Text Preprocessing

PREPARED BY: AHMAD ALAA ALDINE

PRESENTED BY: AHMAD ALAA ALDINE

Definition

Text preprocessing is the process of making the input text more machine-understandable.

Preparing text data for analysis

Enrich data with syntactic information

Text Preprocessing Techniques

Sentence Segmentation

It is the process of breaking text data into a collection of sentences.

Also known as sentence tokenization.

```
['Once upon a time there was an old mother pig who had three little pigs and not enough food to feed them.',
 'So when they were old enough, she sent them out into the world to seek their fortunes.',
 'The first little pig was very lazy.',
 "He didn't want to work at all and he built his house out of straw.",
 'The second little pig worked a little bit harder but he was somewhat lazy too and he built his house out of sticks.',
 'Then, they sang and danced and played together the rest of the day.',
 'The third little pig worked hard all day and built his house with bricks.',
 'It was a sturdy house complete with a fine fireplace and chimney.',
 'It looked like it could withstand the strongest winds.']
```

Word Tokenizing

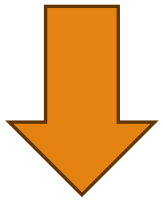
It is the process of breaking text data into a collection of words.

```
['Once', 'upon', 'a', 'time', 'there', 'was', 'an', 'old', 'mother', 'pig', 'who', 'had', 'three', 'little', 'pigs', 'and', 'no
t', 'enough', 'food', 'to', 'feed', 'them', '.', 'So', 'when', 'they', 'were', 'old', 'enough', ',', 'she', 'sent', 'them', 'ou
t', 'into', 'the', 'world', 'to', 'seek', 'their', 'fortunes', '.', 'The', 'first', 'little', 'pig', 'was', 'very', 'lazy',
'.', 'He', 'did', "n't", 'want', 'to', 'work', 'at', 'all', 'and', 'he', 'built', 'his', 'house', 'out', 'of', 'straw', '.', 'T
he', 'second', 'little', 'pig', 'worked', 'a', 'little', 'bit', 'harder', 'but', 'he', 'was', 'somewhat', 'lazy', 'too', 'and',
'he', 'built', 'his', 'house', 'out', 'of', 'sticks', '.', 'Then', ',', 'they', 'sang', 'and', 'danced', 'and', 'played', 'toge
ther', 'the', 'rest', 'of', 'the', 'day', '.', 'The', 'third', 'little', 'pig', 'worked', 'hard', 'all', 'day', 'and', 'built',
'his', 'house', 'with', 'bricks', '.', 'It', 'was', 'a', 'sturdy', 'house', 'complete', 'with', 'a', 'fine', 'fireplace', 'an
d', 'chimney', '.', 'It', 'looked', 'like', 'it', 'could', 'withstand', 'the', 'strongest', 'winds', '.']
```

Stop words Removal

The process of cleaning text from stop words.

```
['Once', 'upon', 'a', 'time', 'there', 'was', 'an', 'old', 'mother', 'pig', 'who', 'had', 'three', 'little', 'pigs', 'and', 'no
t', 'enough', 'food', 'to', 'feed', 'them', '.', 'So', 'when', 'they', 'were', 'old', 'enough', ',', 'she', 'sent', 'them', 'ou
t', 'into', 'the', 'world', 'to', 'seek', 'their', 'fortunes', '.', 'The', 'first', 'little', 'pig', 'was', 'very', 'lazy',
'.', 'He', 'did', "n't", 'want', 'to', 'work', 'at', 'all', 'and', 'he', 'built', 'his', 'house', 'out', 'of', 'straw', '.', 'T
he', 'second', 'little', 'pig', 'worked', 'a', 'little', 'bit', 'harder', 'but', 'he', 'was', 'somewhat', 'lazy', 'too', 'and',
'he', 'built', 'his', 'house', 'out', 'of', 'sticks', '.', 'Then', ',', 'they', 'sang', 'and', 'danced', 'and', 'played', 'toge
ther', 'the', 'rest', 'of', 'the', 'day', '.', 'The', 'third', 'little', 'pig', 'worked', 'hard', 'all', 'day', 'and', 'built',
'his', 'house', 'with', 'bricks', '.', 'It', 'was', 'a', 'sturdy', 'house', 'complete', 'with', 'a', 'fine', 'fireplace', 'an
d', 'chimney', '.', 'It', 'looked', 'like', 'it', 'could', 'withstand', 'the', 'strongest', 'winds', '.']
```



```
['upon', 'time', 'old', 'mother', 'pig', 'three', 'little', 'pigs', 'enough', 'food', 'feed', 'old', 'enough', 'sent', 'world',
'seek', 'fortunes', 'first', 'little', 'pig', 'lazy', 'want', 'work', 'built', 'house', 'straw', 'second', 'little', 'pig', 'wo
rked', 'little', 'bit', 'harder', 'somewhat', 'lazy', 'built', 'house', 'sticks', 'sang', 'danced', 'played', 'together', 'res
t', 'day', 'third', 'little', 'pig', 'worked', 'hard', 'day', 'built', 'house', 'bricks', 'sturdy', 'house', 'complete', 'fin
e', 'fireplace', 'chimney', 'looked', 'like', 'could', 'withstand', 'strongest', 'winds']
```

POS Tagging

POS tagging is the process of assigning part of speech tag for each word.

```
[('Once', 'ADV'), ('upon', 'SCONJ'), ('a', 'DET'), ('time', 'NOUN'), ('there', 'PRON'), ('was', 'VERB'), ('an', 'DET'), ('old',
'ADJ'), ('mother', 'NOUN'), ('pig', 'NOUN'), ('who', 'PRON'), ('had', 'VERB'), ('three', 'NUM'), ('little', 'ADJ'), ('pigs', 'N
OUN'), ('and', 'CCONJ'), ('not', 'PART'), ('enough', 'ADJ'), ('food', 'NOUN'), ('to', 'PART'), ('feed', 'VERB'), ('them', 'PRO
N'), ('.', 'PUNCT'), ('\n', 'SPACE'), ('So', 'ADV'), ('when', 'SCONJ'), ('they', 'PRON'), ('were', 'AUX'), ('old', 'ADJ'), ('en
ough', 'ADV'), (',', 'PUNCT'), ('she', 'PRON'), ('sent', 'VERB'), ('them', 'PRON'), ('out', 'ADP'), ('into', 'ADP'), ('the', 'D
ET'), ('world', 'NOUN'), ('to', 'PART'), ('seek', 'VERB'), ('their', 'PRON'), ('fortunes', 'NOUN'), ('.', 'PUNCT')]
```

Stemming

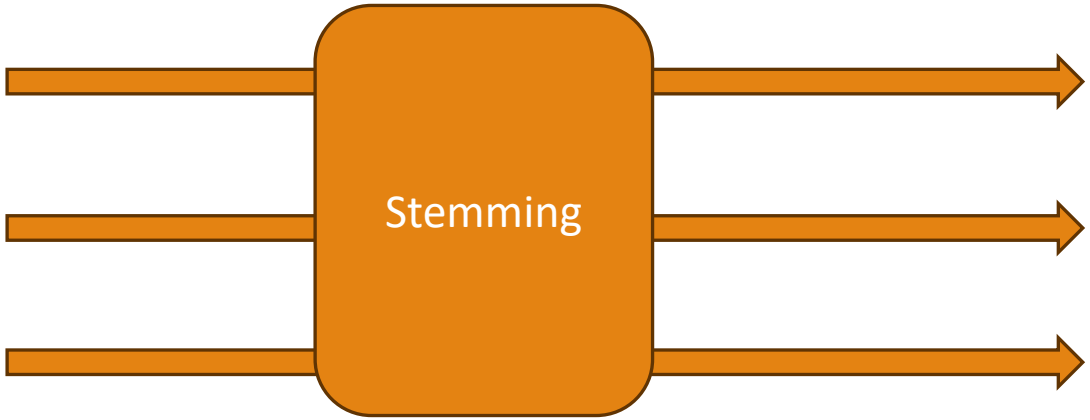
In English and many other languages, a single word can take multiple forms depending upon the context used.

Stemming normalizes the word by truncating the word to its stem word.

i. Studies

ii. Studying

iii. Studied



Studi

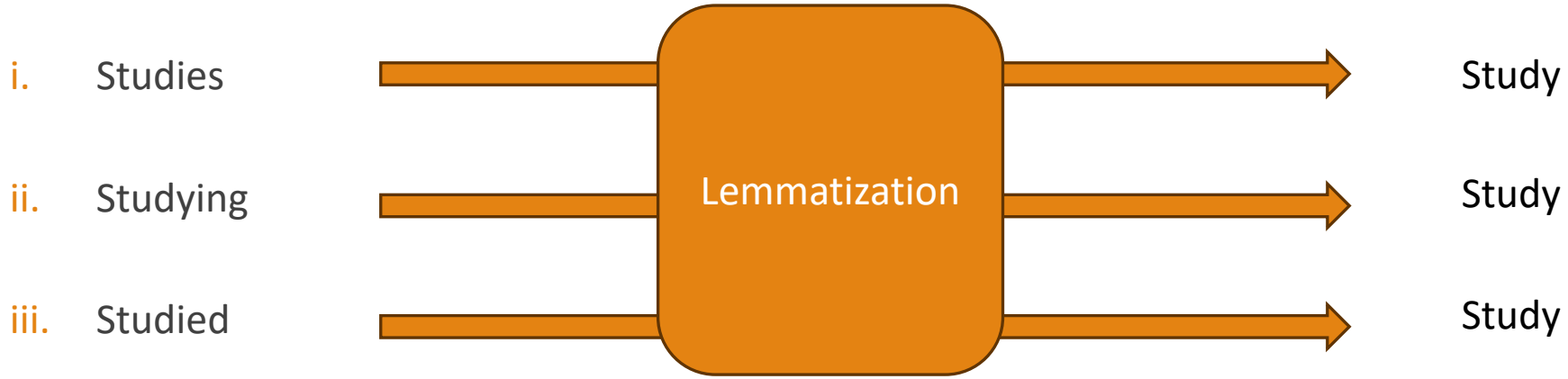
Studi

Studi

Notice that stemming may not give us a dictionary word

Lemmatization

Lemmatization tries to achieve a similar base “stem” for a word. However, what makes it different is that it finds the dictionary word instead of truncating the original word.

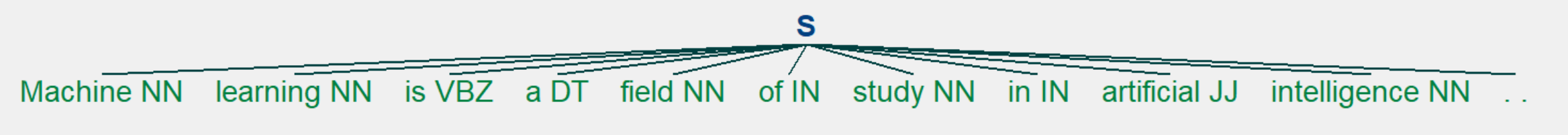


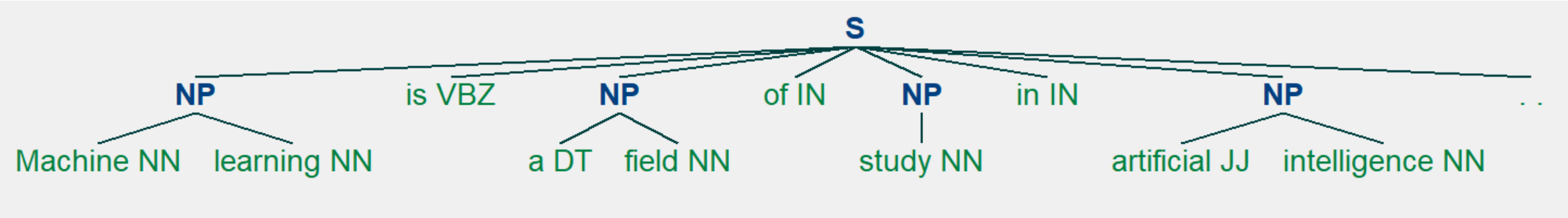
Lemmatization takes into account Part Of Speech (POS) values.

Chunking

Also known as shallow parsing.

It is applied on POS-tagged tokens to get chunks (groups of words) that are more meaningful than individual words.

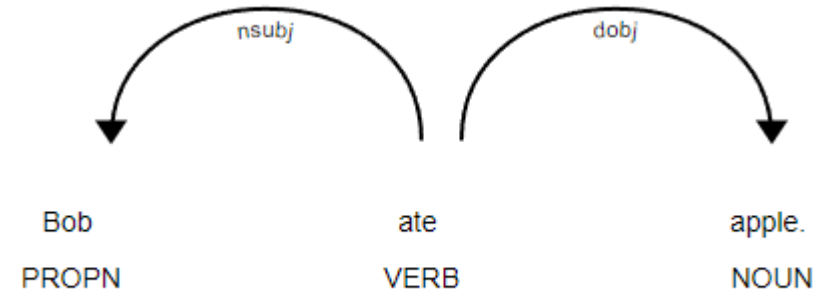




Dependency Parsing

It describes the syntactic structure of sentences in terms of words and binary grammatical relations.

The main advantage of dependency parsers is that their typed dependency structure provides semantic relationships between words of the sentence.



VS

spaCy

NLTK VS spaCy

NLTK	spaCy
Provides several algorithm for a particular task	Best algorithm for a particular task
Poor performance	Good performance
Does not support word vectors	Supports word vectors
Fits more for research purposes	Fits more for development purposes

NLTK: Sentence Segmentation

```
from nltk.tokenize import sent_tokenize
text = """Once upon a time there was an old mother pig who had three little pigs and not enough food to feed them.
So when they were old enough, she sent them out into the world to seek their fortunes.
The first little pig was very lazy. He didn't want to work at all and he built his house out of straw.
The second little pig worked a little bit harder but he was somewhat lazy too and he built his house out of sticks.
Then, they sang and danced and played together the rest of the day.
The third little pig worked hard all day and built his house with bricks.
It was a sturdy house complete with a fine fireplace and chimney.
It looked like it could withstand the strongest winds."""
tokenized_text=sent_tokenize(text)
print(tokenized_text)
```

```
['Once upon a time there was an old mother pig who had three little pigs and not enough food to feed them.', 'So when they were
old enough, she sent them out into the world to seek their fortunes.', 'The first little pig was very lazy.', "He didn't want t
o work at all and he built his house out of straw.", 'The second little pig worked a little bit harder but he was somewhat lazy
too and he built his house out of sticks.', 'Then, they sang and danced and played together the rest of the day.', 'The third l
ittle pig worked hard all day and built his house with bricks.', 'It was a sturdy house complete with a fine fireplace and chim
ney.', 'It looked like it could withstand the strongest winds.']
```

spaCy: Sentence Segmentation

```
import spacy
nlp = spacy.load('en_core_web_sm') # or whatever model you have installed

text = """Once upon a time there was an old mother pig who had three little pigs and not enough food to feed them.
So when they were old enough, she sent them out into the world to seek their fortunes.
The first little pig was very lazy. He didn't want to work at all and he built his house out of straw.
The second little pig worked a little bit harder but he was somewhat lazy too and he built his house out of sticks.
Then, they sang and danced and played together the rest of the day.
The third little pig worked hard all day and built his house with bricks.
It was a sturdy house complete with a fine fireplace and chimney.
It looked like it could withstand the strongest winds."""

doc = nlp(text)
sentences = [sent.text.strip() for sent in doc.sents]
print(sentences)
```

```
['Once upon a time there was an old mother pig who had three little pigs and not enough food to feed them.', 'So when they were
old enough, she sent them out into the world to seek their fortunes.', 'The first little pig was very lazy.', "He didn't want t
o work at all and he built his house out of straw.", 'The second little pig worked a little bit harder but he was somewhat lazy
too and he built his house out of sticks.', 'Then, they sang and danced and played together the rest of the day.', 'The third l
ittle pig worked hard all day and built his house with bricks.', 'It was a sturdy house complete with a fine fireplace and chim
ney.', 'It looked like it could withstand the strongest winds.']
```

NLTK: Word Tokenization

```
from nltk.tokenize import word_tokenize
text = """Once upon a time there was an old mother pig who had three little pigs and not enough food to feed them.
So when they were old enough, she sent them out into the world to seek their fortunes.
The first little pig was very lazy. He didn't want to work at all and he built his house out of straw.
The second little pig worked a little bit harder but he was somewhat lazy too and he built his house out of sticks.
Then, they sang and danced and played together the rest of the day.
The third little pig worked hard all day and built his house with bricks.
It was a sturdy house complete with a fine fireplace and chimney.
It looked like it could withstand the strongest winds."""
tokenized_text=word_tokenize(text)
print(tokenized_text)
```

```
['Once', 'upon', 'a', 'time', 'there', 'was', 'an', 'old', 'mother', 'pig', 'who', 'had', 'three', 'little', 'pigs', 'and', 'no
t', 'enough', 'food', 'to', 'feed', 'them', '.', 'So', 'when', 'they', 'were', 'old', 'enough', ',', 'she', 'sent', 'them', 'ou
t', 'into', 'the', 'world', 'to', 'seek', 'their', 'fortunes', '.', 'The', 'first', 'little', 'pig', 'was', 'very', 'lazy',
'.', 'He', 'did', "n't", 'want', 'to', 'work', 'at', 'all', 'and', 'he', 'built', 'his', 'house', 'out', 'of', 'straw', '.', 'T
he', 'second', 'little', 'pig', 'worked', 'a', 'little', 'bit', 'harder', 'but', 'he', 'was', 'somewhat', 'lazy', 'too', 'and',
'he', 'built', 'his', 'house', 'out', 'of', 'sticks', '.', 'Then', ',', 'they', 'sang', 'and', 'danced', 'and', 'played', 'toge
ther', 'the', 'rest', 'of', 'the', 'day', '.', 'The', 'third', 'little', 'pig', 'worked', 'hard', 'all', 'day', 'and', 'built',
'his', 'house', 'with', 'bricks', '.', 'It', 'was', 'a', 'sturdy', 'house', 'complete', 'with', 'a', 'fine', 'fireplace', 'an
d', 'chimney', '.', 'It', 'looked', 'like', 'it', 'could', 'withstand', 'the', 'strongest', 'winds', '.']
```

spaCy: Word Tokenization

```
import spacy

#load the small English model
nlp = spacy.load("en_core_web_sm")

sentence = """Once upon a time there was an old mother pig who had three little pigs and not enough food to feed them.
So when they were old enough, she sent them out into the world to seek their fortunes.
The first little pig was very lazy. He didn't want to work at all and he built his house out of straw.
The second little pig worked a little bit harder but he was somewhat lazy too and he built his house out of sticks.
Then, they sang and danced and played together the rest of the day.
The third little pig worked hard all day and built his house with bricks.
It was a sturdy house complete with a fine fireplace and chimney.
It looked like it could withstand the strongest winds."""

tokens = [token.text for token in nlp(sentence)]
print(tokens)
```

```
['Once', 'upon', 'a', 'time', 'there', 'was', 'an', 'old', 'mother', 'pig', 'who', 'had', 'three', 'little', 'pigs', 'and', 'no
t', 'enough', 'food', 'to', 'feed', 'them', '.', '\n', 'So', 'when', 'they', 'were', 'old', 'enough', ',', 'she', 'sent', 'the
m', 'out', 'into', 'the', 'world', 'to', 'seek', 'their', 'fortunes', '.', '\n', 'The', 'first', 'little', 'pig', 'was', 'ver
y', 'lazy', '.', 'He', 'did', "n't", 'want', 'to', 'work', 'at', 'all', 'and', 'he', 'built', 'his', 'house', 'out', 'of', 'str
aw', '.', '\n', 'The', 'second', 'little', 'pig', 'worked', 'a', 'little', 'bit', 'harder', 'but', 'he', 'was', 'somewhat', 'la
zy', 'too', 'and', 'he', 'built', 'his', 'house', 'out', 'of', 'sticks', '.', '\n', 'Then', ',', 'they', 'sang', 'and', 'dance
d', 'and', 'played', 'together', 'the', 'rest', 'of', 'the', 'day', '.', '\n', 'The', 'third', 'little', 'pig', 'worked', 'har
d', 'all', 'day', 'and', 'built', 'his', 'house', 'with', 'bricks', '.', '\n', 'It', 'was', 'a', 'sturdy', 'house', 'complete',
'with', 'a', 'fine', 'fireplace', 'and', 'chimney', '.', '\n', 'It', 'looked', 'like', 'it', 'could', 'withstand', 'the', 'stro
ngest', 'winds', '.']
```

NLTK: Stop words removal

```
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
sentence = """Once upon a time there was an old mother pig who had three little pigs and not enough food to feed them.
So when they were old enough, she sent them out into the world to seek their fortunes.
The first little pig was very lazy. He didn't want to work at all and he built his house out of straw.
The second little pig worked a little bit harder but he was somewhat lazy too and he built his house out of sticks.
Then, they sang and danced and played together the rest of the day.
The third little pig worked hard all day and built his house with bricks.
It was a sturdy house complete with a fine fireplace and chimney.
It looked like it could withstand the strongest winds."""

stop_words = set(stopwords.words('english'))

word_tokens = word_tokenize(sentence)
filtered_tokens = [word for word in word_tokens if not word.lower() in stop_words]
print(filtered_tokens)
```

```
['upon', 'time', 'old', 'mother', 'pig', 'three', 'little', 'pigs', 'enough', 'food', 'feed', '.', 'old', 'enough', ',', 'sen
t', 'world', 'seek', 'fortunes', '.', 'first', 'little', 'pig', 'lazy', '.', "n't", 'want', 'work', 'built', 'house', 'straw',
'.', 'second', 'little', 'pig', 'worked', 'little', 'bit', 'harder', 'somewhat', 'lazy', 'built', 'house', 'sticks', '.', ',',
'sang', 'danced', 'played', 'together', 'rest', 'day', '.', 'third', 'little', 'pig', 'worked', 'hard', 'day', 'built', 'hous
e', 'bricks', '.', 'sturdy', 'house', 'complete', 'fine', 'fireplace', 'chimney', '.', 'looked', 'like', 'could', 'withstand',
'strongest', 'winds', '.']
```

spaCy: Stop words removal

```
import spacy

#load the small English model
nlp = spacy.load("en_core_web_sm")

sentence = """Once upon a time there was an old mother pig who had three little pigs and not enough food to feed them.
So when they were old enough, she sent them out into the world to seek their fortunes.
The first little pig was very lazy. He didn't want to work at all and he built his house out of straw.
The second little pig worked a little bit harder but he was somewhat lazy too and he built his house out of sticks.
Then, they sang and danced and played together the rest of the day.
The third little pig worked hard all day and built his house with bricks.
It was a sturdy house complete with a fine fireplace and chimney.
It looked like it could withstand the strongest winds."""

doc = nlp(sentence)
filtered_tokens = [token.text for token in doc if not token.is_stop]
print(filtered_tokens)
```

```
['time', 'old', 'mother', 'pig', 'little', 'pigs', 'food', 'feed', '.', '\n', 'old', ',', 'sent', 'world', 'seek', 'fortunes',
'.', '\n', 'little', 'pig', 'lazy', '.', 'want', 'work', 'built', 'house', 'straw', '.', '\n', 'second', 'little', 'pig', 'work
ed', 'little', 'bit', 'harder', 'somewhat', 'lazy', 'built', 'house', 'sticks', '.', '\n', ',', 'sang', 'danced', 'played', 're
st', 'day', '.', '\n', 'little', 'pig', 'worked', 'hard', 'day', 'built', 'house', 'bricks', '.', '\n', 'sturdy', 'house', 'com
plete', 'fine', 'fireplace', 'chimney', '.', '\n', 'looked', 'like', 'withstand', 'strongest', 'winds', '.']
```

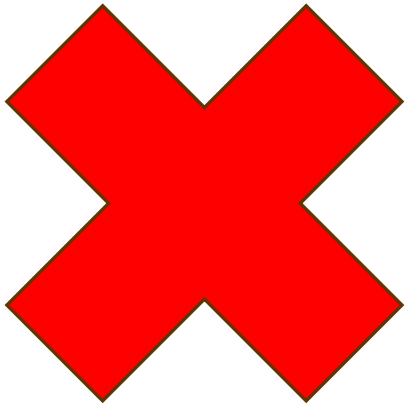
NLTK: Stemming

```
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
sentence = """Once upon a time there was an old mother pig who had three little pigs and not enough food to feed them.
So when they were old enough, she sent them out into the world to seek their fortunes.
The first little pig was very lazy. He didn't want to work at all and he built his house out of straw.
The second little pig worked a little bit harder but he was somewhat lazy too and he built his house out of sticks.
Then, they sang and danced and played together the rest of the day.
The third little pig worked hard all day and built his house with bricks.
It was a sturdy house complete with a fine fireplace and chimney.
It looked like it could withstand the strongest winds."""
ps = PorterStemmer()
word_tokens = word_tokenize(sentence)
stem_tokens = [ps.stem(word) for word in word_tokens]
print(stem_tokens)
```

```
['onc', 'upon', 'a', 'time', 'there', 'wa', 'an', 'old', 'mother', 'pig', 'who', 'had', 'three', 'littl', 'pig', 'and', 'not',
'enough', 'food', 'to', 'feed', 'them', '.', 'So', 'when', 'they', 'were', 'old', 'enough', ',', 'she', 'sent', 'them', 'out',
'into', 'the', 'world', 'to', 'seek', 'their', 'fortun', '.', 'the', 'first', 'littl', 'pig', 'wa', 'veri', 'lazi', '.', 'He',
'did', "n't", 'want', 'to', 'work', 'at', 'all', 'and', 'he', 'built', 'hi', 'hous', 'out', 'of', 'straw', '.', 'the', 'secon
d', 'littl', 'pig', 'work', 'a', 'littl', 'bit', 'harder', 'but', 'he', 'wa', 'somewhat', 'lazi', 'too', 'and', 'he', 'built',
'hi', 'hous', 'out', 'of', 'stick', '.', 'then', ',', 'they', 'sang', 'and', 'danc', 'and', 'play', 'togeth', 'the', 'rest', 'o
f', 'the', 'day', '.', 'the', 'third', 'littl', 'pig', 'work', 'hard', 'all', 'day', 'and', 'built', 'hi', 'hous', 'with', 'bri
ck', '.', 'It', 'wa', 'a', 'sturdi', 'hous', 'complet', 'with', 'a', 'fine', 'fireplac', 'and', 'chimney', '.', 'It', 'look',
'like', 'it', 'could', 'withstand', 'the', 'strongest', 'wind', '.']
```

spaCy: Stemming

spaCy does not support stemming, it relies only on lemmatization.



NLTK: POS tagging

```
from nltk.tokenize import word_tokenize
from nltk import pos_tag
sentence = """Once upon a time there was an old mother pig who had three little pigs and not enough food to feed them.
So when they were old enough, she sent them out into the world to seek their fortunes.
The first little pig was very lazy. He didn't want to work at all and he built his house out of straw.
The second little pig worked a little bit harder but he was somewhat lazy too and he built his house out of sticks.
Then, they sang and danced and played together the rest of the day.
The third little pig worked hard all day and built his house with bricks.
It was a sturdy house complete with a fine fireplace and chimney.
It looked like it could withstand the strongest winds."""
word_tokens = word_tokenize(sentence)
pos_tagging = pos_tag(word_tokens)
print(pos_tagging)
```

```
[('Once', 'RB'), ('upon', 'IN'), ('a', 'DT'), ('time', 'NN'), ('there', 'EX'), ('was', 'VBD'), ('an', 'DT'), ('old', 'JJ'), ('m
other', 'NN'), ('pig', 'NN'), ('who', 'WP'), ('had', 'VBD'), ('three', 'CD'), ('little', 'JJ'), ('pigs', 'NNS'), ('and', 'CC'),
('not', 'RB'), ('enough', 'RB'), ('food', 'NN'), ('to', 'TO'), ('feed', 'VB'), ('them', 'PRP'), ('.', '.'), ('So', 'RB'), ('whe
n', 'WRB'), ('they', 'PRP'), ('were', 'VBD'), ('old', 'JJ'), ('enough', 'RB'), (',', ','), ('she', 'PRP'), ('sent', 'VBD'), ('t
hem', 'PRP'), ('out', 'RP'), ('into', 'IN'), ('the', 'DT'), ('world', 'NN'), ('to', 'TO'), ('seek', 'VB'), ('their', 'PRP$'),
('fortunes', 'NNS'), ('.', '.'), ('The', 'DT'), ('first', 'JJ'), ('little', 'JJ'), ('pig', 'NN'), ('was', 'VBD'), ('very', 'R
B'), ('lazy', 'JJ'), ('.', '.'), ('He', 'PRP'), ('did', 'VBD'), ("n't", 'RB'), ('want', 'VB'), ('to', 'TO'), ('work', 'VB'),
('at', 'IN'), ('all', 'DT'), ('and', 'CC'), ('he', 'PRP'), ('built', 'VBD'), ('his', 'PRP$'), ('house', 'NN'), ('out', 'IN'),
('of', 'IN'), ('straw', 'NN'), ('.', '.'), ('The', 'DT'), ('second', 'JJ'), ('little', 'JJ'), ('pig', 'NN'), ('worked', 'VBD'),
('a', 'DT'), ('little', 'JJ'), ('bit', 'NN'), ('harder', 'RBR'), ('but', 'CC'), ('he', 'PRP'), ('was', 'VBD'), ('somewhat', 'R
B'), ('lazy', 'JJ'), ('too', 'RB'), ('and', 'CC'), ('he', 'PRP'), ('built', 'VBD'), ('his', 'PRP$'), ('house', 'NN'), ('out',
'IN'), ('of', 'IN'), ('sticks', 'NNS'), ('.', '.'), ('Then', 'RB'), (',', ','), ('they', 'PRP'), ('sang', 'VBD'), ('and', 'C
C'), ('danced', 'VBD'), ('and', 'CC'), ('played', 'VBD'), ('together', 'RB'), ('the', 'DT'), ('rest', 'NN'), ('of', 'IN'), ('th
e', 'DT'), ('day', 'NN'), ('.', '.'), ('The', 'DT'), ('third', 'JJ'), ('little', 'JJ'), ('pig', 'NN'), ('worked', 'VBD'), ('har
d', 'JJ'), ('all', 'DT'), ('day', 'NN'), ('and', 'CC'), ('built', 'VBD'), ('his', 'PRP$'), ('house', 'NN'), ('with', 'IN'), ('b
ricks', 'NNS'), ('.', '.'), ('It', 'PRP'), ('was', 'VBD'), ('a', 'DT'), ('sturdy', 'JJ'), ('house', 'NN'), ('complete', 'JJ'),
('with', 'IN'), ('a', 'DT'), ('fine', 'JJ'), ('fireplace', 'NN'), ('and', 'CC'), ('chimney', 'NN'), ('.', '.'), ('It', 'PRP'),
('looked', 'VBD'), ('like', 'IN'), ('it', 'PRP'), ('could', 'MD'), ('withstand', 'VB'), ('the', 'DT'), ('strongest', 'JJS'),
('winds', 'NNS'), ('.', '.')]
```

spaCy: POS tagging

```
import spacy
nlp = spacy.load("en_core_web_sm")
sentence = """Once upon a time there was an old mother pig who had three little pigs and not enough food to feed them.
So when they were old enough, she sent them out into the world to seek their fortunes.
The first little pig was very lazy. He didn't want to work at all and he built his house out of straw.
The second little pig worked a little bit harder but he was somewhat lazy too and he built his house out of sticks.
Then, they sang and danced and played together the rest of the day.
The third little pig worked hard all day and built his house with bricks.
It was a sturdy house complete with a fine fireplace and chimney.
It looked like it could withstand the strongest winds."""
doc = nlp(sentence)
pos_tagging = [(token.text, token.pos_) for token in doc]
print(pos_tagging)
```

```
[('Once', 'ADV'), ('upon', 'SCONJ'), ('a', 'DET'), ('time', 'NOUN'), ('there', 'PRON'), ('was', 'VERB'), ('an', 'DET'), ('old',
'ADJ'), ('mother', 'NOUN'), ('pig', 'NOUN'), ('who', 'PRON'), ('had', 'VERB'), ('three', 'NUM'), ('little', 'ADJ'), ('pigs', 'N
OUN'), ('and', 'CCONJ'), ('not', 'PART'), ('enough', 'ADJ'), ('food', 'NOUN'), ('to', 'PART'), ('feed', 'VERB'), ('them', 'PRO
N'), ('.', 'PUNCT'), ('\n', 'SPACE'), ('So', 'ADV'), ('when', 'SCONJ'), ('they', 'PRON'), ('were', 'AUX'), ('old', 'ADJ'), ('en
ough', 'ADV'), (',', 'PUNCT'), ('she', 'PRON'), ('sent', 'VERB'), ('them', 'PRON'), ('out', 'ADP'), ('into', 'ADP'), ('the', 'D
ET'), ('world', 'NOUN'), ('to', 'PART'), ('seek', 'VERB'), ('their', 'PRON'), ('fortunes', 'NOUN'), ('.', 'PUNCT'), ('\n', 'SPA
CE'), ('The', 'DET'), ('first', 'ADJ'), ('little', 'ADJ'), ('pig', 'NOUN'), ('was', 'AUX'), ('very', 'ADV'), ('lazy', 'ADJ'),
('.', 'PUNCT'), ('He', 'PRON'), ('did', 'AUX'), ("n't", 'PART'), ('want', 'VERB'), ('to', 'PART'), ('work', 'VERB'), ('at', 'AD
V'), ('all', 'ADV'), ('and', 'CCONJ'), ('he', 'PRON'), ('built', 'VERB'), ('his', 'PRON'), ('house', 'NOUN'), ('out', 'ADP'),
('of', 'ADP'), ('straw', 'NOUN'), ('.', 'PUNCT'), ('\n', 'SPACE'), ('The', 'DET'), ('second', 'ADJ'), ('little', 'ADJ'), ('pi
g', 'NOUN'), ('worked', 'VERB'), ('a', 'DET'), ('little', 'ADJ'), ('bit', 'NOUN'), ('harder', 'ADV'), ('but', 'CCONJ'), ('he',
'PRON'), ('was', 'AUX'), ('somewhat', 'ADV'), ('lazy', 'ADJ'), ('too', 'ADV'), ('and', 'CCONJ'), ('he', 'PRON'), ('built', 'VER
B'), ('his', 'PRON'), ('house', 'NOUN'), ('out', 'ADP'), ('of', 'ADP'), ('sticks', 'NOUN'), ('.', 'PUNCT'), ('\n', 'SPACE'),
('Then', 'ADV'), (',', 'PUNCT'), ('they', 'PRON'), ('sang', 'VERB'), ('and', 'CCONJ'), ('danced', 'VERB'), ('and', 'CCONJ'),
('played', 'VERB'), ('together', 'ADV'), ('the', 'DET'), ('rest', 'NOUN'), ('of', 'ADP'), ('the', 'DET'), ('day', 'NOUN'),
('.', 'PUNCT'), ('\n', 'SPACE'), ('The', 'DET'), ('third', 'ADJ'), ('little', 'ADJ'), ('pig', 'NOUN'), ('worked', 'VERB'), ('ha
rd', 'ADV'), ('all', 'DET'), ('day', 'NOUN'), ('and', 'CCONJ'), ('built', 'VERB'), ('his', 'PRON'), ('house', 'NOUN'), ('with',
'ADP'), ('bricks', 'NOUN'), ('.', 'PUNCT'), ('\n', 'SPACE'), ('It', 'PRON'), ('was', 'AUX'), ('a', 'DET'), ('sturdy', 'ADJ'),
('house', 'NOUN'), ('complete', 'ADJ'), ('with', 'ADP'), ('a', 'DET'), ('fine', 'ADJ'), ('fireplace', 'NOUN'), ('and', 'CCON
J'), ('chimney', 'NOUN'), ('.', 'PUNCT'), ('\n', 'SPACE'), ('It', 'PRON'), ('looked', 'VERB'), ('like', 'SCONJ'), ('it', 'PRO
N'), ('could', 'AUX'), ('withstand', 'VERB'), ('the', 'DET'), ('strongest', 'ADJ'), ('winds', 'NOUN'), ('.', 'PUNCT')]
```

NLTK: Lemmatization

```
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
sentence = """Once upon a time there was an old mother pig who had three little pigs and not enough food to feed them.
So when they were old enough, she sent them out into the world to seek their fortunes.
The first little pig was very lazy. He didn't want to work at all and he built his house out of straw.
The second little pig worked a little bit harder but he was somewhat lazy too and he built his house out of sticks.
Then, they sang and danced and played together the rest of the day.
The third little pig worked hard all day and built his house with bricks.
It was a sturdy house complete with a fine fireplace and chimney.
It looked like it could withstand the strongest winds."""
lemmatizer = WordNetLemmatizer()
word_tokens = word_tokenize(sentence)
lemma_tokens = [lemmatizer.lemmatize(word) for word in word_tokens]
print(lemma_tokens)
```

```
['Once', 'upon', 'a', 'time', 'there', 'wa', 'an', 'old', 'mother', 'pig', 'who', 'had', 'three', 'little', 'pig', 'and', 'no
t', 'enough', 'food', 'to', 'feed', 'them', '.', 'So', 'when', 'they', 'were', 'old', 'enough', ',', 'she', 'sent', 'them', 'ou
t', 'into', 'the', 'world', 'to', 'seek', 'their', 'fortune', '.', 'The', 'first', 'little', 'pig', 'wa', 'very', 'lazy', '.',
'He', 'did', "n't", 'want', 'to', 'work', 'at', 'all', 'and', 'he', 'built', 'his', 'house', 'out', 'of', 'straw', '.', 'The',
'second', 'little', 'pig', 'worked', 'a', 'little', 'bit', 'harder', 'but', 'he', 'wa', 'somewhat', 'lazy', 'too', 'and', 'he',
'built', 'his', 'house', 'out', 'of', 'stick', '.', 'Then', ',', 'they', 'sang', 'and', 'danced', 'and', 'played', 'together',
'the', 'rest', 'of', 'the', 'day', '.', 'The', 'third', 'little', 'pig', 'worked', 'hard', 'all', 'day', 'and', 'built', 'his',
'house', 'with', 'brick', '.', 'It', 'wa', 'a', 'sturdy', 'house', 'complete', 'with', 'a', 'fine', 'fireplace', 'and', 'chimne
y', '.', 'It', 'looked', 'like', 'it', 'could', 'withstand', 'the', 'strongest', 'wind', '.']
```

NLTK: Lemmatization with POS tagging

```python
from nltk.corpus import wordnet
def get_wordnet_pos(treebank_tag):
        """
        return WORDNET POS compliance to WORDENT lemmatization (a,n,r,v)
        """
        if treebank_tag.startswith('J'):
            return wordnet.ADJ
        elif treebank_tag.startswith('V'):
            return wordnet.VERB
        elif treebank_tag.startswith('N'):
            return wordnet.NOUN
        elif treebank_tag.startswith('R'):
            return wordnet.ADV
        else:
            # As default pos in lemmatization is Noun
            return wordnet.NOUN
```

```python
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
from nltk import pos_tag
sentence = """Once upon a time there was an old mother pig who had three little pigs and not enough food to feed them.
So when they were old enough, she sent them out into the world to seek their fortunes.
The first little pig was very lazy. He didn't want to work at all and he built his house out of straw.
The second little pig worked a little bit harder but he was somewhat lazy too and he built his house out of sticks.
Then, they sang and danced and played together the rest of the day.
The third little pig worked hard all day and built his house with bricks.
It was a sturdy house complete with a fine fireplace and chimney.
It looked like it could withstand the strongest winds."""
lemmatizer = WordNetLemmatizer()
word_tokens = word_tokenize(sentence)
pos_tagging = pos_tag(word_tokens)

lemma_tokens = [lemmatizer.lemmatize(word, pos=get_wordnet_pos(pos_t)) for word, pos_t in pos_tagging]
print(lemma_tokens)
```

```
['Once', 'upon', 'a', 'time', 'there', 'be', 'an', 'old', 'mother', 'pig', 'who', 'have', 'three', 'little', 'pig', 'and', 'no
t', 'enough', 'food', 'to', 'fee', 'them', '.', 'So', 'when', 'they', 'be', 'old', 'enough', ',', 'she', 'send', 'them', 'out',
'into', 'the', 'world', 'to', 'seek', 'their', 'fortune', '.', 'The', 'first', 'little', 'pig', 'be', 'very', 'lazy', '.', 'H
e', 'do', "n't", 'want', 'to', 'work', 'at', 'all', 'and', 'he', 'build', 'his', 'house', 'out', 'of', 'straw', '.', 'The', 'se
cond', 'little', 'pig', 'work', 'a', 'little', 'bit', 'hard', 'but', 'he', 'be', 'somewhat', 'lazy', 'too', 'and', 'he', 'buil
d', 'his', 'house', 'out', 'of', 'stick', '.', 'Then', ',', 'they', 'sing', 'and', 'dance', 'and', 'play', 'together', 'the',
'rest', 'of', 'the', 'day', '.', 'The', 'third', 'little', 'pig', 'work', 'hard', 'all', 'day', 'and', 'build', 'his', 'house',
'with', 'brick', '.', 'It', 'be', 'a', 'sturdy', 'house', 'complete', 'with', 'a', 'fine', 'fireplace', 'and', 'chimney', '.',
'It', 'look', 'like', 'it', 'could', 'withstand', 'the', 'strong', 'wind', '.']
```

spaCy: Lemmatization

```
import spacy
nlp = spacy.load("en_core_web_sm")
sentence = """Once upon a time there was an old mother pig who had three little pigs and not enough food to feed them.
So when they were old enough, she sent them out into the world to seek their fortunes.
The first little pig was very lazy. He didn't want to work at all and he built his house out of straw.
The second little pig worked a little bit harder but he was somewhat lazy too and he built his house out of sticks.
Then, they sang and danced and played together the rest of the day.
The third little pig worked hard all day and built his house with bricks.
It was a sturdy house complete with a fine fireplace and chimney.
It looked like it could withstand the strongest winds."""
doc = nlp(sentence)
lemma_tokens = [token.lemma_ for token in doc]
print(lemma_tokens)
```

```
['once', 'upon', 'a', 'time', 'there', 'be', 'an', 'old', 'mother', 'pig', 'who', 'have', 'three', 'little', 'pig', 'and', 'no
t', 'enough', 'food', 'to', 'feed', 'they', '.', '\n', 'so', 'when', 'they', 'be', 'old', 'enough', ',', 'she', 'send', 'they',
'out', 'into', 'the', 'world', 'to', 'seek', 'their', 'fortune', '.', '\n', 'the', 'first', 'little', 'pig', 'be', 'very', 'laz
y', '.', 'he', 'do', 'not', 'want', 'to', 'work', 'at', 'all', 'and', 'he', 'build', 'his', 'house', 'out', 'of', 'straw', '.',
'\n', 'the', 'second', 'little', 'pig', 'work', 'a', 'little', 'bit', 'hard', 'but', 'he', 'be', 'somewhat', 'lazy', 'too', 'an
d', 'he', 'build', 'his', 'house', 'out', 'of', 'stick', '.', '\n', 'then', ',', 'they', 'sing', 'and', 'dance', 'and', 'play',
'together', 'the', 'rest', 'of', 'the', 'day', '.', '\n', 'the', 'third', 'little', 'pig', 'work', 'hard', 'all', 'day', 'and',
'build', 'his', 'house', 'with', 'brick', '.', '\n', 'it', 'be', 'a', 'sturdy', 'house', 'complete', 'with', 'a', 'fine', 'fire
place', 'and', 'chimney', '.', '\n', 'it', 'look', 'like', 'it', 'could', 'withstand', 'the', 'strong', 'wind', '.']
```

NLTK: Noun phrase chunking

```
import nltk
from nltk import Tree
from nltk.tokenize import word_tokenize
sentence = """Once upon a time there was an old mother pig who had three little pigs and not enough food to feed them.
So when they were old enough, she sent them out into the world to seek their fortunes.
The first little pig was very lazy. He didn't want to work at all and he built his house out of straw.
The second little pig worked a little bit harder but he was somewhat lazy too and he built his house out of sticks.
Then, they sang and danced and played together the rest of the day.
The third little pig worked hard all day and built his house with bricks.
It was a sturdy house complete with a fine fireplace and chimney.
It looked like it could withstand the strongest winds."""
tokenized_text = nltk.word_tokenize(sentence)
tagged_token = nltk.pos_tag(tokenized_text)
grammer = r"""
            NP: {<DT>?<JJ>+<NN>}
            {<DT>+<NN>}
            {<NN>+}
            """
phrases = nltk.RegexpParser(grammer)
result = phrases.parse(tagged_token)
noun_phrases = []
for child in result:
    if isinstance(child, Tree):
        if child.label() == 'NP':
            np = " ".join([lf[0] for lf in child.leaves()])
            noun_phrases.append(np)
print(noun_phrases)
```

```
['a time', 'an old mother', 'pig', 'food', 'the world', 'The first little pig', 'house', 'straw', 'The second little pig', 'a l
ittle bit', 'house', 'the rest', 'the day', 'The third little pig', 'all day', 'house', 'a sturdy house', 'a fine fireplace',
'chimney']
```

spaCy: Noun phrase chunking

```
import spacy
nlp = spacy.load("en_core_web_sm")
sentence = """Once upon a time there was an old mother pig who had three little pigs and not enough food to feed them.
So when they were old enough, she sent them out into the world to seek their fortunes.
The first little pig was very lazy. He didn't want to work at all and he built his house out of straw.
The second little pig worked a little bit harder but he was somewhat lazy too and he built his house out of sticks.
Then, they sang and danced and played together the rest of the day.
The third little pig worked hard all day and built his house with bricks.
It was a sturdy house complete with a fine fireplace and chimney.
It looked like it could withstand the strongest winds."""
doc = nlp(sentence)
noun_phrases = [np for np in doc.noun_chunks]
print(noun_phrases)
```

```
[a time, an old mother pig, who, three little pigs, not enough food, them, they, she, them, the world, their fortunes, The firs
t little pig, He, he, his house, straw, The second little pig, he, he, his house, sticks, they, the rest, the day, The third li
ttle pig, his house, bricks, It, a sturdy house, a fine fireplace, chimney, It, it, the strongest winds]
```

NLTK: Dependency Parsing

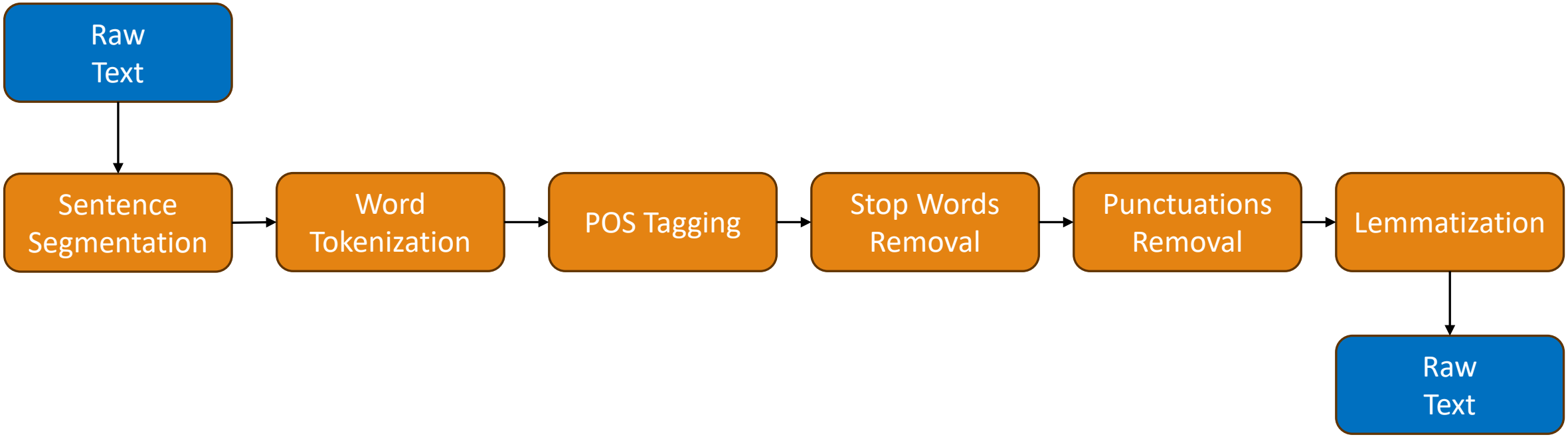
```
from nltk.parse.stanford import StanfordDependencyParser
# Path to CoreNLP jar unzipped

jar_path = '/content/stanford-corenlp-4.2.2/stanford-corenlp-4.2.2.jar'

# Path to CoreNLP model jar
models_jar_path = '/content/stanford-corenlp-4.2.2-models-english.jar'

sentence = 'Deemed universities charge huge fees'

# Initialize StanfordDependency Parser from the path
parser = StanfordDependencyParser(path_to_jar = jar_path, path_to_models_jar = models_jar_path)

# Parse the sentence
result = parser.raw_parse(sentence)
dependency = result.__next__()
```

spaCy: Dependency Parsing

```
import spacy
nlp = spacy.load("en_core_web_sm")
sentence = """Once upon a time there was an old mother pig who had three little pigs and not enough food to feed them.
So when they were old enough, she sent them out into the world to seek their fortunes.
The first little pig was very lazy. He didn't want to work at all and he built his house out of straw.
The second little pig worked a little bit harder but he was somewhat lazy too and he built his house out of sticks.
Then, they sang and danced and played together the rest of the day.
The third little pig worked hard all day and built his house with bricks.
It was a sturdy house complete with a fine fireplace and chimney.
It looked like it could withstand the strongest winds."""
doc = nlp(sentence)
for token in doc:
    print(token.text + ", " + token.dep_ + ", " + token.head.text)
```

```
Once, advmod, was
upon, prep, was
a, det, time
time, pobj, upon
there, expl, was
was, ROOT, was
an, det, pig
old, amod, pig
mother, compound, pig
pig, attr, was
who, nsubj, had
had, relcl, pig
three, nummod, pigs
```

Text Preprocessing Pipeline



NLTK Text Preprocessing Pipeline

```
from nltk.tokenize import sent_tokenize
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
from nltk import pos_tag
from nltk.corpus import stopwords
import string
text = """Once upon a time there was an old mother pig who had three little pigs and not enough food to feed them.
So when they were old enough, she sent them out into the world to seek their fortunes.
The first little pig was very lazy. He didn't want to work at all and he built his house out of straw.
The second little pig worked a little bit harder but he was somewhat lazy too and he built his house out of sticks.
Then, they sang and danced and played together the rest of the day.
The third little pig worked hard all day and built his house with bricks.
It was a sturdy house complete with a fine fireplace and chimney.
It looked like it could withstand the strongest winds."""
lemmatizer = WordNetLemmatizer()
stop_words = set(stopwords.words('english'))
processed_text = []
sentences = sent_tokenize(text)
for sentence in sentences:
    tokens = word_tokenize(sentence)
    pos_tagged_tokens = pos_tag(tokens)
    for token, tag in pos_tagged_tokens:
        if token.lower() in stop_words or token in string.punctuation or token.isdigit():
            continue
        lemma = lemmatizer.lemmatize(token, pos=get_wordnet_pos(tag))
        processed_text.append((lemma, tag))
print(processed_text)
```

```
[('upon', 'IN'), ('time', 'NN'), ('old', 'JJ'), ('mother', 'NN'), ('pig', 'NN'), ('three', 'CD'), ('little', 'JJ'), ('pig', 'NN
S'), ('enough', 'RB'), ('food', 'NN'), ('fee', 'VB'), ('old', 'JJ'), ('enough', 'RB'), ('send', 'VBD'), ('world', 'NN'), ('see
k', 'VB'), ('fortune', 'NNS'), ('first', 'JJ'), ('little', 'JJ'), ('pig', 'NN'), ('lazy', 'JJ'), ("n't", 'RB'), ('want', 'VB'),
('work', 'VB'), ('build', 'VBD'), ('house', 'NN'), ('straw', 'NN'), ('second', 'JJ'), ('little', 'JJ'), ('pig', 'NN'), ('work',
'VBD'), ('little', 'JJ'), ('bit', 'NN'), ('hard', 'RBR'), ('somewhat', 'RB'), ('lazy', 'JJ'), ('build', 'VBD'), ('house', 'N
N'), ('stick', 'NNS'), ('sing', 'VBD'), ('dance', 'VBD'), ('play', 'VBD'), ('together', 'RB'), ('rest', 'NN'), ('day', 'NN'),
('third', 'JJ'), ('little', 'JJ'), ('pig', 'NN'), ('work', 'VBD'), ('hard', 'JJ'), ('day', 'NN'), ('build', 'VBD'), ('house',
'NN'), ('brick', 'NNS'), ('sturdy', 'JJ'), ('house', 'NN'), ('complete', 'JJ'), ('fine', 'JJ'), ('fireplace', 'NN'), ('chimne
y', 'NN'), ('look', 'VBD'), ('like', 'IN'), ('could', 'MD'), ('withstand', 'VB'), ('strong', 'JJS'), ('wind', 'NNS')]
```

spaCy Text Preprocessing Pipeline

```
import spacy
nlp = spacy.load("en_core_web_sm")
sentence = """Once upon a time there was an old mother pig who had three little pigs and not enough food to feed them.
So when they were old enough, she sent them out into the world to seek their fortunes.
The first little pig was very lazy. He didn't want to work at all and he built his house out of straw.
The second little pig worked a little bit harder but he was somewhat lazy too and he built his house out of sticks.
Then, they sang and danced and played together the rest of the day.
The third little pig worked hard all day and built his house with bricks.
It was a sturdy house complete with a fine fireplace and chimney.
It looked like it could withstand the strongest winds."""
doc = nlp(sentence)

processed_text = []
for token in doc:
    if token.is_stop or token.is_punct or token.is_digit or token.is_space:
        continue
    processed_text.append((token.lemma_, token.pos_))

print(processed_text)
```

```
[('time', 'NOUN'), ('old', 'ADJ'), ('mother', 'NOUN'), ('pig', 'NOUN'), ('little', 'ADJ'), ('pig', 'NOUN'), ('food', 'NOUN'),
('feed', 'VERB'), ('old', 'ADJ'), ('send', 'VERB'), ('world', 'NOUN'), ('seek', 'VERB'), ('fortune', 'NOUN'), ('little', 'AD
J'), ('pig', 'NOUN'), ('lazy', 'ADJ'), ('want', 'VERB'), ('work', 'VERB'), ('build', 'VERB'), ('house', 'NOUN'), ('straw', 'NOU
N'), ('second', 'ADJ'), ('little', 'ADJ'), ('pig', 'NOUN'), ('work', 'VERB'), ('little', 'ADJ'), ('bit', 'NOUN'), ('hard', 'AD
V'), ('somewhat', 'ADV'), ('lazy', 'ADJ'), ('build', 'VERB'), ('house', 'NOUN'), ('stick', 'NOUN'), ('sing', 'VERB'), ('dance',
'VERB'), ('play', 'VERB'), ('rest', 'NOUN'), ('day', 'NOUN'), ('little', 'ADJ'), ('pig', 'NOUN'), ('work', 'VERB'), ('hard', 'A
DV'), ('day', 'NOUN'), ('build', 'VERB'), ('house', 'NOUN'), ('brick', 'NOUN'), ('sturdy', 'ADJ'), ('house', 'NOUN'), ('complet
e', 'ADJ'), ('fine', 'ADJ'), ('fireplace', 'NOUN'), ('chimney', 'NOUN'), ('look', 'VERB'), ('like', 'SCONJ'), ('withstand', 'VE
RB'), ('strong', 'ADJ'), ('wind', 'NOUN')]
```