Finding Appropriate Subjects for Recognition of Prior Learning using Sentence Transformers

By: Matthew Ghannoum

Introduction

This research project delves into the efficient application of Sentence Transformers to streamline the process of Recognition of Prior Learning (RPL). RPL, an essential practice that acknowledges the skills and knowledge gained from different institutions or work experience, often necessitates extensive review and mapping of a learner's past experiences to new learning opportunities. The goal of this project is to automate and refine this manual process by deploying subject document embeddings instead of Named Entity Recognition of skills as shown in previous research. A novel RPL-NA (Not Applicable) classification model is developed, which helps discern whether a subject is definitely not applicable for RPL for another subject, acting as an automated filter for similarity search. The project details the creation of a new dataset via web scraping, the selection and effectiveness of specific Transformer models, and the integration of these models in a Decision Support System (DSS) application. It also provides insight into the challenges faced during implementation and discusses the results derived from various testing methodologies.

Related Work and Limitations

Common approaches to automating the Recognition of Prior Learning process, is to use an API that maps a corpus to a set of skills in a given ontology using Named Entity Recognition. From there either some kind of overlap metric, or predefined rules are used to determine if RPL can be approved. In later research, it was found that creating a vector space of skills and calculating similarity functioned better [1]. However, issues arise when the skill ontology doesn't contain all the skills available in subjects or when the mapping/Name Entity Recognition isn't accurate. This research project aims to approve upon these methods by focusing on using document embeddings instead of Named Entity Recognition of skills.

Dataset

This research project contributes a new dataset, created through web scraping subjects from the websites of the University of Sydney (USYD) and the University of Technology Sydney (UTS). It includes the raw HTML for a given subject, a subject markdown file and a subject plain text file (with the subject code being stored in the file name). As well as JSON files containing subject-degree, subject-major and subject-recommended year pairs.

The subjects collected include all current subjects (as of 13/04/2024) of the Bachelor of Computer Science (Honours) (UTS), Bachelor of Advanced Computing (USYD), Bachelor of Engineering (Honours) (UTS and USYD), Bachelor of Nursing (UTS) and Bachelor of Nursing (Advanced Studies) (USYD).

Moreover, we contribute a novel Python CLI tool specifically made for extracting data from university websites, processing the page data and generating document embeddings from them.

It works by defining a university configuration, an example of which is shown below:

```yaml
university:
  name: University of Sydney
  abbreviation: usyd
  degrees:
    - name: "Bachelor of Advanced Computing"
      url: "https://www.sydney.edu.au/handbooks/engineering/advanced_computing/advanced_computing_table.html"
      majors:
        - name: Computational Data Science
          url:
"https://www.sydney.edu.au/handbooks/engineering/advanced_computing/majors/computational_data_science_table.html"
        - name: Computer Science
          url:
"https://www.sydney.edu.au/handbooks/engineering/advanced_computing/majors/computer_science_table.html"
        - name: Cybersecurity
          url: "https://www.sydney.edu.au/handbooks/engineering/advanced_computing/majors/cybersecurity_table.html"
        - name: Software Development
          url:
"https://www.sydney.edu.au/handbooks/engineering/advanced_computing/majors/software_development_table.html"
    - name: "Bachelor of Engineering (Honours)"
      url: "https://www.sydney.edu.au/handbooks/engineering/engineering_honours/engineering_core_table.html"
      majors:
        - name: Civil
          url: "https://www.sydney.edu.au/handbooks/engineering/engineering_honours/civil/table.html"
        - name: Mechanical
          url: "https://www.sydney.edu.au/handbooks/engineering/engineering_honours/mechanical/table.html"
        - name: Mechatronic
          url: "https://www.sydney.edu.au/handbooks/engineering/engineering_honours/mechatronic/table.html"
        - name: Software
          url: "https://www.sydney.edu.au/handbooks/engineering/engineering_honours/software/table.html"
        - name: Electrical
          url: "https://www.sydney.edu.au/handbooks/engineering/engineering_honours/electrical/table.html"
        - name: Biomedical
          url: "https://www.sydney.edu.au/handbooks/engineering/engineering_honours/biomedical/table.html"
    - name: Bachelor of Nursing (Advanced Studies)
      url:
"https://www.sydney.edu.au/handbooks/medicine_health/coursework/nursing_advanced_studies/nursing_advanced_studies_ta
bles_full_time.html"
  subject_options:
    url_criteria:
      prefix: "https://www.sydney.edu.au/units/"
      container_type: tag
      container_selector: strong
    start_line:
      value: "# "
      match_type: startsWith
    end_line:
      value: "Leadership for good starts here"
      match_type: equals
```

Then running the main Python file, either with or without the provided flags to enable or disable certain functionality.

```
python main.py
Loading university configurations...
University configurations loaded successfully!
Number of universities: 2

Would you like to continue (Y/n)? y
Scraping university degree pages...
University degree pages saved successfully!
Number of NEW pages scraped: 0

Would you like to continue (Y/n)? y
Getting set of subject codes...
Subject codes retrieved successfully!
Number of unique subject codes for UTS: 287
Number of NEW subject codes for UTS: 50
    Number of unique subject codes for Bachelor of Computing Science (Honours): 98
    Number of unique subject codes for Bachelor of Engineering (Honours): 155
    Number of unique subject codes for Bachelor of Nursing: 50
Number of unique subject codes for USYD: 364
Number of NEW subject codes for USYD: 238
    Number of unique subject codes for Bachelor of Advanced Computing: 109
    Number of unique subject codes for Bachelor of Engineering (Honours): 304
    Number of unique subject codes for Bachelor of Nursing (Advanced Studies): 24
Number of unique subject codes: 651
Number of NEW subject codes: 288
```

Methodology

Generating Subject Document Embeddings for STS

Pre-trained Sentence Transformer models are being used to generate embeddings from subject page information. This is because they have been shown to perform the best on the downstream task of document similarity, which is what this RPL recommendation relies on. Moreover, they are far smaller and more efficient than popular Transformer models used for Generative AI such as OpenAI's GPT and Meta's Llama.

There are many Pre-trained Sentence Transformers available from HuggingFace (the platform that hosts the files required for the transformers), all with their own unique attributes and performance characteristics. As such, the Massive Text Embedding Benchmark (MTEB) was used to select the best performing Semantic Textual Similarity (STS) models [2][3]. The models selected for this research were also required to be open source, so they could be run locally for reproducible results without cost. As such, we selected the mxbai-embed-large-v1 and instructor-xl Transformers.

Also included in our tests is the paraphrase-MiniLM-L6-v2 Transformer as it was utilized in prototyping solutions earlier in this research project. It will also serve as representation for small Transformers with smaller embedding sizes. Furthermore, older non-transformer based embedding models such as Doc2Vec (based on Word2Vec) and GloVe (using mean aggregation to form a document embedding) were included to proved a point of reference for baseline performance.

Name	Type	Architecture	Embedding size
mxbai-embed-large-v1	Sentence Transformer	BERT	1024
instructor-xl	Sentence Transformer	BERT	768
paraphrase-MiniLM-L6-v2	Sentence Transformer	Siamese BERT	384
Glove (mean aggregation)	Statistical Model	Statistical (Co-occurence)	300
Doc2Vec	CNN	Skip-gram	384

Similar subjects are discovered using Hierarchical Navigable Small World (HNSW) graphs to conduct an Approximate Nearest Neighbour (ANN) search. Cosine similarity is used as the similarity metric between two embeddings. The database this functionality is implemented in is called "Weaviate".

Testing Methodology

Determining how well our models "recommend" subjects in RPL applications is not straightforward, as there are potentially multiple outcomes to optimise for. Given there existed a testing methodology which accurately evaluated the similarity of a source subject to a suggested subject. This in itself may not be an accurate evaluation of the suggested subject's likelihood of being approved for RPL credit.

For example, given a foundational subject at Institution A and an advanced subject Institution B may cover the same content (at different levels) and be semantically similar. However, you would not expect to be able to claim credit for the advanced subject using the foundational subject. Moreover, a given subject may be very niche and simply not have many (if any) equivalent subjects available at other institutions. This means the testing methodology will need to evaluate multiple qualities of a subject, rather than just the semantic similarity.

The testing methodology is made more convoluted by the fact that ground truth data cannot be solely used for evaluation. This is because RPL precedence data may not have been collected by an institution or may not be accessible in a given timeframe (as is the case in this study). However, if the this precedence data were available, it may not include records for all subjects pairs (hence a source and suggested subject may not be able to be evaluated as correct or incorrect).

Ergo, the testing methodology adopted in this study utilises multiple tests to give a well rounded depiction of the performance of the provided models.

RPL-NA Classification

Previously, a method for measuring the similarity of two subject documents was provided. It was also mentioned that the "subject similarity", may not directly predict whether a source subject is RPL applicable for another subject accurately. As such, a model for RPL-NA classification will be trained and evaluated. This model could potentially work as an ensemble with the previous method, to improve the final RPL recommendations.

RPL-NA (RPL Not Applicable) classification, is the task of classifying if a subject is definitely not applicable for RPL for another subject. If true, the subject pair is definitely not applicable for RPL but if false it is unclear whether or not RPL is appropriate.

The problem is formulated using the negation of the original problem as ground truth data of whether or not a subject pair is RPL valid isn't available. However, a dataset for RPL-NA can be constructed using known RPL rules. These rules are that:

1. A USYD subject is applicable for RPL for a UTS subject, if the minimum year the USYD subject can be taken is greater than or equal to the UTS subject minimum year. And:
2. A USYD subject is applicable for RPL for a UTS subject, if they share at least one major (or either are a core subject).

Training the RPL-NA classification model is a better solution than just statically applying the aforementioned rules as the model can generalise. As such, it can be used on data that doesn't explicitly define the year and major(s) of a given subject pair.

The classifier being trained on the subject document embeddings is the Random Forest Classifier from the Scikit-Learn Python package. With the number of estimators being 100.

Implementation in a Decision Support System

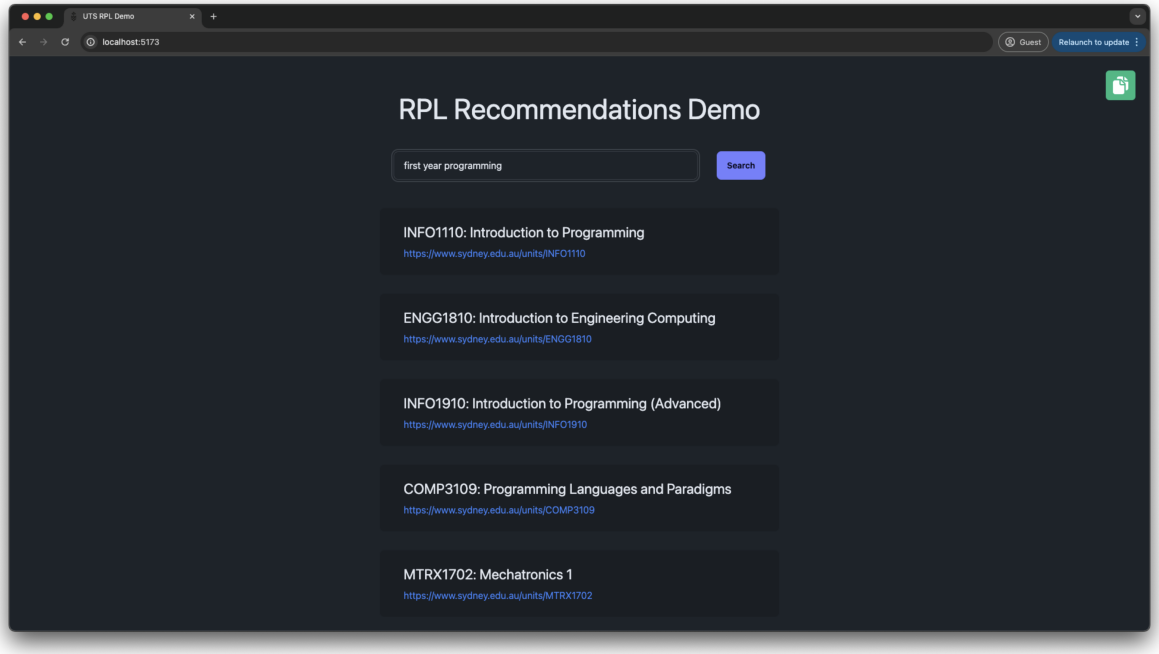
The main objective of the NLP models developed is to support students in searching for subjects that could likely get RPL approval for. While there are some stakeholders that would benefit indirectly from this DSS (Decision Support System) or the integrated models. For instance, administration staff responsible for processing RPL applications could use the RPL-NA classification model to automatically reject certain subjects from an application. Students are the target focus for the application itself.

Despite the focus being on students, the onus of data collection (required for the models and pre-filling the application) could either be placed on students or universities. The entity responsible for data collection is referred to as being "centred" to that entity (e.g. university centred meaning data collection is done by the university).

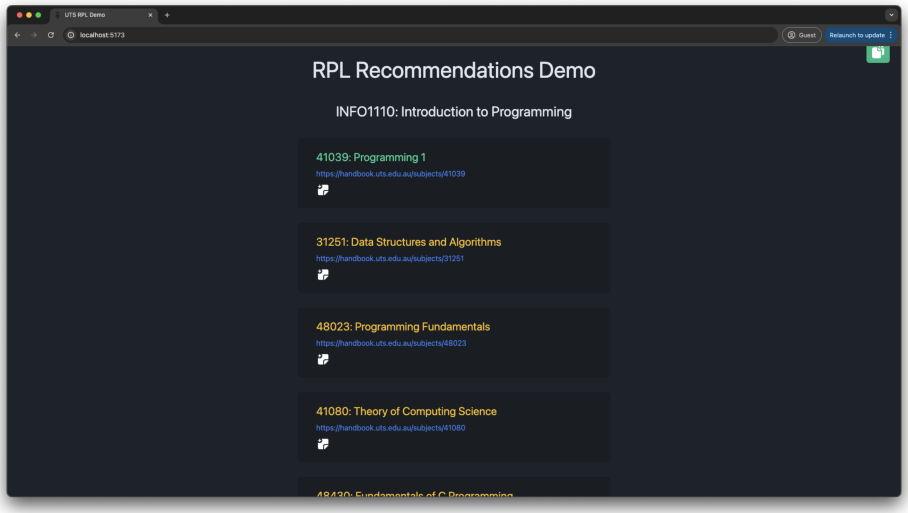
University Centred RPL Recommendation Application

The University Centred RPL Recommendation application uses a search bar and subject-modals as the primary mechanism for selecting subjects.

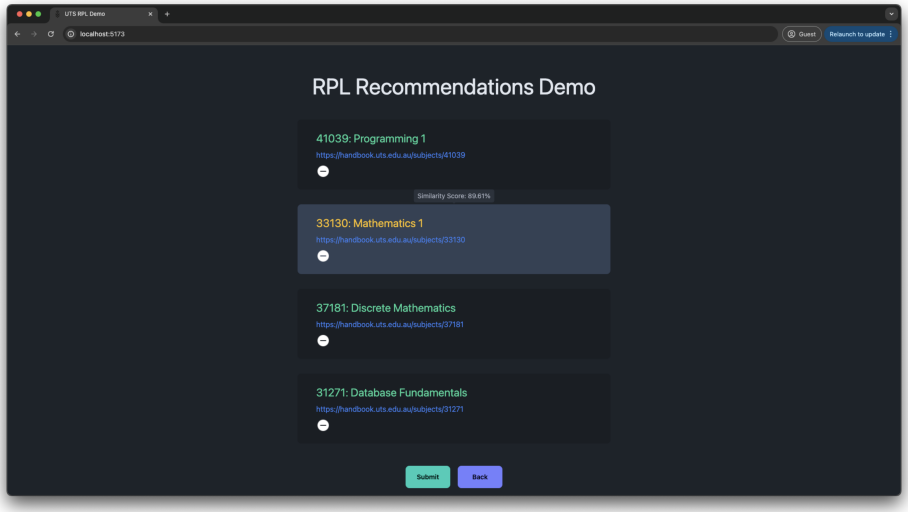
For example, if a student wishes to get recommendations for the "first year programming" subject at USYD, they can first search that prompt. The USYD subjects that are similar to the phrase are then loaded in as shown below.



The student then clicks on “Introduction to Programming” as that is the subject they intended to apply for RPL with. After which, they get 5 suggestions, the top one is the UTS subject “Programming 1”, which has the highest similarity (green colour means subject has a similarity score of 90% or higher).

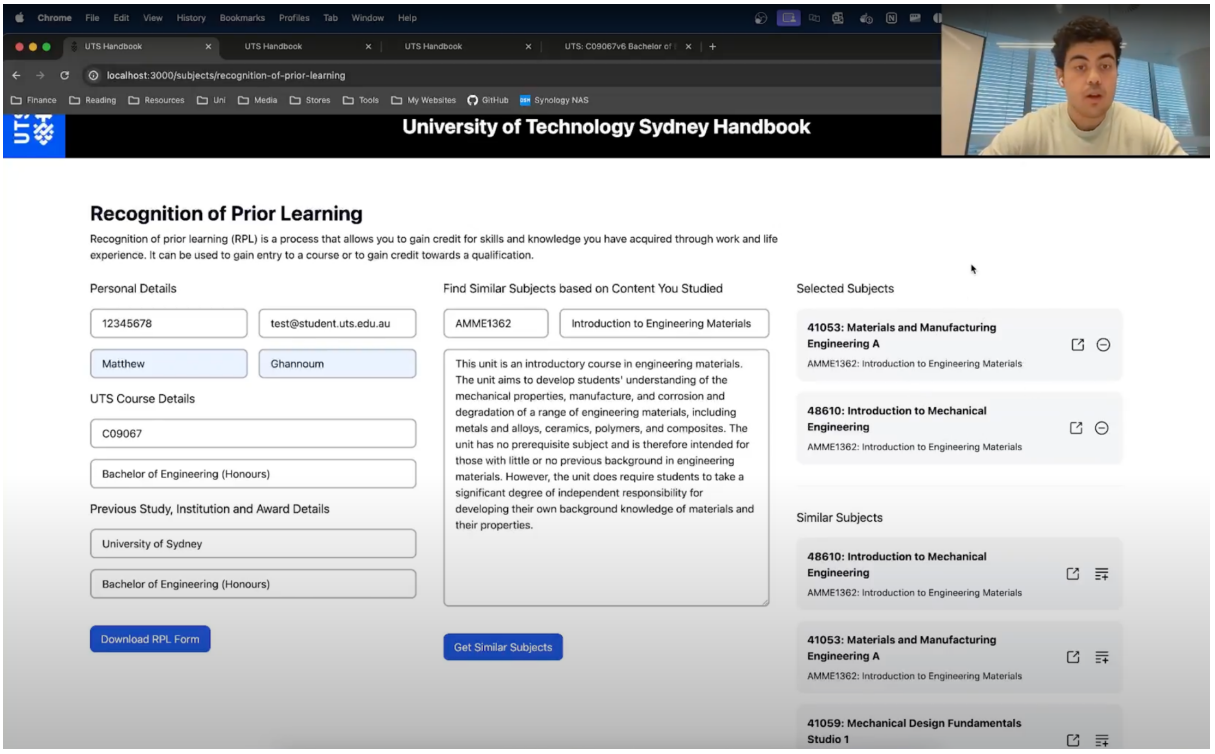


The user can then click on the subjects they wish to apply for RPL, given the provided USYD-UTS subject pair, which can be viewed as shown below. From which they can click subject and get a pre-filled RPL form.

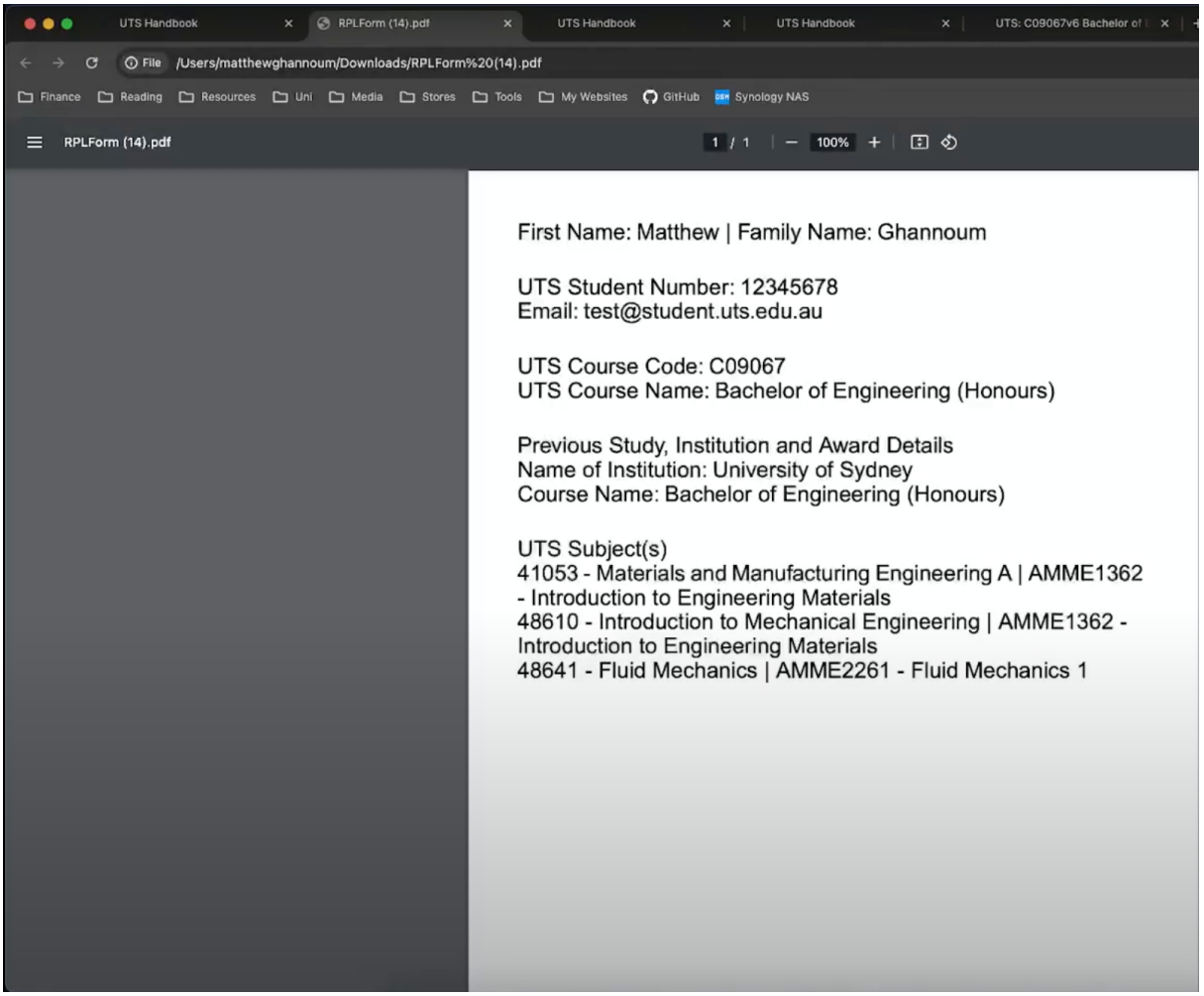


Student Centred RPL Recommendation Application

The core of the Student Centred RPL Recommendation application is the previous institution subject form, which is shown in the middle column before. Unlike the University Centred version of the application, this application doesn’t have any subject document embeddings or data from subjects that aren’t from UTS. This also means this application doesn’t use the RPL-NA classification model (only STS based on embeddings). However, it enables for subjects recommendations to be made from any institution.



The pre-filled RPL application PDF is shown below.



Results and Discussion

Evaluation using Subject Metadata

The two approaches to evaluating subject similarity, are using an accuracy/error metric and visualisation of the embeddings.

Accuracy can be measured by the proportion of recommended subjects with the same metadata value as the source subject to the total collection of subjects recommended. If the metadata value is numerical rather than categorical (as is the case with the year attribute), error can also be calculated.

For each test, we calculate metrics based on the top 2, top 5, top 10 most similar subjects to the source subject. This enables models to be compared with smaller and wider tolerances.

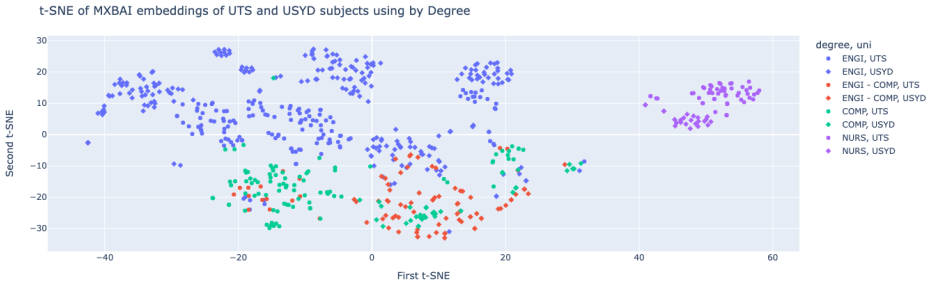
Meanwhile the subject document embeddings can be visualised on a scatter plot, so that the effectiveness of subject clustering can be observed. Since the embeddings are in very high dimensional space (embedding size ≥ 300), T-distributed Stochastic Neighbour Embedding (t-SNE) is used to project each embedding into a two dimensional space, so it can be visualised in a scatter plot.

Shared Degree Accuracy

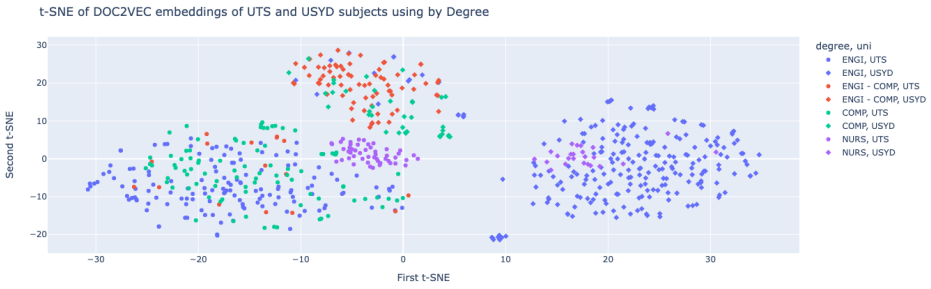
	Top 2 Subjects	Top 5 Subjects	Top 10 Subjects
MXBAI	93.54	94.23	93.32
INSTRUCTOR	92.86	92.36	91.70
SBERT	87.64	87.58	87.09
GLOVE	74.31	71.04	68.08
DOC2VEC	33.65	33.68	29.73

The Mxbai model achieves the highest accuracy across each set size, with the highest accuracy being achieved at 94.23% in the top 5 recommended subjects. However, the Instructor model performed very closely, within 1-2% of Mxbai model.

This is reflected in the t-SNE visualisation where we can see subjects of the same degree being closely located. The only significant overlap being subjects that are both present in Engineering and Computer Science degrees (points coloured red).



Contrastingly, the Doc2Vec model clearly performed the worst achieving 33.68% accuracy at best. This is mirrored in the corresponding Doc2Vec t-SNE graph, as many subjects from different degrees are located very closely to each other. For example, a cluster of Nursing subjects appearing within the larger cluster of Engineering subjects, on the right side of the graph.



Shared Major Accuracy

	Top 2		Top 5		Top 10		
	COMP	ENG	COMP	ENG	COMP	ENG	Mean Ac
MXBAI	57.34	75.99	**60.37**	73.82	**59.27**	75.20	67.00
INSTRUCTOR	**58.26**	**78.45**	57.43	**77.04**	57.43	**75.76**	**67.40**
SBERT	**58.26**	72.20	58.72	74.54	57.16	72.66	65.59
GLOVE	54.13	76.32	52.48	75.26	52.11	69.61	63.32
DOC2VEC	35.32	43.09	33.39	48.36	40.28	49.54	41.66

The table above assumes that a core subject (one with no assigned major) is equivalent to a subject with a major. **Nursing is not included as it does not have any majors available (at both UTS and USYD).**

The results from the shared major test show again, that the Mxbai and Instructor models perform the best. The Mxbai model having the highest mean accuracy of 67.40 and the Instructor model having the highest median accuracy of 67.10.

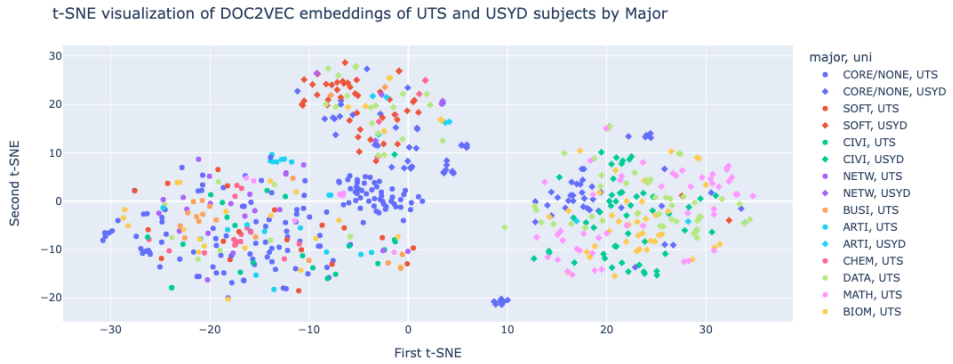
However, unlike the previous shared degree test, the Sbert and GloVe models perform quite closely as well, achieving 65.59% and 63.32% mean accuracy respectively.

	Top 2		Top 5		Top 10		
	COMP	ENG	COMP	ENG	COMP	ENG	Mean Ac
MXBAI	34.86	34.70	32.29	34.87	29.17	32.14	33.01
INSTRUCTOR	**36.70**	**43.09**	**34.50**	**38.62**	**31.28**	**34.57**	**36.46**
SBERT	29.36	31.74	30.09	29.93	30.18	28.12	29.90
GLOVE	22.48	23.19	20.55	20.26	20.73	17.86	20.85
DOC2VEC	6.42	10.03	6.42	11.91	10.83	12.07	9.61

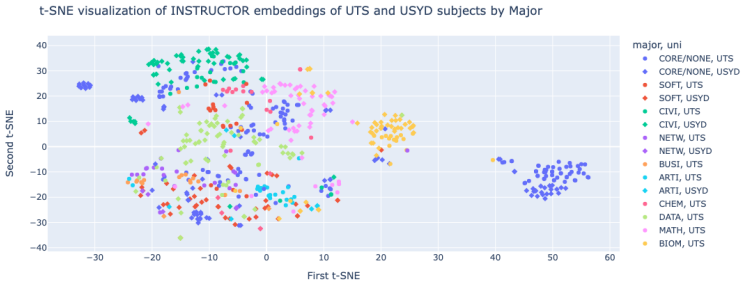
The following table assumes that a core subject (one with no assigned major) is not equivalent to a subject with a major.

Removing the assumption that a core subject is equivalent to any major (hence a core subject can be thought of as being of the “core major”), accuracy is reduced significantly for each model. This test is a measurement of “worst case performance” as in reality some core subjects with no assigned major, are actually quite similar to subjects of a given major and are thought of as such.

This also wides the accuracy gap as Instructor model performs the best in both mean and median accuracy, with Mxbai within 3% of it. While Sbert and GloVe are 7% and 15% less accurate on average than the Instructor model.



Again it is evident that Doc2Vec performs the worst by a clear margin, which is echoed in the lack of clear clustering the it's t-SNE visualisation.

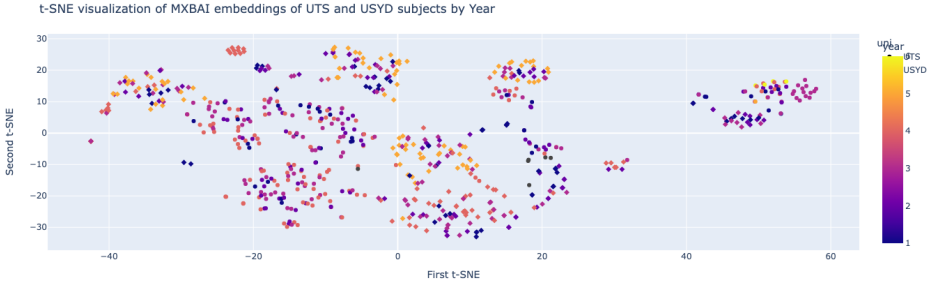


As opposed to the Instructor model which clearly locates subjects with the same major closely, with similar majors being located closely together as well. For example, Civil subjects (green) and Mechanical subject (pink).

Moreover, it visualises how core/no major subjects are scattered across clusters of different majors and how that similarity may be inherently true.

Equal Year Accuracy and Year MAE

	Top 2		Top 5		Top 10	
	Accuracy	MAE	Accuracy	MAE	Accuracy	MAE
MXBAI	28.73	1.21	**24.46**	1.3	21.93	1.39
INSTRUCTOR	24.34	1.28	23.98	1.32	23.53	1.34
SBERT	23.31	1.23	23.47	**1.27**	22.84	1.32
GLOVE	21.43	1.34	21.04	1.34	20.05	1.36
DOC2VEC	20.3	1.6	18.15	1.63	17.52	1.6



Recommending subjects of the same or similar year as the source subject proved to be difficult for all models. While, this could be indicative of the models poorly understanding the subject level (and a relation of level to year), it could also be caused by a number of other factors.

Probably the largest factor being that the year a subject is recommended to be taken is not provided in the USYD Units of Study. Rather only the level of the subject (1000, 2000, 3000, etc). Hence, the starting digit of the level was used to approximate the year it was taken (which may not be accurate in some cases).

Moreover, subjects can be taken at various times in a degree, and there are usually multiple recommended times/years. As such, the minimum year recommended was used for UTS subjects.

Unlike the previous tests where the intuition is that similar subjects should have share a degree or major. This intuition does not hold true with the subject year. There many levels a subject's content can be taught at over the years of a degree (e.g. foundations/fundamentals, intermediate, advanced, etc). As such, it is likely that subjects from different recommended years would be more semantically similar than subjects taken in the same year.

Evaluation Non-RPL Classification Model

The following results are generated using a Random Forest classifier trained on the Instructor document embeddings created previously. The hyperparameter of n-estimators was set to 100 and the train-test split was 80-20.

	Precision	Recall	F1-Score	Support
False	0.97	0.98	0.97	5178
True	0.97	0.95	0.96	3751
Accuracy			0.97	8929
Macro Average	0.97	0.97	0.97	8929
Weighted Average	0.97	0.97	0.97	8929

Given the F1-Score is 97%, this indicates the model can reliably classify subject document embeddings as RPL-NA or not.

However, evaluation using Stratified K-folds is required to ensure this model performs well on unseen subject documents.

Future Improvements and Work

- Combining word embeddings with node embeddings for better representation of program structure and time.
- Train a classifier or a transformer on text classification of ground truth data.
- Survey group of students and staff on their opinion of the quality of recommendations.
- Survey group of students on the user experience of the application.
- Contain to refine and build the application.
- Move from Weaviate to PgVector.

Conclusion

In conclusion, this research project has successfully demonstrated the use of Sentence Transformers and various NLP techniques to develop an efficient and accurate system for recommending subjects in Recognition of Prior Learning (RPL) applications. The system's ability to generate meaningful subject recommendations based on semantic textual similarity is promising, and the use of machine learning models to classify RPL applicability further enhances its effectiveness. The user-centric design of the system also ensures that it is practical and beneficial for end-users, primarily the students. However, there are opportunities for further improvements and refinements, particularly in enhancing the representation of program structure and time, and in collecting more ground truth data for model training. Future work may also include obtaining more user feedback to refine the system and potentially transitioning to PgVector for better performance. This research contributes to the ongoing efforts to leverage artificial intelligence in higher education and opens up new possibilities for enhancing the student experience through personalized learning pathways.

References

1. Kitto, K, Sarathy, N, Gromov, A, Liu, M, Musial, K & Buckingham Shum, S 2020, 'Towards skills-based curriculum analytics', *Proceedings of the Tenth International Conference on Learning Analytics & Knowledge*.

2. Niklas Muennighoff, Tazi, N, Magne, L & Reimers, N 2022, 'MTEB: Massive Text Embedding Benchmark', *arXiv (Cornell University)*, Cornell University.
3. *MTEB Leaderboard - a Hugging Face Space by mteb* n.d., huggingface.co.