Nihar Ranjan Murudi - 210273 - CSE 2

```
#Input text (combination of simple sentences, complex sentences, and paragraphs.)
input_texts = [
    "The sun shines brightly in the sky.",
    "Although he had studied hard, he failed the exam because the questions were too difficult.",
    "John woke up early in the morning. After a quick breakfast, he decided to go for a run in the park. As he jogged along the path, he not
"The software development team implemented the new feature using agile methodology. Despite encountering some initial challenges, they w
"Sarah walked to the park with her dog Max. The sun was shining brightly, and the birds were chirping in the trees. Max wagged his tail ]
```

```
#Tokenization
import nltk
from nltk.tokenize import word_tokenize
nltk.download('punkt')
tokenized_texts = [word_tokenize(text) for text in input_texts]
for idx, tokens in enumerate(tokenized_texts, 1):
    print(f"Text {idx} Tokens:", tokens)
```

```
 Text 1 Tokens: ['The', 'sun', 'shines', 'brightly', 'in', 'the', 'sky', '.']
 Text 2 Tokens: ['Although', 'he', 'had', 'studied', 'hard', ',', 'he', 'failed', 'the', 'exam', 'because', 'the', 'questions', 'were',
 Text 3 Tokens: ['John', 'woke', 'up', 'early', 'in', 'the', 'morning', '.', 'After', 'a', 'quick', 'breakfast', ',', 'he', 'decided', '
 Text 4 Tokens: ['The', 'software', 'development', 'team', 'implemented', 'the', 'new', 'feature', 'using', 'agile', 'methodology', '.',
 Text 5 Tokens: ['Sarah', 'walked', 'to', 'the', 'park', 'with', 'her', 'dog', 'Max', '.', 'The', 'sun', 'was', 'shining', 'brightly', '
 [nltk_data] Downloading package punkt to /root/nltk_data...
 [nltk_data] Package punkt is already up-to-date!
```

```
#Stopwords Removing
from nltk.corpus import stopwords
import string
nltk.download('stopwords')
stop_words = set(stopwords.words('english'))
punctuation = set(string.punctuation)
filtered_texts = [[word for word in tokens if word.lower() not in stop_words and word.lower() not in punctuation] for tokens in tokenized_te
for idx, tokens in enumerate(filtered_texts, 1):
    print(f"Text {idx} Tokens without stopwords:", tokens)
```

```
 Text 1 Tokens without stopwords: ['sun', 'shines', 'brightly', 'sky']
 Text 2 Tokens without stopwords: ['Although', 'studied', 'hard', 'failed', 'exam', 'questions', 'difficult'] Text 3 Tokens without
 stopwords: ['John', 'woke', 'early', 'morning', 'quick', 'breakfast', 'decided', 'go', 'run', 'park', 'jogged', ' Text 4 Tokens without
 stopwords: ['software', 'development', 'team', 'implemented', 'new', 'feature', 'using', 'agile', 'methodology',  Text 5 Tokens without
 stopwords: ['Sarah', 'walked', 'park', 'dog', 'Max', 'sun', 'shining', 'brightly', 'birds', 'chirping', 'trees',  [nltk_data]
 Downloading package stopwords to /root/nltk_data...
 [nltk_data] Package stopwords is already up-to-date!
```

```
#Lancaster Stemmer
from nltk.stem import LancasterStemmer
lancaster = LancasterStemmer()
stemmed_texts = [[lancaster.stem(word) for word in tokens] for tokens in filtered_texts]
for idx, stemmed_tokens in enumerate(stemmed_texts, 1):
    print(f"Text {idx} Stemmed Tokens:", stemmed_tokens)
```

```
 Text 1 Stemmed Tokens: ['sun', 'shin', 'bright', 'sky']
 Text 2 Stemmed Tokens: ['although', 'study', 'hard', 'fail', 'exam', 'quest', 'difficult']
 Text 3 Stemmed Tokens: ['john', 'wok', 'ear', 'morn', 'quick', 'breakfast', 'decid', 'go', 'run', 'park', 'jog', 'along', 'path', 'not'
 Text 4 Stemmed Tokens: ['softw', 'develop', 'team', 'impl', 'new', 'feat', 'us', 'agil', 'methodolog', 'despit', 'encount', 'init', 'ch
 Text 5 Stemmed Tokens: ['sarah', 'walk', 'park', 'dog', 'max', 'sun', 'shin', 'bright', 'bird', 'chirp', 'tre', 'max', 'wag', 'tail', '
```

```
#Snowball Stemmer
from nltk.stem import SnowballStemmer
snowball = SnowballStemmer('english')
stemmed_texts = [[snowball.stem(word) for word in tokens] for tokens in filtered_texts]
for idx, stemmed_tokens in enumerate(stemmed_texts, 1):
    print(f"Text {idx} Stemmed Tokens:", stemmed_tokens)
```

```
 Text 1 Stemmed Tokens: ['sun', 'shine', 'bright', 'sky']
 Text 2 Stemmed Tokens: ['although', 'studi', 'hard', 'fail', 'exam', 'question', 'difficult']
 Text 3 Stemmed Tokens: ['john', 'woke', 'earli', 'morn', 'quick', 'breakfast', 'decid', 'go', 'run', 'park', 'jog', 'along', 'path', 'n
```

```
 Text 4 Stemmed Tokens: ['softwar', 'develop', 'team', 'implement', 'new', 'featur', 'use', 'agil', 'methodolog', 'despit', 'encount', '
 Text 5 Stemmed Tokens: ['sarah', 'walk', 'park', 'dog', 'max', 'sun', 'shine', 'bright', 'bird', 'chirp', 'tree', 'max', 'wag', 'tail',
```

```
#Porter Stemmer
from nltk.stem import PorterStemmer
porter = PorterStemmer()
stemmed_texts = [[porter.stem(word) for word in tokens] for tokens in filtered_texts]
for idx, stemmed_tokens in enumerate(stemmed_texts, 1):
    print(f"Text {idx} Stemmed Tokens:", stemmed_tokens)
```

```
 Text 1 Stemmed Tokens: ['sun', 'shine', 'brightli', 'sky']
 Text 2 Stemmed Tokens: ['although', 'studi', 'hard', 'fail', 'exam', 'question', 'difficult']
 Text 3 Stemmed Tokens: ['john', 'woke', 'earli', 'morn', 'quick', 'breakfast', 'decid', 'go', 'run', 'park', 'jog', 'along', 'path', 'n
 Text 4 Stemmed Tokens: ['softwar', 'develop', 'team', 'implement', 'new', 'featur', 'use', 'agil', 'methodolog', 'despit', 'encount', '
 Text 5 Stemmed Tokens: ['sarah', 'walk', 'park', 'dog', 'max', 'sun', 'shine', 'brightli', 'bird', 'chirp', 'tree', 'max', 'wag', 'tail
```

```
#WordNet lemmatizer
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
import string
import nltk
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
lemmatizer = WordNetLemmatizer()
stop_words = set(stopwords.words('english'))
punctuation = set(string.punctuation)
lemmatized_texts = []
for text in input_texts:
    tokens = word_tokenize(text)
    filtered_tokens = []
    for word in tokens:
        word = word.lower()
        if word not in stop_words and word not in punctuation:
            filtered_tokens.append(word)
    lemmatized_tokens = [lemmatizer.lemmatize(word) for word in filtered_tokens]
    lemmatized_texts.append(lemmatized_tokens)
for idx, tokens in enumerate(lemmatized_texts, 1):
    print(f"Text {idx} Tokens after lemmatization:", tokens)
```

```
 Text 1 Tokens after lemmatization: ['sun', 'shine', 'brightly', 'sky']
 Text 2 Tokens after lemmatization: ['although', 'studied', 'hard', 'failed', 'exam', 'question', 'difficult'] Text 3 Tokens after
 lemmatization: ['john', 'woke', 'early', 'morning', 'quick', 'breakfast', 'decided', 'go', 'run', 'park', 'jogged', Text 4 Tokens after
 lemmatization: ['software', 'development', 'team', 'implemented', 'new', 'feature', 'using', 'agile', 'methodology' Text 5 Tokens after
 lemmatization: ['sarah', 'walked', 'park', 'dog', 'max', 'sun', 'shining', 'brightly', 'bird', 'chirping', 'tree',  [nltk_data]
 Downloading package punkt to /root/nltk_data...
 [nltk_data] Package punkt is already up-to-date!
 [nltk_data] Downloading package stopwords to /root/nltk_data...
 [nltk_data] Package stopwords is already up-to-date!
 [nltk_data] Downloading package wordnet to /root/nltk_data...
 [nltk_data] Package wordnet is already up-to-date!
```

```
#spaCy lemmatization
import spacy
from nltk.corpus import stopwords
import string
import nltk
nltk.download('stopwords')
nlp = spacy.load('en_core_web_sm')
def spacy_lemmatize(text):
    doc = nlp(text)
    lemmatized_tokens = [token.lemma_ for token in doc if token.text.lower() not in stop_words and token.text.lower() not in punctuation]
    return lemmatized_tokens
input_texts = [
    "The sun shines brightly in the sky.",
    "Although he had studied hard, he failed the exam because the questions were too difficult.",
    "John woke up early in the morning. After a quick breakfast, he decided to go for a run in the park. As he jogged along the path, he not
"The software development team implemented the new feature using agile methodology. Despite encountering some initial challenges, they w
"Sarah walked to the park with her dog Max. The sun was shining brightly, and the birds were chirping in the trees. Max wagged his tail ]
stop_words = set(stopwords.words('english'))
punctuation = set(string.punctuation)
spacy_lemmatized_texts = []
for text in input_texts:
    tokens = text.split()
    lemmatized_tokens = spacy_lemmatize(text)
    spacy_lemmatized_texts.append(lemmatized_tokens)
for idx, tokens in enumerate(spacy_lemmatized_texts, 1):
    print(f"Text {idx} Tokens after spaCy lemmatization and removing stopwords:", tokens)
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Package stopwords is already up-to-date!
Text 1 Tokens after spaCy lemmatization and removing stopwords: ['sun', 'shine', 'brightly', 'sky']
Text 2 Tokens after spaCy lemmatization and removing stopwords: ['although', 'study', 'hard', 'fail', 'exam', 'question', 'difficult']
Text 3 Tokens after spaCy lemmatization and removing stopwords: ['John', 'wake', 'early', 'morning', 'quick', 'breakfast', 'decide', 'g
Text 4 Tokens after spaCy lemmatization and removing stopwords: ['software', 'development', 'team', 'implement', 'new', 'feature', 'use
Text 5 Tokens after spaCy lemmatization and removing stopwords: ['Sarah', 'walk', 'park', 'dog', 'Max', 'sun', 'shine', 'brightly', 'bi
```

```
# TextBlob lemmatized
from nltk.corpus import stopwords
from textblob import Word
import string
import nltk
nltk.download('stopwords')
input_texts = [
    "The sun shines brightly in the sky.",
    "Although he had studied hard, he failed the exam because the questions were too difficult.",
    "John woke up early in the morning. After a quick breakfast, he decided to go for a run in the park. As he jogged along the path, he not
"The software development team implemented the new feature using agile methodology. Despite encountering some initial challenges, they w
"Sarah walked to the park with her dog Max. The sun was shining brightly, and the birds were chirping in the trees. Max wagged his tail ]
stop_words = set(stopwords.words('english'))
punctuation = set(string.punctuation)
textblob_lemmatized_texts = []
for text in input_texts:
```

```
    tokens = text.split()
    lemmatized_tokens = [Word(token).lemmatize() for token in tokens if token.lower() not in stop_words and token.lower() not in punctuatio
    textblob_lemmatized_texts.append(lemmatized_tokens)
for idx, tokens in enumerate(textblob_lemmatized_texts, 1):
    print(f"Text {idx} Tokens after TextBlob lemmatization and removing stopwords:", tokens)
```

```
Text 1 Tokens after TextBlob lemmatization and removing stopwords: ['sun', 'shine', 'brightly', 'sky.']
Text 2 Tokens after TextBlob lemmatization and removing stopwords: ['Although', 'studied', 'hard,', 'failed', 'exam', 'question', 'diff
Text 3 Tokens after TextBlob lemmatization and removing stopwords: ['John', 'woke', 'early', 'morning.', 'quick', 'breakfast,', 'decide
Text 4 Tokens after TextBlob lemmatization and removing stopwords: ['software', 'development', 'team', 'implemented', 'new', 'feature',
Text 5 Tokens after TextBlob lemmatization and removing stopwords: ['Sarah', 'walked', 'park', 'dog', 'Max.', 'sun', 'shining', 'bright
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Package stopwords is already up-to-date!
```

```
from IPython.display import Image
Image(filename='/content/Screenshot 2024-02-22 000113.png')
```


Input	Porter Stemmer	Snowball Stemmer	Lancaster Stemmer	Expected	Validation
sun	sun	sun	sun	sun	Right
shines	shine	shine	shin	shine	Right
brightly	brightli	bright	bright	bright	Right
sky	sky	sky	sky	sky	Right
Although	Although	Although	Although	Although	Right
studied	studi	studi	study	study	Right
hard	hard	hard	hard	hard	Right
failed	fail	fail	fail	fail	Right
exam	exam	exam	exam	exam	Right
questions	question	question	quest	question	Right
difficult	difficult	difficult	difficult	difficult	Right
John	john	John	John	John	Right
woke	woke	woke	wok	woke	Right
early	earli	earli	ear	early	Wrong
morning	morn	morn	morn	morning	Wrong
quick	quick	quick	quick	quick	Right
breakfast	breakfast	breakfast	breakfast	breakfast	Right
decided	decid	decid	decid	decide	Wrong
go	go	go	go	go	Right
run	run	run	run	run	Right
park	park	park	park	park	Right
jogged	jog	jog	jog	jog	Right
along	along	along	along	along	Right
path	path	path	path	path	Right
noticed	notic	notic	not	notice	Wrong
group	group	group	group	group	Right
squirrels	squirrel	squirrel	squirrel	squirrel	Right
playing	play	play	play	play	Right
trees	tree	tree	tre	tree	Right
smiled	smile	smile	smil	smile	Right
continued	continu	continu	continu	continue	Wrong
run	run	run	run	run	Right

```
from IPython.display import Image
Image(filename='/content/Screenshot 2024-02-22 000229.png')
```

Input	WordNet lemmatizer	spaCy lemmatization	TextBlob lemmatization	Expected	Validation
sun	sun	sun	sun	sun	Right
shines	shine	shine	shine	shine	Right
brightly	brightly	brightly	brightly	bright	Wrong
sky	sky	sky	sky	sky	Right
Although	Although	although	Although	Although	Right
studied	studied	study	studied	study	Right
hard	hard	hard	hard	hard	Right
failed	failed	fail	failed	fail	Right
exam	exam	exam	exam	exam	Right
questions	question	question	question	question	Right
difficult	difficult	difficult	difficult	difficult	Right
John	john	John	John	John	Right
woke	woke	wake	woke	wake	Right
early	early	early	early	early	Right
morning	morning	morning	morning	morning	Right
quick	quick	quick	quick	quick	Right
breakfast	breakfast	breakfast	breakfast	breakfast	Right
decided	decided	decide	decided	decide	Right
go	go	go	go	go	Right
run	run	run	run	run	Right
park	park	park	park	park	Right
jogged	jogged	jog	jogged	jog	Right
along	along	along	along	along	Right
path	path	path	path	path	Right
noticed	noticed	notice	noticed	notice	Right
group	group	group	group	group	Right
squirrels	squirrel	squirrel	squirrel	squirrel	Right
playing	playing	play	playing	play	Right
trees	tree	tree	trees	tree	Right
smiled	smiled	smile	smiled	smile	Right
continued	continued	continue	continued	continue	Right
run	run	run	run	run	Right


```
from IPython.display import Image
```

```
Image(filename='/content/Screenshot 2024-02-22 000229.png')
```

Input	WordNet lemmatizer	spaCy lemmatization	TextBlob lemmatization	Expected	Validation
sun	sun	sun	sun	sun	Right
shines	shine	shine	shine	shine	Right
brightly	brightly	brightly	brightly	bright	Wrong
sky	sky	sky	sky	sky	Right
Although	Although	although	Although	Although	Right
studied	studied	study	studied	study	Right
hard	hard	hard	hard	hard	Right
failed	failed	fail	failed	fail	Right
exam	exam	exam	exam	exam	Right
questions	question	question	question	question	Right
difficult	difficult	difficult	difficult	difficult	Right
John	john	John	John	John	Right
woke	woke	wake	woke	wake	Right
early	early	early	early	early	Right
morning	morning	morning	morning	morning	Right
quick	quick	quick	quick	quick	Right
breakfast	breakfast	breakfast	breakfast	breakfast	Right
decided	decided	decide	decided	decide	Right
go	go	go	go	go	Right
run	run	run	run	run	Right
park	park	park	park	park	Right
jogged	jogged	jog	jogged	jog	Right
along	along	along	along	along	Right
path	path	path	path	path	Right
noticed	noticed	notice	noticed	notice	Right
group	group	group	group	group	Right
squirrels	squirrel	squirrel	squirrel	squirrel	Right
playing	playing	play	playing	play	Right
trees	tree	tree	trees	tree	Right
smiled	smiled	smile	smiled	smile	Right
continued	continued	continue	continued	continue	Right
run	run	run	run	run	Right