

# COMP 6721 Applied Artificial Intelligence (Fall 2023)

## Lab Exercise #10: Introduction to NLP

**Question 1** Assume that we are working with the Shloutan language. If you don't know Shloutan, don't worry; it is a simple language made of only 5 words: loola nikee aloka bibi vo.

You want to build a word language model for Shloutan. The training corpus that you use is the following:

“Loola nikee. Aloka bibi vo. Vo bibi loola. Loola nikee bibi vo. Vo. Vo. Aloka bibi loola. Loola aloka aloka. Loola loola. Nikee nikee nikee. Bibi vo. Bibi vo. Vo Vo. Nikee loola.”

You can ignore case distinctions and sentence boundaries when answering the following questions.

(a) What is the value of  $P(\text{vo} \mid \text{bibi})$ ?

(b) What is the value of  $P(\text{bibi vo})$ ?

- (c) Build a bigram language model based on this training corpus. Show the frequencies and the probabilities for each bigram.

**Frequencies**

	loola	nikee	aloka	bibi	vo
loola					
nikee					
aloka					
bibi					
vo					

**Conditional Probabilities**

	loola	nikee	aloka	bibi	vo
loola					
nikee					
aloka					
bibi					
vo					

- (d) Smooth your bigram language model using “add 0.5”. Show the frequencies and the probabilities for each bigram.

**Frequencies**

	loola	nikee	aloka	bibi	vo
loola					
nikee					
aloka					
bibi					
vo					

**Conditional Probabilities**

	loola	nikee	aloka	bibi	vo
loola					
nikee					
aloka					
bibi					
vo					

- (e) Using each language model from parts (c) and (d), which of the following two sentences is more probable. Show all your work.

**sentence 1: Aloka vo nikee aloka.**

**sentence 2: Vo nikee nikee aloka.**

**Question 2** Consider the following context-free grammar:

$S \rightarrow NP VP$	$N \rightarrow \text{lion}$
$VP \rightarrow V$	$N \rightarrow \text{knife}$
$VP \rightarrow V NP$	$N \rightarrow \text{zoo}$
$VP \rightarrow VP PP$	$V \rightarrow \text{attacked}$
$NP \rightarrow DET N$	$PN \rightarrow \text{Jane}$
$NP \rightarrow PN$	$DET \rightarrow \text{the}$
$NP \rightarrow NP PP$	$DET \rightarrow \text{a}$
$PP \rightarrow PREP NP$	$PREP \rightarrow \text{with}$
	$PREP \rightarrow \text{in}$

(a) Generate all possible parse trees for the following sentence:

i *Jane attacked a lion with a knife*

**Question 3** Consider the following sentences as training set for disambiguating the sense of the word *light*.

Sentence	Sense
There is a ray of light even if it is dim.	Sense1
This faint ray of light is barely enough to see.	Sense1
It needs to be light in weight to be portable.	Sense2
I can see dim red light coming from that room.	Sense1
This laptop is light enough to carry around.	Sense2
I am not hurt since I was hit by something as light as a candle.	Sense2
There is a faint hint of light at the end of the street.	Sense1
But the room was bright and light from the candle was warm as it fell on the surroundings.	Sense1
My bright red overcoat is light but fairly warm.	Sense2
This red candle stand is light enough to carry it to the room.	Sense2

Further consider this list of stop words:

*a, and, are, as, at, be, for, from, in, is, it, if, of, that, this, the, to, was*

Using a Naive Bayes approach with:

- a context window of  $\pm 3$  words,
- a vocabulary with the size of 31,
- smoothing with the value of 0.2, and
- stop-word removal

Calculate the scores of each possible sense and find the most probable sense of the word *light* in the following sentence:

*In her dark red room, that dim light from the candle was warm enough.*

**Question 4** Preprocessing text is a very crucial step for any NLP related task. There are many frameworks (GATE<sup>1</sup>, UIMA<sup>2</sup>) and libraries (CoreNLP<sup>3</sup>, NLTK<sup>4</sup>, spaCy<sup>5</sup>) available out there to facilitate text analysis.

Today we will be looking into spaCy to get you started with the basic steps.

**Installing spaCy.** To install spaCy using Conda, use:

---

```
conda install -c conda-forge spacy
```

---

After the installation, execute the following command to verify that you have installed spaCy version 3 or upper.

---

```
python -m spacy info
```

---

**Downloading language models.** Not all languages are preprocessed in the same way. Depending on the language, tokenization, sentence splitting, POS tagging, etc. varies. To facilitate analyzing different languages and genres, spaCy provides different pre-trained language pipelines to work with.<sup>6</sup> For starters, we will be working with the basic English model. We can download the model with the following command:

---

```
python -m spacy download en_core_web_sm
```

---

**Let's start coding!** Use the following to import the spaCy library and load the language model:

---

```
import spacy
nlp = spacy.load("en_core_web_sm")
```

---

Now let's try to preprocess the text in your worksheet and check if the POS tags we assigned are what spaCy returns as well:

---

```
doc = nlp("I prefer a direct flight to Chicago.")
```

---

Depending on the language model and sentence provided as input, spaCy outputs an object `doc`, in our case with a variety of annotations encoded within. Linguistic annotations are available as token attributes. Let's try to print some basic attributes:

---

<sup>1</sup><https://gate.ac.uk/>

<sup>2</sup><http://uima.apache.org/>

<sup>3</sup><https://stanfordnlp.github.io/CoreNLP/index.html>

<sup>4</sup><https://www.nltk.org/book/>

<sup>5</sup><https://spacy.io/usage/spacy-101>

<sup>6</sup><https://spacy.io/usage/models>

---

```
for token in doc:
    print(token.text, token.lemma_, token.tag_, token.dep_)
```

---

- (a) Now try to process the following paragraph to see how `spaCy` performs sentence splitting:

*“Superman was born on the planet Krypton and was given the name Kal-El at birth. As a baby, his parents sent him to Earth in a small spaceship moments before Krypton was destroyed in a natural cataclysm. His ship landed in the American countryside, near the fictional town of Smallville. He was found and adopted by farmers Jonathan and Martha Kent, who named him Clark Kent. Clark developed various superhuman abilities, such as incredible strength and impervious skin. His adoptive parents advised him to use his abilities for the benefit of humanity, and he decided to fight crime as a vigilante. To protect his privacy, he changes into a colorful costume and uses the alias “Superman” when fighting crime. Clark Kent resides in the fictional American city of Metropolis, where he works as a journalist for the Daily Planet. Superman’s supporting characters include his love interest and fellow journalist Lois Lane, Daily Planet photographer Jimmy Olsen and editor-in-chief Perry White. His classic foe is Lex Luthor, who is either a mad scientist or a ruthless businessman, depending on the story.”*

Start by processing the text:

---

```
doc = nlp(text)
```

---

This performs *sentence splitting* as part of the analysis pipeline. Now add code to print all the resulting sentences.

- (b) In addition to the constituent parse trees discussed in the lecture and Question 2 above, dependency parsing<sup>7</sup> is another formalism used in English grammars. In dependency parse trees, each token (word) is connected to another by a single arc, where the arc’s label indicates the syntactic relationship between them. The starting node of the arc is known as the *Governor* (or *head* in `spaCy` terminology), and the endpoint is the *Dependant* (called *child* in `spaCy`). Each ‘Dependant/Child’ can have only one ‘Governor/Head,’ but a ‘Governor/Head’ may have multiple dependents. Every dependency tree also has a root, typically the main verb of the sentence, and the arcs point from heads to their children.

When you process text with `spaCy`’s `nlp` function (like in `doc = nlp(text)`), a dependency parse tree is automatically created for each sentence. Now,

---

<sup>7</sup>See <https://web.stanford.edu/~jura/sky/slp3/18.pdf> and <https://spacy.io/usage/linguistic-features#dependency-parse> for more details on dependency parse trees

try to use **spaCy**’s “displaCy visualizer”<sup>8</sup> to visualize the dependency parse tree for the sentence given in your worksheet — the result should look like Figure 1.

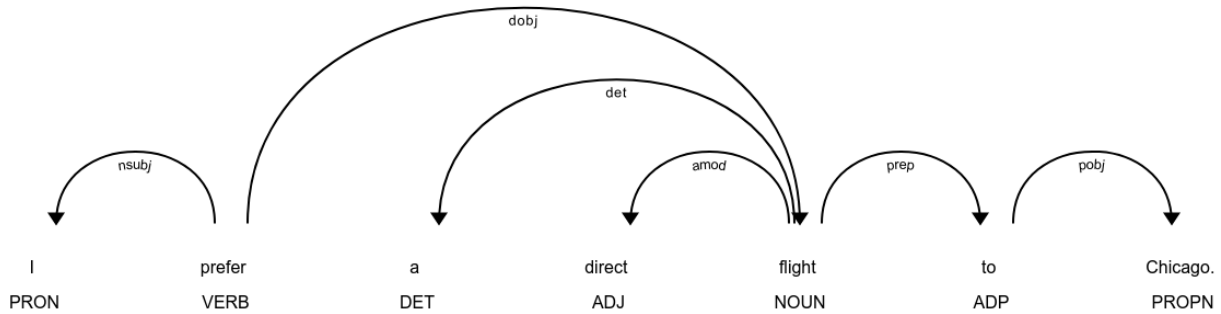


Figure 1: Dependency parse tree created by **spaCy** for the sentence we used on the worksheet

- (c) Now we will look at *Named Entity Recognition* (NER), an essential task in natural language processing (NLP) that involves identifying and classifying proper nouns in text into predefined categories such as names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc. NER is crucial for understanding the content of a document by highlighting the key entities and is widely used in information extraction, content classification, and data analysis.

Using **spaCy**, we can easily perform NER with minimal coding. **spaCy**’s NER model is pre-trained on a large corpus and can recognize a wide range of entity types. Start by processing the following text with **spaCy** and explore the identified named entities:

---

```

text = "Apple is looking at buying U.K. startup for $1 billion. Barack
      ↪ Obama visited Apple headquarters in California."
  
```

---

Process this text using the **spaCy** pipeline and iterate through the named entities in the processed document, which are stored in the **ents** property of a **doc** object. For each entity, print out its text and the entity type it has been classified as. For example, “Apple” should be recognized as an *Organization* and “Barack Obama” as a *Person*.

This lab marks your initial foray into Natural Language Processing (NLP), a foundational aspect of AI that bridges language and technology. The experience gained with **spaCy** is crucial for understanding how machines interpret human language. The practical skills you’ve developed are applicable in various real-world scenarios, from enhancing search engine algorithms to improving

---

<sup>8</sup>See <https://spacy.io/usage/visualizers>



automated customer support systems. NLP's relevance spans multiple industries, including healthcare, finance, and media, where insights from text data are invaluable. As you delve deeper into NLP, you'll find the field replete with opportunities for innovation and advancement. Continue building upon these skills, and you may soon find yourself adept at creating applications that significantly impact the way we interact with technology in language-driven contexts.