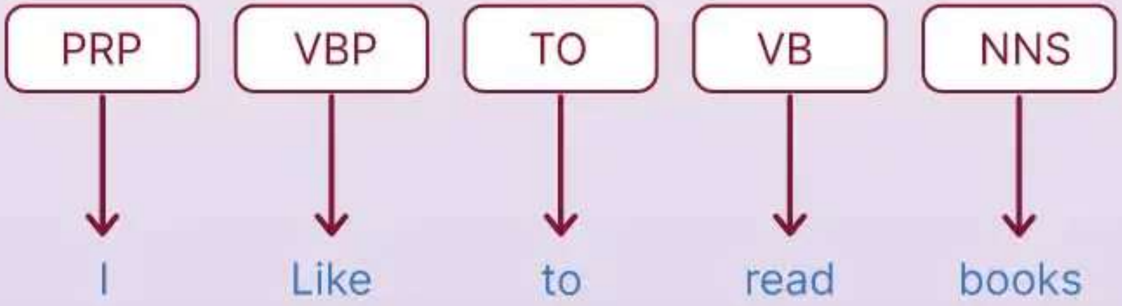
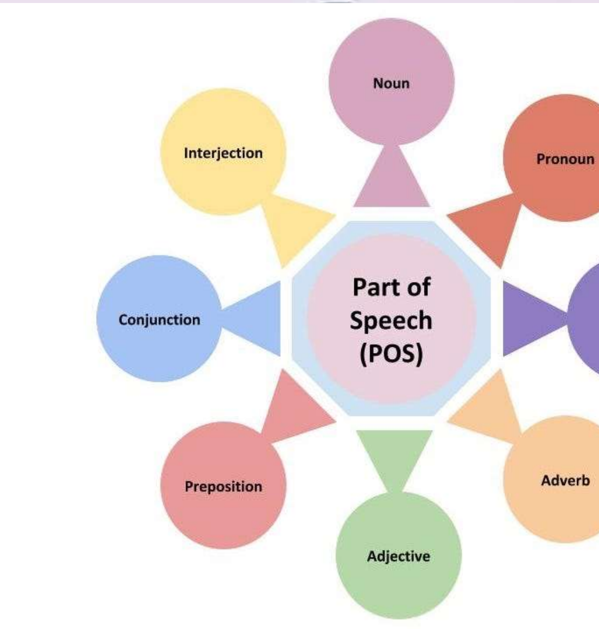
POS Tagging

In NLP



PRP VBP TO VB NNS

I Like to read books



Prepared by:
Nushrat Jahan Ria
Lecturer, Dept of CSE, DIU

What is Part of Speech (POS)

Words can be divided into classes that behave similarly.

Traditionally eight parts of speech in English: noun, verb, pronoun, preposition, adverb, conjunction, adjective and article.

Most recently larger sets have been used: e.g. **Penn Treebank** (45 tags), **Susanne** (353 tags).

Why POS?

POS tell us a lot about a word (and the words near it).

E.g, adjectives often followed by nouns personal pronouns often followed by verbs possessive pronouns by nouns.

Pronunciations depends on POS, e.g. object (first syllable NN, second syllable VM), content, discount.

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.

Categories of POS

Open and closed classes

Closed classes have a fixed membership of words: determiners, pronouns, prepositions.

Closed class words are usually function word: frequently occurring, grammatically important, often short (e.g. of, ot, the, in)

Open classes: nouns, verbs, adjectives and adverbs (allow new addition of word)

Open Class

Nouns:

Proper nouns (Scotland, BBC)

Common nouns

Count nouns (goat, glass)

Mass nouns (snow, pacifism)

Verbs:

Actions and processes (run, hope)

Also auxiliary verbs (is, are, am, will, can)

Adjectives:

Properties and qualities (age, colour, verb)

Adverbs:

Modify verbs or verb phrases or other adverbs- Unfortunately John walked home extremely slowly yesterday.

Sentential adverb: unfortunately

Manner adverb: extremely, slowly

Time adverb: yesterday

Closed class

Prepositions: on, under, over, to, with, by

Determiners: the, a, an, some

Pronouns: she, you, I, who

Conjunctions: and, but, or, as, when, if

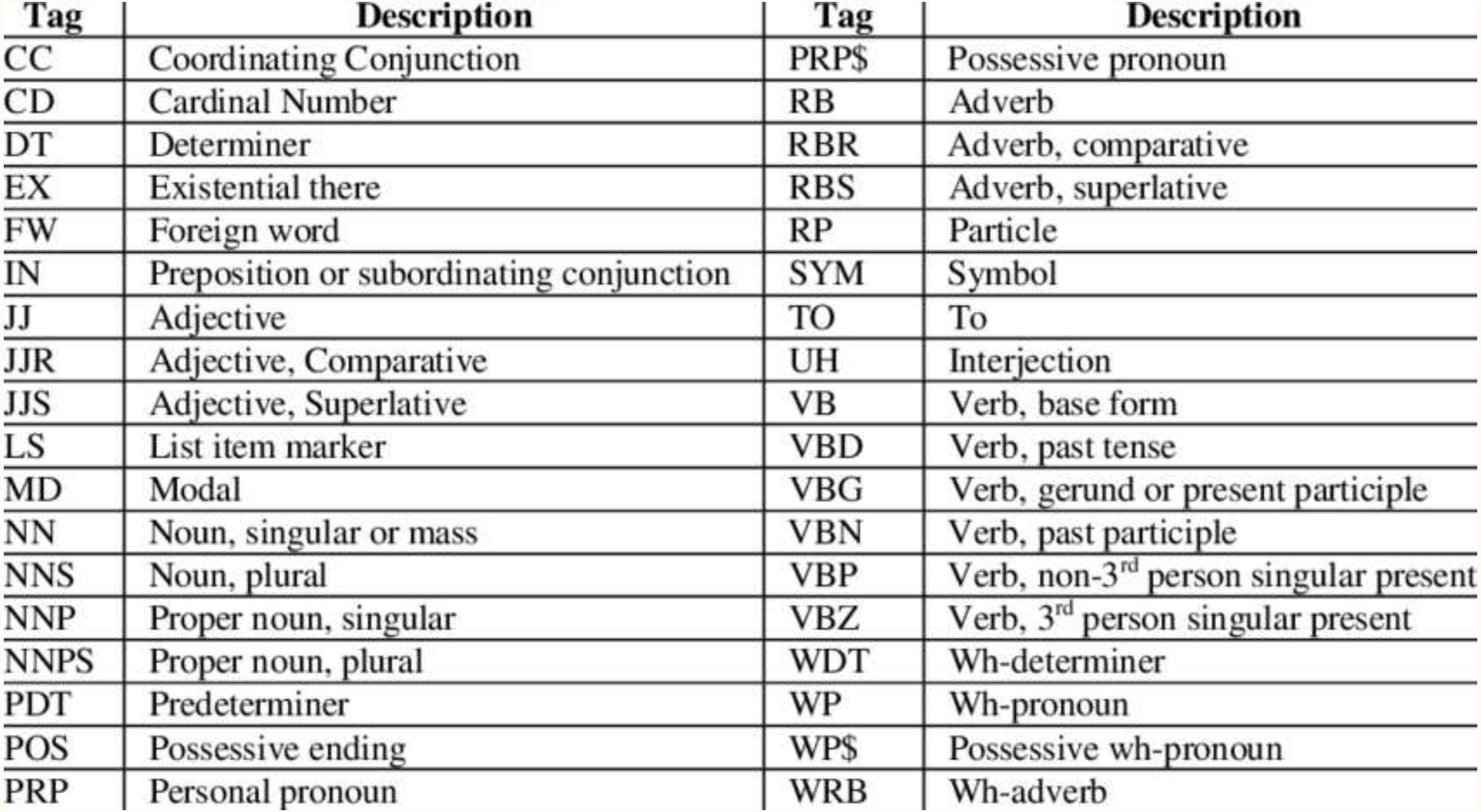
Auxiliary verbs: can, may, are

Tagset

A tagset is a list of part-of-speech tags (POS tags for short), i.e. labels used to indicate the part of speech and sometimes also other grammatical categories (case, tense etc.) of each token in a text corpus

- **Penn Treebank** : An important tagset for English is the 45-tag Penn Treebank tagset which has been used to label many corpora. In such labelings, parts-of-speech are generally represented by placing the tag after each word.
- **Susanne**: A special tagset used in Susanne corpus comprises 353 distinct word tags.
- **British National Corpus Tagset**
- **Brown Corpus Tagset**

Penn tagset



Tag	Description	Tag	Description
CC	Coordinating Conjunction	PRP$	Possessive pronoun
CD	Cardinal Number	RB	Adverb
DT	Determiner	RBR	Adverb, comparative
EX	Existential there	RBS	Adverb, superlative
FW	Foreign word	RP	Particle
IN	Preposition or subordinating conjunction	SYM	Symbol
JJ	Adjective	TO	To
JJR	Adjective, Comparative	UH	Interjection
JJS	Adjective, Superlative	VB	Verb, base form
LS	List item marker	VBD	Verb, past tense
MD	Modal	VBG	Verb, gerund or present participle
NN	Noun, singular or mass	VBN	Verb, past participle
NNS	Noun, plural	VBP	Verb, non-3rd person singular present
NNP	Proper noun, singular	VBZ	Verb, 3rd person singular present
NNPS	Proper noun, plural	WDT	Wh-determiner
PDT	Predeterminer	WP	Wh-pronoun
POS	Possessive ending	WP$	Possessive wh-pronoun
PRP	Personal pronoun	WRB	Wh-adverb

Natural Language Toolkit (NLTK)

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to **over 50 corpora and lexical resources** such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries.

Steps Involved in the POS tagging

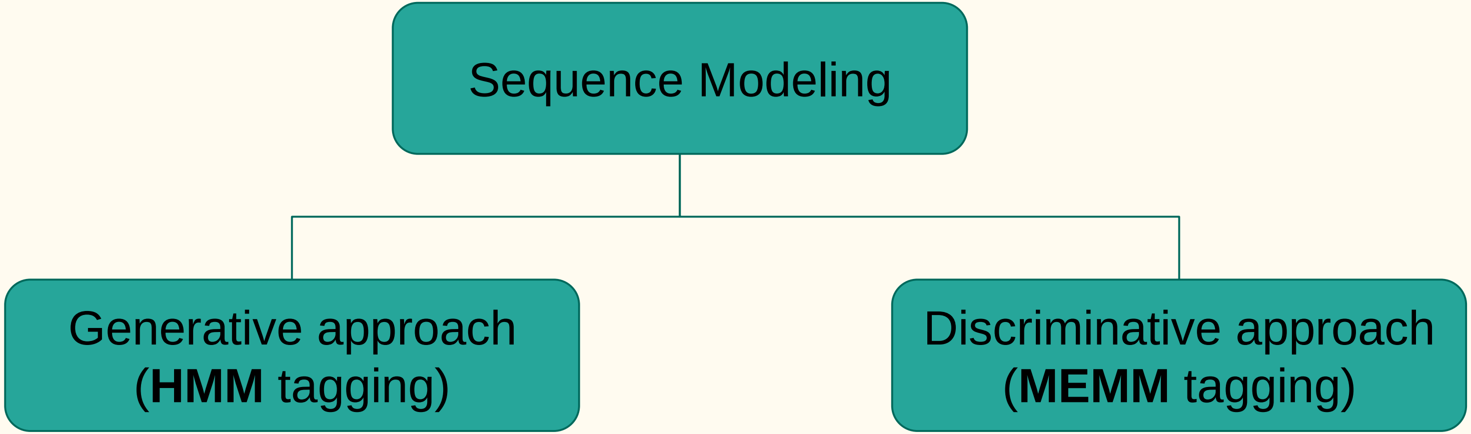
- **Collect a dataset of annotated text:** This dataset will be used to train and test the POS tagger. The text should be annotated with the correct POS tags for each word.
- **Preprocess the text:** This may include tasks such as tokenization (splitting the text into individual words), lowercasing, 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.

Sequence Modeling

Sequence models are the **machine learning models** that input or output **sequences of data**. Sequential data includes text streams, audio clips, video clips, time-series data and etc. Recurrent Neural Networks (RNNs) is a popular algorithm used in sequence models. Applications of Sequence Models:

1. Speech recognition: In speech recognition, an audio clip is given as an input and then the model has to generate its text transcript. Here both the input and output are sequences of data.
2. Sentiment Classification: In sentiment classification opinions expressed in a piece of text is categorized. Here the input is a sequence of words.
3. Video Activity Recognition: In video activity recognition, the model needs to identify the activity in a video clip. A video clip is a sequence of video frames, therefore in case of video activity recognition input is a sequence of data.

Types of Sequence Modeling in POS tagging



Sequence Modeling

Generative approach (**HMM** tagging)

Discriminative approach (**MEMM** tagging)

Generative approach
(Hidden Markov Model or HMM tagging)

HMM (Hidden Markov Model) is a Stochastic technique for POS tagging. Hidden Markov models are known for their applications to **reinforcement learning** and temporal **pattern recognition** such as speech, handwriting, gesture recognition, musical score following, partial discharges, and bioinformatics.

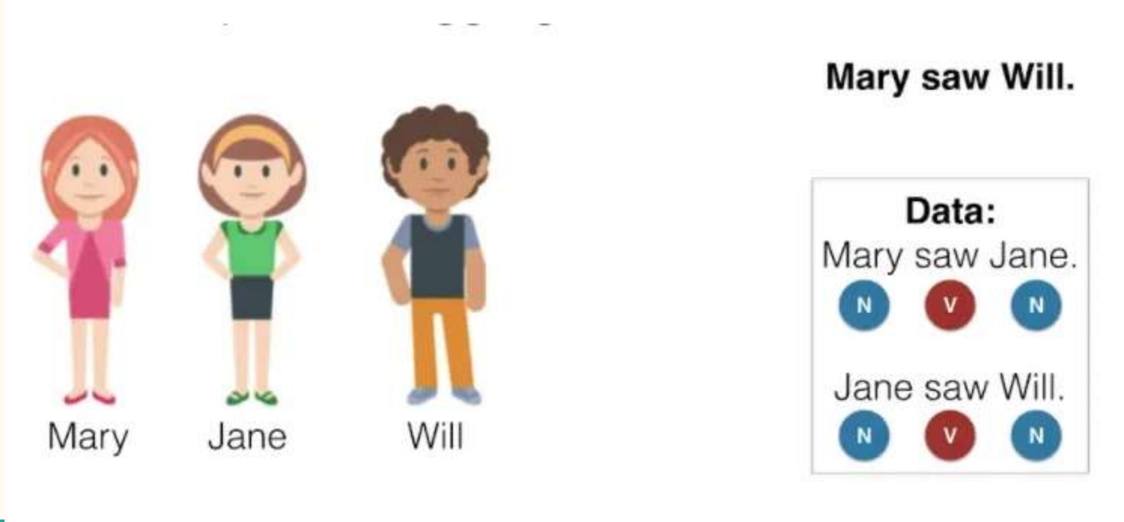
This sequence model or sequence classifier is a model whose job is to assign a label or class to each unit in a sequence, thus mapping a sequence of observations to a sequence of labels. An HMM is a probabilistic sequence model: given a sequence of units (words, letters, morphemes, sentences, whatever), it computes a probability distribution over possible sequences of labels and chooses the best label sequence.

Practice Example:

Let’s take the sentence “Rahul will eat food” where Rahul is a noun, will is a modal, eat is a verb and food is also a noun, so the probability for a word to be in a particular class of part of speech is called the Emission probability.

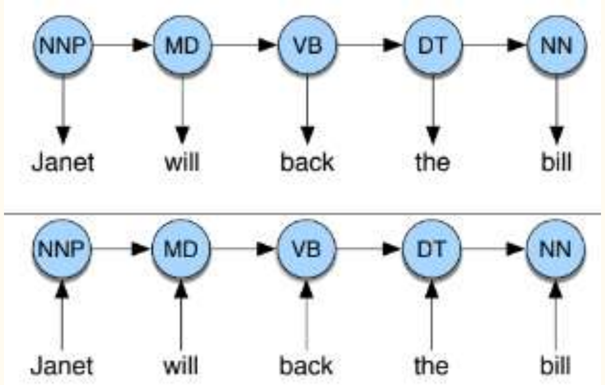
Let’s take a look at how you can calculate these two probabilities for a set of sentences:

- Mary Jane can see will
- The spot will see Mary
- Will Jane spot Mary?
- Mary will pat Spot



Discriminative approach (Maximum Entropy Markov Models or MEMM tagging)

While an HMM can achieve very high accuracy, we saw that it requires a number of architectural innovations to deal with unknown words, backoff, suffixes, and so on. It would be so much easier if we could add arbitrary features directly into the model in a clean way, but that's hard for generative models like HMMs. Logistic regression isn't a sequence model; it assigns a class to a single observation. However, we could turn logistic regression into a discriminative sequence model simply by running it on successive words, using the class assigned to the prior word as a feature in the classification of the next word. When we apply logistic regression in this way, it's called the maximum entropy Markov model or MEMM.



Types of POS Tagging in NLP Cont….

1. **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 predefined 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.

Types of POS Tagging in NLP Cont…

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.

Types of POS Tagging in NLP Cont...

2. 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 predefined 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.

Types of POS Tagging in NLP Cont...

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.

Types of POS Tagging in NLP Cont…

3. 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.

Types of POS Tagging in NLP Cont…

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.

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.