Introduction to AI, LLMs and NLP

Francesca Carlon

July 2026

This two-week course offers an accessible introduction to Artificial Intelligence, Natural Language Processing (NLP), and Large Language Models (LLMs) for bachelor's and master's students from any discipline. Starting from the foundations of human language—how sounds, words, and grammar create meaning—the course gradually moves toward the computational methods that support real-world products like chatbots, translators, and voice assistants. Through interactive labs and a team mini-project, participants gain practical experience with tools such as spaCy, NLTK, Hugging Face Transformers and LLMs platforms. By the end of the course, students will understand both the theoretical and the industry relevance of modern NLP, and will be able to design and present a small working prototype of their own.

Duration: 2 weeks (10 days, 3 hours per day)
Audience: Bachelor's and Master's students, mixed backgrounds.
Structure: Lecture/discussion (75 min) ; Hands-on/lab (90 min) ; Wrap-up (15 min)

1 Day 1 - Human Language and AI

- **Lecture**: What Is Language?
  Exploring Linguistic Levels such as Phonetics and Phonology, Morphology, Syntax, Semantics, Pragmatics with relevant examples from morphology-rich and analytic languages.
  Relevance with NLP and Human to Machine connection from early linguistic approaches (Chomsky, rule-based) to statistical to deep learning.
  Connection of NLP use within industry (chatbots, speech assistants, translation).
- **Lab**: Short multi-language annotation (POS, morphology, syntactic chunks).
- **Outcomes**: Recognize complexity of human language; map linguistic layers to modern NLP tasks.

2 Day 2 - Words, Meaning and Data

- **Lecture**: Tokens, types, lemmas. Frequency, Zipf's law. Corpora: corpus design and sampling bias. Vector space model.
- **Lab**: explore word frequencies and collocations in a news vs. social media corpora (Python/Colab), discuss domain differences.
- **Outcomes**: Link language data to statistical patterns.

3 Day 3 - From Rules to Statistics

- **Lecture**: Show evolution from rule-based NLP to statistical methods. Regular expressions, simple grammars. N-grams, language modeling, probability basics.
- **Lab**: Build a bigram language model on a small corpus, generate "silly sentences". Regex patterns to find dates/emails/hashtags.
- **Outcomes**: Understand the basics of NLP methods.

4 Day 4 - Modern NLP

- **Lecture**: Preprocessing: tokenization, stopwords, stemming vs. lemmatization. POS tagging and named entity recognition (NER). Introduction to spaCy, NLTK.
- **Lab**: groups manually label entities in sample tweets, then compare with NLP libraries output. Explore spaCy pipeline components and visualize dependency trees; explore NLTK tools for NLP analysis.
- **Outcomes**: Understand typical NLP workflow. Understand standard preprocessing and tagging tasks. Get hands-on with spaCy and NLTK.

5 Day 5 - Word Meaning and Embeddings

- **Lecture**: Distributional hypothesis, vector semantics, Word2Vec/GloVe intuition, cosine similarity.
- **Lab**: Explore pre-trained embeddings (gensim or spaCy vectors): nearest neighbors, analogy tasks. Make word clusters.
- **Outcomes**: Understand distributional semantics and vector space models. Experiment with pre-trained embeddings.

6 Day 6 - Deep Learning for NLP

- **Lecture**: Fundamentals of feed-forward nets, RNNs, attention. Sequence-to-sequence tasks (translation, summarization).
- **Lab**: run a pre-built sentiment classifier notebook; tweak hyperparameters or input size to observe changes. Compare RNN vs. bag-of-words baseline.
- **Outcomes**: Grasp fundamentals of neural nets, sequence modeling and attention.

7 Day 7 - Transformers and LLMs

- **Lecture**: Transformer architecture. Pre-training vs. fine-tuning. Prompt Engineering methods. Natural language coding.
- **Lab**: Compare outputs of a small open model on HuggingFace with a large one. Compare outputs from the same prompt. Coding with free models.
- **Outcomes**: Understand the Transformer architecture. Practice basic prompting and model comparison. Understand how natural language coding works and how to apply it to individual cases.

8 Day 8 - Mini-Project Build Day (pt. 1)

- **Lecture**: Project design and dataset selection.
  Teams define a business-relevant NLP/LLM application and outline a processing pipeline.
- **Lab**: Begin implementation. Example options: Chatbot for a local business, Sentiment dashboard, Domain-specific NER, Simple text summarizer.
- **Outcomes**: Apply the week's knowledge to a practical, industry-relevant task. Develop coding skills with natural language coding.

9 Day 9 - Mini-Project (pt. 2) and Presentation

- **Lecture**: Q&A and final mentoring.
- **Lab**: Continue projects. Team presentations (10 min each + 5 min Q&A); peer feedback.
- **Outcomes**: Communicate technical work and relate skills to real-world career paths.

10 Day 10 - Ethics, Bias and Responsible AI

- **Lecture**: AI safety, bias, fairness, privacy, environmental impact, explainability and limitations.
- **Lab**: Inspect bias in a sentiment model (e.g., gendered names) and propose mitigation steps.
- **Outcomes**: Critically evaluate social and ethical implications of NLP and LLMs.