NATURAL LANGUAGE PROCESSING

* Humans communicate through some form of language either by text or speech.
* To make interactions between computers and humans, computers need to understand natural languages used by humans.
* Natural language processing is all about making computers learn, understand, analyse, manipulate and interpret natural(human) languages.
* NLP stands for Natural Language Processing, which is a part of Computer Science, Human language, and Artificial Intelligence.
* Processing of Natural Language is required when you want an intelligent system like robot to perform as per your instructions.
* Applications that include NLP are chatbots, Email classification and spam filters.
* The input and output of an NLP system can be Speech or Written Text

There are two components of NLP, Natural Language Understanding (NLU)and Natural Language Generation (NLG).

|  |  |
| --- | --- |
| NLU | NLG |
| Natural Language Understanding (NLU) which involves transforming human language into a machine- readable format. | Natural Language Generation (NLG) acts as a translator that converts the computerized data into natural language representation. |
| It helps the machine to understand and analyse human language by extracting the text from large data such as keywords, emotions,  relations, and semantics. | It mainly involves Text planning, Sentence planning, and Text realization. |

The NLU is harder than NLG

|  |  |
| --- | --- |
| Phonology | Study of how sounds are organized and used in a language |
| Morphology | Study of structure and formation of words |
| Morpheme | Smallest unit in a language that carries meaning |
| Syntax | Study of rules and principles that govern the structure of sentence. |
| Semantics | Focus on understanding the meaning of words |
| Disclosure | Studies the interaction between sentence in a larger context |
| World Knowledge | General understanding of a person about the world |

There are five general steps:

|  |  |
| --- | --- |
| Lexical Analysis | The first phase of NLP is the Lexical Analysis.  This phase scans the source code as a stream of characters and converts it into meaningful lexemes. |

|  |  |
| --- | --- |
| Syntactic Analysis | Syntactic Analysis is used to check grammar, word arrangements, and shows the relationship among the words. The sentence such as “The school goes to boy” is rejected  by English syntactic analyser. |
| Semantic analysis | Semantic analysis is concerned with the meaning representation.  It mainly focuses on the literal meaning of words, phrases, and sentences.  The semantic analyser disregards sentence such as “hot ice-  cream”. |
| Discourse Integration | Discourse integration involves understanding sentences that come before and after, which helps make sense of the  overall meaning. |
| Pragmatic Analysis | It involves deriving those aspects of language which require real world knowledge.  Example: "Open the door" is interpreted as a request instead of an order. |

Word Tokenization:

* Word tokenization is the process of breaking a continuous stream of text into individual words.
* In many languages, words are considered the smallest units that can form a complete expression by themselves.

Lexemes:

* Lexemes are sets of linguistic forms that represent a concept, along with its alternative forms.
* They make up the lexicon of a language and can include verbs, nouns, adjectives, and other parts of speech.
* The primary form of a lexeme is called its lemma. Lexemes can be inflected (changing forms) and derived (transformed into related forms).
* Here's a simplified example sentence: "Tokenization splits a sentence into separate words. For instance, 'running' and 'ran' are forms of the lexeme 'run'."

Morphological Typology:

* Morphological typology is a way to group languages based on how they build words using morphemes.
* It divides languages into synthetic (combining morphemes) and analytic (using word order and helper words) types.
* Synthetic languages are further split into agglutinative (add clear pieces for meaning) and fusional (combine many meanings in one ending) languages.
* Synthetic languages, ones that are not analytic, are divided into two categories: agglutinative and fusional languages.
* Agglutinative languages rely primarily on discrete particles (prefixes, suffixes, and infixes) for inflection, ex: inter+national = international, international+ize = internationalize. •
* While fusional languages "fuse" inflectional categories together, often allowing one word ending to contain several categories, such that the original root can be difficult to extract (anybody, newspaper).

Issues and Challenges:

Irregularity:

* - Morphological parsing aims to simplify and understand words by finding patterns.
* - Initial descriptions of language data might not be perfect due to errors or complexity.
* - To improve accuracy and clarity, we need better ways to describe linguistic data.
* - The rules guiding morphological models are crucial for making them work effectively.

Ambiguity:

* Morphological ambiguity happens when words could mean different things without context.
* Words that look alike but mean different things are called homonyms.
* Ambiguity is found in all parts of language processing. Morphological Models:
* Designing morphological models has different ways.
* In computational linguistics, various approaches and frameworks have been developed to solve problems in natural and formal languages.
* Let's explore the main types of computational approaches for understanding word structure.

Dictionary Lookup:

* Morphological parsing connects word forms to their linguistic descriptions.
* Some systems list these associations one by one without generalization.
* Similarly, systems that just look up words in lists, dictionaries, or databases lack depth, unless they're in sync with advanced language models."

Finite State Morphology:

* Finite-State Morphology simplifies language analysis using finite-state transducers.
* The two most popular tools supporting this approach, XFST (Xerox Finite-State Tool) and LexTools.
* Finite-state transducers are computational devices extending the power of finite-state automata.
* They consist of a finite set of nodes connected by directed edges labeled with pairs of input and output symbols.
* In such a network or graph, nodes are also called states, while edges are called arcs.
* The set of possible sequences accepted by the transducer defines the input language; the set of possible sequences emitted by the transducer defines the output language.
* In English, a finite-state transducer could analyze the surface string children into the lexical string child [+plural], for instance, or generate women from woman [+plural].

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Functional Morphology** | **Unification-Based Morphology** |
| Approach | Functional programming and type theory principles | Uses unification principles  from linguistics and computer science |
| Representation | Morphological processes as mathematical functions | Morphological processes  represented using unification rules |
| Types | Linguistic elements organized into distinct types and type classes | Involves matching and merging of linguistic features and constraints |
| Usage | Especially useful for languages with complex,  fusional morphologies | Suited for languages with rich inflectional systems and  irregularities |
| Representation | Linguistic concepts like paradigms and categories  intuitively represented | Can handle intricate relationships between  morphemes and features |
| Implementation | Aimed to be reusable as programming libraries in various language  applications | Can capture complex interactions among morphological elements and  generate different word forms |

Difference between Sentence boundary detection and Topic boundary detection.

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Sentence Boundary Detection (SBD)** | **Topic Boundary Detection (TBD)** |
| Task | Identifies boundaries between sentences in a text | Identifies transitions between different topics or segments in  a text |
| Purpose | Helps in segmenting a text into  individual sentences for further analysis or processing | Helps in identifying shifts in  topics or segments within a text |

|  |  |  |
| --- | --- | --- |
| Type of Boundary | Identifies the end of one sentence and the beginning of the next | Identifies the boundary where the topic or segment changes |
| Criteria for Detection | Typically relies on punctuation marks (e.g., period, exclamation, question mark) | Can be based on keywords, changes in writing style, introduction of new concepts,  etc. |
| NLP Applications | Used for various NLP tasks such  as machine translation, sentiment analysis, summarization, etc. | Used in document clustering,  topic modeling, information retrieval, etc. |
| Example | "I went to the store. It was crowded." | "In the first section, we discussed NLP techniques. In the next section, we'll explore  applications." |

|  |  |  |  |
| --- | --- | --- | --- |
| Feature | GPT-3.5 | GPT-4 | Bard |
| **Type of model** | Generative pre-trained transformer | Generative pre-trained transformer | Generative pre-trained transformer |
| **Parameters** | 175B | 100T | 137B |
| **Training data** | Text and code from the Common Crawl, Wikipedia, books, articles, documents, and content scraped from the open internet | Text and code from a variety of sources, including Common Crawl, Wikipedia, books, articles, documents, and code | Text and code from a variety of sources, including Common Crawl, Wikipedia, books, articles, documents, and code |
| **Performance** | Can generate text, translate languages, write different kinds of creative content, and answer your questions in an informative way | Can generate human-quality text, translate languages, write different kinds of creative content, and answer your questions in a comprehensive and informative way | Can generate different creative text formats of text content, like poems, code, scripts, musical pieces, email, letters, etc. |
| **Availability** | OpenAI API | OpenAI API | Google AI API |

### Syntax:

|  |  |  |
| --- | --- | --- |
| **Part-of-Speech (POS) Tagging** | **Constituency and Dependency Parsing** | **Morphological Analysis** |
| Assigns a grammatical category (noun, verb, adjective, etc.) to each word in a sentence. | Analyzes the structure of sentences by identifying phrases or relationships between words. | Analyzes the internal structure and formation of words, including morphemes and word roots |
| "The cat is sleeping." → "Article Noun Verb Gerund." | "The cat is sleeping." → (NP (Det The) (Noun cat)) (VP (Verb is) (Gerund sleeping)). | "Unhappiness" → "Un- (prefix) + Happy (root) + -ness (suffix)." |

### Semantics:

|  |  |  |
| --- | --- | --- |
| **Named Entity Recognition (NER)** | **Semantic Role Labeling (SRL)** | **Word Sense Disambiguation (WSD)** |
| Identifies and classifies entities (such as persons, organizations, locations) in text. | Identifies the predicate-argument structure in a sentence, assigning roles such as "agent," "patient," etc. | Resolves the ambiguity of word meanings based on context. |
| "Apple Inc. is located in Cupertino." → "ORG is located in LOC." | "She ate the apple." → "She (agent) ate (predicate) the apple (patient)." | "Bank" → "Financial institution" or "Side of a river." |

### Pragmatics:

|  |  |  |
| --- | --- | --- |
| **Coreference Resolution** | **Anaphora Resolution** | **Speech Act Recognition** |
| Determines when two expressions in a text refer to the same entity. | Resolves the interpretation of pronouns and other referring expressions. | Identifies the intended speech act or illocutionary force of a sentence (e.g., statement, question, command). |
| "John picked up the phone. He said hello." → "John (Entity 1) said hello." | "Mary called. She left a message." → "Mary (Entity) left a message." | "Can you pass the salt?" → "Question." |