UNIVERSITY OF LONDON

Goldsmiths

UNIVERSITY OF LONDON

london.ac.uk

UNIVERSITY OF LONDON

Goldsmiths
UNIVERSITY OF LONDON

Applications of AI in the BSc CS degree

Professor Matthew Yee-King
Programme Director
BSc Computer Science

Summary

- Some terms
- Local LLMs for survey analysis
- LLM embeddings for collusion detection
- WIP: Generating exam papers
- WIP: course chatbo

Introduction

- PhD AI/CS (creative signal processing) and CS educator
- BSc CS fully online undergraduate programme:
- UoL, Goldsmiths Coursera
- Launched 2019, ~4000 active students

coursera

Goldsmiths
UNIVERSITY OF LONDON

UNIVERSITY OF LONDON

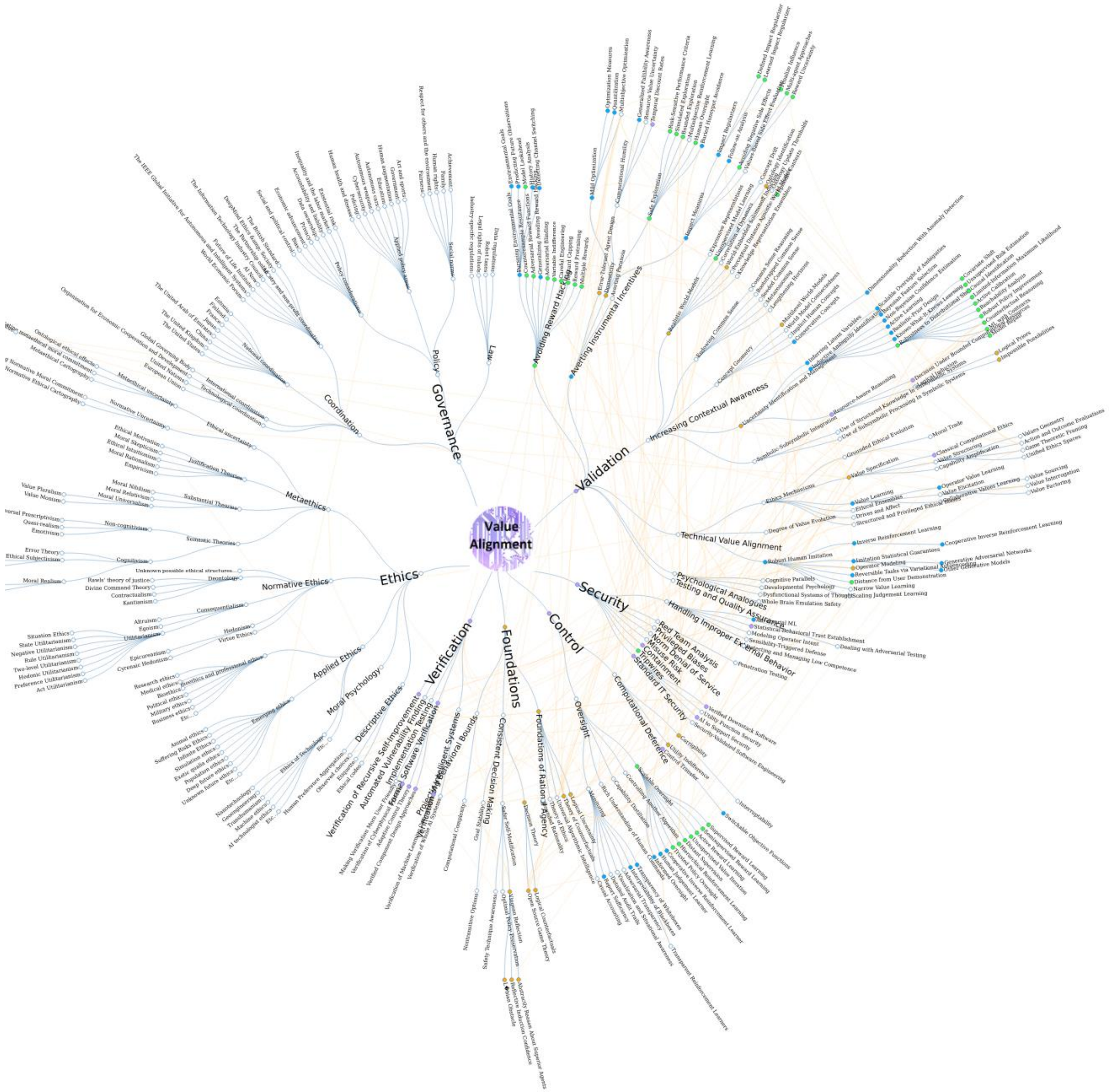
Position statement

- Enhance not replace
- Alignment: how to constrain the current and future generation of AI

Value alignment is hard

"The goal of this workshop is to bring together researchers and thought leaders from various disciplines such as computer science, social sciences, philosophy, law etc. to discuss the ethical considerations and implications of aligning AI to human values."

- https://futureoflife.org/valuealignmentmap/ (2017)



Seven Top level Categories

Governance

Ethics

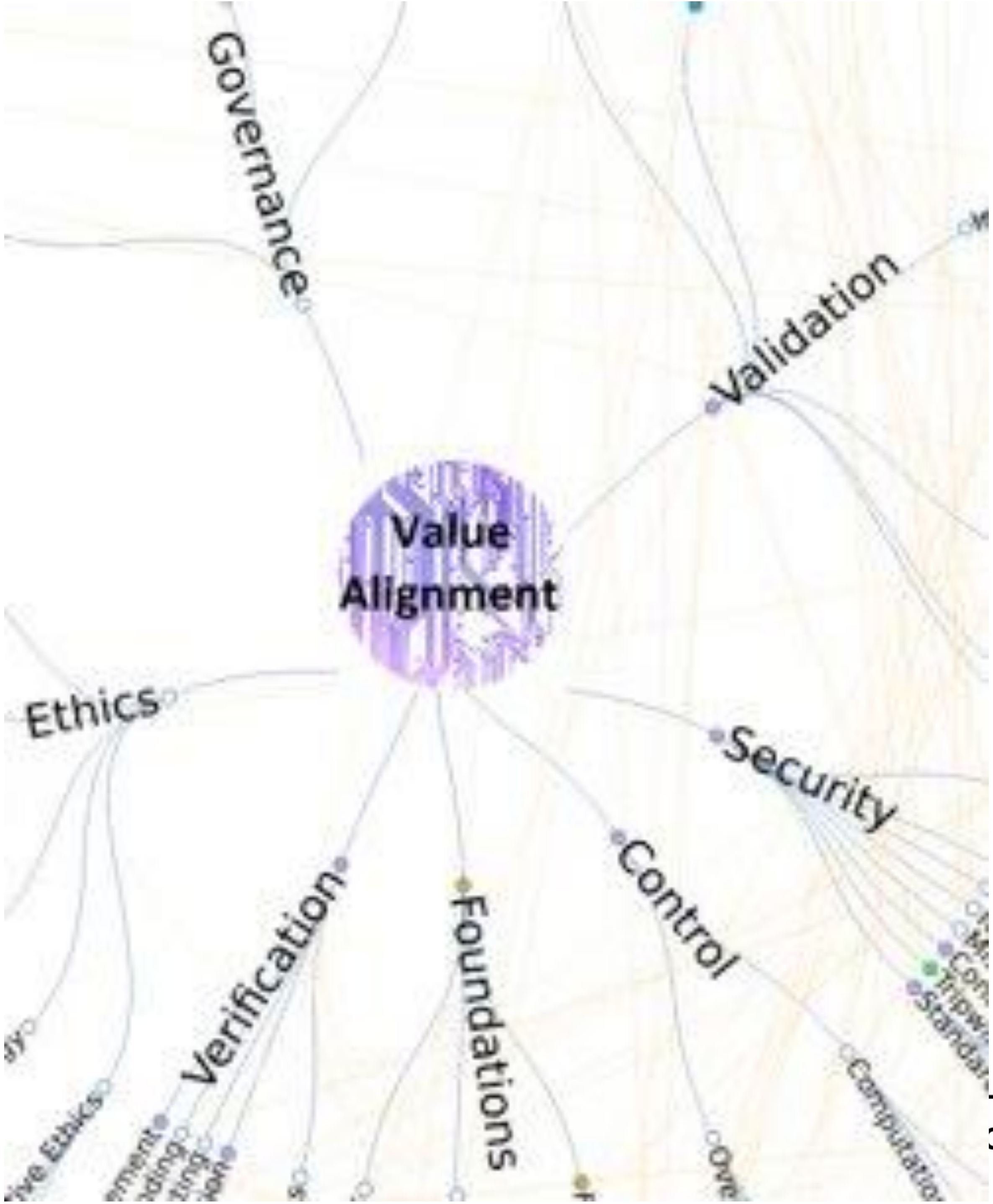
Validation

Verification

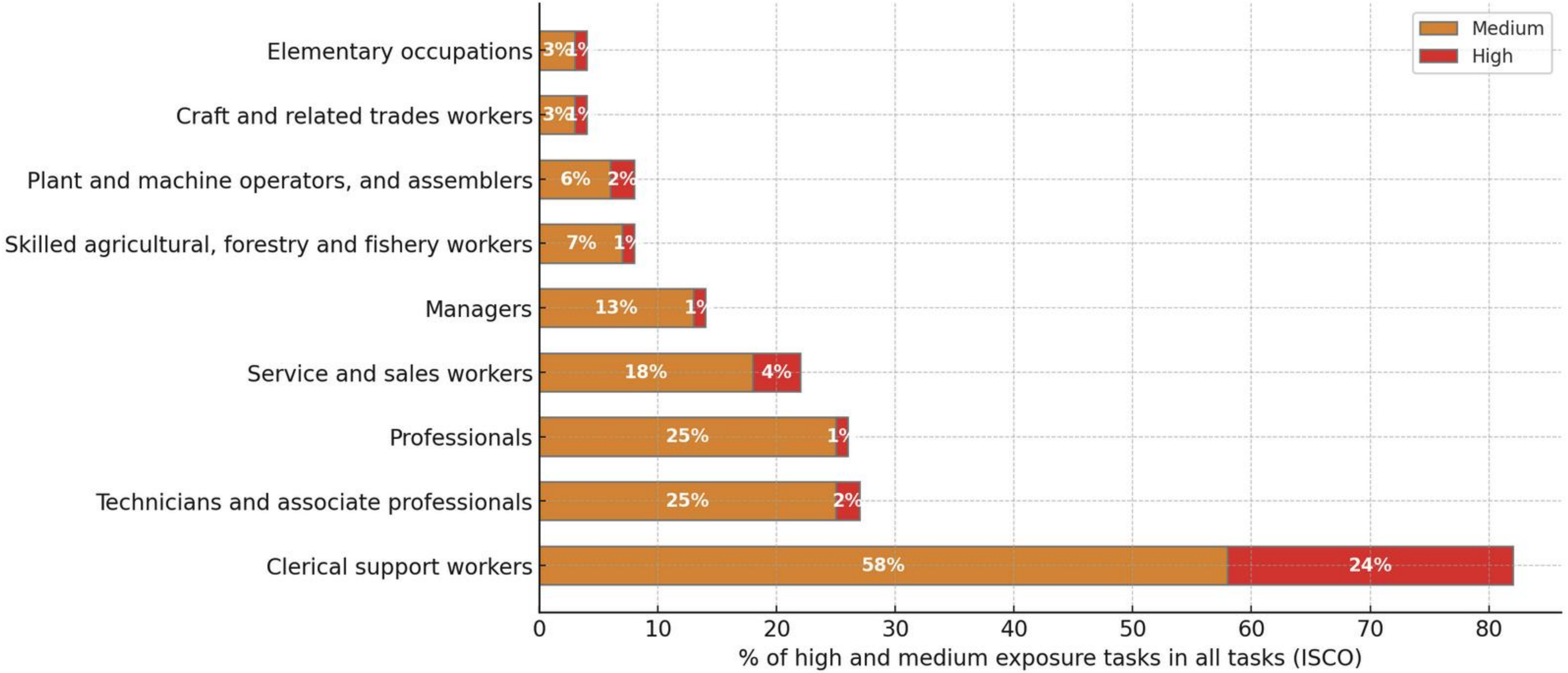
Foundation

Control

Security



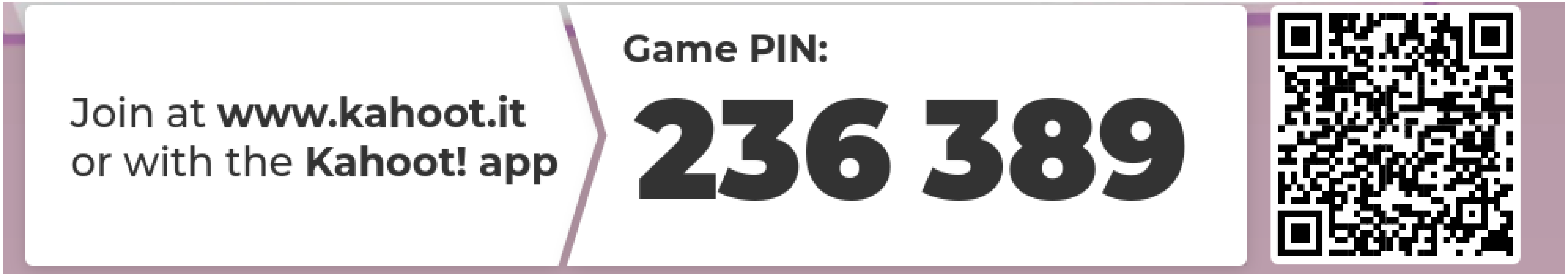
Exposure to AI in different jobs



Tasks with medium and high GPT-exposure, by occupational category

ecreated after: https://www.econstor.eu/bitstream/10419/278614/1/1857683005.pdf (cc license)

Possible interactive thing



Join at **www.kahoot.it**
or with the **Kahoot! app**

Game PIN:

236 389

Ethical HE frameworks

- Lots of people are developing HE AI frameworks

Case study 1: analysing survey data

Some terms

- Natural language processing
- Sentiment analysis
- Embeddings
- Classification
- Zero-shot
- Large language models

Analysing survey data

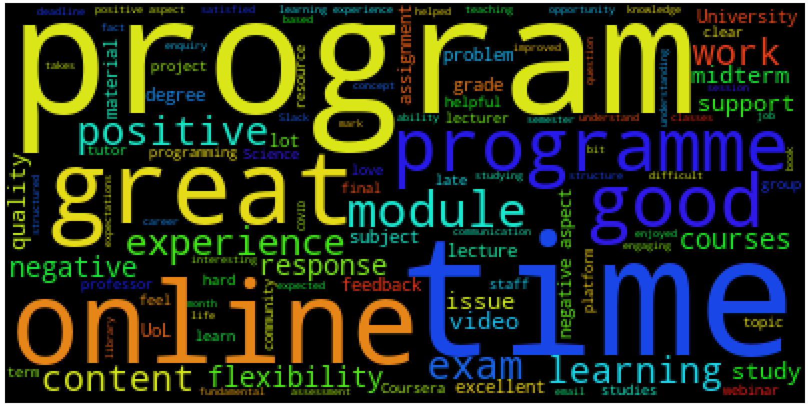
What is it? / Technology used: Semantic analysis, word frequency analysis, zero-shot classification

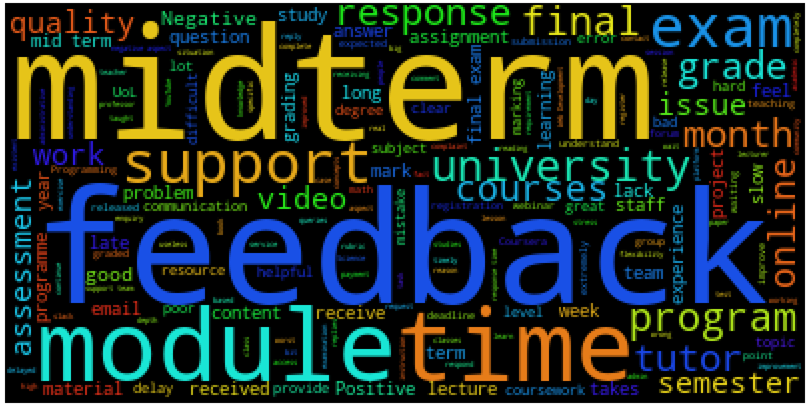
Motivation: Large number of free text comments, need to extract main themes

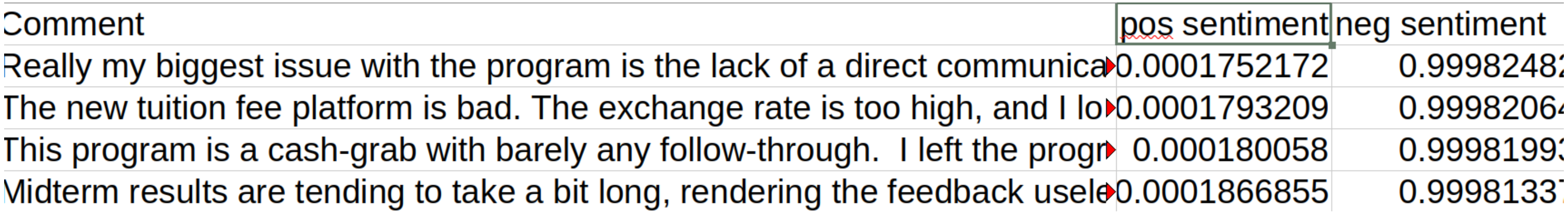
Benefits: Can read and categorise all free text comments, effectively automating qualitative research process

Ethics: Data exposure: use local LLM, exposure of human qual. researcher role (we don't have one, they can design and orchestrate the workflow)

Survey analysis: the beginning... 2022 word clouds and sentiment analysis







Comment	pos sentiment	neg sentiment
Really my biggest issue with the program is the lack of a direct communica	0.0001752172	0.9998248
The new tuition fee platform is bad. The exchange rate is too high, and I lo	0.0001793209	0.9998206
This program is a cash-grab with barely any follow-through. I left the progr	0.000180058	0.999819
Midterm results are tending to take a bit long, rendering the feedback usele	0.0001866855	0.999813

Survey analysis: assigning sentiment to themes

word	high_neg	med_meg	middle	med_pos	high_pos
organis	1.3	5.6	0	0	2.3
communicat	7.6	11.1	5.9	0	4.6
response	16.6	33.3	5.9	0	9.7
reply	3.5	0	5.9	0	1.7
efficien	0.2	0	0	0	0.6
delay	9	0	0	8.3	1.7
flexi	3.5	5.6	0	8.3	13.7
run	0.7	0	0	0	1.7
change	2.8	5.6	11.8	0	2.3

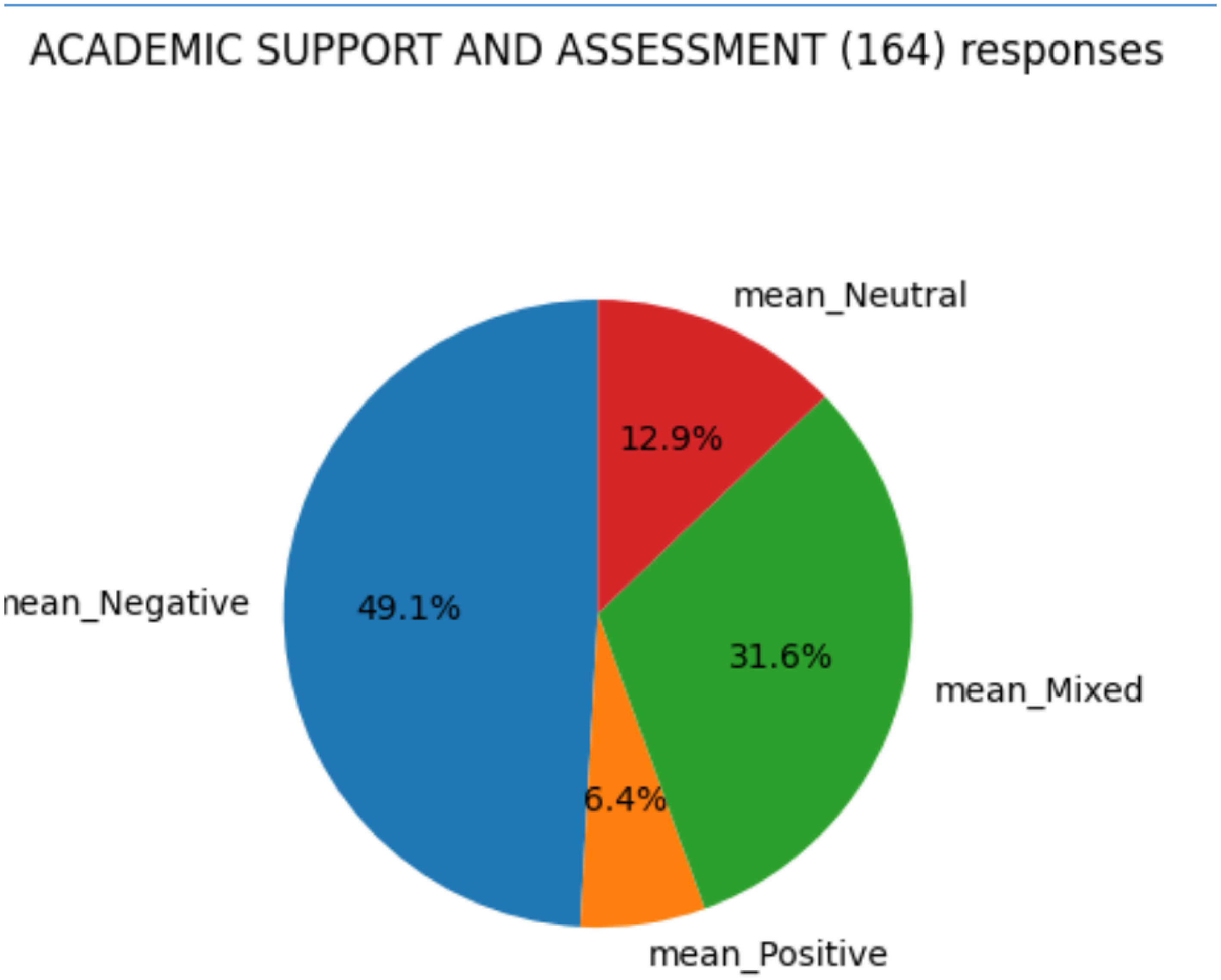
Local LLMs for survey analysis 2023

Local LLM: runs on your own machine but limited capability

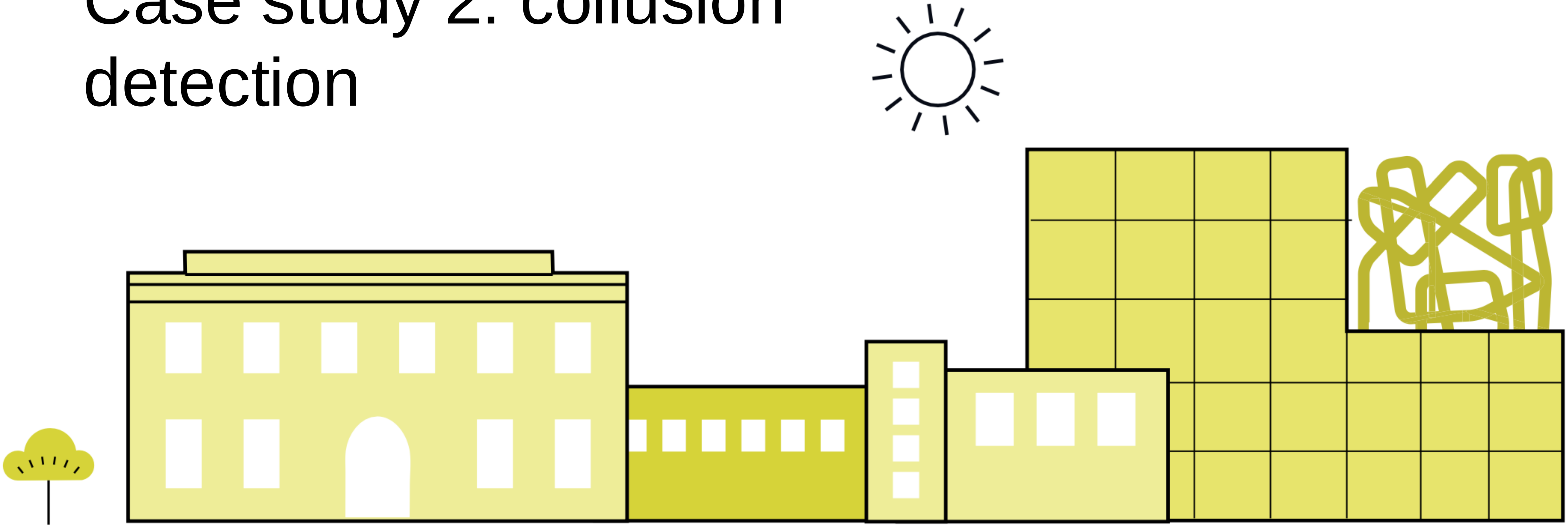
Zero-shot classification with local LLM: assign comments to themes

Sentiment analysis

Summarisation with sets (works with a small model)



Case study 2: collusion detection



Collusion detection

What is it? / Technology used: Semantic embeddings

Motivation: Large number of exams, current system inadequate, impossible for human to porcess all comparisons

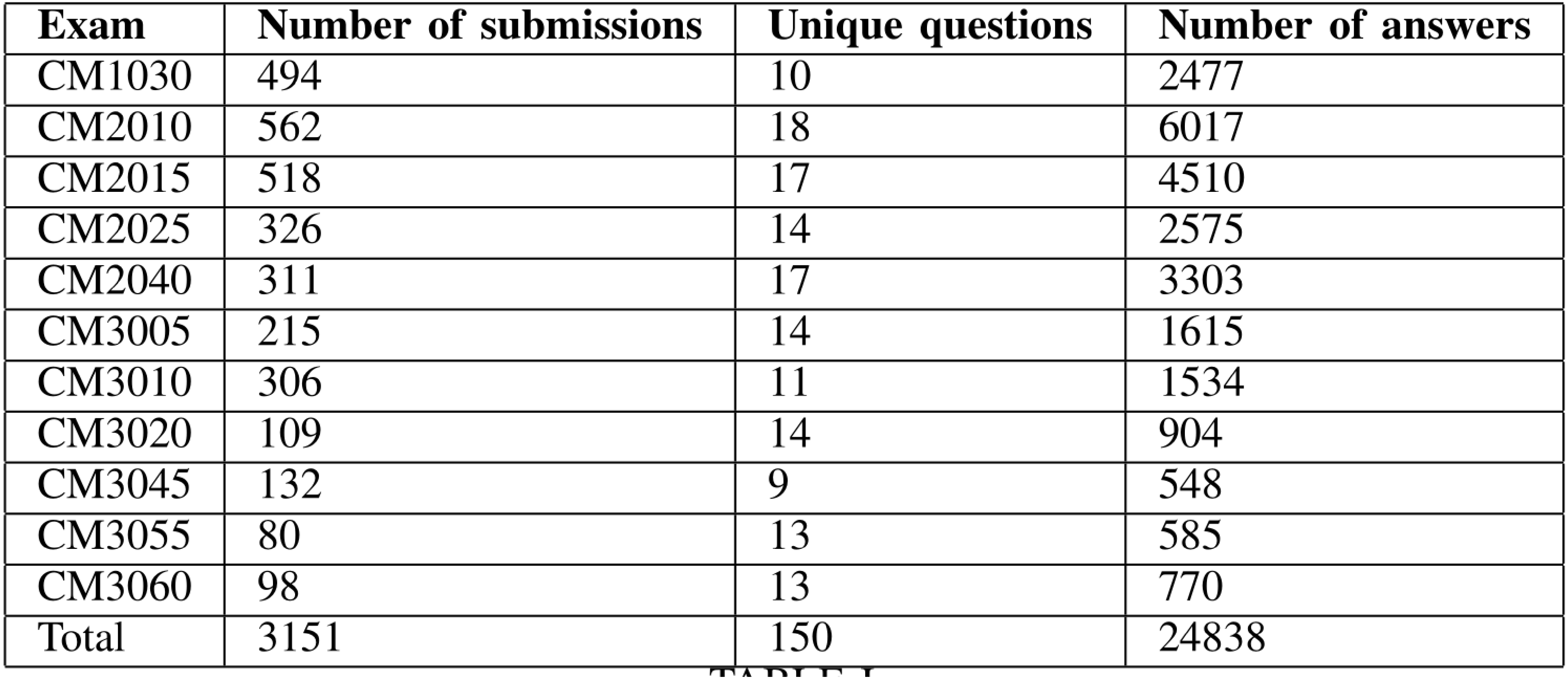
Benefits: Can process all answers to all exams, present information in compact digestible form

Ethics: Data exposure: use local model, exposure of examiners to automation - they didn't do this task before anyway, false positives: examiners have the final say

Collusion detection: research question

Is student A's answer to this question excessively similar to anyone else's?

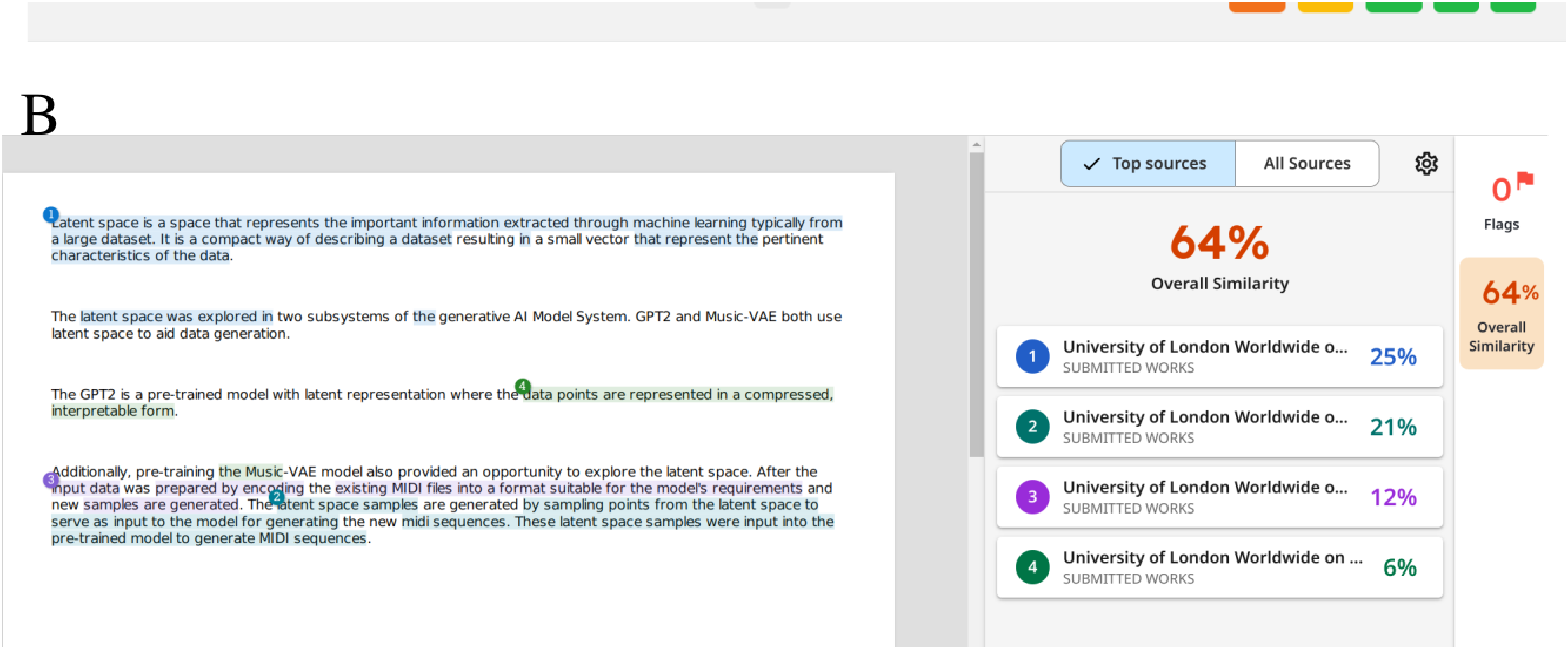
Identify excessively similar pairs - dataset



Exam	Number of submissions	Unique questions	Number of answers
CM1030	494	10	2477
CM2010	562	18	6017
CM2015	518	17	4510
CM2025	326	14	2575
CM2040	311	17	3303
CM3005	215	14	1615
CM3010	306	11	1534
CM3020	109	14	904
CM3045	132	9	548
CM3055	80	13	585
CM3060	98	13	770
Total	3151	150	24838

TABLE I

Collusion detection: turnitin is not enough



Latent space is a space that represents the important information extracted through machine learning typically from a large dataset. It is a compact way of describing a dataset resulting in a small vector that represent the pertinent characteristics of the data.

The latent space was explored in two subsystems of the generative AI Model System. GPT2 and Music-VAE both use latent space to aid data generation.

The GPT2 is a pre-trained model with latent representation where the data points are represented in a compressed, interpretable form.

Additionally, pre-training the Music-VAE model also provided an opportunity to explore the latent space. After the input data was prepared by encoding the existing MIDI files into a format suitable for the model's requirements and new samples are generated. The latent space samples are generated by sampling points from the latent space to serve as input to the model for generating the new midi sequences. These latent space samples were input into the pre-trained model to generate MIDI sequences.

64%
Overall Similarity

1 University of London Worldwide o... SUBMITTED WORKS 25%

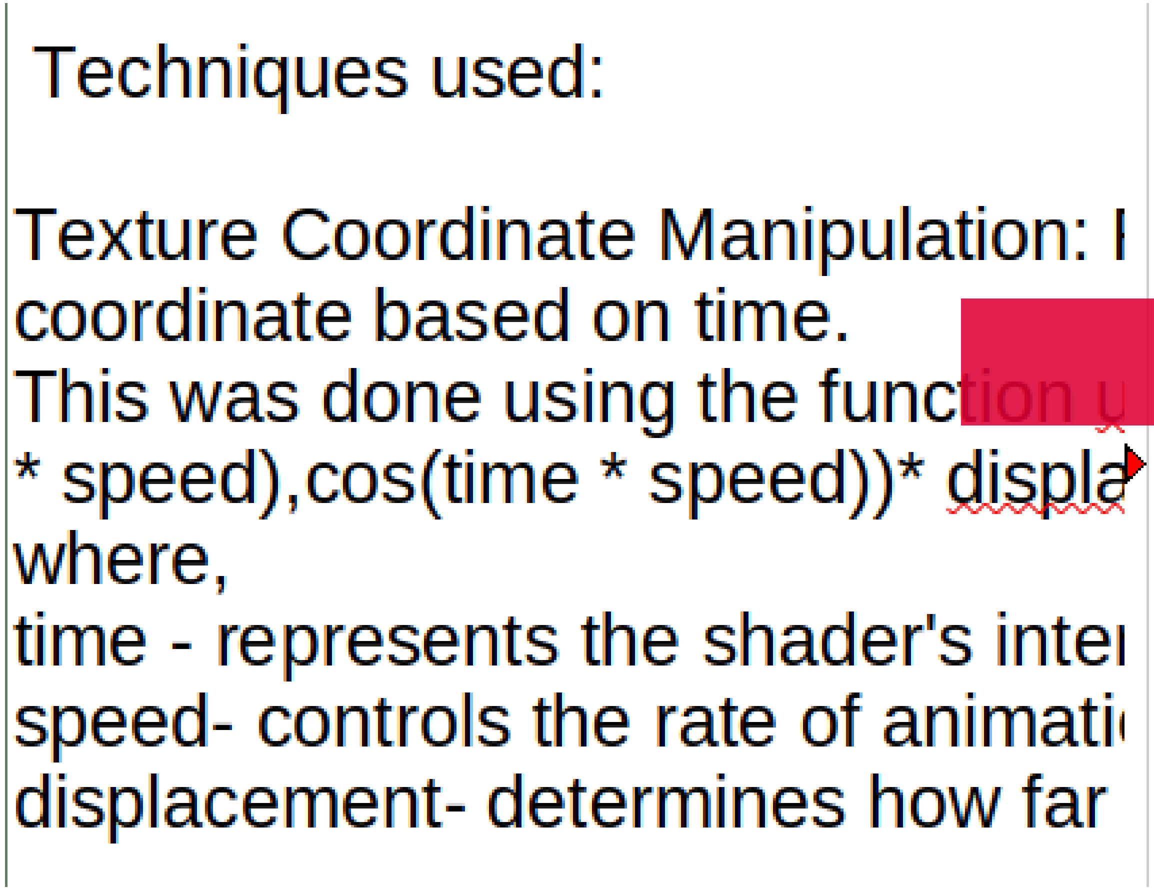
2 University of London Worldwide o... SUBMITTED WORKS 21%

3 University of London Worldwide o... SUBMITTED WORKS 12%

4 University of London Worldwide on ... SUBMITTED WORKS 6%

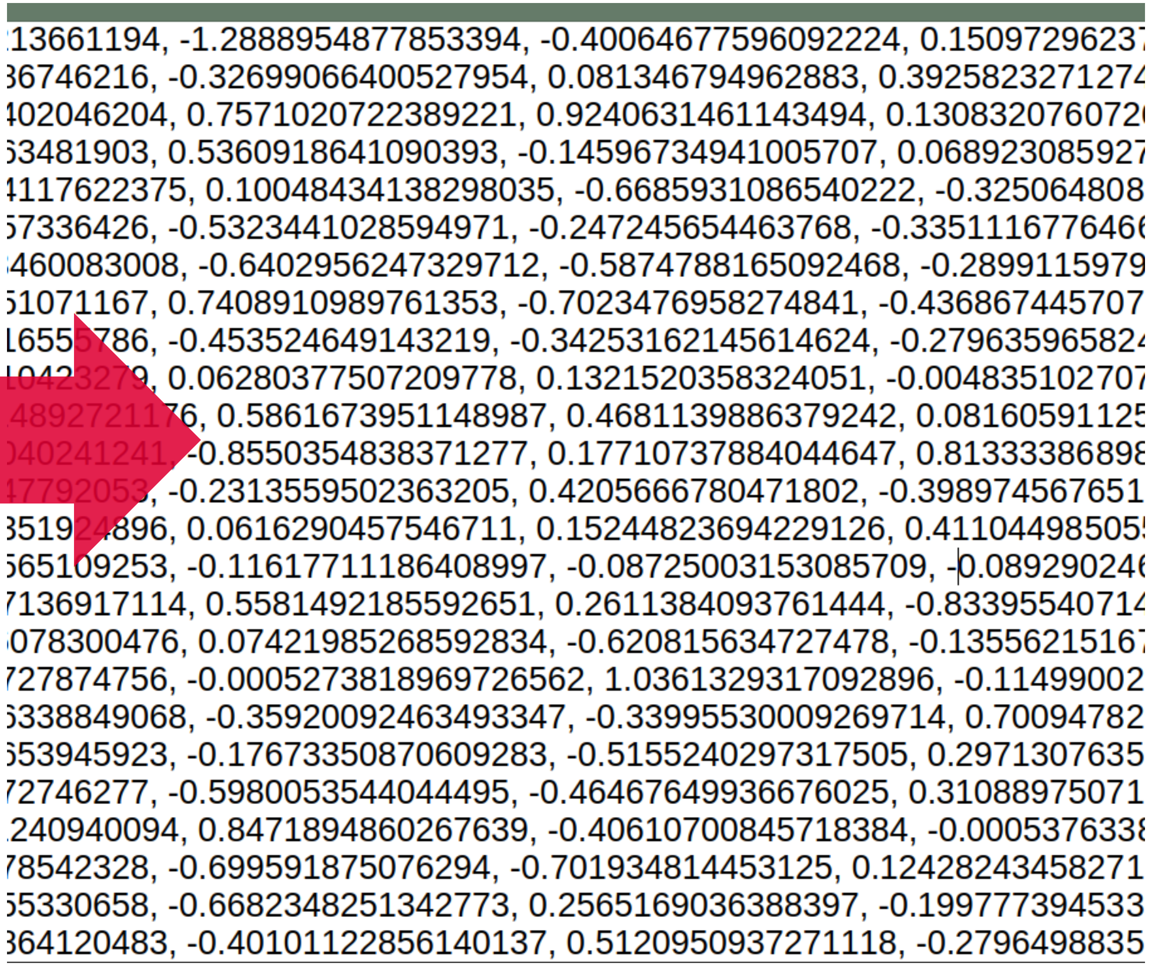
64% Overall Similarity

Collusion detection



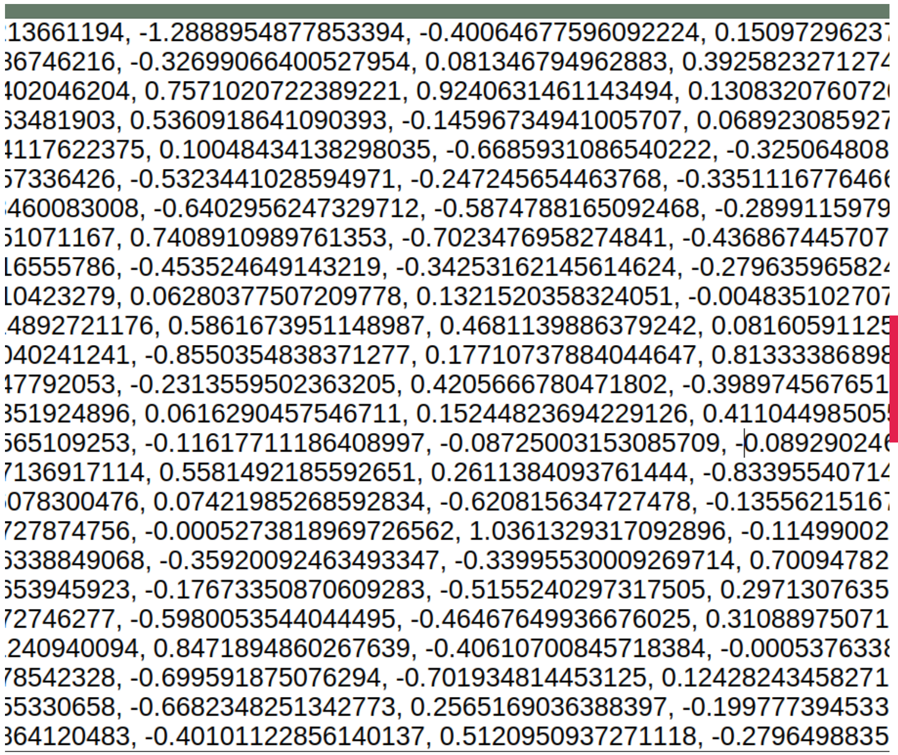
Techniques used:

Texture Coordinate Manipulation: I
coordinate based on time.
This was done using the function u
* speed),cos(time * speed))* displa
where,
time - represents the shader's inte
speed- controls the rate of animati
displacement- determines how far



```
13661194, -1.2888954877853394, -0.40064677596092224, 0.1509729623
36746216, -0.32699066400527954, 0.081346794962883, 0.392582327127
402046204, 0.7571020722389221, 0.9240631461143494, 0.13083207607
53481903, 0.5360918641090393, -0.14596734941005707, 0.06892308592
4117622375, 0.10048434138298035, -0.6685931086540222, -0.325064808
57336426, -0.5323441028594971, -0.247245654463768, -0.33511167764
460083008, -0.6402956247329712, -0.5874788165092468, -0.2899115979
51071167, 0.7408910989761353, -0.7023476958274841, -0.436867445707
16555786, -0.453524649143219, -0.3425316214561462 4, -0.27963596582
10423279, 0.06280377507209778, 0.132152035832405 1, -0.00483510270
4892721176, 0.5861673951148987, 0.4681139886379242, 0.0816059112
040241241, -0.8550354838371277, 0.1771073788404464 7, 0.8133338689
47792053, -0.231355950236320 5, 0.4205666780471802, -0.398974567651
3519244896, 0.0616290457546711, 0.1524482369422912 6, 0.41104498505
565109253, -0.11617711186408997, -0.0872500315308570 9, -0.08929024
7136917114, 0.5581492185592651, 0.2611384093761444, -0.83395540714
078300476, 0.0742198526859283 4, -0.62081563472747 8, -0.1355621516
727874756, -0.000527381896972656 2, 1.0361329317092896, -0.11499002
6338849068, -0.35920092463493347, -0.3399553000926971 4, 0.70094782
553945923, -0.17673350870609283, -0.515524029731750 5, 0.2971307635
72746277, -0.5980053544044495, -0.4646764993667602 5, 0.31088975071
240940094, 0.847189486026763 9, -0.40610700845718384, -0.00053763 38
78542328, -0.699591875076294, -0.701934814453125, 0.1242824345827 1
55330658, -0.6682348251342773, 0.2565169036388397, -0.199777394533
364120483, -0.40101122856140137, 0.5120950937271118, -0.2796498835
```

Embeddings: compute distance in semantic space



```
13661194, -1.2888954877853394, -0.40064677596092224, 0.1509729623
36746216, -0.32699066400527954, 0.081346794962883, 0.3925823271274
402046204, 0.7571020722389221, 0.9240631461143494, 0.130832076072
53481903, 0.5360918641090393, -0.14596734941005707, 0.068923085927
4117622375, 0.10048434138298035, -0.6685931086540222, -0.325064808
57336426, -0.5323441028594971, -0.247245654463768, -0.335111677646
460083008, -0.6402956247329712, -0.5874788165092468, -0.2899115979
51071167, 0.7408910989761353, -0.7023476958274841, -0.436867445707
16555786, -0.453524649143219, -0.34253162145614624, -0.2796359658
10423279, 0.06280377507209778, 0.1321520358324051, -0.004835102707
4892721176, 0.5861673951148987, 0.4681139886379242, 0.0816059112
040241241, -0.8550354838371277, 0.1771073788404464 7, 0.813333868
47792053, -0.2313559502363205, 0.4205666780471802, -0.398974567651
351924896, 0.0616290457546711, 0.15244823694229126, 0.41104498505
565109253, -0.11617711186408997, -0.08725003153085709, -0.08929024
7136917114, 0.5581492185592651, 0.2611384093761444, -0.8339554071
078300476, 0.07421985268592834, -0.620815634727478, -0.1355621516
727874756, -0.0005273818969726562, 1.0361329317092896, -0.11499002
5338849068, -0.35920092463493347, -0.3399553000926971 4, 0.70094782
553945923, -0.17673350870609283, -0.5155240297317505, 0.2971307635
72746277, -0.5980053544044495, -0.46467649936676025, 0.31088975071
240940094, 0.8471894860267639, -0.40610700845718384, -0.000537633
78542328, -0.699591875076294, -0.701934814453125, 0.12428243458271
55330658, -0.6682348251342773, 0.2565169036388397, -0.199777394533
364120483, -0.40101122856140137, 0.5120950937271118, -0.2796498835
```

Student A's answer

Other answers

Identify excessively similar pairs - report

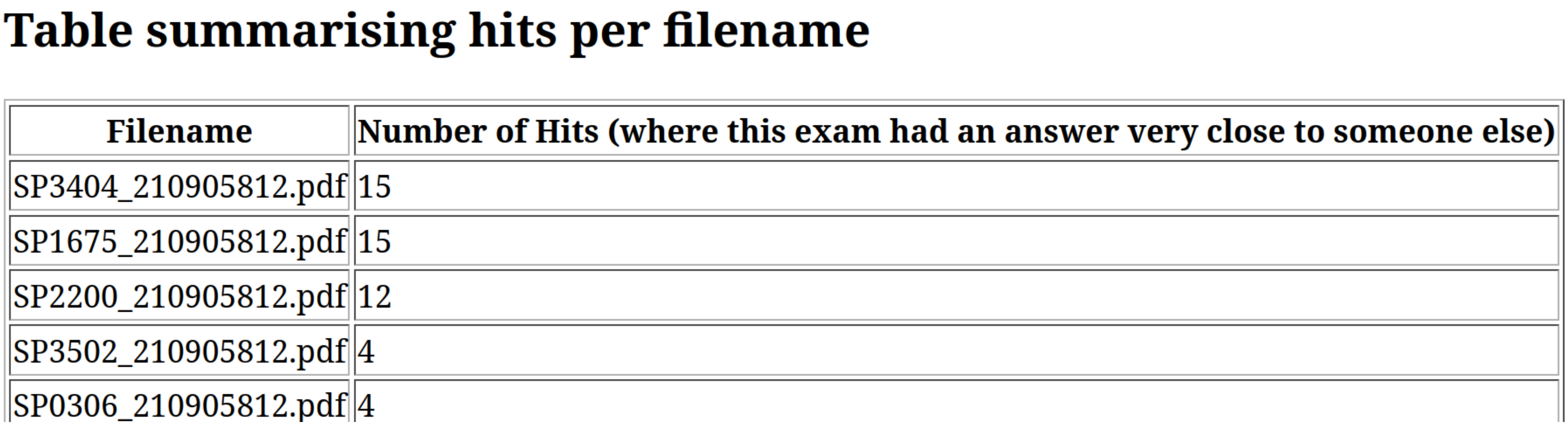
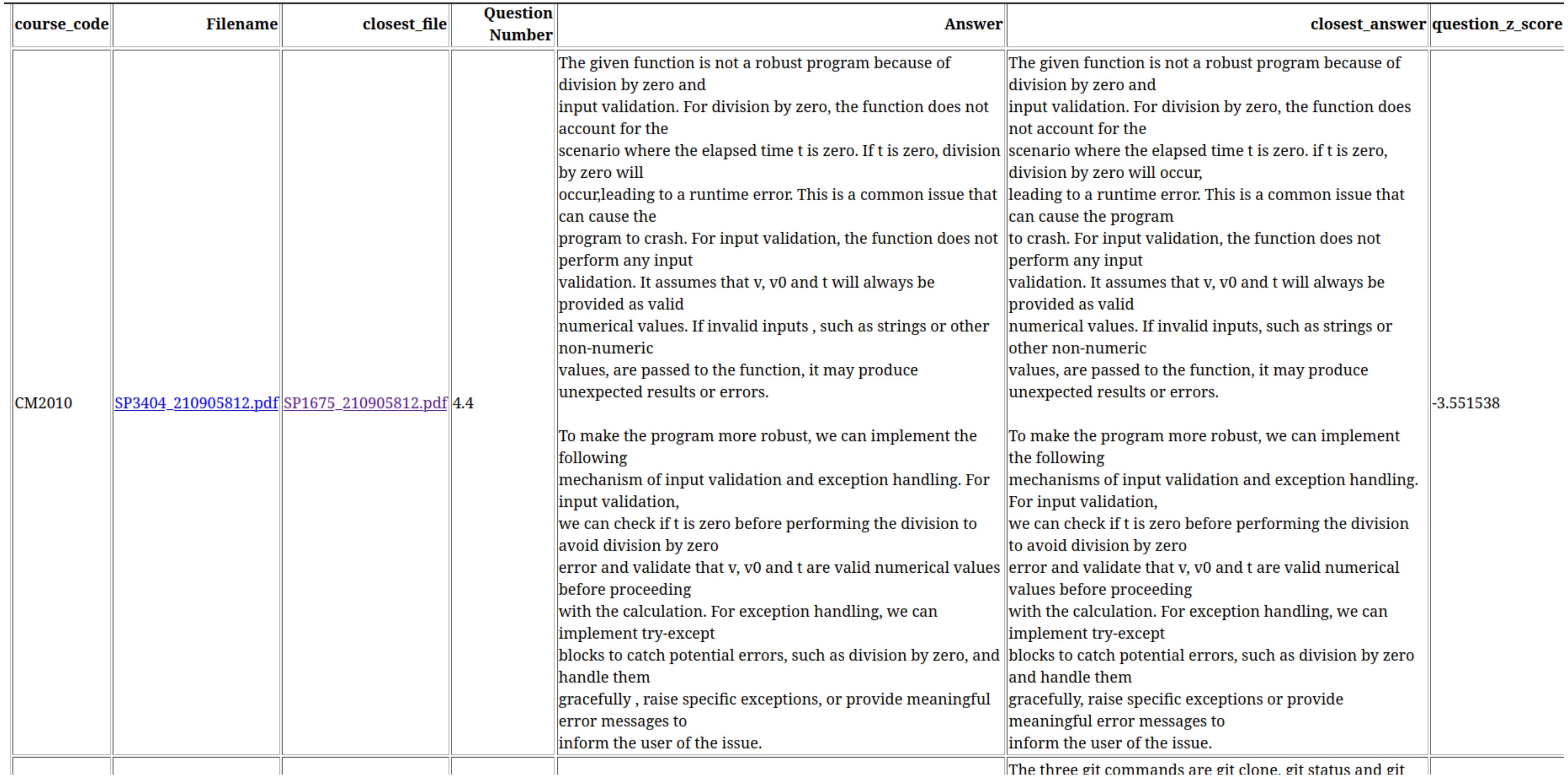


Table summarising hits per filename

Filename	Number of Hits (where this exam had an answer very close to someone else)
SP3404_210905812.pdf	15
SP1675_210905812.pdf	15
SP2200_210905812.pdf	12
SP3502_210905812.pdf	4
SP0306_210905812.pdf	4

Identify excessively similar pairs - report



course_code	Filename	closest_file	Question Number	Answer	closest_answer	question_z_score
CM2010	SP3404_210905812.pdf	SP1675_210905812.pdf	4.4	The given function is not a robust program because of division by zero and input validation. For division by zero, the function does not account for the scenario where the elapsed time t is zero. If t is zero, division by zero will occur,leading to a runtime error. This is a common issue that can cause the program to crash. For input validation, the function does not perform any input validation. It assumes that v, v0 and t will always be provided as valid numerical values. If invalid inputs , such as strings or other non-numeric values, are passed to the function, it may produce unexpected results or errors. To make the program more robust, we can implement the following mechanism of input validation and exception handling. For input validation, we can check if t is zero before performing the division to avoid division by zero error and validate that v, v0 and t are valid numerical values before proceeding with the calculation. For exception handling, we can implement try-except blocks to catch potential errors, such as division by zero, and handle them gracefully , raise specific exceptions, or provide meaningful error messages to inform the user of the issue.	The given function is not a robust program because of division by zero and input validation. For division by zero, the function does not account for the scenario where the elapsed time t is zero. if t is zero, division by zero will occur, leading to a runtime error. This is a common issue that can cause the program to crash. For input validation, the function does not perform any input validation. It assumes that v, v0 and t will always be provided as valid numerical values. If invalid inputs, such as strings or other non-numeric values, are passed to the function, it may produce unexpected results or errors. To make the program more robust, we can implement the following mechanisms of input validation and exception handling. For input validation, we can check if t is zero before performing the division to avoid division by zero error and validate that v, v0 and t are valid numerical values before proceeding with the calculation. For exception handling, we can implement try-except blocks to catch potential errors, such as division by zero and handle them gracefully, raise specific exceptions or provide meaningful error messages to inform the user of the issue.	-3.551538
					The three git commands are git clone, git status and git	

Results

Minimal effort for module leaders:

Cases are auto-detected

Reports are auto-generated

Module leaders have sign-off

14 cases brought forward

Next step: detect AI content

Generate lots of answers to questions with LLMs of various sorts

Is student A’s answer excessively semantically similar to the LLM answers?

Case study 3: generating exams

WIP: generating exams with LLMs

What is it? / Technology used: LocalLLM, pipelines, prompt engineering

Motivation: Continuous exam setting cycle, difficult to complete on time, inconsistency of exam design. Must have exams: gold standard

Benefits: Break perceived barrier of initial draft, exam setting task is simplified to an iteration and improvement task

Ethics: Data exposure: use local model, exposure of examiners to automation or increased workload

Side note: generating quizzes

Various projects to generate quizzes for scaffolding purposeS:

Context: video transcript + learning objectves:

Generate answers + feedback

Example exam structure: not just generate me an exam about x!

Request from the LLM, several questions

1) Fact recall

2) Given a scenario/ problem, provide a high level description of a solution

3) Describe the technical details of the solution

4) Critically evaluate your approach and other possible approaches

Implementation

Use local Llama 70b model via ‘open-ai’-like API

Data on courses all prepared in consistent form: topic lists, learning objectives, weekly descriptions of content

Scripts to iterate over courses and generate exams

Prompt

I am setting an exam for an undergraduate computer science course.

Here are the main topics in the course:

##topis##

Here ar descriptions of each week of content in the course:

##weeklies##

Here are the weekly learning objectives for the course.

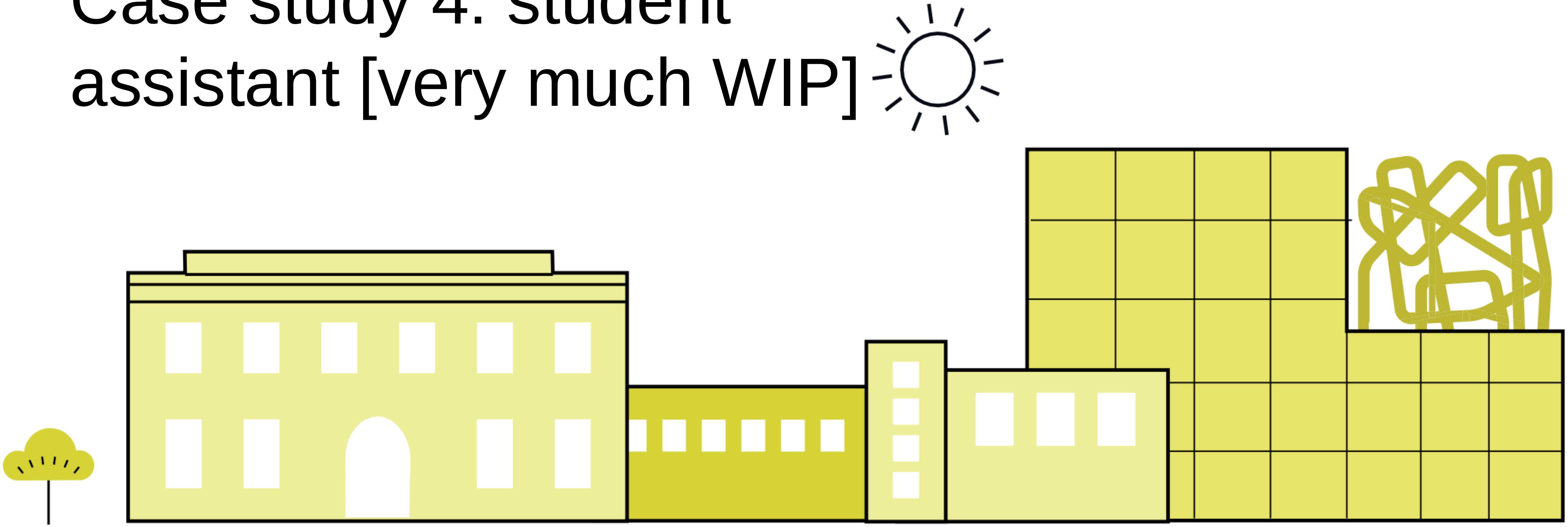
##weeklylos##

Here are the most important descriptors of the course, the course-level learning objectives. These are the items the exam should aim to evaluate the student against:

Prompt

Please write one exam question for me. The question should start with some factual questions which check the student's recall of basic facts taught in the course. The question should then ask the student to apply some skills learnt in the course to a particular case study or problem. The student should have to describe their approach or solution at a high level, then to give a more technical and specific description of their solution, using an appropriate format shown in the course. The final part of the question should ask the student to critically evaluate two approaches to solving a particular problem, highlighting advantages and disadvantages of the different approaches using knowledge gained in the course.

Case study 4: student assistant [very much WIP]



Student assistant

What is it? / Technology used: LocalLLM, RAG pipelines, prompt engineering,

Motivation: salespeople, hype, seems like a good idea, surely we can do it better?, they are already doing it with GPT

Benefits: Structured pedagogic workflows, reduce data exposure for students, proper access to local resources (actually fair use!)

Ethics: Data exposure: use local model, exposure of tutors to automation, cheating on courseworks

Student assistant agent: design

RAG pipeline:

Compose prompts from fragments of course materials through semantic search

Structured interaction: menu of options

Local LLM: Llama 3, Phi etc. Good enough? Fine-tuning? Local RAG?

Running locally: no inference cost for institution

Summary

- Some terms
- Local LLMs for survey analysis
- LLM embeddings for collusion detection
- WIP: Generating exam papers
- WIP: course chatbo

Goldsmiths
UNIVERSITY OF LONDON

UNIVERSITY
OF LONDON

london.ac.uk

Senate House, Malet Street,
London WC1E 7HU