MTH3022 Graphs Networks and Algorithms: Ontology Project

Candidate Numbers: 214389, 187357

May 5, 2024

1 Introduction

In our data-driven society, the organisation and representation of knowledge plays a vital role, especially in an university degree programme where structured and categorised data is essential. This project delves into graph theory, focusing on knowledge representation through directed graphs, to model individuals' academic journeys within the University of Exeter's degree programmes. We will categorise this information into modules based on prerequisites, exams, coursework, and topics, examining their interrelations. Then we will compare students with varying module choices and utilise AI tools to create graphs, finally going further to assess neural networks accuracy in categorising mathematical content. Our investigation will quantify the similarity of these representations. Through analytical methods, we aim to explore how our findings can enhance the student experience.

2 The Semantic Web

The concept of the Semantic Web, often referred to as Web 3.0, traces back to the early development of the World Wide Web by Sir Tim Berners-Lee in 1989. While conceived at that time, its practical implementation awaited further advancements in technology [1]. The Semantic Web stands as an extension of the conventional World Wide Web, offering a unified framework facilitating the seamless sharing and reuse of data across diverse applications, enterprises, and communities. Therefore enhancing collaborative efforts between humans and machines [2]. Defined informally in a May 2001 Scientific American article authored by Berners-Lee and colleagues, the Semantic Web is defined as an augmentation of the existing web infrastructure, imbuing information with precisely defined semantics to facilitate enhanced collaboration between humans and machines [3]. At its core, the Semantic Web hinges upon fundamental standards such as the Resource Description Framework (RDF) and the Web Ontology Language (OWL), enabling the structured representation and organisation of data in a format interpretable by machines.

2.1 RDF

RDF (Resource Description Framework) is fundamental for information representation on the Semantic Web, serving as the standard model for data interchange. It facilitates seamless data merging and supports schema evolution over time. RDF extends the web's linking structure using URLs to form triples—subject-predicate-object statements—succinctly describing resources and their relationships. This approach enables the mixing, exposure, and sharing of structured and semi-structured data across applications. RDF organises data into a directed graph structure, where nodes represent resources/entities, and edges represent relationships between them. Each element within an RDF graph can be precisely labelled, allowing for the creation of labelled directed graphs that encapsulate knowledge effectively. This graphical representation provides an easy model for RDF and is commonly used in visual explanations [4].

In a workplace context, RDF triples can effectively organise information, as exemplified by the following:

RDFTriple[“JohnSmith”, “worksInDepartment”, “Marketing”]

- “JohnSmith” represents an individual entity, likely an employee within the workplace.
- “WorksInDepartment” represents a relationship between John Smith and the department he works in.
- “Marketing” represents the department in which John Smith works.

Official RDF format requires each element of the RDF to have its own unique URL. For example, the previous instance can be represented as follows:

RDFTriple[“http://name#JohnSmith”, “http://connection#worksInDepartment”, “http://department#Marketing ”]

However, for clarity purposes, the RDFs used throughout this report will not include their unique URLs.

2.2 OWL

OWL (Web Ontology Language), developed by the W3C OWL Working Group, serves as an extension of RDF, offering a more robust framework for representing complex knowledge structures within the Semantic Web. By building upon RDF, OWL enables the depiction of intricate relationships and constraints between entities, creating a more expressive framework for defining ontologies on the Semantic Web.

This language allows for the creation of detailed relationships and constraints between resources, facilitating the construction of sophisticated knowledge representations. Similar to RDF, OWL ontologies can be conceptualised as labelled directed graphs, with nodes representing classes, individuals, properties, and restrictions, while edges indicate various relationships such as subclass, property relationships, and class membership.

Furthermore, OWL operates as a computational logic-based language, empowering computer programs to leverage the expressed knowledge for tasks like consistency verification and implicit knowledge extraction. Ontologies formulated in OWL can be disseminated on the World Wide Web and referenced by other OWL ontologies, contributing to the interconnectedness of information. OWL stands as an integral component of the W3C's Semantic Web technology stack, complementing other standards like RDF, RDFS, and SPARQL, to facilitate the realisation of a robust and interoperable Semantic Web ecosystem [5].

Revisiting our RDF example within the workplace scenario, we can illustrate how OWL ontology could be practically applied:

We could define classes to represent entities such as employees and departments, each with their own set of properties and relationships. For instance, we could define a class for employees, where each individual employee is linked to the department they work in through a "worksInDepartment" property. Similarly, we could define a class for departments, with properties defining attributes such as department name, location, etc. This demonstrates the application of OWL ontology in structuring and organising workplace data. By defining classes, properties, and relationships, we create a semantic framework that enables more meaningful interpretation and analysis of the data.

The relationship between these standards and labelled directed graphs is profound. RDF forms the basic structure for representing data, while OWL adds further expressiveness for defining formal ontologies. Both RDF and OWL can be conceptualised and represented as labelled directed graphs, with nodes and edges capturing the entities, relationships, and properties defined within the data or ontology. This enables labelled directed graphs to serve as a suitable model for visualising and comprehending the structure of data and knowledge represented using Semantic Web standards [6].

Hence in our investigations we will use this relationship to create multiple triples where for module prerequisites, topics they use and their exam weightage. We initially depict the module prerequisites within the directed graph structure. Here, each module serves as the subject, represented by the predicate 'requires,' linking it to its prerequisite module:

RDFTriple["MTH2003", "requires", "MTH1002"], ... (and so forth for each module)

Next, we incorporate RDF triples indicating the examination weight age associated with each module. The subject remains the module, with the predicate 'Exam_weight,' and the exam weight age being the object:

RDFTriple["MTH1002", "Exam_weight", "90%"], ... (and similarly for other modules)

Lastly, we introduce RDF triples illustrating the utilisation of specific topics within each module. Here, the module serves as the subject, with the predicate 'uses,' linking it to the topics from the Mathematics MSc (Masters of Science) 2022/23 specification of the University of Exeter. Further using the predicate 'relates to' for how each topic within a module are related. For this project the 'MSc 2022/23 specification' served as our Simple Knowledge Organisation System (SKOS).

RDFTriple["MTH1002", "uses", "Functions"], RDFTriple["An Introduction to Conic Sections", "relates to", "Functions"], ... (and so on for each module and topic).

This systematic approach provides a structured representation of module prerequisites, examination weight age, and the idolisation of topics within each module.

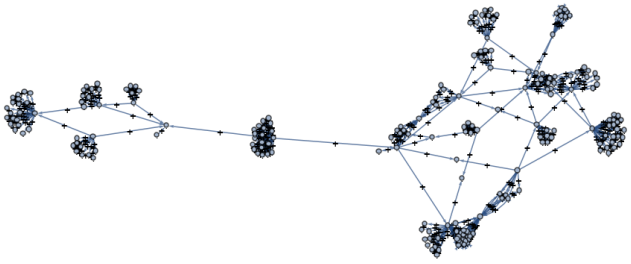
3 The Research

For this investigation, we employed the Resource Description Framework (RDF) to construct knowledge graphs representing the modules undertaken throughout our degree program. These graphs captured the interrelationships between modules through properties such as prerequisites, written exams, coursework, and internal topics. We further strengthened our data by incorporating a collection of student journey graphs obtained from the RDFStore sharing forum. To broaden the scope of our inquiry, we explored the potential of using Artificial Intelligence (AI) languages as well as machine learning methods to categorise mathematical information within a degree program.

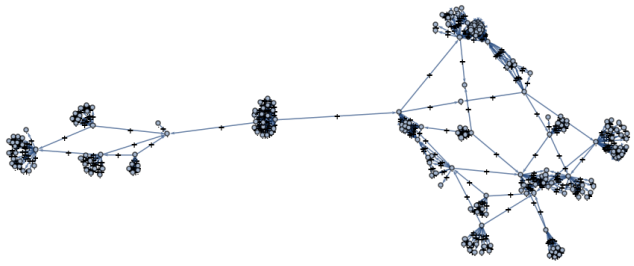
Our initial hypothesis is centred on the notion that graphs representing similar module selections would exhibit higher degrees of similarity due to overlapping content. We anticipated that graphs representing more modelling modules would be more

alike than a graph representing a student who took more pure maths modules or statistical modules. Yet we expect to find differences on how topics relate to each other due to individual preferences. Pertaining to the usage of AI languages and machine learning methods we expect this to show broad similarity to the graphs created by students. Although, we expect AI languages to make assumptions, be over-generalised, or produce incorrect guesses. However, if we improve the prompts used for AI and if we train the Neural network more we would expect the graph similarity to increase.

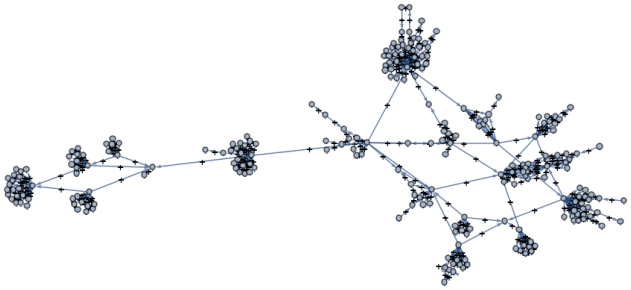
3.1 The Graphs



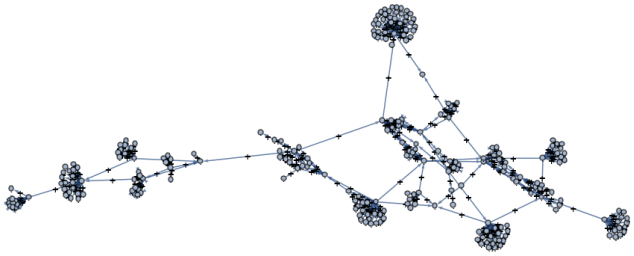
Graph 1: A 3^{rd} Year Mathematics Student Studying Predominantly Modelling Modules



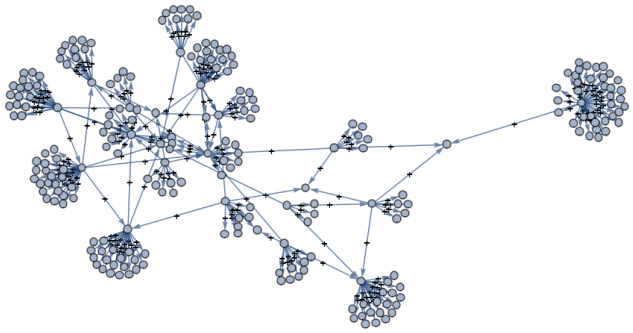
Graph 2: Chat GPT's Interpretation of a 3^{rd} Year Mathematics Student Studying Predominantly Modelling Modules



Graph 3: Gemini's Interpretation of a 3^{rd} Year Mathematics Student Studying Predominantly Modelling Modules



Graph 4: A 3^{rd} Year Mathematics Student Studying Modelling and Business Modules



Graph 5: A 3^{rd} Year Mathematics Student Studying Predominantly Pure Mathematics Modules

3.2 Measuring Similarity and Contrast within graphs

3.2.1 Graph edit distance (Vertex Count)

Graph Edit Distance (GED) measures the similarity between two graphs by quantifying the minimum number of edit operations (such as vertex additions, deletions, substitutions, or edge modifications) needed to transform one graph into another [7]. When focusing on Vertex Count, GED considers only operations related to vertices or nodes. It calculates the minimum number of vertex-level edit operations required to make two graphs equivalent.

3.2.2 Graph distance

Graph Distance, also referred to as Shortest Path Distance, is a fundamental metric in graph theory, quantifying the minimum number of edges needed to traverse between two vertices. It serves as a crucial measure in network analysis and various graph algorithms, offering insights into connectivity and node proximity [8].

3.2.3 Spectral Distance

Spectral Distance is a way to compare how similar two graphs are based on their spectral properties, like eigenvalues or eigenvectors. Eigenvalues of a graph refer to the eigenvalues of its adjacency matrix. This matrix represents the connections between different vertices in the graph. When we find the eigenvalues of this matrix, we get what's known as the 'Graph Spectrum' [9]. It helps to understand the differences in structure between graphs and is often used in tasks like clustering and classification.

In simple terms, given two graphs G_1 and G_2 with n_1 n_2 vertices respectively, the Spectral Distance $\sigma(G_1, G_2)$ is calculated by comparing their eigenvalues. Specifically, it's the sum of the absolute differences between corresponding eigenvalues of the two graphs:

$$\sigma(G_1, G_2) = \sum_{i=1}^{n} |\lambda_i(G_1) - \lambda_i(G_2)|$$

Here, $\lambda_i(G)$ represents the eigenvalues of graph G, ordered from largest to smallest [10].

If the number of vertices differed, $n_1 \neq n_2$, we have to consider how to handle the comparison of their spectral properties fairly. The approach we took was to normalise the eigenvalues before comparing them. Let $n_1 > n_2$:

$$\sigma_{\text{norm}}(G_1, G_2) = \frac{1}{n_1} \sum_{i=1}^{n_1} |\lambda_i(G_1) - \lambda_i(G_2)|$$

3.2.4 Adjacency Matrix

An adjacency matrix is a square matrix representation of a graph, where each row and column corresponds to a vertex, and the entries indicate whether there is an edge between the vertices. For this measure we will compare the eigenvalues of the adjacency matrices and measure their Manhattan distance. Where the 'Manhattan distance' is a metric defining the distance between two vectors as the L1 Norm of their difference [11]. This was achieved by using our 'CompareAdjacency' function in MATHEMATICA. The function 'CompareAdjacency' takes two graphs, 'graph1' and 'graph2', as inputs. It computes a dissimilarity measure between them based on their adjacency matrices. First, it calculates the eigenvalues of the adjacency matrices for both graphs, sorting them in ascending order. Then, it determines the maximum length of the sorted eigenvalue lists. Next, it pads both eigenvalue lists with zeros to ensure they have the same length. After that, it computes the Manhattan distance between the padded eigenvalue lists, which quantifies the dissimilarity between the graphs' connectivity patterns. Additionally, it includes a penalty term, equal to the absolute difference in length between the two eigenvalue lists, to account for differences in the number of vertices between the graphs. Finally, the function returns the sum of the Manhattan distance and the penalty term as the overall dissimilarity measure between the two graphs.

3.2.5 Laplacian Matrix

This measure is similar to the Adjacency Matrix in the way it is computed however we utilise the Laplacian Matrix. The Laplacian matrix, alternatively referred to as the admittance matrix or Kirchhoff matrix, characterises the structural properties of an undirected, unweighted graph $G = (V, E)$, where V denotes the vertex set, $n = |V|$, and E represents the edge set. It is a symmetric $n \times n$ matrix, delineating one row and column for each node, defined as:

$$L = D - A \quad (1)$$

Here, $D = \text{diag}(d_1, \ldots, d_n)$ signifies the degree matrix, comprised of the degrees of vertices along its diagonal, while A symbolises the adjacency matrix. In L, the diagonal elements l_{ij} are equivalent to the degree of vertex v_i, and the off-diagonal elements l_{ij} are set to -1 if vertex v_i is adjacent to v_j, and 0 otherwise [12].

The Laplacian matrix of a graph is implemented in the Wolfram Language as `KirchhoffMatrix[g]`- a built in function. The we used our function 'CompareLaplacian' assesses the dissimilarity between two graphs, 'graph1' and 'graph2', by examining their Laplacian matrices. It begins by computing the eigenvalues of the Laplacian matrices for both graphs using the Wolfram Language's 'KirchhoffMatrix' function. These eigenvalues are then normalised and sorted in ascending order. Next, the function determines the maximum length of the sorted eigenvalue lists. To ensure uniformity, it pads both lists with zeros. A penalty term is then calculated as the absolute difference in length between the two lists of eigenvalues. Finally, the Manhattan distance between the padded eigenvalue lists is computed. This distance reflects the dissimilarity between the Laplacian matrices. The penalty term is added to the Manhattan distance to provide an overall measure of dissimilarity between the graphs. This approach effectively captures structural differences between the graphs, considering both the differences in eigenvalues and the penalty for discrepancies in graph size.

4 Analysis of Similarity Measures

This section comprises of Heat Map matrices in order to compare the similarity of each graph by the listed measures above. For clarification this is the colour scale corresponding to our similarity measures.

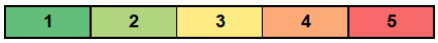


Figure 1: Heat Map Scale (1 Most Similar, 5 Least Similar)

First, most of the percentages were calculated using largest values in each table, as demonstrated by the following equation:

$$= \left(1 - \frac{\text{Value}}{\text{Max Value}}\right) \times 100$$

Whereas the Spectral Distance percentages were calculated using the following equation:

$$= \left(1 - \frac{\text{Norm(norm1} - \text{norm2)}}{\text{meanNorm}}\right) \times 100$$

4.1 Heat Map of Graph Edit Distance

From the provided Heat Map Matrix 1, it's evident that Graph 2 bears the highest similarity to Graph 1, scoring 82.5% using the Graph edit distance measure. In Graph 1, we see the learning trajectory of a Mathematics student primarily focusing on Modelling Modules, while Graph 2 reflects ChatGPT's interpretation of this trajectory, categorised into RDF triples. This indicates that ChatGPT's understanding aligns closely with the typical learning path of a Mathematics student focused on modelling modules, as evidenced by the high similarity score.

In contrast, Gemini's rendition of the learning path depicted in Graph 1 shows significantly lower similarity at 40.3%, hinting at the potential unreliability of AI models in accurately capturing and representing complex data like educational trajectories. This discrepancy underscores the need for further refinement and validation of AI-generated outputs in educational contexts.

Graph Edit Distance

	Graph 1	Graph 2	Graph 3	Graph 4	Graph 5
Graph 1	100%	82.5%	40.3%	18.6%	14.2%
Graph 2	82.5%	100%	27.3%	6.9%	0.0%
Graph 3	40.3%	27.3%	100%	32.0%	56.5%
Graph 4	18.6%	6.9%	32.0%	100%	47.9%
Graph 5	14.2%	0.0%	56.5%	47.9%	100%

Heat Map Matrix 1: Comparing the similarity of each graph using Graph Edit Distance

Furthermore, there is a 0% similarity between Graph 2 and Graph 5, where Graph 5 portrays a Mathematics student concentrating mostly on Pure Modules. This outcome aligns with expectations, considering that Graph 2 is derived from a student taking more modelling modules. However, it's worth noting a 14.2% similarity between Graph 1 and Graph 5, suggesting that ChatGPT might have overlooked some aspects of the learning path that are common to both modelling and pure modules, or that the categorisation into RDF triples might not fully capture the nuances of the educational content.

It's important to emphasise that the equation used to transform the raw data results in the least similar graph registering 0% similarity. Understanding that all percentages are relative to the least similar graph is crucial, as any potential misrepresentations could lead to flawed conclusions. This underscores the importance of careful interpretation and contextualisation of similarity scores when analysing educational trajectories using AI-generated models.

4.2 Heat Map of Graph distance

When employing the Graph distance measure, we find significant alignment with the earlier discussed 'Graph Edit Distance' measure. Once again, Graph 1 emerges as the most similar, boasting an 81.4% resemblance to Graph 2.

In contrast, Graphs 2, 4, and 5 display the lowest similarity, with Graph 2 and Graph 4 showing a mere 1.3% resemblance, and Graph 5 showing 0%. This discrepancy stems from the considerable divergence in module composition compared to Graph 2. Specifically, Graph 4 integrates modelling and business modules, while Graph 5 focuses solely on pure mathematics. Meanwhile, Graph 2 represents ChatGPT's rendition of a student primarily undertaking modelling modules. Additionally, Graphs 4 and 5 depict entirely different students, whereas Graphs 2 and 3 represent an AI model's interpretation of student 1's academic journey from Graph 1. Consequently, achieving Graph 2 from either Graph 4 or 5 necessitates substantial edits, which could involve removing, adding, elongating, contracting, or renaming vertices.

Graph Distance

	Graph 1	Graph 2	Graph 3	Graph 4	Graph 5
Graph 1	100%	81.4%	44.4%	15.8%	14.8%
Graph 2	81.4%	100%	30.3%	1.3%	0.0%
Graph 3	44.4%	30.3%	100%	47.4%	56.2%
Graph 4	15.8%	1.3%	47.4%	100%	54.3%
Graph 5	14.8%	0.0%	56.2%	54.3%	100%

Heat Map Matrix 2: Comparing the similarity of each graph using Graph Distance

4.3 Heat Map of Spectral Distance

Interestingly, Spectral Measure observes Graph 3 and Graph 5 to be the most similar pair, which is intriguing since Graph 3 represents Gemini's version of Graph 1, suggesting potential very broad assumptions and limitations within AI. As it seems over-generalisations would be cause of this unexpected high similarity measure.

However, ChatGPT demonstrates promise, with Graph 1 and Graph 2 being 72% similar according to the spectral distance measure. Conversely, Graph 2 and Graph 5 exhibit the least similarity at 1%, aligning with logical assumptions since Graph 5 represents a different undergraduate student. Aside from the anomaly the data is as expected, with similar modules graphs having higher scores.

Spectral Distance

	Graph 1	Graph 2	Graph 3	Graph 4	Graph 5
Graph 1	100%	72%	38%	51%	31%
Graph 2	72%	100%	8%	21%	1%
Graph 3	38%	8%	100%	89%	95%
Graph 4	51%	21%	89%	100%	82%
Graph 5	31%	1%	95%	82%	100%

Heat Map Matrix 3: Comparing the similarity of each graph using Spectral Distance

4.4 Heat Map of Laplacian Matrix

Using the Laplacian Matrix measure, the Heat Map demonstrates that Graph 1 and Graph 2 were the most similar at 86.4%, while Graph 2 and Graph 5 showed no similarity at 0%. Likely due to ChatGPT containing too much data as it was the largest RDF Store with ≈1000 different entries, causing the vast difference in eigenvalues. Graph 4 and Graph 2 display minimal similarity at 16.1%, along with Graph 5 and Graph 1, which are also not very similar at 12.6%.

Unexpectedly, Graph 5 and Graph 3 exhibit relatively high similarity at 74.5%. This unexpected result can be attributed to the methodology of the Laplacian Matrix measure, which may emphasise structural aspects of the graphs that lead to this unexpected similarity, despite the explicit differences between the graph stores.

Laplace Comparison

	Graph 1	Graph 2	Graph 3	Graph 4	Graph 5
Graph 1	100%	86.4%	58.5%	29.7%	12.6%
Graph 2	86.4%	100%	34.0%	16.1%	0.0%
Graph 3	58.5%	34.0%	100%	22.1%	74.5%
Graph 4	29.7%	16.1%	22.1%	100%	56.4%
Graph 5	12.6%	0.0%	74.5%	56.4%	100%

Heat Map Matrix 4: Comparing the similarity of each graph using Laplacian Matrix Measure

4.5 Heat Map of Adjacency Matrix

In the Heat Map generated from the adjacency matrix measures, Graph 1 and Graph 2 emerge as the most similar, boasting a high similarity score of 86.5%. Conversely, Graph 1 and Graph 5 exhibit the lowest similarity, registering at 0%, indicating significant dissimilarity between them. Similarly, Graph 2 and Graph 5 are notably dissimilar at 0.7%.

This suggests that ChatGPT has performed well in this measure, as we would ideally expect Graph 1 and Graph 2, representing similar academic journeys, to exhibit high similarity scores with each other. However, the slight discrepancy of 0.7% between Graph 2 and Graph 1 might stem from overgeneralising or broad assumptions made by the AI language model.

Furthermore, there is an anomaly where Graph 3 and Graph 5 show relatively high similarity, contrary to expectations. This anomaly is not as pronounced when inspecting the 'Adjacency Matrix' measure, indicating potential nuances in the data representation.

Additionally, Graph 4 demonstrates consistent similarity levels across all other graphs, indicating minimal variation in similarity measures between each pair of graphs. This consistency suggests that the academic journey depicted in Graph 4 shares similarities not only with the student in Graph 1 but also with the student in Graph 5. This observation implies that Graph 4 may encompass elements common to both the academic paths represented in Graphs 1 and 5, for example from shared modules.

Adjacency Comparison

	Graph 1	Graph 2	Graph 3	Graph 4	Graph 5
Graph 1	100%	86.5%	37.8%	27.7%	0.0%
Graph 2	86.5%	100%	32.1%	24.2%	0.7%
Graph 3	37.8%	32.1%	100%	34.6%	51.4%
Graph 4	27.7%	24.2%	34.6%	100%	26.2%
Graph 5	0.0%	0.7%	51.4%	26.2%	100%

Heat Map Matrix 5: Comparing the similarity of each graph using Adjacency Matrix Measure

5 Evaluation

Graph Similarity

	Graph 1	Graph 2	Graph 3	Graph 4	Graph 5
Graph 1	100%	80%	45%	33%	18%
Graph 2	80%	100%	23%	15%	0%
Graph 3	45%	23%	100%	53%	74%
Graph 4	33%	15%	53%	100%	61%
Graph 5	18%	0%	74%	61%	100%

Heat Map Matrix 6: Comparing each graph using multiple similarity measures weighted to reach quantifiable percentages

When computing the overall similarity score, we took all of the previous similarity measures and weighted them individually. The breakdown was as such:

- Laplace Matrix = 25%
- Adjacency Matrix = 10%
- Graph Edit Distance = 10%
- Graph Distance = 20%
- Spectral Distance = 35%

Both the 'Adjacency Matrix' and 'Graph Edit Distance' were both weighted at 10%. They measure the presence/absence of edges between vertices and dissimilarity between two graphs. Although they add value to the overall understanding of graph similarity, their accuracy was questionable as they drew unusual connections between graphs 3 and 4 when, in reality these were not correct. Graphs 3 and 4 represent different module paths and ultimately this lead to the two similarity measures being of lower importance.

Then on to 'Graph Distance' and 'Laplace Matrix' which received a weighting of 20% and 25% respectively. The 'Laplace Matrix' captures structural information about the graph and the 'Graph Distance' the shortest path lengths. These systemic measures demonstrate a more realistic representation of the graphs and capture a better, more accurate view than previously mentioned measures. Hence why they are more heavily weighted in the final similarity score.

Finally the 'Spectral Distance', this measures the dissimilarity between graphs based on their eigenvalues. This traditionally more mathematical approach captures subtle structural differences that may not be evident from other measures, and produce a more appropriate score as the percentages are calculated using a different method. Spectral methods are powerful tools in graph analysis, and help to reveal important structural characteristics. Assigning a higher weight to 'Spectral Distance' acknowledges its ability to capture nuanced similarities between graphs that other similarity measures cannot achieve.

Overall, the chosen weights reflect a balanced consideration of different aspects of graph similarity, form structural to spectral properties, weighing them individually allows us to utilise the strongest aspects of each similarity measure, leading to the best possible 'Graph Similarity' measure.

Evaluating the Similarity Between Each Students Graphs (1,4 and 5)

Beginning with the comparison between the modelling and business-oriented student (Graph 4) and the pure mathematics student (Graph 5), a substantial 61% similarity is observed. This may be attributed to the foundational core modules shared by all first-year mathematics students or to inherent similarities within the structure of the mathematics degree program. Expanding this research to encompass a wider pool of mathematics and non-mathematics students, coupled with a more comprehensive dataset detailing the skills imparted, could enable academic institutions to predict the curriculum students can expect and the skills potential students will be taught.

Moving forward, when juxtaposing the graphs of the student focusing solely on modelling (Graph 1) with that of the student delving into both modelling and business modules (Graph 4), a moderate 33% similarity is evident. While modelling and

pure maths components will be encountered by both students, the incorporation of business modules introduces a discernible deviation in the data.

Lastly, the graphs representing students with contrasting primary emphases — modelling (Graph 1) and pure mathematics (Graph 5) — exhibit the lowest similarity at 18%. This stark contrast implies that these students opt for markedly distinct modules beyond their first year, highlighting the diverse array of courses available within the mathematics curriculum at the University of Exeter.

Evaluating the Similarity AI produced Graphs

An essential aspect to consider is the striking 80% similarity discovered between the modules and topics of Student 1, who focuses in modelling (Graph 1), and ChatGPT's interpretation and classification of said modules and topics. This indicates that ChatGPT proved highly beneficial and dependable in categorising mathematical knowledge. Such a tool holds promising potential for broader application across various educational contexts which will be discussed in detail later.

When contrasting Graph 3, which represents Gemini's interpretation of Student 1's topics and modules, with ChatGPT's classification, a notable disparity arises with only a 45% similarity observed compared to ChatGPT's 80%. This suggests that Gemini lacks the accuracy required for effectively categorising mathematical data. This discrepancy becomes more pronounced when comparing Graph 3 with Graph 5, where a surprising 74% similarity is observed. This unexpected result indicates that Gemini's categorisation based on Student 1's data is less accurate than anticipated, as the similarity with a student focusing on pure mathematics (Graph 5) is higher than that with the actual student (Graph 1).

Touching on the prompts we used for both ChatGPT and Gemini, which were identical for fair comparison. We used evidence from articles to make these as efficient as possible, by using the 'Situation', 'Task', 'Appearance' and 'Refine' method [13]. An example of a prompt we used followed this template:

`Input:` "You are a neural network model trained on categorising mathematical content to a professional standard for academic purposes. Your task is to categorise the following topics within the module Groups, Rings and Fields on whether the mathematically relate to each other. The output put should be in the form: RDFTriple[".....", "relates to", "....."], "

Then we inputted the output into Mathematica to compute Graph 2 for ChatGPT and Graph 3 for Gemini.

In summary, the utilisation of AI models in Graphs 2 and 3 underscores the significance of precise prompts and training data. ChatGPT demonstrates a superior response to the prompts provided for categorising the data from Student 1 in Graph 1 into RDF triples. Conversely, Gemini requires further refinement to better comprehend the specific characteristics of the student's data. Future iterations may benefit from additional prompts or examples to enhance the training of this AI language model. For example, If we had more time we would like to see if we could keep adding to prompts to get more accurate results, pay to see if ChatGPT4 (a more powerful and up to date AI model) gives more accurate results as well as give critiques until we were more satisfied with the output. Overall, these findings highlight the inherent inconsistencies in AI models and emphasise the necessity for meticulous training and precision, particularly when employing different AI languages.

6 Machine Learning & Neural Networks

We conducted an investigation into the effectiveness of neural networks in categorising mathematical content by developing our own neural network using Mathematica. The objective was to assess its capability in categorising data utilised to generate graphs 1, 4, and 5. Initially, we curated training data from fellow students' data repositories accessible through the RDF sharing forum linked to the module's ELE page. We formatted this data into the appropriate element of a Rule using MapApply[Rule]. Our neural network construction involved creating a basic model to assign MSC codes to strings and establishing suitable encoders and decoders. Following this, we trained the neural network, a process performed on desktops due to computational requirements. We evaluated the accuracy of the trained network using statistics, such as NetMeasurements[trainedNet,trainingData,"Accuracy"], and compared it with the accuracy obtained through the classify function. Despite achieving initial accuracy's of 0.73 and 0.8 for graph 1, further training was necessary. We enhanced the neural network's performance by incorporating a pre-trained neural network, GloVe, until achieving accuracy's above 0.9 for graphs 1, 4, and 5.

To evaluate our network's effectiveness, we used different approaches. Firstly, we compared its classifications with our own and other students' manual ones to check consistency and accuracy. Secondly, we tweaked the network's structure, adjusting parameters like layer count, to see how it affected its ability to categorise maths content accurately. This helped us find the best setup and we confirmed our results using cross-validation. Then most importantly we evaluated the efficiency of the neural network by testing it on untrained data. For this, we gathered additional mathematical content from sources not included in the training dataset to evaluate the network's performance. In this case we used the Imperial College London's Mathematics Programme Structure, and Imperial's Module handbook. We tested the neural network's efficiently in categorising modules by their pre-requisites and topics within modules that relate. By observation we saw the neural network was inefficient in classifying the data by pre-requisites and would require a lot more training to be used in real-world

applications. To quantify this, we used the methods of calculating accuracy stayed above and only achieved a 0.54 and 0.59. Whereas the neural network showed more capabilities in categorising the how topic relate withing modules. By observation we concluded that most topics it categorised together with relations made sense but a couple were over-sighted. Again finding the accuracy, we obtained 0.62 and 0.68, demonstrating the need for more training but also the potential for wider use. By presenting the network with new and unseen data, we assessed its ability to generalise and accurately categorise content beyond what it was trained on. This approach helps validate the network's robustness and provides insights into its effectiveness in real-world applications.

7 Practical Applications of Research in Improving the Student Experience

Understanding the connections between different mathematical topics and resources within the curriculum can profoundly impact the development of personalised learning paths. By meticulously mapping the network relationships between topics and identifying prerequisite connections, students can receive tailored advice on selecting modules that align with their interests and future aspirations. Utilising knowledge graphs to chart academic journeys not only enhances students' ability to visualise and plan their degree paths, but also uncovers common pathways and potential roadblocks, enabling proactive interventions when needed.

Moreover, comprehending the relationships between topics within a module can significantly enhance learning efficiency and serve as a vital revision tool. Visualising dependencies and connections as a network empowers students to pinpoint key concepts and understand their significance within broader knowledge structures. This approach fosters a methodical approach to learning, allowing students to prioritise foundational topics before delving into more complex concepts. The creation of graphs with links between each topic within a module reinforces this approach, facilitating efficient identification of foundational topics and enhancing the revision process. Students can revisit key concepts, identify missed connections, and test their understanding in an engaging manner, thus promoting deeper comprehension.

Semantic Web technologies play a pivotal role in data integration, streamlining the sharing of course information across different platforms and departments, thereby enhancing the overall student experience. This integration could extend further to facilitate the exchange of information between different universities and academic repositories. Additionally, using graph theory to model academic networks and collaborations enables the formation of study groups and peer support networks, promoting interdisciplinary research and collaboration within the academic community. Expanding these efforts to incorporate data from multiple universities can enrich academic networks and collaborations on a broader scale, resulting in greater innovation and collaboration within the academic sphere.

Furthermore, the use of AI models to categorise mathematical content and generate RDF triples, subsequently creating graphs after extensive training, holds immense potential in streamlining this process. This can significantly enhance the student experience by providing personalised recommendations, identifying learning patterns, and offering insights for curriculum refinement. By using neural networks, we can further optimise the student experience, enabling efficient analysis and delivery of insights on a wider scale. This demonstrates how neural networks can be harnessed to improve the student experience and enhance learning outcomes.

8 Further Considerations and Future Research

While the applications outlined present promising possibilities, it's essential to acknowledge potential challenges and limitations. Integrating these approaches into existing educational frameworks while ensuring student privacy and addressing potential biases in network representations will require careful consideration. Ethical considerations are paramount when employing these methods.

Future research avenues could explore the long-term impact and effectiveness of these applications in real-world educational settings. Evaluating their influence on student learning outcomes and refining the approaches based on user feedback will be crucial for successful implementation. This may involve conducting pilot studies in classrooms, gathering student feedback, and analysing learning outcome data to measure the effectiveness of network-based approaches compared to traditional methods.

To enhance future research in education, it is essential to expand datasets by including various sources like public datasets and textbooks, thereby improving adaptability. Given more time, expanding this report to include graphs representing academic journeys of students pursuing different degrees as well as from different institutions would be valuable. Additionally, integrating real-time data, such as course availability and performance, can empower students with timely information for informed decision-making.

Moreover, collaborating with educational institutions will expedite the implementation of research findings and address practical challenges, further advancing the integration of knowledge graphs and AI in education.

References

[1] Lawton G. What is the Semantic Web? Definition, History and Timeline. TechTarget. 2023. [Last accessed on 2024-04-05].

[2] Sandro Hawke PAEPIH Ivan Herman. W3C SEMANTIC WEB ACTIVITY. W3C. 2013. [Last accessed on 2024-04-10].

[3] Authors SWA. The Semantic Web Made Easy. W3C Technology and Scoiety Domain. [Last accessed on 2024-04-10].

[4] Group RW. Resource Description Framework (RDF). Semantic Web Standards. 2014. [Last accessed on 2024-04-07].

[5] Group OW. Web Ontology Language (OWL). Semantic Web Standards. 2012. [Last accessed on 2024-04-07].

[6] Semantics C. Learn OWL and RDFS. Cambridge Semantics. 2024. [Last accessed on 2024-04-07].

[7] Chen X, Huo H, Huan J, Vitter JS. An efficient algorithm for graph edit distance computation. Knowledge-Based Systems. 2019. [Last accessed on 2024-04-11].

[8] Towfic F, Vanderplas S, Oliver C, Couture O, Tuggle C, Greenlee MH, et al. Detection of gene orthology from gene co-expression and protein interaction networks. BMC bioinformatics. 2010. [Last accessed on 2024-04-24].

[9] Weisstein EW. Graph Eigenvalue. Wolfram MathWorld. 2024. [Last accessed on 2024-05-01].

[10] Jovanović I, Stanić Z. Spectral distances of graphs. Linear Algebra and its Applications. 2012. [Last accessed on 2024-05-01].

[11] Fred E Szabo PhD iTLASG. Manhattan Distance. Science Direct. 2015. [Last accessed on 2024-05-01].

[12] Weisstein EW. Laplacian Matrix. Wolfram MathWorld From MathWorld–A Wolfram Web Resource. 2024. [Last accessed on 2024-05-01].

[13] SCOTLAND C. Master ChatGPT Prompts: The STAR Method For ChatGPT. 2023. [Last accessed on 2024-04-20].