**1.Extract Sample document and apply following document preprocessing methods: Tokenization, POS Tagging, stop words removal, Stemming and Lemmatization.**

**2.Create representation of document by calculating Term Frequency and Inverse Document Frequency.**

# Sample Sentences



Sample\_Sentences **=** "I played the play playfully as the players were playing in the play with playfullness"

Tokenization



[nltk\_data] Downloading package averaged\_perceptron\_tagger to

[nltk\_data] C:\Users\aades\AppData\Roaming\nltk\_data...

[nltk\_data] Package averaged\_perceptron\_tagger is already up-to- [nltk\_data] date!

**import** nltk nltk**.**download('averaged\_perceptron\_tagger')

True



**from** nltk.tokenize **import** sent\_tokenize



sentences **=** sent\_tokenize(Sample\_Sentences)



['I played the play playfully as the players were playing in the play with playfullness']

sentences



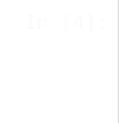
**from** nltk **import** word\_tokenize, sent\_tokenize sentences **=** sent\_tokenize(Sample\_Sentences)

tokenized\_words **=** [word\_tokenize(sentence) **for** sentence **in** sentences] print('sentences words: ',sentences )

print('Tokenized words:', tokenized\_words)

sentences words: ['I played the play playfully as the players were playing in the play with playfullness'] Tokenized words: [['I', 'played', 'the', 'play', 'playfully', 'as', 'the', 'players', 'were', 'playing', 'in', ' the', 'play', 'with', 'playfullness']]

# POS Tagging



**from** nltk **import** pos\_tag

tokenized\_words **=** word\_tokenize(Sample\_Sentences) pos\_tags **=** pos\_tag(tokenized\_words) print("Tagging Parts of Speech:", pos\_tags)

Tagging Parts of Speech: [('I', 'PRP'), ('played', 'VBD'), ('the', 'DT'), ('play', 'NN'), ('playfully', 'RB'), (

'as', 'IN'), ('the', 'DT'), ('players', 'NNS'), ('were', 'VBD'), ('playing', 'VBG'), ('in', 'IN'), ('the', 'DT')

, ('play', 'NN'), ('with', 'IN'), ('playfullness', 'NN')]

# Stop-Words Removal



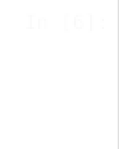
**from** nltk.corpus **import** stopwords

stop\_words **=** set(stopwords**.**words('english'))

filtered\_tokens **=** [word **for** word **in** tokenized\_words **if** word**.**lower() **not in** stop\_words] print("Filtered Tokens after Stop Words Removal:", filtered\_tokens)

Filtered Tokens after Stop Words Removal: ['played', 'play', 'playfully', 'players', 'playing', 'play', 'playful lness']

# Stemming



**from** nltk.stem **import** PorterStemmer

stemmer **=** PorterStemmer()

stemmed\_tokens **=** [stemmer**.**stem(word) **for** word **in** filtered\_tokens] print("Stemmed Tokens:", stemmed\_tokens)

Stemmed Tokens: ['play', 'play', 'play', 'player', 'play', 'play', 'playful']

# Lemmatization



**from** nltk.stem **import** WordNetLemmatizer

lemmatizer **=** WordNetLemmatizer()

lemmatized\_tokens **=** [lemmatizer**.**lemmatize(word) **for** word **in** filtered\_tokens] print("Lemmatized Tokens:", lemmatized\_tokens)

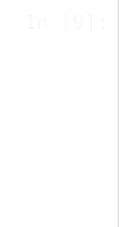
Lemmatized Tokens: ['played', 'play', 'playfully', 'player', 'playing', 'play', 'playfullness']

# 2) Create representation of document by calculating Term Frequency and Inverse Document Frequency.



preprocessed\_text **=** ' '**.**join(lemmatized\_tokens)

Preprocessed Text: played play playfully player playing play playfullness TF-IDF Representation:



**from** sklearn.feature\_extraction.text **import** TfidfVectorizer

tfidf\_vectorizer **=** TfidfVectorizer()

tfidf\_representation **=** tfidf\_vectorizer**.**fit\_transform([preprocessed\_text])

print("Preprocessed Text:", preprocessed\_text) print("\nTF-IDF Representation:") print(tfidf\_representation**.**toarray())

[[0.66666667 0.33333333 0.33333333 0.33333333 0.33333333 0.33333333]]