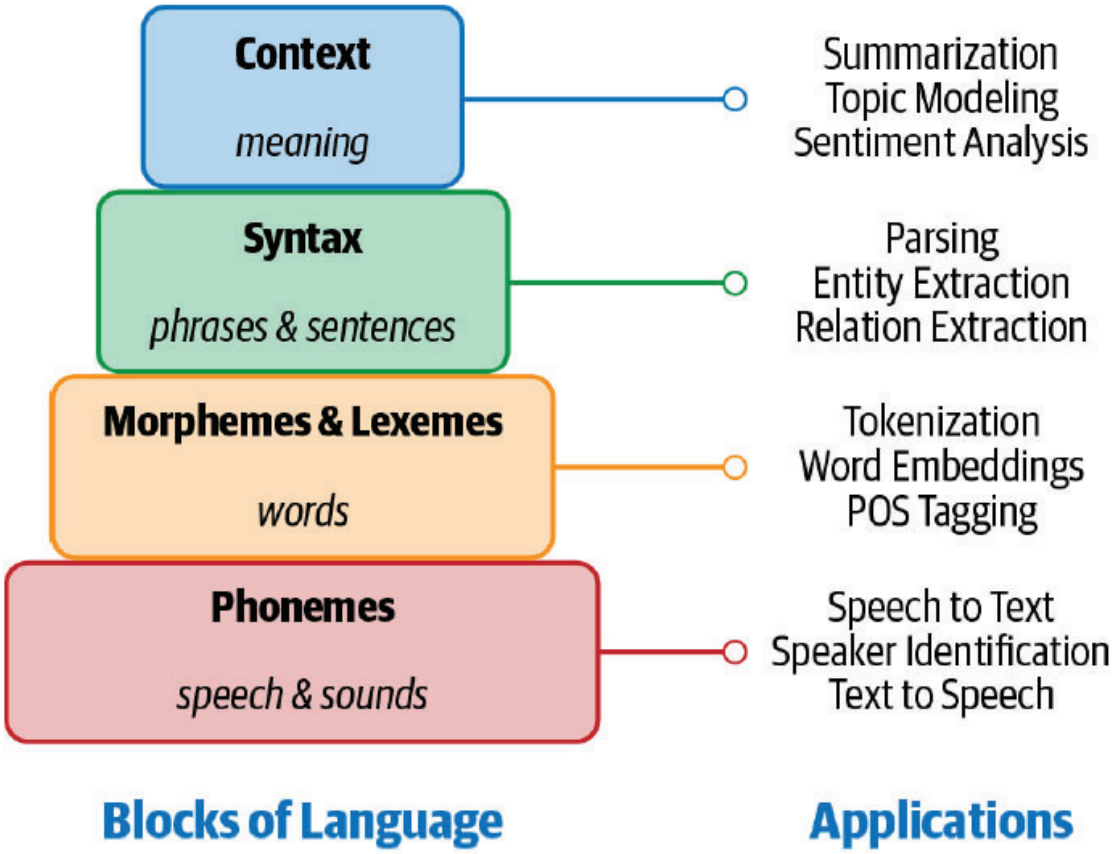
Language is a structured system of communication that involves complex combinations of its constituent components, such as characters, words, sentences, etc.

To study NLP, it is important to understand key concepts from linguistics.
Linguistics is the study of language and its structure, including the ways in which words are formed, how they combine to form sentences, and how meaning is conveyed through language. Understanding linguistic concepts is essential for developing effective NLP algorithms, as it provides a foundation for understanding the underlying structure of language and how it can be processed and analyzed by computers. Some linguistic concepts that are important in NLP include syntax, semantics, pragmatics, and discourse analysis. By drawing on these concepts, NLP researchers and developers can create algorithms that accurately analyze, interpret, and generate natural language text.

Language is a complex system of communication that involves various building blocks, such as **phonemes, morphemes, syntax, and context.** *Phonemes* are the smallest units of sound in a language, *morphemes* are the smallest units of language with meaning, and *syntax* refers to the rules for constructing grammatically correct sentences. *Context* is also crucial for understanding language, as the meaning of words and phrases can change based on their context.

To study NLP, it is important to have a solid understanding of these linguistic concepts, as they provide the foundation for developing effective NLP algorithms. For example, morphological analysis is the basis for many NLP tasks, such as tokenization, stemming, and part-of-speech tagging. Syntax is also critical for tasks such as parsing and named entity recognition, while context is essential for tasks such as sentiment analysis and text summarization.

By drawing on these linguistic concepts, NLP researchers and developers can create algorithms that accurately analyze, interpret, and generate natural language text, making it possible to automate a wide range of language-related tasks and applications.



Why is NLP challenging?

NLP is challenging for several reasons:

- Ambiguity: Natural language is inherently ambiguous and often requires context to understand its meaning. The same word or phrase can have different meanings depending on the context in which it is used.
- Synonymy: Different words can have the same or similar meanings, making it difficult for NLP models to accurately understand and interpret language.
- Coreference: Referring to the same entity or concept using different words or phrases can create confusion for NLP models.

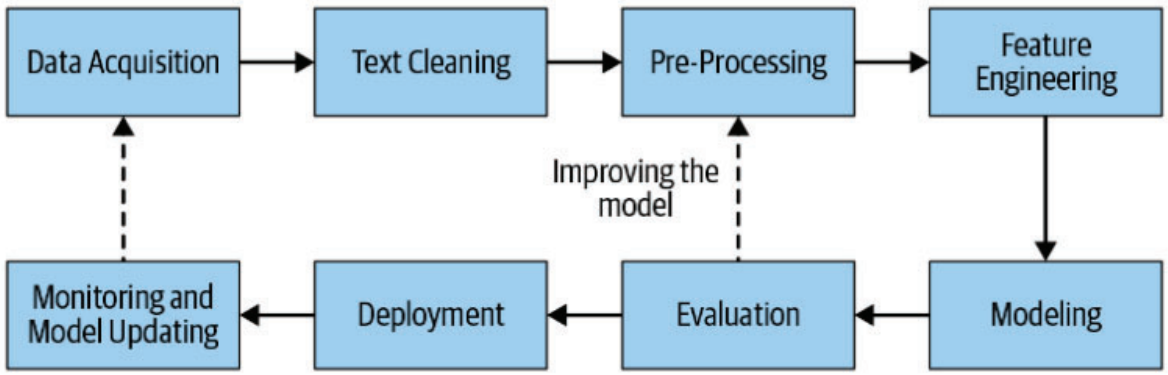
- Diversity across languages: Different languages have different structures, vocabularies, and grammatical rules, making it challenging to develop NLP solutions that work well across multiple languages.

  Additionally, some languages may have limited resources available for training NLP models, making it difficult to achieve high levels of accuracy and performance for low-resource languages.

These challenges make NLP a complex and evolving field that requires ongoing research and innovation to develop effective solutions.

Pipeline of NLP

A typical pipeline for an NLP application may include the following steps:



Data Acquisition (Collection), Text Cleaning, Pre-processing: After data acquisition or collection, there is a need to clean and pre-process the text, such as converting all text to lowercase, removing punctuation and numbers, and splitting the text into individual words or tokens.

Feature extraction: In this step, relevant features are extracted from the preprocessed text data. This can include things like part-of-speech tagging, named entity recognition, and term frequency-inverse document frequency (TF-IDF) vectors.

Model training: In this step, a machine learning model is trained on the extracted features to perform a specific task, such as sentiment analysis or language translation.

Model evaluation: After the model has been trained, it is typically evaluated on a separate test dataset (unseen by the model during training) to measure its performance and determine how well it can generalize to unseen data.

Model deployment: If the model performs well on the evaluation dataset, it can be deployed in a production environment, where it can be used to process real-world data.
The model should also be monitored and updated.

References for Chapter 1

1. "Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems", Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta, Harshit Surana, 2020
2. "Natural Language Processing with Transformers: Building Language Applications with Hugging Face", Lewis Tunstall, Leandro von Werra, Thomas Wolf, 2022
3. "Speech and Language Processing", Dan Jurafsky and James H. Martin, 2022
4. The Hugging Face Course, 2022, link here

Chapter 2: Text Preprocessing

By the end of the lesson, students should be able to:
(1) explain text preprocessing techniques in NLP
(2) perform text preprocessing in NLP

Why do we need to do text preprocessing?

Text preprocessing is a crucial step in Natural Language Processing (NLP) that involves transforming raw text data into a format that is more suitable for analysis. Text data is often unstructured and messy, containing noise, inconsistencies, and irrelevant information. Text preprocessing helps to clean and transform the text data to a standardized and consistent format, making it easier for NLP algorithms to extract meaningful insights and patterns. By removing stopwords, lowercasing, removing punctuation, tokenizing, and stemming as well as lemmatizing the text, the resulting data is easier to analyze, which can improve the accuracy

and effectiveness of NLP models.Thus, text preprocessing is essential in NLP to ensure that text data is properly formatted for effective analysis and insights.

Common text preprocessing

These are common text preprocessing techniques used in natural language processing:

- Lowercasing: Convert all text to lowercase to avoid different casing for the same word.
- Removing punctuation: Remove all punctuation marks such as periods, commas, and semicolons as they usually do not add much meaning to the text.
- Removing stopwords: Remove common words that do not add much meaning to the text such as "the," "a," "an," "in," and "is".
- Removal of frequent words: This is a common text preprocessing step that involves removing frequently occurring words from the text data, which are also known as stopwords. Stopwords are usually common words that do not add much meaning to the text and can be safely removed. Examples of stopwords include "a", "an", "the", "and", "or", "but", etc.
- Removal of emojis/emoticons: Emojis and emoticons are graphical symbols used to express emotions, and are commonly used in social media and online communication. However, they can add noise to text data and may not be relevant to the analysis, especially in tasks such as sentiment analysis or topic modeling. Removing emojis and emoticons can help improve the accuracy of the analysis.
- Removal of URLs: URLs (Uniform Resource Locators) are web addresses that point to a specific resource on the internet, such as a website, a document, or an image. URLs can add noise to text data, especially in social media or online forums where users often share links. Removing URLs can help clean up the text data and make it more relevant for analysis.
- Removal of HTML tags: HTML (Hypertext Markup Language) is a markup language used to create web pages and format text. HTML tags are used to define the structure and format of the text, such as headings, paragraphs, and links. However, HTML tags can add noise to the text data and may not be relevant for analysis. Removing HTML tags can help clean up the text data and make it more suitable for analysis.
- Tokenization: Split the text into individual words or phrases (tokens) to facilitate analysis.
- Stemming / Lemmatization: Reduce words to their base form to avoid treating similar words as distinct entities. Stemming involves stripping words down to their stem, whereas lemmatization involves converting words to their base form based on their part of speech.

- Spelling correction: Spelling errors are common in text data, and can negatively affect the accuracy of analysis, especially in tasks such as sentiment analysis or topic modeling. Spelling correction is a text preprocessing step that involves correcting spelling errors in the text data using methods such as dictionary lookups or machine learning algorithms. Correcting spelling errors can help improve the accuracy of the analysis and make the text data more suitable for further processing.

- Conversion of emojis/emoticons to words: Emojis and emoticons are graphical symbols used to express emotions, and are commonly used in social media and online communication. Converting emojis and emoticons to words is a text preprocessing step that involves replacing these symbols with their corresponding textual representation, such as replacing the smiling face emoji with the word "smile". This can help improve the accuracy of the analysis, especially in tasks such as sentiment analysis or topic modeling.

- Conversion of chat words to words: Chat words, also known as internet slang, are informal language expressions commonly used in online communication and social media. These words may not be recognized by standard language processing tools and can negatively affect the accuracy of analysis. Conversion of chat words to words is a text preprocessing step that involves replacing these slang words with their standard English counterparts. This can help improve the accuracy of the analysis and make the text data more suitable for further processing.

Python libraries used for text preprocessing

NLTK (Natural Language Toolkit)

NLTK (Natural Language Toolkit) is a popular Python library used for natural language processing. It provides a variety of tools and methods for text preprocessing, which can be used to clean and transform text data for analysis.

One of the most common text preprocessing tasks using NLTK is tokenization, which is the process of splitting text into individual words or phrases (tokens). NLTK provides several methods for tokenization, including word_tokenize, which splits text into words, and sent_tokenize, which splits text into sentences.

NLTK also provides methods for removing stop words, which are common words that do not add much meaning to the text. By removing stop words, the resulting data is more meaningful and easier to analyze. Additionally, NLTK provides methods for stemming and lemmatization, which can be used to reduce words to their base form, making it easier to analyze text that contains variations of the same word.

SpaCy

Spacy is another popular Python library for natural language processing, which provides advanced features for text preprocessing. One of the unique features of Spacy is its efficient processing of text data, which makes it faster and more scalable than NLTK.

Spacy provides a variety of tools for text preprocessing, including tokenization, part-of-speech tagging, named entity recognition, and dependency parsing. These features allow users to extract meaningful information from text data and create structured representations of text.

In addition, Spacy also provides built-in models for multiple languages, making it a versatile tool for analyzing text data in different languages. It also offers an interactive visualization tool, called "displacy," which helps users to understand the structure and meaning of text data visually.

Overall, Spacy is a powerful tool for text preprocessing in NLP, which provides advanced features for extracting meaningful information from text data. Its efficient processing, built-in models for multiple languages, and interactive visualization tools make it a valuable resource for analyzing text data.

Regular expressions

Regular expressions, or regex, is a powerful tool used for text preprocessing in natural language processing. It is a pattern-matching language that allows users to identify and extract specific patterns of text data. Regex can be used for a variety of text preprocessing tasks, such as extracting specific words or phrases, removing special characters, and validating data format.

For example, suppose you want to extract all email addresses from a large dataset. In that case, you can use a regex pattern to identify the pattern of email addresses, such as "[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+.[a-zA-Z]{2,}". Similarly, if you want to remove all special characters from the text, you can use a regex pattern to match all non-alphanumeric characters and replace them with spaces.

Thus, regex is a powerful tool for text preprocessing, which can help to clean and transform text data for analysis in NLP.

Considerations for text preprocessing

Which library to use?

Choosing the right text preprocessing library depends on various factors, such as the task at hand, the format of the text, and the default performance of the tokenizers. It is recommended to start with NLTK or spacy for basic text preprocessing tasks, as they provide a range of tools and methods for tokenization, part-of-speech tagging, named entity recognition, and more. If the performance is not satisfactory, then you may consider using regular expressions or building your own custom tokenizer. Hence, the choice of text preprocessing library depends on the specific requirements of the task at hand and the type of text data being analyzed.

Different languages have different tokenisation methods

Different languages have different rules for sentence structure and word formation, which require different tokenization methods. For example, some languages like Chinese do not use

spaces between words, making it difficult to identify individual words without a specialized tokenizer. Similarly, languages like Japanese and Korean use complex scripts that require special tokenization methods to separate words and sentences.

In this course, we will focus on text preprocessing for the English language, which is the most common language used in NLP research and applications. In English, the most common tokenization methods are sentence tokenization, also known as sentence segmentation, and word tokenization. Sentence tokenization is the process of identifying and separating individual sentences in a document or text corpus. This is typically done using a combination of punctuation and grammatical rules. Word tokenization, on the other hand, involves breaking down a sentence or text corpus into individual words or tokens, which can then be used for further analysis. The most common method for word tokenization is to split the text at spaces, but other methods such as using punctuation and word morphology may also be used in some cases.

Extent of Text preprocessing

Text preprocessing is a critical step in natural language processing, and the extent of preprocessing required depends on various factors. While some preprocessing tasks, such as noise removal and spelling correction, are common across all tasks, the extent of preprocessing required is primarily task-dependent. For instance, tasks such as sentiment analysis may require additional preprocessing steps such as stemming or lemmatization to extract the root form of words, whereas text classification tasks may require more advanced feature extraction techniques such as named entity recognition.

Furthermore, the amount of preprocessing required also depends on the quality of text obtained. Text obtained from sources such as social media or user-generated content may contain more noise, spelling errors, and abbreviations, requiring more extensive preprocessing compared to text obtained from professional sources. While some preprocessing steps are common across all tasks, the extent of preprocessing required can vary significantly depending on the task and the quality of the text data.

[Lab] MS9007 Lesson 2 Text Preprocessing.ipynb

[Lab] MS9007 Lesson 2 Social Media Disaster Tweets.ipynb