**Assignment-1**

1. **Definition and scope of NLP?**

* **Definition of NLP:** Natural language processing (NLP) is a subset of artificial intelligence, [computer science](https://www.coursera.org/articles/what-is-computer-science), and linguistics focused on making human communication, such as speech and text, comprehensible to computers. NLP is used in a wide variety of everyday products and services. Some of the most common ways NLP is used are through voice-activated digital assistants on smartphones, email-scanning programs used to identify spam, and translation apps that decipher foreign languages.
* **Scope of NLP:** NLP encompasses a wide range of techniques to analyze human language. Some of the most common techniques you will likely encounter in the field include:
* **Sentiment analysis:** An NLP technique that analyzes text to identify its sentiments, such as “positive,” “negative,” or “neutral.” Sentiment analysis is commonly used by businesses to better understand customer feedback.
* **Summarization:** An NLP technique that summarizes a longer text, in order to make it more manageable for time-sensitive readers. Some common texts that are summarized include reports and articles.
* **Keyword extraction:** An NLP technique that analyzes a text to identify the most important keywords or phrases. Keyword extraction is commonly used for [search engine optimization (SEO)](https://www.coursera.org/articles/seo-marketing), social media monitoring, and business intelligence purposes.
* **Tokenization:** The process of breaking characters, words, or subwords down into “tokens” that can be analyzed by a program. Tokenization undergirds common NLP tasks like word modeling, vocabulary building, and frequent word occurrence.

1. **Applications and real-world examples of NLP?**

* **Applications and real-world examples of NLP:** Natural Language Processing (NLP) is a field of artificial intelligence that focuses on the interaction between humans and computers using natural language. NLP has a wide range of applications in various industries and real-world scenarios. Here are some examples:

1. **Sentiment Analysis:** NLP is used to determine the sentiment or emotional tone of text data, such as social media posts, product reviews, or customer feedback. Companies can use sentiment analysis to gauge customer opinions and make informed decisions about their products or services.
2. **Chatbots and Virtual Assistants:** NLP powers chatbots and virtual assistants like Siri, Alexa, and Google Assistant. These systems can understand and respond to natural language queries, making them valuable for customer support, information retrieval, and task automation.
3. **Language Translation:** NLP is used for machine translation services like Google Translate. These systems can automatically translate text from one language to another, facilitating communication across language barriers.
4. **Text Summarization:** NLP can automatically generate concise summaries of long texts, which is useful for quickly extracting key information from articles, research papers, or news stories.
5. **Speech Recognition:** NLP technologies enable speech recognition systems like those used in voice assistants and transcription services. They convert spoken language into text, making it possible to interact with devices and transcribe spoken content.
6. **Information Retrieval:** NLP helps search engines like Google understand user queries and retrieve relevant search results. Search engines use techniques like keyword extraction and semantic analysis to improve search accuracy.
7. **Spam Detection:** NLP is employed in email filtering systems to identify and filter out spam messages. It analyzes the content and structure of emails to distinguish between legitimate messages and unsolicited spam.
8. **Linguistic Fundamentals: Regular Expressions, Words, Corpus?**

* **Regular Expressions:** Regular expressions, often abbreviated as "regex" or "regexp," are powerful tools used for pattern matching within text strings. They are a way to describe and search for specific patterns in text data. In simple words, we can say that a regular expression is a set of characters or a pattern that is used to find substrings in a given string. A regular expression (RE) is a language for specifying text search strings. It helps us to match or extract other strings or sets of strings, with the help of a specialized syntax present in a pattern.
* **Usage:** Regular expressions are commonly used for tasks such as text search, text extraction, data validation, and text manipulation. They are used in programming languages like Python, JavaScript, and Perl, as well as in text editors and command-line utilities.
* **Example:** For Example, extracting all hashtags from a tweet, getting email iD or phone numbers, etc from large unstructured text content. The regular expression \d{2}-\d{2}-\d{4} can be used to match dates in the format "dd-mm-yyyy."
* **Corpus:** Definition: A "corpus" (plural: corpora) is a large and structured collection of written or spoken text. It serves as a representative sample of a language or a specific domain of discourse. Corpora are used for linguistic research, language analysis, and training natural language processing models.
* **Usage:** Corpora are employed in various linguistic and NLP applications, including language modeling, sentiment analysis, text classification, and machine translation. Linguists analyze corpora to study language patterns and trends. The Corpus of Contemporary American English (COCA) is a large collection of texts, including books, newspapers, and spoken conversations, used for linguistic research and analysis.
* **Example:**

1. Corpus creation for Marathi language or native languages etc.
2. Corpus creation of foreign language.

* **Words:** In linguistics, a "word" is the smallest meaningful unit of language. Words are typically composed of letters or phonemes and carry specific meanings. They can stand alone or be combined to form sentences.
* **Usage**: Words are the building blocks of language and are used to convey information, express thoughts, and communicate. Understanding the structure and meaning of words is essential for language analysis and natural language processing (NLP).
* **Example:** In the sentence "The quick brown fox jumps over the lazy dog," the words are "The," "quick," "brown," "fox," "jumps," "over," "the," "lazy," and "dog."

1. **Text Normalization: Tokenization and segmentation, Stop word removal?**

* Text normalization is an essential preprocessing step in natural language processing (NLP) that involves transforming text data into a consistent and standardized format. This process helps make text analysis more effective by reducing noise and variation in the data. Text normalization typically includes tokenization and segmentation, as well as the removal of stop words. Let's delve into each of these components with examples:
* **Tokenization and Segmentation:** Tokenization is the process of breaking a text into smaller units, often words or phrases, known as tokens. Segmentation, on the other hand, refers to the splitting of text into sentences or other meaningful segments. Tokenization and segmentation are crucial for various NLP tasks, such as text classification, information retrieval, and machine translation.
* **Example:**

**Input Text:** "Text normalization is an important step in NLP. It involves tokenization and segmentation."

* **Tokenization:**
* **Tokens:** ["Text", "normalization", "is", "an", "important", "step", "in", "NLP", ".", "It", "involves", "tokenization", "and", "segmentation", "."]
* **Segmentation:**
* **Sentences:** ["Text normalization is an important step in NLP.", "It involves tokenization and segmentation."]
* **Stop Word Removal:** Stop words are common words in a language that often occur frequently but do not carry significant meaning on their own. Examples of stop words in English include "the," "and," "of," "in," and "is." Removing stop words is a crucial text normalization step because it reduces the dimensionality of the data and eliminates noise.
* **Example:**
* **Input Text:** "Text normalization is an important step in NLP. It involves tokenization and segmentation."
* **After stop word removal:**
* **Cleaned Text:** "Text normalization important step NLP. involves tokenization segmentation."
* By removing stop words, we have reduced the input text to its essential content, which can be more useful for various NLP tasks. Stop words removal helps in improving the efficiency of algorithms, reducing storage requirements, and enhancing the performance of tasks like text classification and information retrieval.
* Text normalization may also include other steps such as stemming (reducing words to their root form, e.g., "running" to "run") or lemmatization (reducing words to their base or dictionary form, e.g., "better" to "good"). These additional steps depend on the specific requirements of the NLP task.
* Text normalization, which includes tokenization, segmentation, and stop word removal, is a critical preprocessing step in NLP that helps prepare text data for analysis and machine learning tasks by making it more structured and meaningful.

1. **Stemming and lemmatization, Handling capitalization and punctuation, Minimum Edit distance?**
2. **Stemming and lemmatization** are two text normalization techniques used in natural language processing (NLP) to reduce words to their base or root forms. These techniques help in standardizing and simplifying text data, making it easier to analyze, search, and process.

* **Stemming:** Stemming is a heuristic process that aims to remove prefixes and suffixes from words to obtain their root or base form. The resulting word, known as the "stem," may or may not be a valid word in the language. Stemming is typically used to achieve text normalization by reducing inflected or derived words to a common form.
* **Example of stemming:**
* Original Word: "jumping"
* Stem: "jump"
* Common stemming algorithms include the Porter Stemmer and the Snowball Stemmer.
* **Lemmatization:** Lemmatization, unlike stemming, reduces words to their base or dictionary form, known as the "lemma." Lemmatization ensures that the resulting word is a valid word in the language and is considered a more accurate and linguistically informed approach to text normalization.
* **Example of lemmatization:**
* Original Word: "better"
* Lemma: "good"
* Lemmatization algorithms consider the context and part of speech (e.g., noun, verb, adjective) of a word to determine its lemma accurately.
* **Stemming:** Use stemming when speed is crucial, and you can tolerate some loss of accuracy or when you want to reduce words to a common base form for information retrieval or indexing tasks.
* **Lemmatization:** Use lemmatization when you need a higher level of linguistic accuracy, and valid words are critical, such as in applications like sentiment analysis or machine translation.
* Stemming and lemmatization are text normalization techniques in NLP that reduce words to their base forms. Stemming is faster but may not always produce valid words, while lemmatization is more accurate but computationally more intensive. The choice between them depends on the specific requirements of your NLP task.

1. **Handling capitalization and punctuation:** is an essential step in text preprocessing for natural language processing (NLP) tasks. It involves ensuring that text data is in a consistent and standardized format, where capitalization and punctuation marks are either removed or treated in a way that best suits the specific NLP task. Here are the common approaches:
2. **Capitalization Normalization:** Capitalization normalization involves converting all text to a common case, typically lowercase. This is done to ensure that words are treated as the same regardless of their original capitalization. Here's why it's important:

* **Consistency:** Treating all text in lowercase ensures consistency in the data, making it easier to compare and process.
* **Eliminating Case Sensitivity:** It eliminates case sensitivity issues, ensuring that "word" and "Word" are considered the same.
* **Reducing Dimensionality:** By standardizing the case, you reduce the number of unique tokens, which can help improve the efficiency of NLP algorithms.
* **Example: Original Text:** "The quick Brown Fox" **Normalized Text:** "the quick brown fox"

1. **Punctuation Handling:** Punctuation handling involves deciding how to treat punctuation marks in text data. The approach can vary based on the specific NLP task, but there are two common strategies:

* **Punctuation Removal:** In many cases, punctuation marks are removed from the text altogether. This is particularly common when analyzing the semantic content of text and when punctuation does not convey valuable information for the task.
* **Example: Original Text:** "Hello, world! How are you?" **Normalized Text:** "Hello world How are you"
* **Sentence Boundary Detection:** In some cases, you may want to retain certain punctuation marks, such as periods, question marks, or exclamation marks, to detect sentence boundaries. This is crucial for tasks that require understanding the structure of text.**Example:**
* Original Text: "This is a sentence. It has two sentences!" Sentences: ["This is a sentence.", "It has two sentences!"]
* The choice of punctuation handling strategy depends on the specific NLP task you are performing. For example, for sentiment analysis, you may want to retain punctuation for better context, while for topic modeling, you may remove most punctuation to focus on word frequencies. Handling capitalization and punctuation is a crucial text preprocessing step in NLP. It involves normalizing the text to a consistent case (usually lowercase) and deciding how to handle punctuation marks based on the specific requirements of the task. These steps help in making text data more uniform and suitable for various NLP applications.

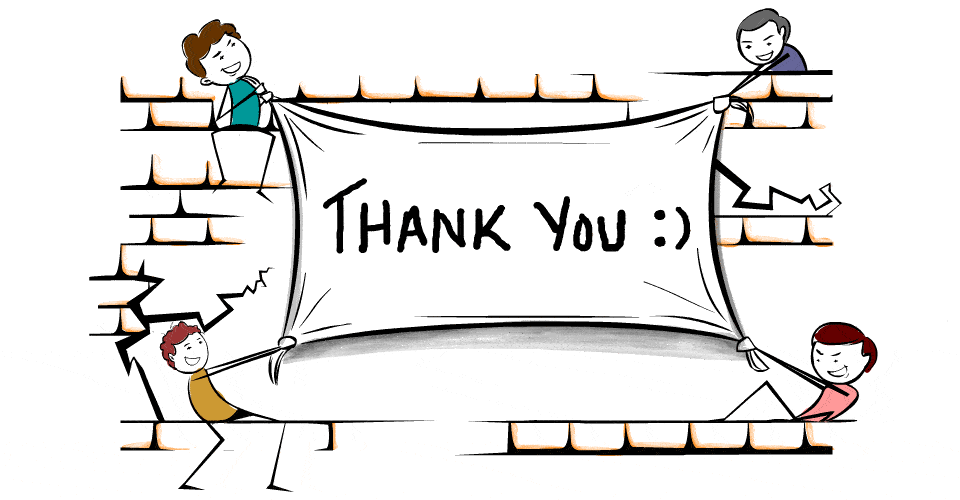
1. **Minimum Edit distance:** Minimum Edit Distance (also known as Levenshtein distance) is a metric used to measure the similarity or dissimilarity between two strings by calculating the minimum number of single-character operations needed to transform one string into the other. These operations include insertion, deletion, and substitution. Let's break down each of these operations in the context of Minimum Edit Distance:

* **Insertion (I):** Insertion refers to adding a character to one of the strings to match it with the other string. The cost associated with insertion is typically set to 1 in the Minimum Edit Distance calculation.
* **Example:**
* String A: "cat"
* String B: "cart"
* Minimum Edit Distance: 1 (Insert 'r' into string A)
* **Deletion (D):** Deletion involves removing a character from one of the strings to match it with the other string. The cost of deletion is also usually set to 1 in the Minimum Edit Distance calculation.
* **Example:**
* String A: "cat"
* String B: "car"
* Minimum Edit Distance: 1 (Delete 't' from string A)
* **Substitution (S):** Substitution is the operation of replacing one character with another to match both strings. Like insertion and deletion, the cost of substitution is typically set to 1.
* **Example:**
* String A: "cat"
* String B: "cut"
* Minimum Edit Distance: 1 (Substitute 'a' with 'u' in string A)
* The Minimum Edit Distance algorithm considers these three basic operations to determine the minimum number of operations required to make two strings identical. It builds a dynamic programming matrix where each cell represents the cost of transforming a substring of one string into a substring of the other string. By considering the costs of insertion, deletion, and substitution, the algorithm finds the minimum path through the matrix, which yields the minimum edit distance between the two strings.

|  |
| --- |
| https://miro.medium.com/v2/resize:fit:358/1*xyoq20suqByW8wzlKe9O-A.png |

* **Here's a simple example of the Minimum Edit Distance calculation for the strings "kitten" and "sitting":** In this example, the Minimum Edit Distance between "kitten" and "sitting" is 3, as it takes three operations (insert 's,' substitute 'k' with 's,' and substitute 'e' with 'i') to transform "kitten" into "sitting." Minimum Edit Distance is a fundamental concept used in various NLP applications, including spell checking, string similarity measurement, and fuzzy string matching. It allows for quantifying the similarity between two strings based on the cost of transforming one into the other using insertion, deletion, and substitution operations.

**(PPT Give you more understanding than PDF)** The material for the PDF has been compiled from various sources such as books, tutorials (offline and online), lecture notes, several resources available on Internet. The information contained in this PDF is for general information and education purpose only. While we endeavor to keep the information up to date and correct, **we make no representation of any kind about the completeness and accuracy of the material.** The information shared through this PDF material should be used for educational purpose only.

****