In [2]:

```
# 1.    Load the basic libraries and packages

import spacy
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize, sent_tokenize
from spacy import displacy

nltk.download('stopwords')
nltk.download('punkt')
nltk.download('punkt_tab')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.
```

Out[2]:

```
True
```

In [3]:

```
# Sample Text

text = "Natural Language Processing (NLP) is a field of AI that focuses on the inte
```

In [4]:

```
# 1. Tokenization

sent_tokens = sent_tokenize(text)
word_tokens = word_tokenize(text)

print("\nSentence Tokenization:")
print(sent_tokens)

print("\nWord Tokenization:")
print(word_tokens)
```

```
Sentence Tokenization:
['Natural Language Processing (NLP) is a field of AI that focuses on the interaction
between computers and human language.', 'It enables machines to read, understand, an
d interpret human language.', 'NLP techniques include tokenization, stemming, lemmat
ization, and part-of-speech tagging to process text data effectively.', 'Sentiment a
nalysis is a popular NLP application used to determine the emotional tone of text.',
'Chatbots and virtual assistants leverage NLP to engage in human-like conversation
s.', 'Named Entity Recognition (NER) identifies key entities like names, dates, and
locations in text.', 'Machine translation, such as Google Translate, is powered by N
LP to convert text between languages.', 'NLP models like GPT and BERT have revolutio
nized text generation and understanding.', 'Speech recognition systems rely on NLP t
o convert spoken words into text.', 'Ethical concerns in NLP include bias in languag
e models and the misuse of AI-generated text.']

Word Tokenization:
['Natural', 'Language', 'Processing', '(', 'NLP', ')', 'is', 'a', 'field', 'of', 'A
I', 'that', 'focuses', 'on', 'the', 'interaction', 'between', 'computers', 'and', 'h
uman', 'language', '.', 'It', 'enables', 'machines', 'to', 'read', ',', 'understan
d', ',', 'and', 'interpret', 'human', 'language', '.', 'NLP', 'techniques', 'includ
e', 'tokenization', ',', 'stemming', ',', 'lemmatization', ',', 'and', 'part-of-spee
ch', 'tagging', 'to', 'process', 'text', 'data', 'effectively', '.', 'Sentiment', 'a
nalysis', 'is', 'a', 'popular', 'NLP', 'application', 'used', 'to', 'determine', 'th
e', 'emotional', 'tone', 'of', 'text', '.', 'Chatbots', 'and', 'virtual', 'assistant
s', 'leverage', 'NLP', 'to', 'engage', 'in', 'human-like', 'conversations', '.', 'Na
med', 'Entity', 'Recognition', '(', 'NER', ')', 'identifies', 'key', 'entities', 'li
ke', 'names', ',', 'dates', ',', 'and', 'locations', 'in', 'text', '.', 'Machine',
'translation', ',', 'such', 'as', 'Google', 'Translate', ',', 'is', 'powered', 'by',
'NLP', 'to', 'convert', 'text', 'between', 'languages', '.', 'NLP', 'models', 'lik
e', 'GPT', 'and', 'BERT', 'have', 'revolutionized', 'text', 'generation', 'and', 'un
derstanding', '.', 'Speech', 'recognition', 'systems', 'rely', 'on', 'NLP', 'to', 'c
onvert', 'spoken', 'words', 'into', 'text', '.', 'Ethical', 'concerns', 'in', 'NLP',
'include', 'bias', 'in', 'language', 'models', 'and', 'the', 'misuse', 'of', 'AI-gen
erated', 'text', '.']
```

In [5]:

```python
# 2. Filtration

filtered_tokens = [word for word in word_tokens if word.isalpha()]
print("After Filtration (Only Words):")
print(filtered_tokens)
```

```
After Filtration (Only Words):
['Natural', 'Language', 'Processing', 'NLP', 'is', 'a', 'field', 'of', 'AI', 'that',
'focuses', 'on', 'the', 'interaction', 'between', 'computers', 'and', 'human', 'lang
uage', 'It', 'enables', 'machines', 'to', 'read', 'understand', 'and', 'interpret',
'human', 'language', 'NLP', 'techniques', 'include', 'tokenization', 'stemming', 'le
mmatization', 'and', 'tagging', 'to', 'process', 'text', 'data', 'effectively', 'Sen
timent', 'analysis', 'is', 'a', 'popular', 'NLP', 'application', 'used', 'to', 'dete
rmine', 'the', 'emotional', 'tone', 'of', 'text', 'Chatbots', 'and', 'virtual', 'ass
istants', 'leverage', 'NLP', 'to', 'engage', 'in', 'conversations', 'Named', 'Entit
y', 'Recognition', 'NER', 'identifies', 'key', 'entities', 'like', 'names', 'dates',
'and', 'locations', 'in', 'text', 'Machine', 'translation', 'such', 'as', 'Google',
'Translate', 'is', 'powered', 'by', 'NLP', 'to', 'convert', 'text', 'between', 'lang
uages', 'NLP', 'models', 'like', 'GPT', 'and', 'BERT', 'have', 'revolutionized', 'te
xt', 'generation', 'and', 'understanding', 'Speech', 'recognition', 'systems', 'rel
y', 'on', 'NLP', 'to', 'convert', 'spoken', 'words', 'into', 'text', 'Ethical', 'con
cerns', 'in', 'NLP', 'include', 'bias', 'in', 'language', 'models', 'and', 'the', 'm
isuse', 'of', 'text']
```

In [6]:

```python
# 3. Stopwords Removal

stop_words = set(stopwords.words('english'))
tokens_without_stopwords = [word for word in filtered_tokens if word.lower() not in
print("After Stopwords Removal:")
print(tokens_without_stopwords)
```

```
After Stopwords Removal:
['Natural', 'Language', 'Processing', 'NLP', 'field', 'AI', 'focuses', 'interactio
n', 'computers', 'human', 'language', 'enables', 'machines', 'read', 'understand',
'interpret', 'human', 'language', 'NLP', 'techniques', 'include', 'tokenization', 's
temming', 'lemmatization', 'tagging', 'process', 'text', 'data', 'effectively', 'Sen
timent', 'analysis', 'popular', 'NLP', 'application', 'used', 'determine', 'emotiona
l', 'tone', 'text', 'Chatbots', 'virtual', 'assistants', 'leverage', 'NLP', 'engag
e', 'conversations', 'Named', 'Entity', 'Recognition', 'NER', 'identifies', 'key',
'entities', 'like', 'names', 'dates', 'locations', 'text', 'Machine', 'translation',
'Google', 'Translate', 'powered', 'NLP', 'convert', 'text', 'languages', 'NLP', 'mod
els', 'like', 'GPT', 'BERT', 'revolutionized', 'text', 'generation', 'understandin
g', 'Speech', 'recognition', 'systems', 'rely', 'NLP', 'convert', 'spoken', 'words',
'text', 'Ethical', 'concerns', 'NLP', 'include', 'bias', 'language', 'models', 'misu
se', 'text']
```

In [7]:

```python
# 4. PoS Tagging

nlp = spacy.load("en_core_web_sm")
doc = nlp(text)

print("Part-of-Speech (PoS) Tagging:")
for token in doc:
    print(f"{token.text:<15} {token.pos_:<10} {token.dep_:<10}")
```

```
Part-of-Speech (PoS) Tagging:
Natural         PROPN      compound
Language        PROPN      compound
Processing      PROPN      nsubj
(               PUNCT      punct
NLP             PROPN      appos
)               PUNCT      punct
is              AUX        ROOT
a               DET        det
field           NOUN       attr
of              ADP        prep
AI              PROPN      pobj
that            PRON       nsubj
focuses         VERB       relcl
on              ADP        prep
the             DET        det
interaction     NOUN       pobj
between         ADP        prep
computers       NOUN       pobj
and             CCONJ      cc
human           ADJ        amod
language        NOUN       conj
.               PUNCT      punct
It              PRON       nsubj
enables         VERB       ROOT
machines        NOUN       nsubj
to              PART       aux
read            VERB       ccomp
,               PUNCT      punct
understand      VERB       conj
,               PUNCT      punct
and             CCONJ      cc
interpret       VERB       conj
human           ADJ        amod
language        NOUN       dobj
.               PUNCT      punct
NLP             PROPN      compound
techniques      NOUN       nsubj
include         VERB       ROOT
tokenization    NOUN       dobj
,               PUNCT      punct
stemming        NOUN       conj
,               PUNCT      punct
lemmatization   NOUN       conj
,               PUNCT      punct
and             CCONJ      cc
part            NOUN       nmod
-               PUNCT      punct
of              ADP        prep
-               PUNCT      punct
speech          NOUN       pobj
tagging         NOUN       conj
to              PART       aux
process         VERB       xcomp
text            NOUN       compound
data            NOUN       dobj
```

```
effectively     ADV        advmod
.               PUNCT      punct
Sentiment       NOUN       compound
analysis        NOUN       nsubj
is              AUX        ROOT
a               DET        det
popular         ADJ        amod
NLP             PROPN      compound
application     NOUN       attr
used            VERB       acl
to              PART       aux
determine       VERB       xcomp
the             DET        det
emotional       ADJ        amod
tone            NOUN       dobj
of              ADP        prep
text            NOUN       pobj
.               PUNCT      punct
Chatbots        NOUN       nsubj
and             CCONJ      cc
virtual         ADJ        amod
assistants      NOUN       conj
leverage        VERB       ROOT
NLP             PROPN      dobj
to              PART       aux
engage          VERB       xcomp
in              ADP        prep
human           NOUN       npadvmod
-               PUNCT      punct
like            ADJ        amod
conversations   NOUN       pobj
.               PUNCT      punct
Named           VERB       csubj
Entity          PROPN      compound
Recognition     PROPN      nsubj
(               PUNCT      punct
NER             PROPN      appos
)               PUNCT      punct
identifies      VERB       ROOT
key             ADJ        amod
entities        NOUN       dobj
like            ADP        prep
names           NOUN       pobj
,               PUNCT      punct
dates           NOUN       conj
,               PUNCT      punct
and             CCONJ      cc
locations       NOUN       conj
in              ADP        prep
text            NOUN       pobj
.               PUNCT      punct
Machine         NOUN       compound
translation     NOUN       nsubjpass
,               PUNCT      punct
such            ADJ        amod
as              ADP        prep
```

```
Google          PROPN      compound
Translate       PROPN      pobj
,               PUNCT      punct
is              AUX        auxpass
powered         VERB       ROOT
by              ADP        agent
NLP             PROPN      pobj
to              PART       aux
convert         VERB       xcomp
text            NOUN       dobj
between         ADP        prep
languages       NOUN       pobj
.               PUNCT      punct
NLP             PROPN      compound
models          NOUN       nsubj
like            ADP        prep
GPT             PROPN      pobj
and             CCONJ      cc
BERT            PROPN      conj
have            AUX        aux
revolutionized  VERB       ROOT
text            NOUN       compound
generation      NOUN       dobj
and             CCONJ      cc
understanding   NOUN       conj
.               PUNCT      punct
Speech          PROPN      compound
recognition     NOUN       compound
systems         NOUN       nsubj
rely            VERB       ROOT
on              ADP        prep
NLP             PROPN      pobj
to              PART       aux
convert         VERB       xcomp
spoken          VERB       amod
words           NOUN       dobj
into            ADP        prep
text            NOUN       pobj
.               PUNCT      punct
Ethical         ADJ        amod
concerns        NOUN       nsubj
in              ADP        prep
NLP             PROPN      pobj
include         VERB       ROOT
bias            NOUN       dobj
in              ADP        prep
language        NOUN       compound
models          NOUN       pobj
and             CCONJ      cc
the             DET        det
misuse          NOUN       conj
of              ADP        prep
AI              PROPN      npadvmod
-               PUNCT      punct
generated       VERB       amod
```

```
text            NOUN      pobj
.               PUNCT     punct
```

In [8]:

```
# 5. Noun Phrase Chunking

print("\nNoun Phrase Chunking:")
for chunk in doc.noun_chunks:
    print(f"Chunk: {chunk.text} | Root: {chunk.root.text} | Dep: {chunk.root.dep_}
Noun Phrase Chunking:
Chunk: Natural Language Processing | Root: Processing | Dep: nsubj | Head: is
Chunk: NLP | Root: NLP | Dep: appos | Head: Processing
Chunk: a field | Root: field | Dep: attr | Head: is
Chunk: AI | Root: AI | Dep: pobj | Head: of
Chunk: that | Root: that | Dep: nsubj | Head: focuses
Chunk: the interaction | Root: interaction | Dep: pobj | Head: on
Chunk: computers | Root: computers | Dep: pobj | Head: between
Chunk: human language | Root: language | Dep: conj | Head: computers
Chunk: It | Root: It | Dep: nsubj | Head: enables
Chunk: machines | Root: machines | Dep: nsubj | Head: read
Chunk: human language | Root: language | Dep: dobj | Head: interpret
Chunk: NLP techniques | Root: techniques | Dep: nsubj | Head: include
Chunk: tokenization | Root: tokenization | Dep: dobj | Head: include
Chunk: stemming | Root: stemming | Dep: conj | Head: tokenization
Chunk: lemmatization | Root: lemmatization | Dep: conj | Head: stemming
Chunk: speech | Root: speech | Dep: pobj | Head: of
Chunk: text data | Root: data | Dep: dobj | Head: process
Chunk: Sentiment analysis | Root: analysis | Dep: nsubj | Head: is
Chunk: a popular NLP application | Root: application | Dep: attr | Head: is
Chunk: the emotional tone | Root: tone | Dep: dobj | Head: determine
Chunk: text | Root: text | Dep: pobj | Head: of
Chunk: Chatbots | Root: Chatbots | Dep: nsubj | Head: leverage
Chunk: virtual assistants | Root: assistants | Dep: conj | Head: Chatbots
Chunk: NLP | Root: NLP | Dep: dobj | Head: leverage
Chunk: human-like conversations | Root: conversations | Dep: pobj | Head: in
Chunk: Entity Recognition | Root: Recognition | Dep: nsubj | Head: identifies
Chunk: NER | Root: NER | Dep: appos | Head: Recognition
Chunk: key entities | Root: entities | Dep: dobj | Head: identifies
Chunk: names | Root: names | Dep: pobj | Head: like
Chunk: dates | Root: dates | Dep: conj | Head: names
Chunk: locations | Root: locations | Dep: conj | Head: dates
Chunk: text | Root: text | Dep: pobj | Head: in
Chunk: Machine translation | Root: translation | Dep: nsubjpass | Head: powered
Chunk: Google Translate | Root: Translate | Dep: pobj | Head: as
Chunk: NLP | Root: NLP | Dep: pobj | Head: by
Chunk: text | Root: text | Dep: dobj | Head: convert
Chunk: languages | Root: languages | Dep: pobj | Head: between
Chunk: NLP models | Root: models | Dep: nsubj | Head: revolutionized
Chunk: GPT | Root: GPT | Dep: pobj | Head: like
Chunk: BERT | Root: BERT | Dep: conj | Head: GPT
Chunk: text generation | Root: generation | Dep: dobj | Head: revolutionized
Chunk: understanding | Root: understanding | Dep: conj | Head: generation
Chunk: Speech recognition systems | Root: systems | Dep: nsubj | Head: rely
Chunk: NLP | Root: NLP | Dep: pobj | Head: on
Chunk: spoken words | Root: words | Dep: dobj | Head: convert
Chunk: text | Root: text | Dep: pobj | Head: into
Chunk: Ethical concerns | Root: concerns | Dep: nsubj | Head: include
Chunk: NLP | Root: NLP | Dep: pobj | Head: in
Chunk: bias | Root: bias | Dep: dobj | Head: include
Chunk: language models | Root: models | Dep: pobj | Head: in
Chunk: the misuse | Root: misuse | Dep: conj | Head: bias
Chunk: AI-generated text | Root: text | Dep: pobj | Head: of
```

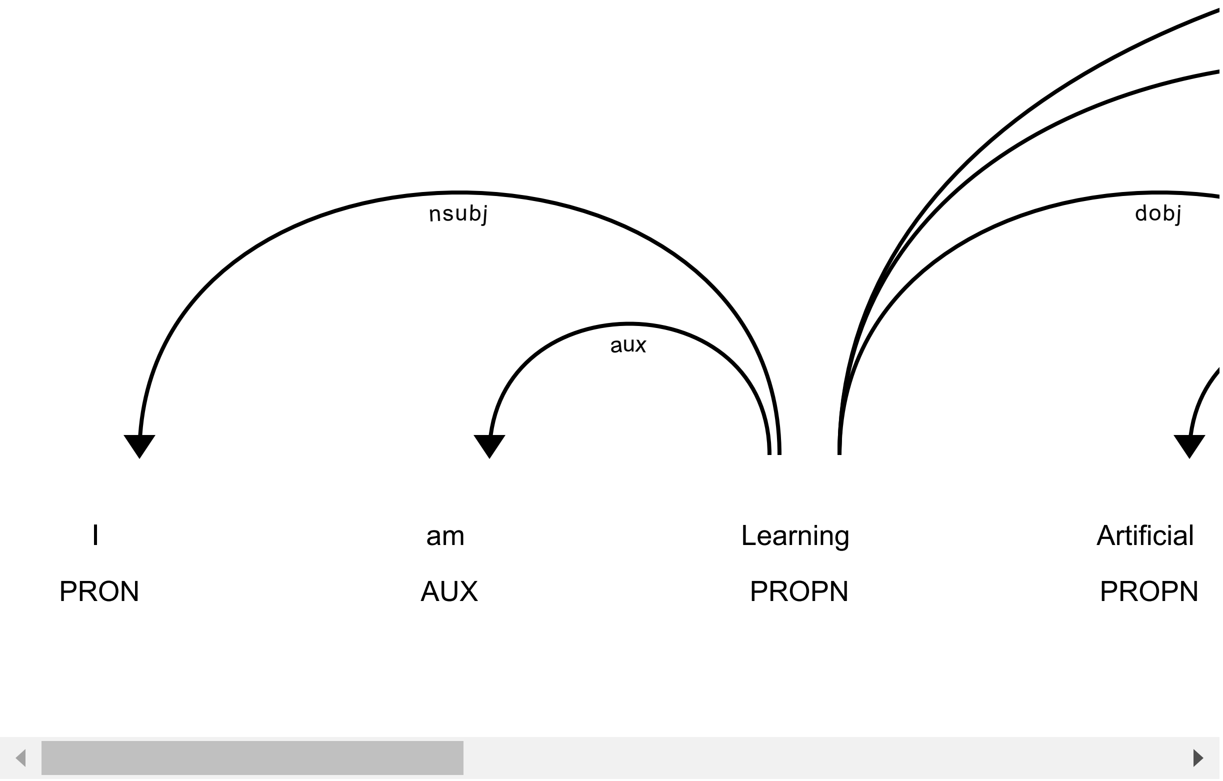
In [9]:

```
# 6. Dependency Parsing
```

```
print("Dependency Parsing Visualization:")

displacy.render(nlp("I am Learning Artificial Intelligence at 11:40AM in MA112.") ,
```

```
Dependency Parsing Visualization:
```


nsubj
aux
dobj
I
PRON
am
AUX
Learning
PROPN
Artificial
PROPN