# Higher-Order Thinking in the Age of AI

Opposition or Allies?

Nadim Bakhshov – Senior Teaching Fellow, School of Computing, University of Portsmouth

**Student** thinking →

**Student** produces work →

Teacher evaluates work →

Teacher **infers learning + understanding** of student

Student thinking **+ AI** →

Student **+ AI** produces work →

Teacher evaluates work →

Teacher **infers learning + understanding** of student

# Assessment as Proxy for Learning

The deeper question is: What actually counts as evidence that learning has occurred? Does output in assessment artefacts prove learning has taken place?

# The Learning Question

But what has happened to **learning** in this age of LLMs?

# LLMs are designed to function as helpful conversational partners

To do so, they often exhibit **default interaction tendencies** that **reduce** friction in dialogue

In learning contexts, however, those **same tendencies** may narrow inquiry or **prematurely close down uncertainty**

Let us examine some of those default conversational tendencies in LLMs

# Resolution/Completion/Closure Defaults

**Uncertainty —** which often drives deeper learning — is treated as something to be **resolved** rather than **sustained**

Conversational systems are **optimized for closure**, but learning often depends on **sustained uncertainty**

# Interpretive Default

The **model** does the interpretive work, so the learner may **skip the reasoning** that produces understanding

# Expansion Default

The model often **over-explains**, adding material that may increase **extraneous cognitive load** rather than deepen understanding

# Cognitive Outsourcing

The issue is not simply assistance; it is the **relocation** of **core cognitive labour** from student to system.

There's a deep **risk** here

How do we know **learning** is still taking place?

How do we **counteract** these tendencies in LLMs?

# Prototype Ecosystem

Learning Coaches (IDA architecture)

**Programming:** Python 1 Coach, Python 2 Coach, Haskell Coach

**Computing Foundations:** Architecture & Logic Coach, Data Structures & Algorithms Coach

**Mathematics:** Mathematics 1 Coach

**Academic Writing:** CORA Student, CORA Staff, MIRA Reflection Tool:

**Security:** SECRYPT Lab Assistant

**Philosophy / Text Analysis:** Plato & Aristotle Readers

# Deployment Activity 1

**Recorded conversations across the prototype tools**

Programming: Python 1 Coach — **800**+ chats, Python 2 Coach — **500**+ chats

Computer Science: Architecture & Logic — **200**+ chats, Data Structures & Algorithms — **100**+ chats

Mathematics: Mathematics 1 Coach — **100**+ chats

# Deployment Activity 2

Functional Programming: Haskell Coach — **70**+ chats

Security Labs: SECRYPT Assistant — **80**+ chats

Academic Writing & Reflection:  CORA Student — **100**+ chats, CORA Staff — **100**+ chats, MIRA — **10**+ chats

**Total interactions: 2000+** over a few months (currently closer to 4000+)

# Key Quantitative Feedback

**Student Feedback — Early Pilot (n = 102**)

**Recommendation: 92%** would recommend the learning coaches

**User Ratings: 78%** rated interaction easy, **73%** rated explanations clear and helpful, **73%** rated overall experience highly helpful

**Student comments:** “It breaks problems down really well letting me catch my logical errors.”, “The coach gives hints rather than the whole answer.”, “Clear step-by-step explanations.”, “Very helpful for understanding difficult concepts.”, “You can ask basic questions and build up your understanding.”

# Positive Themes Emerging from Student Feedback

# 1. Step-by-Step Reasoning Support

Students repeatedly emphasised that the coach breaks problems down into **manageable reasoning steps**

Example quotes:

“It breaks problems down really well letting me catch my logical or code error.”

“They explain step by step what is going on.”

“Breaking down topics into clearer steps helped me understand.”

This aligns directly with the **metacognitive architecture design**

# 2. Hint-Based Guidance Instead of Giving Answers

Students noticed that the system does not simply give solutions

Quotes:

“The fact that it gives you a hint for the problems instead of the whole thing.”

“Guiding you instead of just telling you the answer.”

“It helps structure how to approach the problem.”

This strongly supports your argument about **preventing cognitive outsourcing**

# 3. Support for Understanding Difficult Concepts

Students highlighted conceptual clarity

Quotes

“Very helpful with understanding concepts.”

“Clear and simple explanations.”

“Using it to understand difficult maths / concepts.”

This indicates the system is functioning as a l**earning scaffold rather than an answer generator**

# 4. Alignment with Course Material

Students valued that the coach is built around course worksheets and tasks

Quotes:

“It has all the worksheets so it's built for the worksheet problems.”

“It knows the concepts we need to learn.”

This is important evidence that **domain-specific AI tools outperform generic assistants**

# 5. Interactive Questioning

Students appreciated the dialogic interaction

Quotes

“Helpful for asking questions continuously.”

“You can ask very basic questions and build up to the required level.”

This supports your claim that the tools function as **dialogic learning partners**

# There is an opportunity here

## Three levels make this clearer

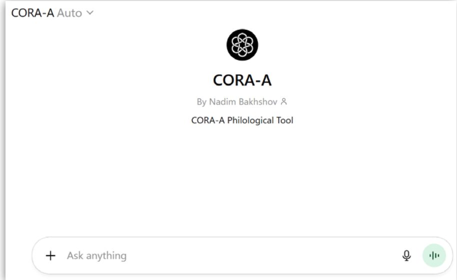
## 1 - a full university-built LLM

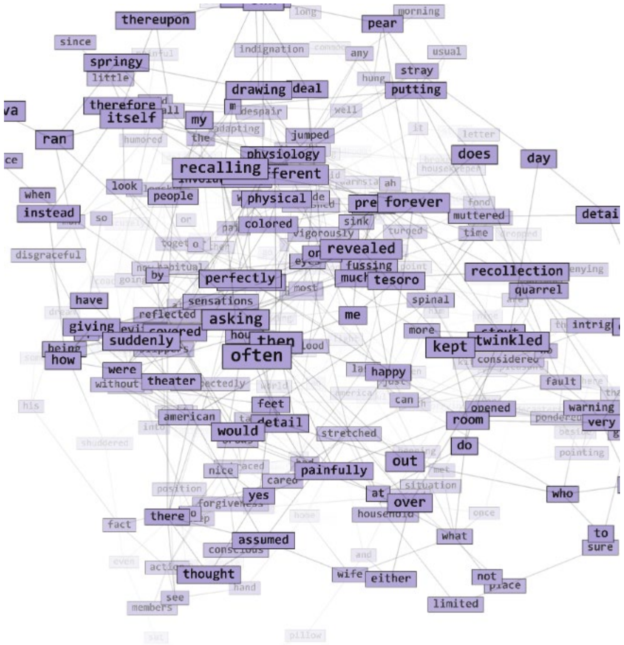
## 2 - an institutional model adapted from an existing base

## 3 - and a controlled pedagogical system built on top of an imported model

The third is usually the most feasible and the most useful, because the university can encode its own IFR logic, syllabus content, and learning behaviours without having to train a foundation model from scratch.

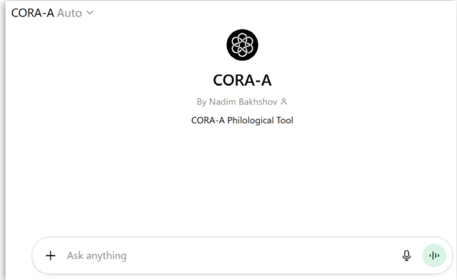
# Current

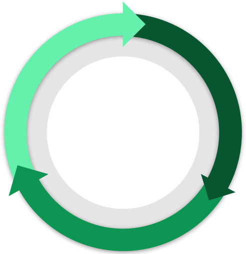


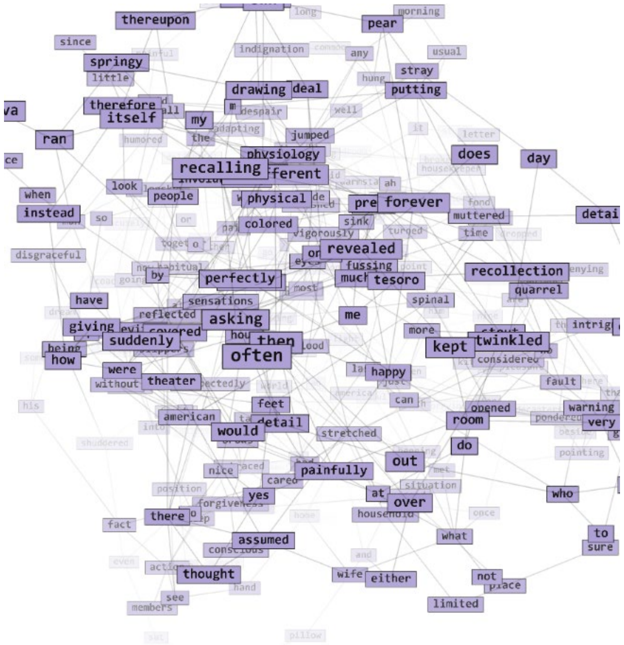




UI ↔ LLM

# Prototype Design





UI ⟷ Control Interface Layer ⟷ LLM