**POS(Parts-Of-Speech) Tagging in NLP**

One of the core tasks in **Natural Language Processing (NLP)** is **Parts of Speech (PoS) tagging**, which is giving each word in a text a grammatical category, such as nouns, verbs, adjectives, and adverbs. Through improved comprehension of phrase structure and semantics, this technique makes it possible for machines to study and comprehend human language more accurately.

In many NLP applications, including machine translation, sentiment analysis, and information retrieval, PoS tagging is essential. PoS tagging serves as a link between language and machine understanding, enabling the creation of complex language processing systems and serving as the foundation for advanced linguistic analysis.

**What is POS(Parts-Of-Speech) Tagging?**

Parts of Speech tagging is a linguistic activity in Natural Language Processing (NLP) wherein each word in a document is given a particular part of speech (adverb, adjective, verb, etc.) or grammatical category. Through the addition of a layer of syntactic and semantic information to the words, this procedure makes it easier to comprehend the sentence’s structure and meaning.

In NLP applications, POS tagging is useful for machine translation, named entity recognition, and information extraction, among other things. It also works well for clearing out ambiguity in terms with numerous meanings and revealing a sentence’s grammatical structure.

Default tagging is a basic step for the part-of-speech tagging. It is performed using the DefaultTagger class. The DefaultTagger class takes ‘tag’ as a single argument. NN is the tag for a singular noun. DefaultTagger is most useful when it gets to work with most common part-of-speech tag. that’s why a noun tag is recommended.Example of POS Tagging

Consider the sentence: “The quick brown fox jumps over the lazy dog.”

**After performing POS Tagging:**

“The” is tagged as determiner (DT)

“quick” is tagged as adjective (JJ)

“brown” is tagged as adjective (JJ)

“fox” is tagged as noun (NN)

“jumps” is tagged as verb (VBZ)

“over” is tagged as preposition (IN)

“the” is tagged as determiner (DT)

“lazy” is tagged as adjective (JJ)

“dog” is tagged as noun (NN)

By offering insights into the grammatical structure, this tagging aids machines in comprehending not just individual words but also the connections between them inside a phrase. For many NLP applications, like text summarization, sentiment analysis, and machine translation, this kind of data is essential.

**Workflow of POS Tagging in NLP**

The following are the processes in a typical natural language processing (NLP) example of part-of-speech (POS) tagging:

Tokenization: Divide the input text into discrete tokens, which are usually units of words or subwords. The first stage in NLP tasks is tokenization.

Loading Language Models: To utilize a library such as NLTK or SpaCy, be sure to load the relevant language model. These models offer a foundation for comprehending a language’s grammatical structure since they have been trained on a vast amount of linguistic data.

Text Processing: If required, preprocess the text to handle special characters, convert it to lowercase, or eliminate superfluous information. Correct PoS labeling is aided by clear text.

Linguistic Analysis: To determine the text’s grammatical structure, use linguistic analysis. This entails understanding each word’s purpose inside the sentence, including whether it is an adjective, verb, noun, or other.

Part-of-Speech Tagging: To determine the text’s grammatical structure, use linguistic analysis. This entails understanding each word’s purpose inside the sentence, including whether it is an adjective, verb, noun, or other.

Results Analysis: Verify the accuracy and consistency of the PoS tagging findings with the source text. Determine and correct any possible problems or mistagging.

### **Implementation of Parts-of-Speech tagging using NLTK in Python**

#### **Installing packages**

nltk.download('punkt')  
nltk.download('averaged\_perceptron\_tagger')

*# Importing the NLTK library*

*import nltk*

*from nltk.tokenize import word\_tokenize*

*from nltk import pos\_tag*

*# Sample text*

*text = "NLTK is a powerful library for natural language processing."*

*# Performing PoS tagging*

*pos\_tags = pos\_tag(words)*

*# Displaying the PoS tagged result in separate lines*

*print("Original Text:")*

*print(text)*

*print("\nPoS Tagging Result:")*

*for word, pos\_tag in pos\_tags:*

*print(f"{word}: {pos\_tag}")*

### **Implementation of Parts-of-Speech tagging using Spacy in Python**

#### Installing Packages

!pip install spacy  
!python -m spacy download en\_core\_web\_sm

#importing libraries

import spacy

# Load the English language model

nlp = spacy.load("en\_core\_web\_sm")

# Sample text

text = "SpaCy is a popular natural language processing library."

# Process the text with SpaCy

doc = nlp(text)

# Display the PoS tagged result

print("Original Text: ", text)

print("PoS Tagging Result:")

for token in doc:

print(f"{token.text}: {token.pos\_}")

Import the SpaCy library and load the English language model “en\_core\_web\_sm” using spacy.load(“en\_core\_web\_sm”). Process the sample text using the loaded SpaCy model to obtain a Doc object containing linguistic annotations. Print the original text and iterate through the tokens in the processed Doc, displaying each token’s text and its associated part-of-speech tag (token.pos\_).

### **Types of POS Tagging in NLP**

Assigning grammatical categories to words in a text is known as Part-of-Speech (PoS) tagging, and it is an essential aspect of Natural Language Processing (NLP). Different PoS tagging approaches exist, each with a unique methodology. Here are a few typical kinds:

#### 1. **Rule-Based Tagging**

Rule-based part-of-speech (POS) tagging involves assigning words their respective parts of speech using predetermined rules, contrasting with machine learning-based POS tagging that requires training on annotated text corpora. In a rule-based system, POS tags are assigned based on specific word characteristics and contextual cues.

For instance, a rule-based POS tagger could designate the “noun” tag to words ending in “‑tion” or “‑ment,” recognizing common noun-forming suffixes. This approach offers transparency and interpretability, as it doesn’t rely on training data.

Let’s consider an example of how a rule-based part-of-speech (POS) tagger might operate:  
**Rule:** Assign the POS tag “noun” to words ending in “-tion” or “-ment.”

**Text:**“The presentation highlighted the key achievements of the project’s development.”

**Rule based Tags:**

* “The” – Determiner (DET)
* “presentation” – Noun (N)
* “highlighted” – Verb (V)
* “the” – Determiner (DET)
* “key” – Adjective (ADJ)
* “achievements” – Noun (N)
* “of” – Preposition (PREP)
* “the” – Determiner (DET)
* “project’s” – Noun (N)
* “development” – Noun (N)

In this instance, the predetermined rule is followed by the rule-based POS tagger to label words. “Noun” tags are applied to words like “presentation,” “achievements,” and “development” because of the aforementioned restriction. Despite the simplicity of this example, rule-based taggers may handle a broad variety of linguistic patterns by incorporating different rules, which makes the tagging process transparent and comprehensible.

#### 2. **Transformation Based tagging**

Transformation-based tagging (TBT) is a part-of-speech (POS) tagging method that uses a set of rules to change the tags that are applied to words inside a text. In contrast, statistical POS tagging uses trained algorithms to predict tags probabilistically, while rule-based POS tagging assigns tags directly based on predefined rules.

To change word tags in TBT, a set of rules is created depending on contextual information. A rule could, for example, change a verb’s tag to a noun if it comes after a determiner like “the.” The text is systematically subjected to these criteria, and after each transformation, the tags are updated.

When compared to rule-based tagging, TBT can provide higher accuracy, especially when dealing with complex grammatical structures. To attain ideal performance, nevertheless, it might require a large rule set and additional computer power.

Consider the transformation rule: Change the tag of a verb to a noun if it follows a determiner like “the.”

**Text:** “The cat chased the mouse”.

**Initial Tags:**

* “The” – Determiner (DET)
* “cat” – Noun (N)
* “chased” – Verb (V)
* “the” – Determiner (DET)
* “mouse” – Noun (N)

**Transformation rule applied:**

Change the tag of “chased” from Verb (V) to Noun (N) because it follows the determiner “the.”

**Updated tags:**

* “The” – Determiner (DET)
* “cat” – Noun (N)
* “chased” – Noun (N)
* “the” – Determiner (DET)
* “mouse” – Noun (N)

In this instance, the tag “chased” was changed from a verb to a noun by the TBT system using a transformation rule based on the contextual pattern. The tagging is updated iteratively and the rules are applied sequentially. Although this example is simple, given a well-defined set of transformation rules, TBT systems can handle more complex grammatical patterns.

#### **3. Statistical POS Tagging**

Utilizing probabilistic models, statistical part-of-speech (POS) tagging is a computer linguistics technique that places grammatical categories on words inside a text. If rule-based tagging uses massive annotated corpora to train its algorithms, statistical tagging uses machine learning.

In order to capture the statistical linkages present in language, these algorithms learn the probability distribution of word-tag sequences. CRFs (conditional random fields) and Hidden Markov Model (HMMs) are popular models for statistical point-of-sale classification. The algorithm estimates the chance of observing a specific tag given the current word and its context by learning from labeled samples during training.

The most likely tags for text that hasn’t been seen are then predicted using the trained model. Statistical POS tagging works especially well for languages with complicated grammatical structures because it is exceptionally good at handling linguistic ambiguity and catching subtle language trends.

* **Hidden Markov Model POS tagging:**Hidden Markov Models (HMMs) serve as a statistical framework for part-of-speech (POS) tagging in natural language processing (NLP). In HMM-based POS tagging, the model undergoes training on a sizable annotated text corpus to discern patterns in various parts of speech. Leveraging this training, the model predicts the POS tag for a given word based on the probabilities associated with different tags within its context.  
  Comprising states for potential POS tags and transitions between them, the HMM-based POS tagger learns transition probabilities and word-emission probabilities during training. To tag new text, the model, employing the Viterbi algorithm, calculates the most probable sequence of POS tags based on the learned probabilities.  
  Widely applied in NLP, HMMs excel at modeling intricate sequential data, yet their performance may hinge on the quality and quantity of annotated training data.

### **Advantages of POS Tagging**

There are several advantages of Parts-Of-Speech (POS) Tagging including:

* **Text Simplification:**Breaking complex sentences down into their constituent parts makes the material easier to understand and easier to simplify.
* **Information Retrieval:**Information retrieval systems are enhanced by point-of-sale (POS) tagging, which allows for more precise indexing and search based on grammatical categories.
* **Named Entity Recognition:**POS tagging helps to identify entities such as names, locations, and organizations inside text and is a precondition for named entity identification.
* **Syntactic Parsing:**It facilitates syntactic parsing, which helps with phrase structure analysis and word link identification.

### **Disadvantages of POS Tagging**

Some common disadvantages in part-of-speech (POS) tagging include:

* **Ambiguity:**The inherent ambiguity of language makes POS tagging difficult since words can signify different things depending on the context, which can result in misunderstandings.
* **Idiomatic Expressions:**Slang, colloquialisms, and idiomatic phrases can be problematic for POS tagging systems since they don’t always follow formal grammar standards.
* **Out-of-Vocabulary Words:**Out-of-vocabulary words (words not included in the training corpus) can be difficult to handle since the model might have trouble assigning the correct POS tags.
* **Domain Dependence:**For best results, POS tagging models trained on a single domain should have a lot of domain-specific training data because they might not generalize well to other domains.

## Introduction to POS Tagging

Part-of-speech (POS) tagging is a process in [natural language processing](https://www.shiksha.com/online-courses/articles/introduction-to-natural-language-processing/) (NLP) where each word in a text is labeled with its corresponding part of speech. This can include nouns, verbs, adjectives, and other grammatical categories.

POS tagging is useful for a variety of NLP tasks, such as information extraction, named entity recognition, and machine translation. It can also be used to identify the grammatical structure of a sentence and to disambiguate words that have multiple meanings.

POS tagging is typically performed using machine learning algorithms, which are trained on a large annotated corpus of text. The algorithm learns to predict the correct POS tag for a given word based on the context in which it appears.

There are various POS tagging schemes that have been developed, each with its own set of tags and rules. Some common POS tagging schemes include the [Penn Treebank tagset](https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html) and the [Universal Dependencies tagset](https://universaldependencies.org/u/pos/).

Let’s take an example,

Text: “The cat sat on the mat.”

POS tags:

* The: determiner
* cat: noun
* sat: verb
* on: preposition
* the: determiner
* mat: noun

In this example, each word in the sentence has been labeled with its corresponding part of speech. The determiner “the” is used to identify specific nouns, while the noun “cat” refers to a specific animal. The verb “sat” describes an action, and the preposition “on” describes the relationship between the cat and the mat.

POS tagging is a useful tool in natural language processing (NLP) as it allows algorithms to understand the grammatical structure of a sentence and to disambiguate words that have multiple meanings. It is typically performed using machine learning algorithms that are trained on a large annotated corpus of text.

Identifying part of speech of word is not just mapping words to their respective POS tags. Same word might have different part of speech tag based on different context. Thus it is not possible to have common mapping for parts of speech tags.

When you have a huge corpus manually finding different part-of-speech for each word is a scalable solution. As tagging itself might take days. This is why we rely on tool-based POS tagging.

But why are we tagging these words with their parts of speech?

## Use of Parts of Speech Tagging in NLP

There are several reasons why we might tag words with their parts of speech (POS) in natural language processing (NLP):

* **To understand the grammatical structure of a sentence:**By labeling each word with its POS, we can better understand the syntax and structure of a sentence. This is useful for tasks such as machine translation and information extraction, where it is important to know how words relate to each other in the sentence.
* **To disambiguate words with multiple meanings:**Some words, such as “bank,” can have multiple meanings depending on the context in which they are used. By labeling each word with its POS, we can disambiguate these words and better understand their intended meaning.
* **To improve the accuracy of NLP tasks:** POS tagging can help improve the performance of various NLP tasks, such as named entity recognition and text classification. By providing additional context and information about the words in a text, we can build more accurate and sophisticated algorithms.
* **To facilitate research in linguistics:**POS tagging can also be used to study the patterns and characteristics of language use and to gain insights into the structure and function of different parts of speech.

## Steps Involved in the POS tagging

Here are the steps involved in a typical example of part-of-speech (POS) tagging in natural language processing (NLP):

* **Collect a dataset of annotated text:**This dataset will be used to train and test the POS tagger. The text should be annotated with the correct POS tags for each word.
* **Preprocess the text:** This may include tasks such as tokenization (splitting the text into individual words), lowercasing, and removing punctuation.
* **Divide the dataset into training and testing sets:** The training set will be used to train the POS tagger, and the testing set will be used to evaluate its performance.
* **Train the POS tagger:**This may involve building a statistical model, such as a hidden Markov model (HMM), or defining a set of rules for a rule-based or transformation-based tagger. The model or rules will be trained on the annotated text in the training set.
* **Test the POS tagger:** Use the trained model or rules to predict the POS tags of the words in the testing set. Compare the predicted tags to the true tags and calculate metrics such as precision and recall to evaluate the performance of the tagger.
* **Fine-tune the POS tagger:**If the performance of the tagger is not satisfactory, adjust the model or rules and repeat the training and testing process until the desired level of accuracy is achieved.
* **Use the POS tagger:**Once the tagger is trained and tested, it can be used to perform POS tagging on new, unseen text. This may involve preprocessing the text and inputting it into the trained model or applying the rules to the text. The output will be the predicted POS tags for each word in the text.

## Application of POS Tagging

There are several real-life applications of part-of-speech (POS) tagging in natural language processing (NLP):

* **Information extraction:**POS tagging can be used to identify specific types of information in a text, such as names, locations, and organizations. This is useful for tasks such as extracting data from news articles or building knowledge bases for artificial intelligence systems.
* **Named entity recognition:** POS tagging can be used to identify and classify named entities in a text, such as people, places, and organizations. This is useful for tasks such as building customer profiles or identifying key figures in a news story.
* **Text classification:**POS tagging can be used to help classify texts into different categories, such as spam emails or sentiment analysis. By analyzing the POS tags of the words in a text, algorithms can better understand the content and tone of the text.
* **Machine translation:** POS tagging can be used to help translate texts from one language to another by identifying the grammatical structure and relationships between words in the source language and mapping them to the target language.
* **Natural language generation:**POS tagging can be used to generate natural-sounding text by selecting appropriate words and constructing grammatically correct sentences. This is useful for tasks such as chatbots and virtual assistants.

## Types of POS Tagging in NLP

### Rule Based POS Tagging

Rule-based part-of-speech (POS) tagging is a method of labeling words with their corresponding parts of speech using a set of pre-defined rules. This is in contrast to machine learning-based POS tagging, which relies on training a model on a large annotated corpus of text.

In a rule-based POS tagging system, words are assigned POS tags based on their characteristics and the context in which they appear. For example, a rule-based POS tagger might assign the tag “noun” to any word that ends in “-tion” or “-ment,” as these suffixes are often used to form nouns.

Rule-based POS taggers can be relatively simple to implement and are often used as a starting point for more complex machine learning-based taggers. However, they can be less accurate and less efficient than machine learning-based taggers, especially for tasks with large or complex datasets.

Here is an example of how a rule-based POS tagger might work:

* Define a set of rules for assigning POS tags to words. For example:
* If the word ends in “-tion,” assign the tag “noun.”
* If the word ends in “-ment,” assign the tag “noun.”
* If the word is all uppercase, assign the tag “proper noun.”
* If the word is a verb ending in “-ing,” assign the tag “verb.”
* Iterate through the words in the text and apply the rules to each word in turn. For example:
* “Nation” would be tagged as “noun” based on the first rule.
* “Investment” would be tagged as “noun” based on the second rule.
* “UNITED” would be tagged as “proper noun” based on the third rule.
* “Running” would be tagged as “verb” based on the fourth rule.
* Output the POS tags for each word in the text.

This is a very basic example of a rule-based POS tagger, and more complex systems can include additional rules and logic to handle more varied and nuanced text.

### Statistical POS Tagging

Statistical part-of-speech (POS) tagging is a method of labeling words with their corresponding parts of speech using statistical techniques. This is in contrast to rule-based POS tagging, which relies on pre-defined rules, and to unsupervised learning-based POS tagging, which does not use any annotated training data.

In statistical POS tagging, a model is trained on a large annotated corpus of text to learn the patterns and characteristics of different parts of speech. The model uses this training data to predict the POS tag of a given word based on the context in which it appears and the probability of different POS tags occurring in that context.

Statistical POS taggers can be more accurate and efficient than rule-based taggers, especially for tasks with large or complex datasets. However, they require a large amount of annotated training data and can be computationally intensive to train.

Here is an example of how a statistical POS tagger might work:

* Collect a large annotated corpus of text and divide it into training and testing sets.
* Train a statistical model on the training data, using techniques such as maximum likelihood estimation or hidden Markov models.
* Use the trained model to predict the POS tags of the words in the testing data.
* Evaluate the performance of the model by comparing the predicted tags to the true tags in the testing data and calculating metrics such as precision and recall.
* Fine-tune the model and repeat the process until the desired level of accuracy is achieved.
* Use the trained model to perform POS tagging on new, unseen text.

There are various statistical techniques that can be used for POS tagging, and the choice of technique will depend on the specific characteristics of the dataset and the desired level of accuracy.

### Transformation-based tagging (TBT)

Transformation-based tagging (TBT) is a method of part-of-speech (POS) tagging that uses a series of rules to transform the tags of words in a text. This is in contrast to rule-based POS tagging, which assigns tags to words based on pre-defined rules, and to statistical POS tagging, which relies on a trained model to predict tags based on probability.

In TBT, a set of rules is defined to transform the tags of words in a text based on the context in which they appear. For example, a rule might change the tag of a verb to a noun if it appears after a determiner such as “the.” The rules are applied to the text in a specific order, and the tags are updated after each transformation.

TBT can be more accurate than rule-based tagging, especially for tasks with complex grammatical structures. However, it can be more computationally intensive and requires a larger set of rules to achieve good performance.

Here is an example of how a TBT system might work:

* Define a set of rules for transforming the tags of words in the text. For example:
* If the word is a verb and appears after a determiner, change the tag to “noun.”
* If the word is a noun and appears after an adjective, change the tag to “adjective.”
* Iterate through the words in the text and apply the rules in a specific order. For example:
* In the sentence “The cat sat on the mat,” the word “sat” would be changed from a verb to a noun based on the first rule.
* In the sentence “The red cat sat on the mat,” the word “red” would be changed from an adjective to a noun based on the second rule.
* Output the transformed tags for each word in the text.

This is a very basic example of a TBT system, and more complex systems can include additional rules and logic to handle more varied and nuanced text.

### Hidden Markov Model POS tagging

Hidden Markov models (HMMs) are a type of statistical model that can be used for part-of-speech (POS) tagging in natural language processing (NLP). In an HMM-based POS tagger, a model is trained on a large annotated corpus of text to learn the patterns and characteristics of different parts of speech. The model uses this training data to predict the POS tag of a given word based on the probability of different tags occurring in the context of the word.

An HMM-based POS tagger consists of a set of states, each corresponding to a possible POS tag, and a set of transitions between the states. The model is trained on the training data to learn the probabilities of transitioning from one state to another and the probabilities of observing different words given a particular state.

To perform POS tagging on a new text using an HMM-based tagger, the model uses the probabilities learned during training to compute the most likely sequence of POS tags for the words in the text. This is typically done using the Viterbi algorithm, which calculates the probability of each possible sequence of tags and selects the most likely one.

HMMs are widely used for POS tagging and other tasks in NLP due to their ability to model complex sequential data and their efficiency in computation. However, they can be sensitive to the quality of the training data and may require a large amount of annotated data to achieve good performance.

## Challenges in POS Tagging

Some common challenges in part-of-speech (POS) tagging include:

* **Ambiguity:** Some words can have multiple POS tags depending on the context in which they appear, making it difficult to determine their correct tag. For example, the word “bass” can be a noun (a type of fish) or an adjective (having a low frequency or pitch).
* **Out-of-vocabulary (OOV) words:** Words that are not present in the training data of a POS tagger can be difficult to tag accurately, especially if they are rare or specific to a particular domain.
* **Complex grammatical structures:**Languages with complex grammatical structures, such as languages with many inflections or free word order, can be more challenging to tag accurately.
* **Lack of annotated training data:** Some languages or domains may have limited annotated training data, making it difficult to train a high-performing POS tagger.
* **Inconsistencies in annotated data:** Annotated data can sometimes contain errors or inconsistencies, which can negatively impact the performance of a POS tagger.

## Conclusion

Part-of-speech (POS) tagging is a crucial step in natural language processing (NLP), as it allows algorithms to understand the grammatical structure and meaning of a text. There are several methods for performing POS tagging, including rule-based, statistical, transformation-based, and hidden Markov model (HMM) tagging.

Rule-based POS tagging relies on a set of pre-defined rules to assign tags to words, while statistical POS tagging uses a trained model to predict tags based on probability. Transformation-based tagging (TBT) uses a series of rules to transform the tags of words based on context, and HMM tagging uses an HMM to learn the patterns and characteristics of different parts of speech.