

Davide Modolo

MSC GRADUATE IN ARTIFICIAL INTELLIGENCE SYSTEMS @ UNIVERSITY OF TRENTO

✉ davide.modolo98@gmail.com | github.com/davidemodolo | [in linkedin.com/in/davide-modolo](https://www.linkedin.com/in/davide-modolo)

Education

University of Trento

Trento, Italy

Master's Degree in Artificial Intelligence Systems

September 2021 - March 2025

- **Thesis:** Exploring the Use of LLMs for Agent Planning: Strengths and Weaknesses. [[report](#)] [[code](#)]
- **Relevant Courses:** Machine Learning, Deep Learning, Natural Language Understanding, Automated Planning, Law & Ethics in AI, Fundamentals of AI (Reasoning and Planning), Signal Processing, AI for Finance, High-Performance Computing.

University of Trento

Trento, Italy

Bachelor's Degree in Computer Science

September 2017 - June 2021

- **Thesis:** Healthy Plus - Redesign and evolution of an Android application for monitoring healthy lifestyles.
- **Relevant Courses:** Algorithms and Data Structures, Advanced Algorithms, Database, Software Engineering, Networks, Geometry and Linear Algebra, Calculus 1, Computer Architectures, Operating Systems, Mobile Programming (Android).

Work Experience

Eurecat Technology Center

Barcelona, Spain

INTERN / AI ENGINEER

April 2024 - June 2024

- Explored log-probability-based uncertainty of Large Language Models (LLMs) applied to planning and choice-selection. By generating only 1 token I was able to reconstruct the uncertainty of the model on a specific choice (up to 20 choices per single prompt).
- Approach tried with *GPT 3.5* and *GPT 4o* using OpenAI API and *Llama 3.1* with llama.cpp.
- Created a CSV-based collection of preferences (using *gte-large-en-v1.5* embeddings) that would help reducing the uncertainty in recurrent tasks thanks to Retrieval-Augmented Generation (RAG).
- Evaluated the implementation on 5 simulated scenarios plus many non-standardized scenarios.
- Started the implementation of a Multimodal Model (LLaVA) to include in the system.

Python, Jupyter Notebook, GitHub, Hugging Face, OpenAI API, llama.cpp, Ollama

University of Trento

Trento, Italy

INTERN / ANDROID APP DEVELOPER

January 2021 - May 2021

- Rebuilt a modular Android 11 app for tracking healthy habits using Kotlin and Firebase.
- Implemented Google login, NoSQL storage, and designed 23+ UI screens.
- Enabled future extensibility; at least one module added post-graduation by other students.

Kotlin, Firebase, NoSQL, XLM, GitHub, Figma

Projects

COVID-19 Lung Ultrasound Images classification - Medical Imaging Diagnostic

[[slides](#)] [[report](#)] [[code](#)]

- Designed a Multi-stage deep learning model to classify Lung Ultrasound images based on a 0-to-3 illness score.
- Developed the 3 components of the model: a multi-class frame classifier, an uncertainty detection model and a similarity module.
- Dataset composed by anonymized images from 14 real patients for a total of 47k frames used to train and fine-tune various Computer Vision models.

Python, Jupyter Notebooks, PyTorch, sci-kit learn, Pandas, NumPy, Azure

Domain Adaptation / Transfer Learning - Deep Learning - Team Project

[[code](#)]

- Built, trained and evaluated a deep learning model on a standard setting of Unsupervised Domain Adaptation.
- Solution based on ResNet34 plus a custom adaptation layer with loss based on direct 3rd-order and grouped 4th-order statistics.
- Approach tested on the Adaptope dataset, reaching a +11.53% improvement with respect to the baseline non-adapted model.

Python, PyTorch, Google Colab

Joint Intent Detection and Slot Filling - Natural Language Understanding

[[report](#)] [[code](#)]

- Implemented 4 Deep Learning models to simultaneously learn Intent Detection and Slot Filling tasks.
- Reached the goal to improve consistently by at least 2% the base results on the ATIS dataset in both tasks, improved the baseline by 13% on the SNIPS dataset in the slot filling task.
- Fine-tuned BERT and ERNIE, built from scratch a model based on Bidirectional LSTM and an Encoder-Decoder model.

Python, Jupyter Notebooks, PyTorch

Autonomous Delivery BDI Agent - Autonomous Software Agents - Team Project

[[slides](#)] [[report](#)] [[code](#)]

- Designed a solution for a Delivery problem using an autonomous agent following the Belief-Desire-Intent framework.
- Developed and tested both single-agent and collaborative multi-agent environments.
- Successfully run the agents in a total of 7 challenges with 2 different multi-agent approaches.

JavaScript, Node.js, PDDL

Parallel Closest Pair of Points - High Performance Computing - Team Project

[\[slides \]](#) [\[report \]](#) [\[code \]](#)

- Provided a solution to Closest Pair of Points problem in N dimensions with a parallel implementation.
- Compared both Bruteforce and Divide et Impera approaches using Message Passing Interface (MPI) in C to their sequential approaches respectively.
- Tested with 500k to 50M points, with 1 to 80 CPU cores in 4 different CPU configurations.

C, MPI, Python

Skills

Programming Languages Python, C/C++, JavaScript, Kotlin, Flutter (Dart), Lua, Git, Docker, Azure, MPI, Firebase, ~~Android~~
English - Professional, **Italian** - Native