




Welcome to the CoGrammar Theory Summary II

The session will start shortly...

Questions? Drop them in the chat. We'll have dedicated moderators answering questions.



Data Science Session Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly.
(Fundamental British Values: Mutual Respect and Tolerance)
- No question is daft or silly - **ask them!**
- There are **Q&A sessions** midway and at the end of the session, should you wish to ask any follow-up questions. Moderators are going to be answering questions as the session progresses as well.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Academic Sessions. You can submit these questions here: [Questions](#)

Data Science Session Housekeeping cont.

- For all **non-academic questions**, please submit a query: www.hyperiondev.com/support
- Report a **safeguarding** incident: www.hyperiondev.com/safeguardreporting
- We would love your **feedback** on lectures: [Feedback on Lectures](#)

Skills Bootcamp

8-Week Progression Overview

Fulfil 4 Criteria to Graduation

✓ Criterion 1: Initial Requirements

Timeframe: First 2 Weeks

Guided Learning Hours (GLH):

Minimum of 15 hours

Task Completion: First four tasks

Due Date: 24 March 2024

✓ Criterion 2: Mid-Course Progress

60 Guided Learning Hours

Data Science - **13 tasks**

Software Engineering - **13 tasks**

Web Development - **13 tasks**

Due Date: 28 April 2024

Skills Bootcamp Progression Overview

✓ Criterion 3: Course Progress

Completion: All mandatory tasks, including Build Your Brand and resubmissions by study period end
Interview Invitation: Within 4 weeks post-course
Guided Learning Hours: Minimum of 112 hours by support end date (10.5 hours average, each week)

✓ Criterion 4: Demonstrating Employability

Final Job or Apprenticeship Outcome: Document within 12 weeks post-graduation
Relevance: Progression to employment or related opportunity

CoGrammar

Theory Summary II

April 2024

Learning objectives

- ❖ Grasp the concept of object-oriented programming
- ❖ Understand fundamental concepts of Natural Language Processing

Object-Oriented Programming (OOP)



Introduction

- ❖ OOP is a programming paradigm that revolves around the concept of objects and classes.
 - **Classes:** the blueprint for creating objects e.g., 'Car' class with the property 'colour' and the method 'drive()'
 - **Objects:** an instance of a class e.g., an object of the 'Car' class is 'red Toyota' with the method 'drive()'
- ❖ Key idea: objects are instances of classes, which define the data and behaviour of the objects

Key Concepts of OOP

- ❖ **Encapsulation**: the bundling of data and methods that operate on the data within one unit, e.g., a class.
 - Purpose: To hide the internal state of the object and only expose a controlled interface.
- ❖ **Abstraction** which involves simplifying complex systems by modeling classes appropriate to the problem.
 - Purpose: To reduce complexity and allow the programmer to focus on interactions at a higher level.
- ❖ **Inheritance** is a mechanism by which one class (child) can inherit the properties and methods of another class (parent).
 - To promote code reuse and establish a natural hierarchy.
- ❖ **Polymorphism** allows objects of different classes to be treated as objects of a common superclass.
 - To enable one interface to be used for a general class of actions.



Benefits of OOP

Think of OOP as a divide and conquer approach to programming. It breaks down complex problems into manageable, interacting objects.

Benefits of this approach are:

- ❖ **Modularity**: code can be written and maintained independently
- ❖ **Reusability**: classes can be reused across programs
- ❖ **Scalability**: easy to add new features or change existing ones
- ❖ **Maintainability**: simplifies troubleshooting and debugging



Example

- ❖ We want to create a banking system
- ❖ We create an 'Account' class with the following properties and methods:
 - Properties: 'balance' and 'accountNumber'
 - Methods: 'deposit()', 'withdraw()' and 'get_balance()'
- ❖ We can create an instance of this 'Account' class by opening an account for a customer - this customer's account becomes an object of the 'Account' class.
- ❖ Principles of OOP demonstrated:
 - Encapsulation: 'balance' in 'Account' can be private and accessed via 'get_balance()'
 - Abstraction: customers can interact with 'Account' methods without knowing the internal workings i.e., they can deposit and withdraw money
 - Inheritance: we can extend the 'Account' class by creating 'SavingsAccount' and 'CheckingAccount' which inherit from 'Account'

Natural Language Processing

CoGrammar



NLP Components and Levels

Natural Language Processing

Natural Language Understanding

Natural Language Generation

amazon

- ❖ **Morphological/Lexical analysis:** processing and understanding POS.
- ❖ **Syntactic analysis:** understanding the sentence structure.
- ❖ **Semantic analysis:** understanding literal meaning of words, phrases, sentences.
- ❖ **Discourse analysis:** understanding units larger than single sentence
- ❖ **Pragmatic analysis:** using real-world knowledge to understand the bigger context of the sentence.

Lexical Analysis

Syntax Analysis

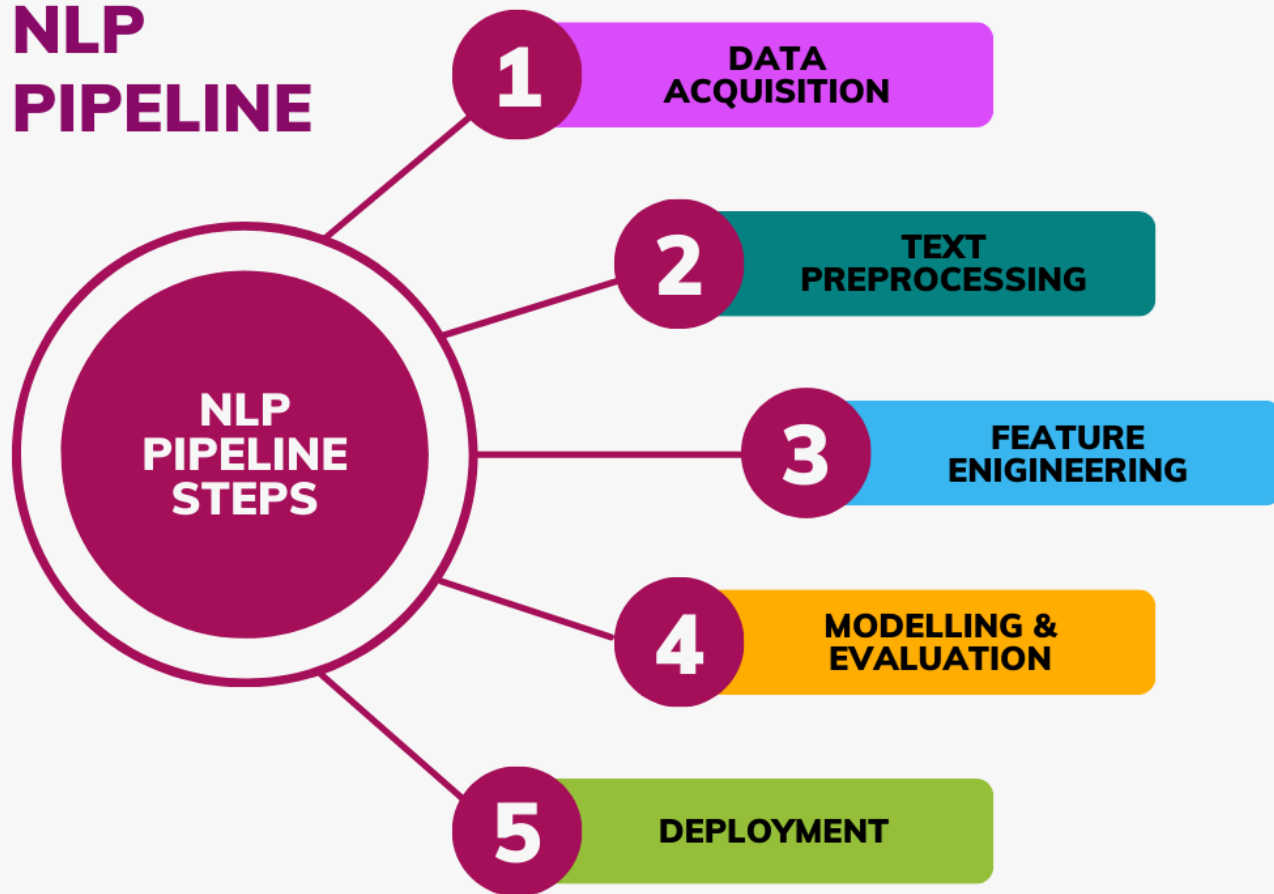
Semantic Analysis

Discourse Analysis

Pragmatic Analysis

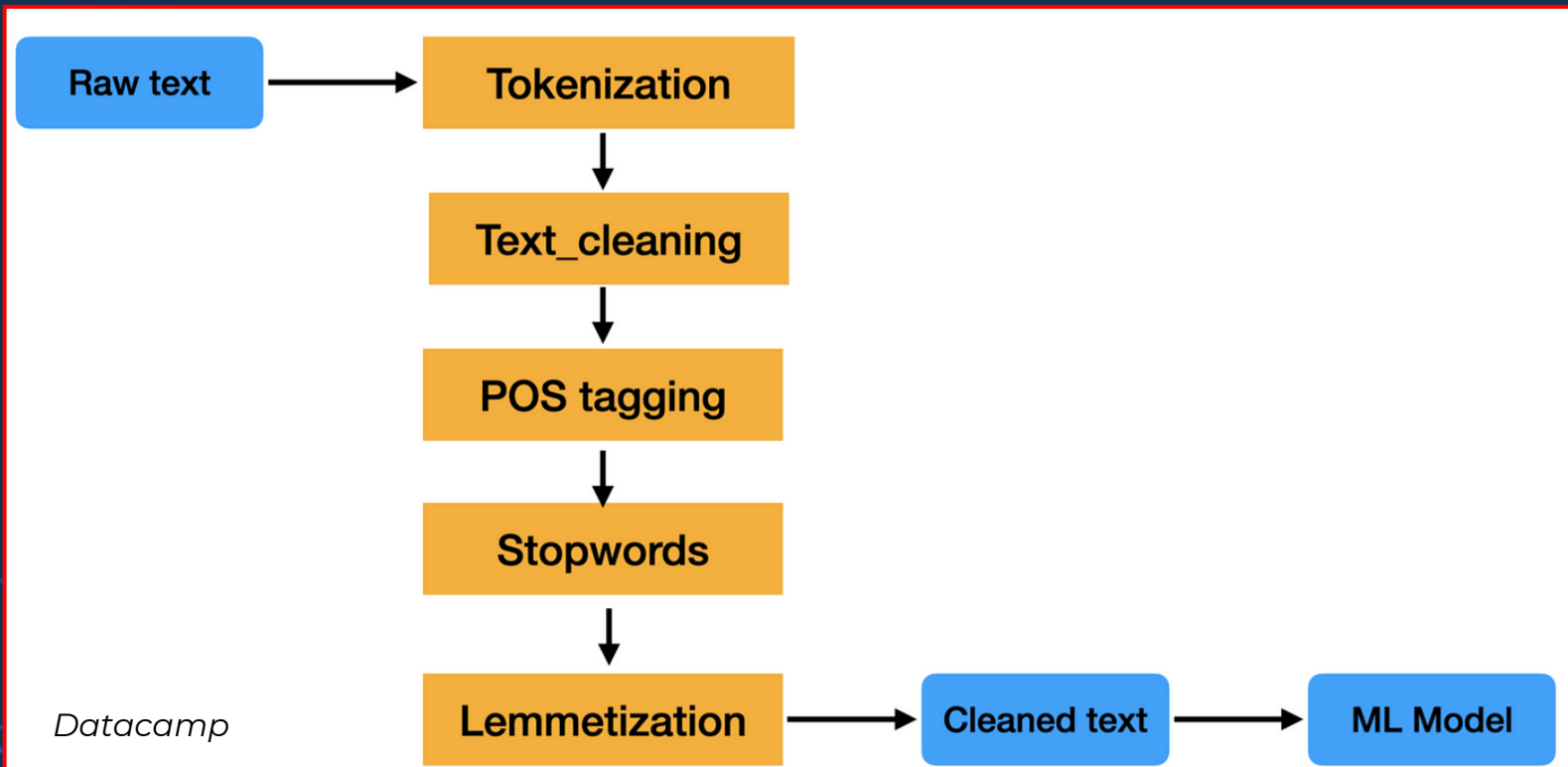
NLP Pipeline

NLP PIPELINE



medium

Text Preprocessing



Semantic Similarity

- ❖ **Semantic similarity** is about the meaning closeness, and **lexical similarity** is about the closeness of the word set.
 - *“The dog bites the man”* and *“The man bites the dog”*
 - Identical considering lexical similarity; however entirely different considering semantic similarity
- ❖ **Cosine similarity** in NLP domain: measures the cosine of the angle between vectors of two points.
- ❖ Use **spacy** pre-trained model with embeddings “**en_core_web_md**” and ‘**similarity**’ to calculate the similarities between embeddings.

Questions and Answers



Thank you for attending



Department
for Education

CoGrammar

