

Lab 1: The objective of this lab is to make use of the knowledge-based pipeline model for natural language processing (NLP).

Learning objectives:

- Utilize existing NLP libraries (e.g. spacy) to:
 - Observe the components that make up the knowledge-based pipeline
 - Steadily add each component of the NLP pipeline
 - Observe in action the purpose and operations of each component
- Experiment with a blank pipeline and a pretrained model, and observe the performance difference between them

Procedure:

- Download the zip file called *Lab1_Week2* and extract it. This zip file contains:
 - An empty Python code for you to fill in
 - This manual for the lab
- The Python code consists of several comments that indicate where you should fill in
 - First, observe the NLP pipeline. Note that even though the pipeline shows nothing, it still has the tokenizer by default.
 - Then, add in the following components to the NLP pipeline:
 - PoS-tagging
 - Parser
 - Named Entity Recognizer (NER)
 - Observe the results of each of the above components
 - Load in a pre-trained model of the NLP pipeline (i.e. `spacy.load("en_core_web_sm")`) and test its performance for each of the following components:
 - PoS-tagging
 - Parser
 - NER

Question:

1. Please explain what the following lines do (note that `text` is a string, as indicated in the code):
 - `nlp = spacy.blank("en")`
 - `doc = nlp(text)`
2. If you start with a blank pipeline, and then add in the components manually (i.e. PoS-tagging, Parser, NER), what is the corresponding performance?
 - Is the pipeline able to do PoS-tagging, Parsing, and Named-Entity Recognizing? If not, then why? Consult online documentations for the Spacy library for answers
3. How do you think the model identified the named-entity recognition?

Deliverables:

- Your code that fills in the blanks as indicated by the comments in *Lab1_Week2_empty.py*
- A report that details the results you have found. It should contain the following information:
 - Introduction to the NLP pipeline
 - Answers to the above questions. Include the results from your code (as specified by the questions)
 - Concluding remark of what you have learned from this lab

Your implemented code should have the following (you will be marked for both):

- Correctness: The code should compile without error
- Style: There should be comments and proper formatting