Predicting Prerequisites of Learning Concepts in Educational Data

Quynh Anh Nguyen
ETH Zurich - Swiss Federal Institute of Technology
quynguyen@student.ethz.ch

Abstract

The project aims to adapt a pretrained prompting and prediction system to handle the task of predicting prerequisite dependencies in educational data, representing a fundamental step towards automatic extraction and detection of prerequisites in educational texts. Diverging from previous approaches that treated this task as a network science puzzle, the author leveraged the capabilities of Large Language Models (LLMs), such as GPT-3.5 and Llama 2, and compared their performance to smaller pretrained models, including T5 and GPT-2. Results indicate that utilizing smaller pre-trained models through fine-tuning and prompting can yield significantly improved results, surpassing not only those Large language models but also demonstrating higher predictive performance compared to prior methodologies. This underscores the potential advantages of harnessing ordinary pre-trained models over LLMs in terms of performance and computational resources, prompting intriguing considerations regarding the trade-offs between model size and depth.

1 Introduction

The accurate dissemination of prerequisite information for each course in a study program is crucial in helping students navigate their educational curriculum efficiently. With this information, students can make well-informed decisions when choosing classes, ensuring they have the necessary skills for positive outcomes. Therefore, this project aims to develop a comprehensive framework for automatically extracting learning concepts from various educational materials, such as textbooks, video descriptions, and university course slides. Additionally, within this framework, we propose a model to effectively identify prerequisite dependencies among the extracted learning concepts. By achieving these objectives, we seek to improve students' educational journeys and streamline decision-making processes.

Problem Formulation This section addresses two primary challenges, namely: (1) the automated detection of learning concepts within textbooks and (2) the identification of prerequisite dependencies among the recognized concepts. In this projects, we focus on the later problem which is to establish a network of learning concept in the textbook and then predict the prerequisite dependencies of them.

Task 1: Learning Concept Extraction To develop an automated model for extracting learning concepts from educational text data, we formulate it as a token classification task, specifically, a keyword extraction task. Given a set of text data, the task is to extract the keywords from each chapter or lesson. In this project, we would use the given data which all the learning concepts are already extracted.

Task 2: Prerequisite Dependencies Prediction The second task can be formulated as a binary classification problem. To effectively leverage word embeddings and the network of learning concepts within the corpus, we represent them mathematically as a graph. The following notations and problem formulation are provided:

- Denotation: The data is composed by $D = \{o_1, o_2, o_3, ..., o_i\}$ in which o_i is the i^{th} object. In particular, a book chapter or a video is regarded as an object, with each keyword extracted from Task 1 being considered as a concept. Typically, each object contains multiple concepts.

- Problem Formulation: The task of detecting prerequisite dependencies among the extracted concepts has been framed as a binary classification problem. Given a corpus $\mathcal{D}$ and its associated course concepts $\mathcal{K}$, the goal is to learn a function $\mathcal{P}$: $\mathcal{K}^2 \rightarrow \{0, 1\}$ that maps a concept pair $\langle a, b \rangle$, where $a, b \in \mathcal{K}$, to a binary class for prediction.

The project introduces state-of-the art methods to the old problem of predicting prerequisite dependencies in educational data. It compares the performance of large language models (LLMs) with smaller pre-trained models (LMs). A key finding is that smaller models, through fine-tuning and prompting, can surpass the performance of larger models, offering better predictive performance with less computational resources. This approach also aims to mitigate the hallucination issues associated with LLMs, showcasing the potential of smaller models in specific tasks. The distinguishing feature of this approach differs from traditional methods that may require manual identification and extraction of learning concepts from educational materials. By using pre-extracted learning concepts, the project can focus more on the predictive modeling aspect, potentially improving efficiency and accuracy in identifying prerequisite relationships. The contributions of the project include:

- The aggregation and creation of an educational dataset suitable for extracting learning concepts in textbook and detecting the prerequisite dependencies between them, sourced from various types of educational materials.

- Applying cutting-edge technology, we leverage the power of large language models for detecting learning concepts in textbooks. This approach involves a comparison between two strategies: utilizing large language models with few-shot learning, and leveraging the power of smaller models through fine-tuning and prompting.

2 Related works

In this section, a summary of research works that are relevant to extracting learning concepts, predicting the prerequisites dependencies in educational data are provided. In the last part, several works about the performance and how to adapt LLMs model and prompting technique are current used for keywork extraction and classification problem.

2.1 Learning Concept Extraction

The task of automatically extracting learning concepts from textbooks has gained significant attention in the field of natural language processing and education technology. This process involves identifying and extracting key terms or concepts from educational materials to enhance content understanding and facilitate knowledge organization. In this review, we explore related works that employ methods such as keyword extraction and entity linking to achieve this goal.

The work of EntQA (Zhang et al., 2021) has garnered attention for its innovative approach to entity linking. Notably, the EntQA model (Zhang et al., 2021) achieves an impressive F1 score of 85.8%, firmly establishing itself as the state-of-the-art performer on the AIDA-CoNLL dataset (Hoffart et al., 2011). In contrast, the pre-GENRE model (De Cao et al., 2021) achieves an F1 score of 85.5%, placing it among the competitive methods for the AIDA-CoNLL dataset (Hoffart et al., 2011) but falling short of the current pinnacle. Additionally, the GENRE model (De Cao et al., 2021) brings a novel perspective by incorporating pretraining and performing on the Mewsli-9 dataset (Botha et al., 2020), attains an F1 score of 87.2%.

Furthermore, learning concept sub-task could be re-framed as keyword extraction, with the potential to utilize pre-trained information in a specific domain and customize it for previously unseen datasets. Several shared tasks have been published in SemEval over the past years, addressing this problem and resulting in the development of multiple proposed models that achieve significant performance improvements. Some of these models include Phraseformer (Nikzad-Khasmakhi et al., 2021) which achieved an F1 score of 69.87% in the keyword extraction task in SemEval 2010 (Kim et al., 2010), SemEval 2017 (Augenstein et al., 2017), and the Inspec (Hulth, 2003) dataset. The method described in this paper combines the latent representation of each keyword candidate, derived from BERT (Devlin et al., 2019) and information obtained from various graph embedding techniques such as ExEm (Nikzad-Khasmakhi et al., 2021), Node2vec (Grover and Leskovec, 2016), and DeepWalk (Perozzi et al., 2014). The latent representation is then fed into a classifier for token classification, with Random Forest (Liaw and Wiener, 2002) outperforming other classifiers. According to Kulkarni et al. (Kulkarni et al., 2022) , the study involves experiments with various masking strategies for pre-training transformer language models in both discriminative and generative settings. In the discriminative setting, Keyphrase Boundary Infilling with Replacement (KBIR) is introduced which resulting in significant performance gains

over the previous state-of-the-art methods, with F1 scores of 62.72%, 40.15%, and 62.56% for the Inspec (Hulth, 2003), SemEval 2010 (Kim et al., 2010), and SemEval 2017 (Augenstein et al., 2017) dataset, respectively. Authors also fine-tuned a language model pre-trained using KBIR (Kulkarni et al., 2022) for the key-phrase extraction task. Promptrank (Kong et al., 2023) proposed an effective unsupervised approach based on a pre-trained language model. It involves feeding the document into the encoder and calculating the probability of generating candidates using a designed prompt by the decoder, resulting in a relative improvement in F1 scores of 34.18%, 24.87%, and 17.57% for 5, 10, and 15 returned results on the same set of data as in Phraseformer (Nikzad-Khasmakhi et al., 2021) and KBIR (Kulkarni et al., 2022) model.

2.2 Prerequisite dependencies prediction

The field of learning concept prerequisite relation has seen significant progress in various model approaches. In prior studies, the problem is frequently framed as binary classification, aimed at determining whether a dependency relationship exists between pairs of learning concepts. Most existing models have attempted to measure the latent representation of learning concepts and the network embedding of the extracted learning concepts.

The RefD model (Pan et al., 2017) achieved an impressive F1 score of 72.6 on the MOOC dataset by employing seven distinct features to measure multiple latent representations. It further employed classifiers like regression models, SVM (Hearst et al., 1998), and Random Forest (Liaw and Wiener, 2002) for binary classification.

Li et al.(Sun et al., 2022) propose ConLearn model, which measures concept embeddings by fine-tuning pretrained language models and constructs concept graphs from pairs of concepts. The model updates concept representations using Gated GNN (Li et al., 2016) and leverages information from related concepts through a self-attention network. Like the CPRL framework (Jia et al., 2021), ConLearn employs a Siamese network for the classification of concept pairs. the model outperforms CPRL (Jia et al., 2021) in most tasks, though it falls short of RefD's performance (Pan et al., 2017) in two specific datasets.

2.3 Pre-trained, Prompt, Predict

Concerning keyword extraction and classification, Large Language Models (LLMs), such as Llama 2 (Touvron et al., 2023) and GPT-4 (Brown et al., 2020) have been pivotal, particularly through techniques such as prompting, retrieval-augmented generation (RAG) (Gao et al., 2024), and fine-tuning. Prompting involves guiding LLMs to generate desired outputs by framing tasks as 'questions' or 'query', effectively utilizing the model's pre-trained knowledge (Liu et al., 2021). RAG method (Balaguer et al., 2024) combines the power of LLMs with external knowledge sources, which could be knowledge base or extra documents, enhancing the model's ability to extract and classify keywords by providing it with relevant context. Fine-tuning, on the other hand, adapts LLMs or Language models such as T5 (Raffel et al., 2023) or GPT-2 (Radford et al., 2019) to specific tasks by training them on a targeted dataset, improving their performance on specialized keyword extraction and classification problems. These methods showcase the versatility and adaptability of LLMs in handling nuanced tasks in natural language processing, demonstrating significant advancements in the field ().

3 Dataset

In this section, author introduces an educational data collection that has been aggregated and prepared for use in both extracting learning concepts and predicting prerequisite dependencies. Additionally, we provide details on the dataset used for predicting prerequisite dependencies experiments by prompting pre-trained models.

3.1 Educational Data Collection

Throughout the project, multiple datasets were aggregated, annotated, and pre-processed to be used for predicting the prerequisite dependencies of learning concepts. We collected and aggregated various types of educational data, including video descriptions, slides, and textbooks. These included lists of documents and lists of keywords. The text was tokenized, and BIO-tag labels (Beginning, Inside, Outside) were assigned to each token. The data for extracting learning concepts resembled typical datasets for a token classification task, where each tokenized token from the provided text was accompanied by a corresponding gold label in the BIO format. Examples of the anticipated data for Task 1 can be found in Table 3, while examples of the data for predicting the prerequisites dependencies are provided in Table 4. The data collections include:

- University Course (Roy et al., 2019) The dataset includes text data from course descriptions, a list of learning concepts, prerequisite relationships between each pair of these learning concepts, and the corresponding Wikipedia article for each learning concept.
  - Courses: Names and course descriptions in text format.
  - General list of: Learning concepts, prerequisite relations, and number of annotators.
- MOOC ML (Yu et al., 2020) The dataset initially comprised lecture descriptions of videos and a list of extracted learning concepts. Following this, the original MOOC dataset was annotated to incorporate information regarding prerequisite relationships between the learning concepts, while the later MOOCCube (Yu et al., 2020) did provide a prerequisite relationship annotation between extracted learning concepts. Therefore, MOOCCube (Yu et al., 2020) is suitable for addressing both Task 1 and Task 2. There are two files: Here is the corrected LaTeX itemization:
  - Candidates: Labels (1 if the candidate is a learning concept, doubly annotated), k-grams information, text - a list of learning concepts.
  - Captions: course_id, caption in text, POS (Part-Of-Speech) tags.
- LectureBank (Li et al., 2019) The dataset comprises extracted learning concepts from online courses and their associated prerequisite dependencies. Due to a unique agreement, the author has chosen to provide only URLs, rather than the actual text data collection. To obtain the data, I performed web crawling and acquired it in the form of PDF and PowerPoint files. My next step for this particular dataset involves extracting text from these files using the tools pdftotext and pptx2text, followed by the addition of BIO-tags to each token within the corpus.
  - Text from PDF and PPTX files.
  - vocabulary.txt: A list of learning concepts.
  - Others: prerequisite_annotation.csv and taxonomy.csv, which show information about the dependencies of these learning concepts.

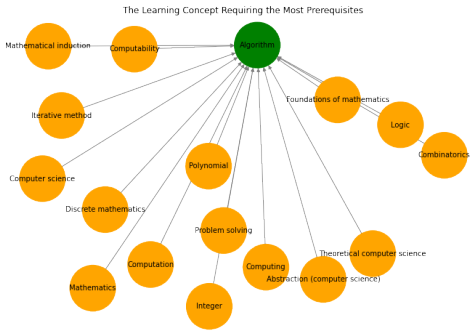

Figure 1: The prerequisite of learning 'Algorithm'

- Fredin's thesis dataset (Thazhathukunnel, 2021) In the context of Task 1, the Complete OpenStax dataset, which had been previously manually annotated by Fredin's team, was further included. This dataset was subsequently integrated into the dataset collections to address Task 1. Additionally, for Task 2, I conducted annotations with five annotators for 100 pairs of learning concepts that had been previously extracted, potentially for use in this task.

3.2 Dataset for Prerequisite Prediction

In the prerequisite prediction task, the final target is to establish a network of learning concepts based on their content dependencies. In other words, the networks would tell us which concepts should be known before learning a certain concept. For example, figure 5 shows that 'Algorithm' is a fundamental learning concept for a vast number of other computer science learning concepts, i.e., Algorithm should be studied first in most Computer Science degrees. Besides, Algorithm also has its own prerequisite requirements as shown in 1. In order to predict prerequisite dependency given extracted learning concepts, we try to detect all pairs of learning concepts that have this prerequisite relationship.

Data Processing

For the prerequisite prediction task, we utilized the University Course dataset (Roy et al., 2019). This dataset comprises 345 extracted learning concepts from 654 documents, each representing a university course description. These descriptions were analyzed to extract learning concepts contained within each. Subsequently, each course description was

tokenized into a bag of concepts, with individual concepts being annotated as learning concepts if they correspond to unique English Wikipedia articles. However, there is a huge imbalance in the distribution of labels between pairs with and without dependencies. Figure 2 shows that 99.2% of all concept pairs are without any dependencies, and only 0.8% have a prerequisite relationship. The learning concept pairs, called shortly "prerequisite pairs" amount to 1008 pairs out of the total 118,680 possible pairs sampled from 345 learning concepts.

For the sake of mitigating the effects of the imbalanced dataset and to be comparable with previous research conducted on the University Course dataset, we employed the oversampling technique for the prerequisite pairs (preq(+)) and applied negative sampling for the pairs without dependencies (preq(-)).

Dataset Base The dataset base is used for finetuning and prompting smaller models as mentioned in section 4. The data is splitted into train/dev/test set. The data distribution is revealed in figure 4. The prerequisite pairs preq(+) are oversampled 1.5 times and this is executed only in the training dataset; meanwhile, the preq(-) pairs are negatively sampled such that the number of preq(-) is equal to that of preq(+).

Dataset Fewshot The dataset base is used for few-shot learning models, as mentioned in Section 4. The data is split into train, test, and sample sets. The number of samples for each time the model is trained is variable, ranging from 5 samples, 10 samples, etc. The data distribution is revealed in Figure 3. There are 100 prerequisite pairs (preq(+)) in the sample set and 302 pairs in the test set, matching the number in the DataBase dataset; meanwhile, the non-prerequisite pairs (preq(-)) are negatively sampled such that the number of preq(-) is equal to that of preq(+).

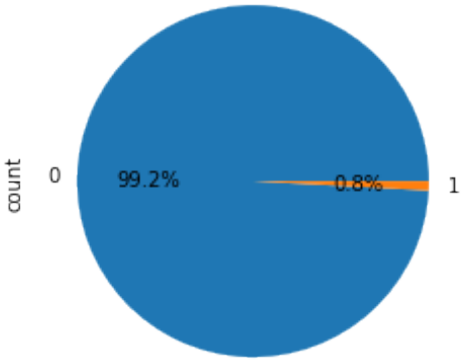

Figure 2: The Distribution of Labels for Concept Pairs

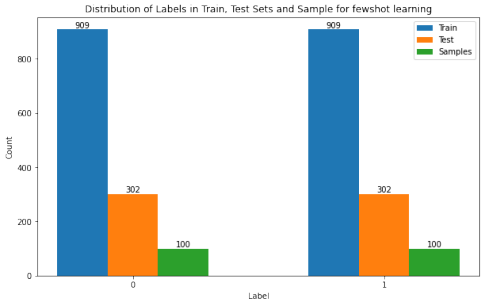

Figure 3: Fewshot learning dataset

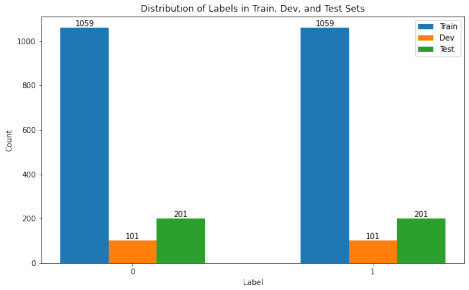

Figure 4: Finetuning + Prompting dataset

4 Approaches

In this section, we outline the Prompting design employed to address the challenges in predicting learning concept prerequisites. We detail the prompting framework, namely the pretrained-prompt-predict approach, and compare the performance of shallow tuning large language models (LLMs) with more radical approaches using smaller pretrained models that we have adapted to handle the classification problem.

Design Considerations for Prompting The prompt design, along with the chosen pretrained models, demonstrates how input can be inserted into the model. The prompt template is shown in the table 1. Samples are provided in the form of a prefix and a verbalizer for the classification problem.

- Pre-trained Model Choice: GPT-3.5 (Brown et al., 2020), LLAMA2-7b-chat (Touvron et al., 2023), T5(Raffel et al., 2023), GPT-2(Radford et al., 2019).

Table 1: Terminology and notation of prompting methods for predicting prerequisite dependencies.

Name	Notation	Example	Description
Input	X	'Fast Fourier transform', 'Signal processing'	Two concepts used as input for the model.
Output	y	'Yes'	Output label indicating the presence (1) or absence (0) of a prerequisite relationship.
Prompt function	$f_{prompt}(x)$	'Is the concept learning {Fast Fourier transform} a prerequisite of the {Signal processing} concept? Response: $[Z]'$	A function that converts the input into a specific form by inserting the input x and adding a slot $[Z]$ where answer z may be filled later
Prompt	x'	'Is the concept learning {Fast Fourier transform} a prerequisite of the {Signal processing} concept? Response: $[Z]$'	Prompt instantiated by input, where $[Z]$ is a placeholder for the model's response.
Filled Prompt	$f_{fill}(x', z)$	'Is the concept learning {Fast Fourier transform} a prerequisite of the {Signal processing} concept? Response: no'	A prompt where the slot $[Z]$ is filled with any arbitrary answer.
Answered Prompt	$f_{fill}(x', z^*)$	'Is the concept learning {Fast Fourier transform} a prerequisite of the {Signal processing} concept? Response: yes'	A prompt where the slot $[Z]$ is filled with the correct answer based on the model's prediction.
Answer	Z	'Yes'	The actual answer that fills the placeholder [Z], representing the model's response.

- Prompt Engineering: Choosing a proper prompt has a large effect not only on the accuracy, but also on which task the model performs in the first place. To handle this specific task, $fprompt(x)$ is a cloze prompts (Petroni et al., 2019),(Cui et al., 2021), which fill in the blanks of a textual string.

- Answer Engineering: Z is designed to be yes, no answer, corresponding to 1, 0 label of each pair

- Expanding the Paradigm: As stated above, the above equations represent only the simplest of the various underlying frameworks that have been proposed to do this variety of prompting. In §6 we discuss ways to expand this underlying paradigm to further improve results or applicability.

- Prompt-based Training Strategies: For Llama 2 and GPT-3.5, few shot tuning is executed. The number of sample given to prompt model is increasing from 5 samples to 100 samples. For the Pre-trained model of T5 and GPT-2, we fine-tune and train parameter the LMs as well of the promt.

4.1 Pretrained-prompt-predict with LMs

The prompt model, integrating fine-tuning with T5 and GPT-2 architectures, employs a learning rate of 0.1 and undergoes training for a variable number of epochs ranging from 5 to 10. With a maximum sequence length of 60 tokens and Adam optimizer utilized, the model is prompted with the template "Is concept 1 a prerequisite of concept 2?" This concise methodology enables efficient exploration of relational dependencies between concepts, facilitating nuanced understanding and inference within a constrained computational framework.

4.2 Few-Shot Tuning of LLMs

In the context of few-shot learning, the integration of the Llama2 algorithm with the GPT-3.5 pre-trained model offers a robust methodology for training models with limited examples per class. This approach involves prompting the model with varying quantities of samples, while configuring

the temperature parameter to 0.2 to enhance precision in binary classification tasks. Additionally, the maximum token length is constrained to 2000 tokens to optimize computational efficiency. Samples and queries are consistently structured, typically presenting a relational context between two concepts, such as a prerequisite association between concept 1 and concept 2. This methodology exhibits promising potential in addressing scenarios where access to abundant labeled data is limited, thereby advancing the field of machine learning by enabling effective knowledge transfer and generalization.

5 Result & Discussion

In this section, the results will be analyzed, comparing two strategies for leveraging large language models: the shallow approaches of few-shot learning versus the deeper approaches of exploiting language models through fine-tuning and prompting techniques. Additionally, the author will analyze the confusion matrix to determine the types of concept pairs the models incorrectly predicted, identifying whether errors were predominantly false negatives or false positives. Lastly, an error analysis will be presented, highlighting the learning concept pairs that were incorrectly predicted across different versions of the models. Through this analysis, we can discern whether the models exhibit diversity in their predictive decisions or if they demonstrate similar predictive patterns.

5.1 Model's Performance

Overall, it is shown that while larger models like GPT-3.5 Brown et al. (2020) and Llama 2 (Touvron et al., 2023) demonstrate promising scalability and performance, they also present challenges such as hallucination which leads to the low performance in the task required the precise answer such as classification task. Smaller pre-trained models including T5 (Raffel et al., 2023) and GPT-2 (Radford et al., 2019) highlight the importance of exploiting the smaller model by fine-tuning pre-trained and adapting prompting technique, resulting a better model performance. The observed trends and performance metrics compared to previous works underscore advancements in adapting pre-trained models and prompting techniques, guiding the choice of the most suitable model based on specific problem characteristics, available data, and computational resources.

Table 2 presents the F_1 scores of different models on a classification problem. The methods are grouped into Large Language Models (LLMs), Language Models (LMs), comparing them to the introduced models. Compared to previous works, prompting pre-trained models tends to result superior performance.

The few-shot prompting method on pre-trained LLMs including GPT-3.5 and Llamma 2 generally show good scalability with the increase in data, particularly GPT-3.5 which peaks at 50 samples. GPT-3.5 pre-trained model (Brown et al., 2020) shows an improving performance in F_1 score as the number of few-shot prompting samples increases from 5 to 100, with F_1 scores ranging from 0.752 to 0.805 before a slight drop to 0.778 at 100 samples. This suggests that pre-trained GPT-3.5 prompting model generally benefits from more examples, up to a certain point, indicating good scalability with data size but possibly a plateau or over-fitting beyond a certain number of samples. On the other hand, Llama 2 (Touvron et al., 2023) pre-trained model shows a significant increase in performance as the number of samples increases from 0 to 5, with F1 scores jumping from 0.590 to 0.624. However, in overall, model using Llama 2 yields a much lower performance compared to GPT-3.5 models. By prompting the *LLama2-7b-chat*, the model faced a severe problem with hallucination. In fact, 121 predictions in the test data out of 604 pairs was predicted with non sense answers, i.e. the answers are different to yes, no or 0,1.

The smaller pre-trained LMs, T5 and GPT-2, demonstrate that training duration (epochs) has a significant impact on performance, with T5 showing gradual improvement and GPT-2 showing potential signs of over-fitting past a certain epoch. T5 model exhibits a consistent increase in F_1 score with more training epochs, moving from 0.822 at 5 epochs to 0.851 at 10 epochs. This suggests that T5 benefits from longer training times, indicating robust learning capabilities. GPT-2 model show higher F_1 scores compared to T5, with the best score at 5 epochs (0.875), which then slightly decreases at 10 epochs to 0.827. This indicates that while GPT-2 starts strong, it may start overfitting or losing generalization beyond a certain point.

5.2 Confusion matrix

GPT-3.5 The figure 6 shows the performance of GPT-3.5 across various sample sizes using confu-

Table 2: F1 score comparison of LLMs, LMs models and introduced models on predicting prerequisite dependencies

LLMs	**GPT-3.5**					**Llama 2**	
# samples	*5*	*10*	*20*	*50*	*100*	*0*	*5*
F_1	0.752	0.757	0.765	**0.805**	0.778	0.590	0.624
LMs	**T5**	**T5**	**GPT-2**	**GPT-2**	-	-	-
# epoch	*5*	*10*	*5*	*10*	-	-	-
F_1	0.822	0.851	**0.875**	0.827	-	-	-
Baseline models	**PREREQ**	**CPRL**	**PPR for unseen concepts**	**ConLearn**	-	-	-
F_1	0.599	0.723	**0.842**	0.749	-	-	-

sion matrices. Concerning smaller Sample Sizes (5, 10, 20 samples), the improvement in model performance with increasing sample sizes suggest better positive case identification. For the larger sample sizes (50, 100 samples), the model results optimal performance at 50 samples, with a slight decrease at 100 samples, indicating potential challenges in maintaining consistency.

Llama-2 The figure 7 compare the performance of a model Llama-2 under two different settings: zero-shot and few-shot learning. In the zero-shot learning scenario, the model demonstrates a high number of True Negatives (TN) with 283 instances correctly identified as class 0, but struggles significantly with False Negatives (FN), incorrectly classifying 205 instances of class 1 as class 0. This indicates a strong bias towards predicting class 0 over class 1, evidenced further by the relatively lower number of True Positives (TP) at 97, showing it has difficulty correctly identifying class 1 instances without prior examples. Conversely, in the few-shot learning scenario, where the model is provided with a small number of examples (five shots), there's a noticeable improvement in balancing its predictions. The TN count decreases slightly to 264, suggesting a minor reduction in its ability to identify class 0 instances correctly, but more importantly, there's a significant improvement in TP, increasing to 125. This indicates that with just a few examples, the model becomes better at correctly identifying class 1 instances. Overall, the shift from zero-shot to few-shot learning demonstrates the model's improved ability to correctly identify class 1 instances with the introduction of a few training examples, albeit at the cost of a slight increase in FP. This illustrates the benefit of few-shot learning in enhancing model sensitivity towards less represented or more challenging classes.

T5 and GPT-2 The figure 8 reveals significant insights into their performance dynamics. The analysis of True Positives (TP) and True Negatives (TN) across the GPT-2 and T5 models reveals interesting trends as epochs increase from 5 to 10. An increase in TP for the T5 model from 79 to 85 and for GPT-2 from 70 to 81 indicates an enhanced capability in correctly identifying positive cases. Similarly, the TN values show slight fluctuations, suggesting nuanced impacts on the models' accuracy over time in identifying negative classes. Besides, the confusion matrix shows an ideal balance between sensitivity (recall) and specificity is pivotal for model optimization. The observed metrics from the confusion matrices highlight the inherent trade-off between these two aspects. As epochs increase, both models demonstrate a shift in this balance, emphasizing the challenges in maximizing TP and TN while minimizing FP and FN.

5.3 Error Analysis

Through error analysis, we can determine which type of pre-trained model yields more robust predictions. Tables 5 and 6 illustrate the pairs of learning concepts that were incorrectly predicted by models using LLM (Large Language Model) pre-trained models and traditional LM (Language Model) pre-trained models, respectively. The T5 and GPT-2 models made only three pairs of incorrect predictions across four models, whereas few-shot methods employing pre-trained LLMs exhibited a wider range of incorrect predictions, with 13 pairs of learning concepts. The fact that all four models referenced in Table 6 made common mistakes in only three instances indicates that they exhibit distinct prediction patterns, implying that these models are less robust in terms of prediction power.

6 Conclusion

In conclusion, this project has successfully demonstrated the effectiveness of adapting pre-trained models for the task of identifying prerequisite dependencies in educational texts. By moving away from conventional network science approaches, this study utilized the advanced capabilities of Large Language Models (LLMs) such as GPT-3.5 and Llama 2, while also exploring the potential of smaller pre-trained models like T5 and GPT-2. The findings reveal that smaller models, when fine-tuned and applied with precise prompting strategies, not only outperform their larger counterparts but also exceed the predictive capabilities of previous methods. This revelation highlights the substantial benefits of employing more compact pre-trained models, which offer comparable or superior performance with less computational demand. The project's outcomes invite further reflection on the optimal balance between model size and effectiveness, suggesting a promising direction for future research in automatic prerequisite extraction and educational data analysis.

7 Future Work & Limitation

In future work, we aim to develop a comprehensive framework for the automatic extraction of learning concepts and the establishment of a prerequisite network for these concepts. We plan to enhance prediction performance by delving deeper into fine-tuning large language models (LLMs) and exploring a variety of pre-trained language models to better adapt to educational data. Additionally, the retrieval-augmented generation (RAG) method holds promise as a predictive model by leveraging knowledge from external databases or directly from educational texts themselves. However, despite the project releasing an extensive collection of educational data, there remains a notable gap in the availability of textbook data specifically tailored for extracting learning concepts and identifying prerequisite relationships. Therefore, we intend to annotate a comprehensive dataset and evaluate our framework on this specific type of data, addressing the current limitations and expanding the applicability of our methods in educational settings.

Limitation This project has several limitations. First, the dataset is limited, which could lead to over-fitting. Second, the utilization of the LLaMA 2 model could be expanded to include more cases with varying sample sizes to enable a comprehensive comparison with GPT-3.5.

Resources

- Github Repository
- Dataset

References

Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017. SemEval 2017 task 10: ScienceIE - extracting keyphrases and relations from scientific publications. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 546–555, Vancouver, Canada. Association for Computational Linguistics.

Angels Balaguer, Vinamra Benara, Renato Luiz de Freitas Cunha, Roberto de M. Estevão Filho, Todd Hendry, Daniel Holstein, Jennifer Marsman, Nick Mecklenburg, Sara Malvar, Leonardo O. Nunes, Rafael Padilha, Morris Sharp, Bruno Silva, Swati Sharma, Vijay Aski, and Ranveer Chandra. 2024. Rag vs fine-tuning: Pipelines, tradeoffs, and a case study on agriculture.

Jan A. Botha, Zifei Shan, and Daniel Gillick. 2020. Entity Linking in 100 Languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7833–7845, Online. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *CoRR*, abs/2005.14165.

Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021. Template-based named entity recognition using BART. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1835–1845, Online. Association for Computational Linguistics.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Highly parallel autoregressive entity linking with discriminative correction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7662–7669, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey.

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864.

M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. 1998. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 782–792, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, EMNLP '03, page 216–223, USA. Association for Computational Linguistics.

Chenghao Jia, Yongliang Shen, Yechun Tang, Lu Sun, and Weiming Lu. 2021. Heterogeneous graph neural networks for concept prerequisite relation learning in educational data. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2036–2047, Online. Association for Computational Linguistics.

Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. 2010. SemEval-2010 task 5 : Automatic keyphrase extraction from scientific articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 21–26, Uppsala, Sweden. Association for Computational Linguistics.

Aobo Kong, Shiwan Zhao, Hao Chen, Qicheng Li, Yong Qin, Ruiqi Sun, and Xiaoyan Bai. 2023. PromptRank: Unsupervised keyphrase extraction using prompt. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9788–9801, Toronto, Canada. Association for Computational Linguistics.

Mayank Kulkarni, Debanjan Mahata, Ravneet Arora, and Rajarshi Bhowmik. 2022. Learning rich representation of keyphrases from text. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 891–906, Seattle, United States. Association for Computational Linguistics.

Irene Li, Alexander R Fabbri, Robert R Tung, and Dragomir R Radev. 2019. What should i learn first: Introducing lecturebank for nlp education and prerequisite chain learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6674–6681.

Yujia Li, Richard Zemel, Marc Brockschmidt, and Daniel Tarlow. 2016. Gated graph sequence neural networks. In *Proceedings of ICLR'16*.

Andy Liaw and Matthew Wiener. 2002. Classification and regression by randomforest. *R News*, 2(3):18–22.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55:1 – 35.

Narjes Nikzad-Khasmakhi, Mohammadali Balafar, M. Reza Feizi-Derakhshi, and Cina Motamed. 2021. Exem: Expert embedding using dominating set theory with deep learning approaches. *Expert Syst. Appl.*, 177(C).

Narjes Nikzad-Khasmakhi, Mohammad-Reza Feizi-Derakhshi, Meysam Asgari-Chenaghlu, Mohammad Ali Balafar, Ali-Reza Feizi-Derakhshi, Taymaz Rahkar-Farshi, Majid Ramezani, Zoleikha Jahanbakhsh-Nagadeh, Elnaz Zafarani-Moattar, and Mehrdad Ranjbar-Khadivi. 2021. Phraseformer: Multimodal key-phrase extraction using transformer and graph embedding. *CoRR*, abs/2106.04939.

Liangming Pan, Chengjiang Li, Juanzi Li, and Jie Tang. 2017. Prerequisite relation learning for concepts in MOOCs. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1447–1456, Vancouver, Canada. Association for Computational Linguistics.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, page 701–710, New York, NY, USA. Association for Computing Machinery.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer.

Sudeshna Roy, Meghana Madhyastha, Sheril Lawrence, and Vaibhav Rajan. 2019. Inferring concept prerequisite relations from online educational resources. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):9589–9594.

Hao Sun, Yuntao Li, and Yan Zhang. 2022. Conlearn: Contextual-knowledge-aware concept prerequisite relation learning with graph neural network. In *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*, pages 118–126.

F. Thazhathukunnel. 2021. Entity Linking on Textbooks. Master's thesis, ETH Zürich.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Jifan Yu, Gan Luo, Tong Xiao, Qingyang Zhong, Yuquan Wang, Wenzheng Feng, Junyi Luo, Chenyu Wang, Lei Hou, Juanzi Li, Zhiyuan Liu, and Jie Tang. 2020. MOOCCube: A large-scale data repository for NLP applications in MOOCs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3135–3142, Online. Association for Computational Linguistics.

Wenzheng Zhang, Wenyue Hua, and Karl Stratos. 2021. Entqa: Entity linking as question answering. *ArXiv*, abs/2110.02369.

Appendix A: Examples of Dataset

Tokens	'Calculus'	'is'	'a'	'requirement'	'to'	'study'	'advanced'	'physics'
BIO-tag	'B'	'O'	'O'	'O'	'O'	'O'	'B'	'I'

Table 3: An example of the (expected) data for task 1.

Learning concept 1	Learning concept 2	Prerequisite relation
'calculus'	'advanced physics'	1
'calculus'	'history'	0

Table 4: An example of the (expected) data for task 2.

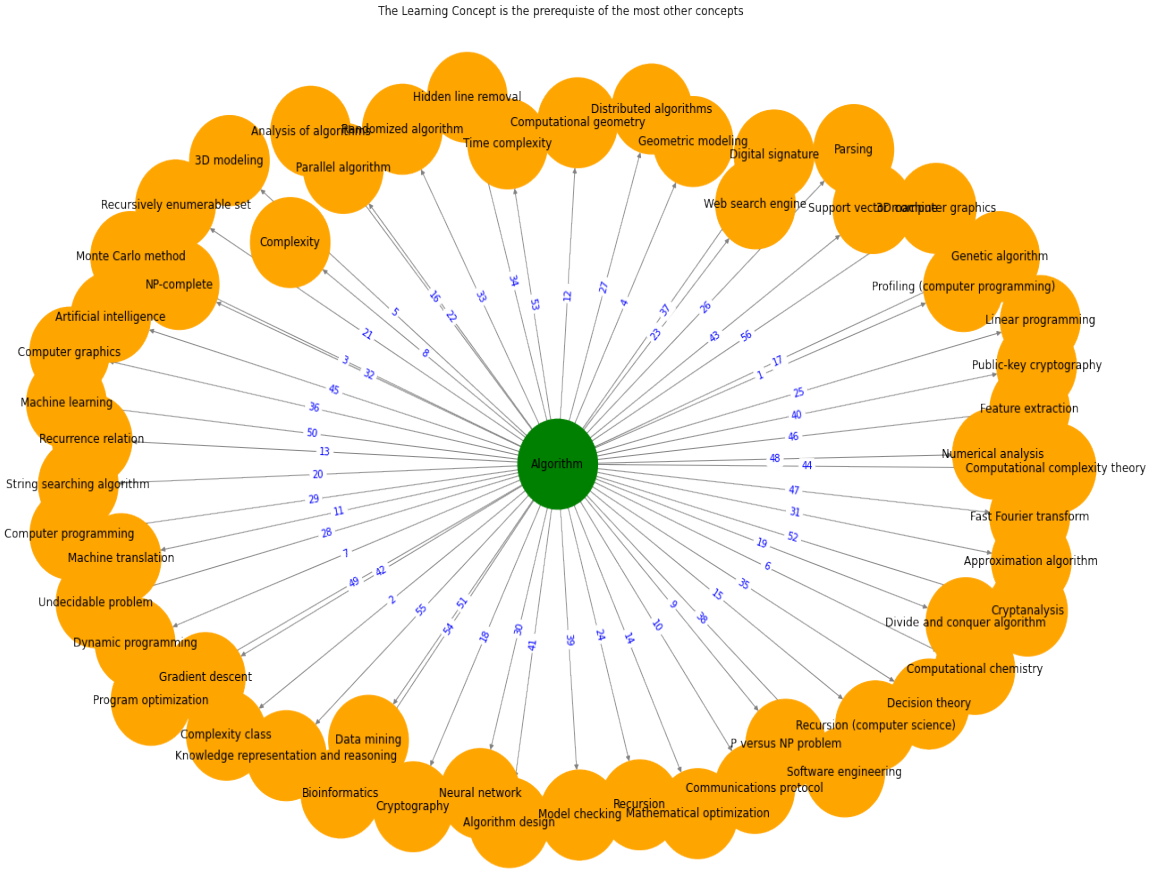

Figure 5: ’Algorithm’ is a prerequisite concept of multiple other learning concepts

Appendix B: Confusion matrix

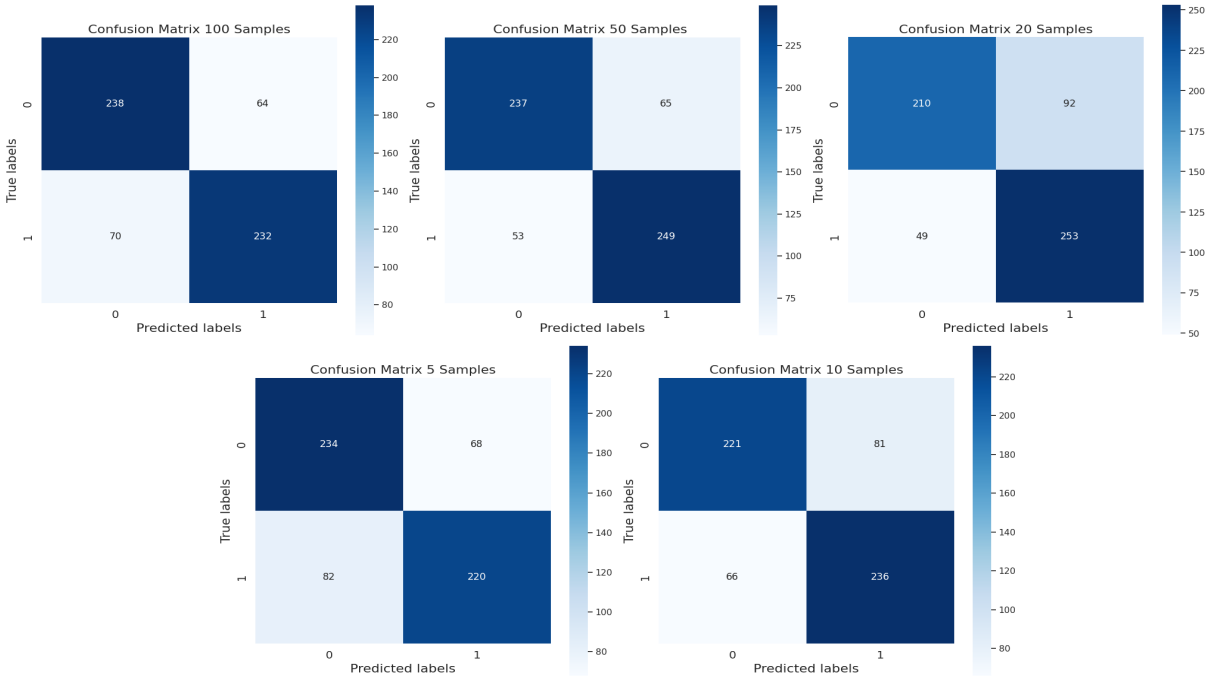

Figure 6: Confusion matrix GPT-3.5

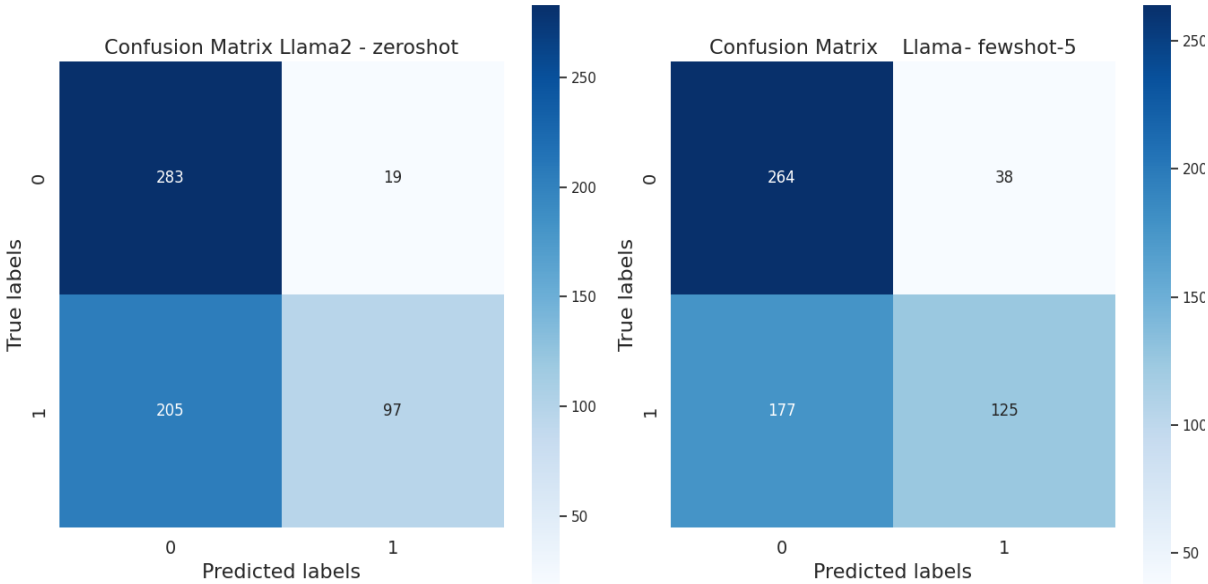

Figure 7: Confusion matrix Llama 2

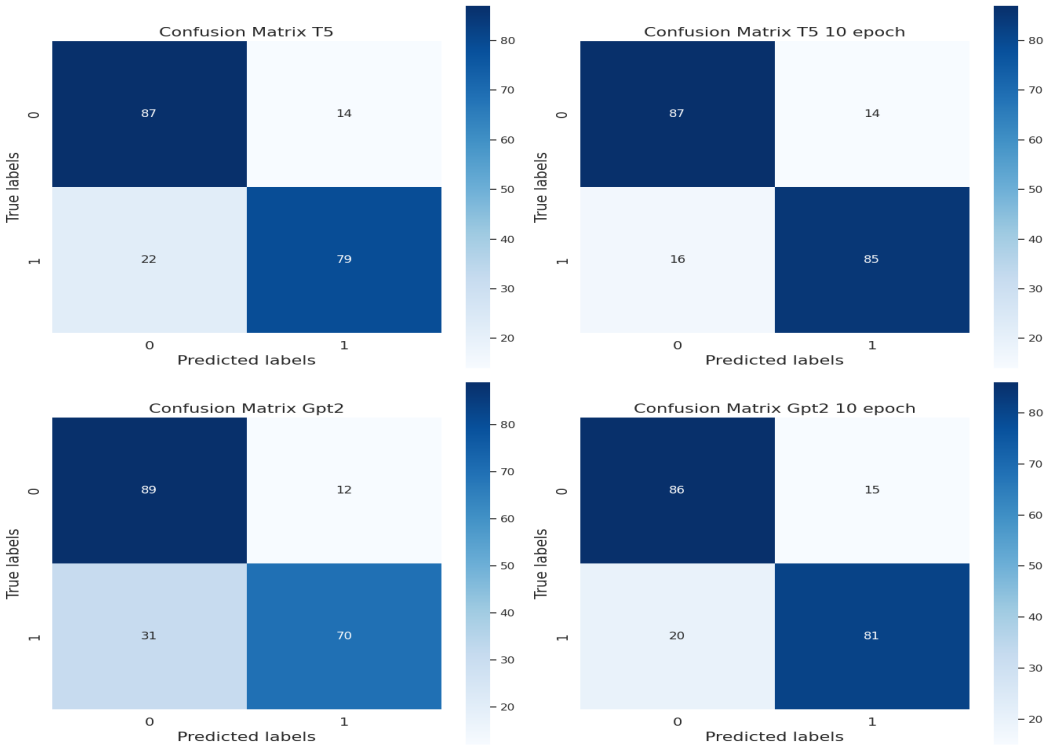

Figure 8: Confusion matrix T5 and GP-T2

Appendix C: Error analysis

Table 5: Error analysis GPT-3.5 and Llama 2

Concept1	**Concept2**
Robotics	Analysis of algorithms
Computer	Arithmetic
Computer graphics	User interface design
Natural language processing	Graph theory
Algorithm	Polynomial
Document classification	Computer science
Algorithm	Integer
Computer graphics	Personal computer
Quantum mechanics	Continuous function
Robotics	Computer programming
Computer programming	Arithmetic
Machine learning	Graph theory

Table 6: Error analysis T5 and GPT-2

Concept1	**Concept2**
Numerical Analysis	Floating point
Stochastic process	Pattern recognition
Round-off error	Data analysis