A New Competency Tagging Method through Semantic Matching with Fine-tuned LLM

Anonymous ACL submission

Abstract

Competency tagging is essential in both academic and industrial domains, facilitating alignment of learning content, job posting and resumes with specific competencies. However, the manual tagging process is time-consuming, labor-intensive, and expensive. In this study, we propose semantic matching-based method for automated competency tagging. Particularly, we explore the potential of large language models (LLMs) to encode text data from learning content and competency descriptions. Subsequently, we employ similarity search to retrieve the most pertinent competency tags corresponding to a given learning content document. We investigated semantic search at different levels of granularity: per document, per paragraph, and per sentence. We further fine-tuned the LLM using the Low-Rank Adaptation (LoRA) technique. Our method yielded promising results, achieving a recall@10 of 80.29% when tested on 164 pages of learning content associated with 96 competencies. These findings highlight the effectiveness of fine-tuned LLMs, which enhanced recall@10 by 5%.

1 Introduction

Competency tagging plays a significant role in both academia and industry. In education, skill tagging enhances educational programs and curricula, ensuring their alignment with the evolving demands of the job market (Holmboe et al., 2010; Roegiers, 2016). In the job market, competency tagging facilitates matching job seekers with relevant opportunities and analyzing market trends. This bridging of opportunities for individuals and optimization of resource allocation for employers helps to ameliorate imbalances between supply and demand within the job market (Danielle et al., 2020).

However, manual tagging is impractical given the vast amount of available data. It is time-consuming, labor-intensive, and costly, as it requires qualified experts. Non-experts are generally unable to accurately identify skills (Moore et al., 2022; Ren et al., 2024). Moreover, even among experts, consistency can be challenging; for instance, two expert teachers who identified knowledge components of a state-wide math test only agreed on 35% of the items (Patikorn et al., 2019).

Therefore, implementing an automated method has the potential to substantially decrease both time and costs; however, challenges persist regarding the accuracy. Numerous approaches have been introduced to address competency or skill tagging task, leveraging both traditional machine learning algorithms (Desmarais, 2012; Zhao et al., 2015) and more recent neural network models (Patikorn et al., 2019; Shen et al., 2021a). Nevertheless, leveraging advanced LLMs remains largely unexplored and presents an intriguing avenue for investigation. Furthermore, more attention should be directed towards semantic search, particularly regarding levels of granularity.

In this paper, we present a semantic matching-based method for competency tagging of learning content. Our method leverages pre-trained LLM (Vaswani et al., 2017) to encode both learning content and competency descriptions, subsequently employing similarity search to retrieve the most relevant competencies corresponding to the content. We investigate various levels of semantic search granularity, including sentence-level, paragraph-level, and document-level. Additionally, we explore the potential of parameterized-efficient finetinning the LLM using (LORA) (Hu et al., 2021) with custom data. To outline, our study addresses the following research questions (RQs): RQ1 : *Does semantic matching using pre-trained LLM for competency tagging prove to be efficient?* RQ2 : *Which level of granularity in semantic matching yields the most effective results?* RQ3 : *Does fine-tuning the LLM enhance competency-tagging performance?* The rest of this paper is arranged as follows: Section 2 provides an overview of the related works on semantic matching; Section 3 describes the proposed method; Section 4 illustrates employed dataset and implementations; Section 5 presents the findings; finally, Section 6 provides concluding remarks.

2 Prior work

Numerous methods have been introduced to address competency tagging task. Early studies employed machine learning algorithms to tag competencies within educational materials. (Desmarais, 2012) suggested mapping question items to skills using Non-negative Matrix Factorization with simulated data. (Karlovčec et al., 2012) proposed a method for knowledge component suggestion for untagged content in an intelligent tutoring system, utilizing text mining and SVM classification which demonstrated promising performance using data from the ASSISTments platform. In a more recent study by (Zhao et al., 2015), Word2Vec algorithm was used to encode data for tagging of skills from a comprehensive taxonomy comprising 50,000 skills. Using a random sampling-based end-user evaluation, the system tagged resumes submitted by job applicants and provided the top 10 skills identified. With a substantial dataset comprising 3,000 responses from users, the current system demonstrated a commendable recall rate of 70%.

Within research on tagging educational learning material, (Pardos and Dadu, 2017) conducted a study on skill tagging from problem texts. The research focused on imputing knowledge components (KC) from untagged problem texts, utilizing the ASSISTments 2012 public dataset. Interestingly, the study compared the skip-gram based approach with the bag-of-words (BOW) method, revealing that the latter yielded superior results in skill prediction. In a similar vein, (Patikorn et al., 2019) conducted a study on skill tagging utilizing 65,120 problems sourced from 336 problem sets, encompassing 173 distinct skill standards. Patikorn et al. employed decision trees, neural networks (NN), and random forest algorithms for skill classification. While neural networks demonstrated promising results, the evaluation on new dataset for testing purposes revealed a notable drop in accuracy, suggesting limitations in generalizability. Despite the performance of all models surpassing chance levels, their utility in real-world applications remains questionable.

(Shen et al., 2021b,a) applied multinomial classification techniques using finetuned BERT models. They initially trained BERT using unlabeled data encompassing various sources. Then, employed the Task-adaptive Pretrained (TAPT) BERT model to finedtuned the model with labeled data extracted from description texts, video titles, and problem texts. In their evaluation, exact matching was replaced by semantic or structural similarity assessments. The researchers used 385 math knowledge components spanning from kindergarten to 12th grade. While the multinomial classification approach yielded promising results, its implementation necessitated a considerable corpus of annotated text problems. Moreover, concerns were raised regarding its generalizability, particularly in scenarios where new data deviates substantially from the training dataset. A recent study (Li et al., 2024) focused on aligning open educational resources with new taxonomies, using various modalities including videos (encoded with U3D), images (processed with EfficientNet-B7), and text (utilizing SentenceBERT). Employing both classification and similarity matching techniques, on datasets comprising 21,475 problems from Khan Academy and 19,996 problems from CK12, and utilizing taxonomies such as Common Core skills, Khan Academy, and CK12. Results indicated that while the classification model exhibited superior performance when using the Common Core taxonomy, similarity matching was more effective with other taxonomies. While the studies explored the similarity matching approach for competency tagging, they have limitation of using the pre-trained SentenceBERT model which tend to lack specificity. Our proposed method offers the advantage of leveraging a more powerful Large Language Model (LLM) and fine-tuning it to enrich its knowledge base, consequently enhancing its performance. By fine-tuning the LLM, we can tailor it to the specific requirements of our task, enabling it to capture intricate nuances and patterns within the data.

3 Methods

This section outlines the design and implementation of our method. Figure 1 shows the overall diagram. Our approach consists of two main components: an offline block and an online competency tagging process. In the offline block, competencies descriptions are embedded using a Large Language Model (LLM) to encode text data into dense vectors. These vectors are subsequently indexed using

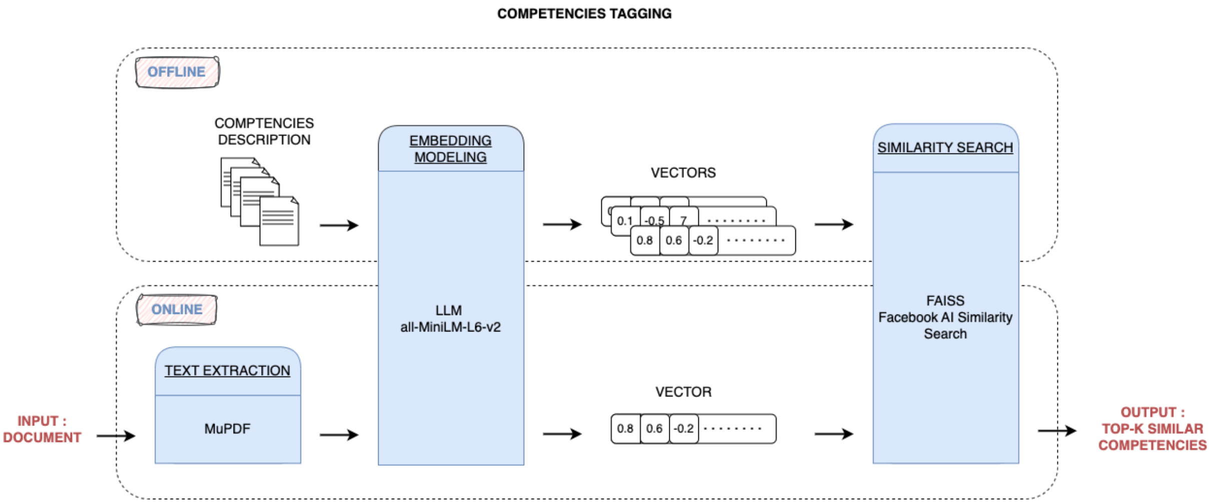

Figure 1: A block diagram of the proposed method for competency tagging of learning content.

facebook AI similarity search (FAISS) library. In the online competency tagging process, when a document is inputted, its text is extracted and transformed into vectorized form using the same LLM. The system then performs a search to identify the closest vectors based on distance, outputting the top-k relevant competencies.

3.1 Embedding modeling

For encoding sentences and paragraphs into dense vectors, we employ the Large language model all-miniLM-L6-v2 variant from the huggingface Transformer library (**?**). The model is designed to convert long textual inputs into a 384-dimensional embedding space, facilitating efficient similarity calculations. The training data for this model includes a diverse collection of datasets, such as Reddit comments, S2ORC citation pairs, WikiAnswers, PAQ, MS MARCO, GOOAQ, Yahoo Answers, Code Search, COCO, SPECTER, and more, amounting to over one billion tuples. The model parameters include 22.7 million parameters with a maximum token limit of 128 per input.

3.2 Similarity Search with FAISS

To perform similarity searches among the dense vectors, we utilize Facebook AI Similarity Search (FAISS). FAISS is an efficient library for searching similar vectors within large datasets. It constructs compressed indexes using techniques like dimensionality reduction and quantization, allowing rapid nearest-neighbor searches based on various distance metrics, such as Euclidean distance and cosine similarity. In this study, we use cosine similarity, calculated as:

$$\cos\theta = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} \tag{1}$$

where $\mathbf{A}\cdot\mathbf{B}$ *is the dot product of vectors* $\mathbf{A}$ *and* $\mathbf{B}$*, and* $\|\mathbf{A}\|$ *and* $\|\mathbf{B}\|$ *are their respective magnitudes.*

We selected FAISS for its capability to efficiently retrieve vectors that closely match a specified query vector, thus avoiding the need for brute-force calculation and comparison of similarity scores.

3.3 Fine-Tuning with LoRA

In this study, textual data encoding relies on leveraging a Large Language Model (LLM) to effectively capture and process linguistic nuances. To enhance the model's performance, LoRa fine-tuning technique was employed, allowing for optimized adaptation to domain-specific datasets. We adopt Parameter-Efficient Fine-Tuning (PEFT) techniques, specifically Low-Rank Adaptation (LoRA). PEFT methods are designed to overcome the challenges of training large language models (LLMs) on low-resource hardware by fine-tuning only a subset of the model's parameters while keeping the majority frozen. This method not only reduces computational and storage costs but also enhances performance in low-data scenarios and improves generalization to out-of-domain data.

Indeed, LoRA involve a low-rank decomposition into the weight matrix W_0 of the pre-trained model. Instead of directly optimizing all parameters, LoRA approximates the update ΔW with a low-rank representation:

$$\Delta W \approx \alpha \cdot A \cdot B^T \tag{2}$$

	Documents	Pages	sentences
Total	35	164	1968
Mean number of words	409.00	41.39	7.72

Table 1
Overview of the learning content dataset used for competency tagging.

Where A is a matrix of size $m \times r$, B is a matrix of size $n \times r$, $r \ll \min(m, n)$ represents the rank of the decomposition, and α is a scalar scaling factor. This results in significantly fewer trainable parameters, $r(m + n) + 1$, compared to the full parameter set mn. During inference, the original weight matrix W_0 is updated as follows:

$$W = W_0 + \alpha \cdot A \cdot B^T \quad (3)$$

In conclusion, LoRA allows efficient fine-tuning while maintaining the integrity of the original pre-trained weights. The small number of newly added trainable parameters makes the training process faster and more memory-efficient, yielding much smaller model weights, typically a few hundred megabytes.

By integrating these methods, our method ensures efficient and scalable skill tagging, leveraging advanced state-of-the-art techniques in text encoding, similarity search, and model fine-tuning.

4 Experiment Setup

4.1 Datasets

Comeptency tagging dataset We evaluate our method on a private dataset provided by a company. This dataset contains 35 course materials in PDF format in PDF format, created and manually annotated by experts using 96 competencies. These annotations involved 96 competencies and served as the ground truth for assessing and enhancing the performance of our approach. The dataset statistics are summarized in table 1 The competencies are specific to Project Manager job, categorized into 14 domains. Each competency entry includes a unique reference code, a name, a detailed definition, and relevant keywords. For instance, the competency with the reference code "DETDEVA" is named "Determine strategic approach to deliver the project." and it is defined as "determining the appropriate development approach and life cycle, such as predictive, adaptive, or hybrid, to deliver value from start to finish". Keywords associated with this competency include "Agile," "scrum," "iterative," and "waterfall."

Fine-tuning data To fine-tune the large language model (LLM), we developed a custom dataset comprising sentence pairs labeled based on their competency components—name, statement, and definition. This dataset includes two subsets containing 2,500 and 3,500 sentence pairs, respectively. Each pair was labeled as similar or different, facilitating binary classification. Pairs deemed similar were assigned a label of 1, while dissimilar pairs were labeled as 0. For example, a similar pair would be represented as (comp_name, comp_def, 1), whereas a different pair would be represented as (compX_name, compY_name, 0).

4.2 Evaluation metrics

To assess the performance of our approach in competency tagging within learning content, we utilized the following evaluation metrics: the $Recall@k$ and $MAP@k$.

$Recall@k$ is a metric used to evaluate the effectiveness of an information matching system by measuring the proportion of relevant items retrieved in the top k results. It is defined as the number of relevant items in the top k results divided by the total number of relevant items in the dataset. Recall@k can be expressed as:

$$\text{Recall@k} = \frac{\text{Number of relevant competencies retrieved in } k \text{ competencies}}{\text{Total number of relevant competencies}} \quad (4)$$

$MAP@k$ (Mean Average Precision) is a metric used to evaluate the precision of an information matching system. It measures the average precision of the relevant competencies at each rank position up to k, providing a single numerical value that summarizes the quality of the ranking. It is defined as:

$$\text{MAP@k} = \frac{1}{|Q|} \sum_{q=1}^{|Q|} \frac{1}{\min(m, k)} \sum_{i=1}^{k} P(i) \times \text{rel}(i) \quad (5)$$

where, Q represents the set of queries, m is the total number of relevant competencies for a query, k is the maximum number of competencies to consider, $P(i)$ is the precision at cutoff i, and rel(i) is an indicator function that equals 1 if the competency at rank i is relevant and 0 otherwise.

4.3 Implementation details

For text extraction from documents, we utilized the MuPDF library in Python. The Large Language Model (LLM) was imported from Huggingface (Wolf et al., 2019). The fine-tuning process involved various LoRA parameters, including rank values of 8, 16, and 32, with the scaling factor set to twice the rank (Alpha = 2 * R). Learning parameters included learning rates of 0.001 and 0.00001, batch sizes of 8 and 16, and training epochs set to 3, 5, and 10. We opted for a rank value of 8, a learning rate of 0.001, and trained for 10 epochs.

All training and experiments were conducted on Kaggle, utilizing two NVIDIA T4 GPUs to ensure efficient processing of the dataset and accelerate the fine-tuning process. All code for competency alignment and finetuning the LLM will be made available on GitHub for the purpose of reproducibility.

5 Results and Discussion

To address our research questions, we conducted a series of experiments.

1. First, we evaluated the performance of our approach using 164 annotated pages, where each page represents a unit of learned content and tagged competencies were recommended by our system. The results were compared against expert annotations to evaluate the performance. This evaluation aimed to determine the efficiency of semantic matching using pre-trained models for competencies tagging(RQ1).

2. The second part of our study focused on exploring competency tagging across multiple levels of granularity: document-level, paragraph-level (page-level), and sentence-level. This phase aimed to tag competencies within documents using different units of analysis. Specifically, the experiments involved tagging competencies using the entire document text, paragraphs corresponding to pages, and individual sentences. Results from paragraphs and sentences were aggregated to recommend competencies for each document. This approach was intended to determine the optimal unit of analysis for accurate competency tagging (RQ2).

3. Finally, to enhance the model's competency-related understanding, we employed Low-Rank Adaptation (LoRA) to fine-tune a large language model. This involved utilizing two datasets, the first containing 2,500 and the second 3,500 pairs respectively, to refine the model's capability in understanding and predicting competencies. Subsequently, using the fine-tuned model, we re-evaluated our approach on the 164 annotated pages. Additionally, we assessed the competency tagging module across different levels of granularity: document-level, page-level, and sentence-level. This evaluation aimed to validate the effectiveness of fine-tuning a pre-trained LLM (RQ3).

5.1 RQ1: Does semantic matching using pre-trained models for competency prove to be efficient?

Evaluating our approach using 164 pages from different material learning documents yielded a recall@10 of 74.14% and MAP@10 of 47.21%. Examples illustrating the results are shown in Table 2.

The high recall rate demonstrates the effectiveness of our approach in assisting experts with data tagging. However, the approach wasn't able to tag all associated competencies, confirming the study by (Ren et al., 2024), which claims that AI helped save time but sacrificed accuracy. In their study, they found that AI saved almost 50% of the time compared to manual tagging but sacrificed 35% of accuracy.

One key challenge faced by our algorithm is the highly refined nature of competencies, which can complicate the accurate tagging of all relevant competencies. For instance, as shown in example 2 in Table 2 (line 2), the competencies recommended for a page from the module "Engage Stakeholders" included a broader range of competencies than those identified by the experts. While the expert-selected competencies were "Engage stakeholders" and "Monitor stakeholder engagement" the algorithm additionally recommended competencies such as "Analyze stakeholders", "Identify relevant stakeholders", "Detect stakeholders attitude", "Prioritize stakeholders" and others. Although these recommendations are closely related, they highlight the algorithm's difficulty in precisely identifying and prioritizing the correct competencies without expert intervention.

The relatively low MAP@10 value of 47.21% further suggests a significant presence of false pos-

Module title	Page text	Actual competencies	Recommended competencies
Preparing an effective presentation	What are the elements of a great presentation? list the key elements, based on your experience and what you have seen from others	"Present project performance information", "Provide quality information"	"Present project performance information", "Provide quality information", "Tailors communication to audience", "Facilitate open communication", "Influence others to gain support and commitment", "Demonstrate leadership", "Demonstrate empathy", "Encourage others to share", "Promote and sell project", "Gaining value from learning"
Engage stakeholders	Do you - manage proactively stakeholder expectations to ensure the project's objectives are achieved? - Engage stakeholders at appropriate stages to obtain or confirm their continued commitment to the success of the project? - seek out potential conflicts among stakeholders to detect new risks and issues? -clarify and respond to issues raised by stakeholders? - ensure stakeholders understand the project's goals, objectives, and risks throughout the life of the project? - monitor overall project stakeholder relationships and adjust engagement strategies and plans accordingly? - review and update stakeholder management plan throughout the life of the project? - evaluate stakeholder level of engagement and confirm it's at appropriate level?	"Engage stakeholders", "Monitor stakeholder engagement"	"Engage stakeholders", "Monitor stakeholder engagement", "Analyze stakeholders", "Identify relevant stakeholders", "Detect stakeholders attitude", "Prioritize stakeholders", "Establish the strategic positioning of the project", "Build trust based relationships", "Demonstrate leadership", "Plan communications"
Team leadership	The abilene paradox : teams frequently take collective action contrary to the individual wishes of any of their members and therefore defeat the very purposes they set out to achieve	"Develop team", "Encourage others to share"	"Ensure successful teamwork", "Demonstrate leadership", "Develop team", "Determine team composition and structure", "Maintain project team focus", "Influence others to gain support and commitment", "Encourage others to share", "Lead change through people", "Develop others", "Adapting and responding to change"

Table 2: Examples of competency tagging for pages across different modules.

	Recall@10	MAP@10
Document-level	62.17%	34.17%
Paragraph-level	70.14%	48.09%
Sentence-level	56.00%	29.95%

Table 3: Competency tagging results at different levels of granularity.

	Recall@10	MAP@10
Pre-trained llm	74.14%	47.21%
Fine-tuned llm with 2500 data	75.82%	49.71%
Fine-tuned llm with 3500 data	**80.29%**	**52.48%**

Table 4: Competency tagging results with pre-trained and fine-tuned all-miniLM-L6-v2 at different levels of granularity: document-level, paragraph-level, and sentence-level.

itives. This implies that while the AI-based approach can substantially aid the tagging process, it may not be reliable enough for fully automated tagging where accuracy is paramount. Consequently, a more effective application of AI in this context might be as an assisted system that supports experts in the tagging process, rather than relying on a fully automated approach.

5.2 RQ2: Which granularity of semantic matching yields the most effective results?

The second step involved evaluating the performance of tagging competencies for individual learning material documents. Leveraging the advantage of expert-annotated competencies for each module, we assessed the tagging performance at three levels of granularity: document-level, paragraph-level (page-level), and sentence-level. The results of this performance evaluation are summarized in Table 3. Semantic matching at the paragraph level yields the most effective results, achieving a recall@10 of 70.14%. This level of granularity proves superior because it captures the essential idea of each paragraph. By focusing on paragraphs, the model can better understand and tag competencies accurately within each paragraph. These results are then aggregated to provide a competency tagging for the entire document.

Conversely, matching at the document level presents significant challenges due to token limitations and the potential for data truncation. When the input exceeds the model's maximum token limit, the model truncates the input to fit within this limit, leading to the loss of crucial context and information, as highlighted by Levy et al. (Levy et al., 2024).

Despite the fact that sentence-level semantic matching can identify over half of the competencies, it is less effective compared to paragraph-level matching, as it tends to focus more on granular details rather than capturing the general idea of the text. While it can be beneficial in identifying specific competencies, it often misses the broader context and overall themes that are crucial for accurate competency tagging. In contrast, paragraph-level matching provides a more comprehensive view, encapsulating the essential meaning of each paragraph. This allows for a more accurate aggregation of results, leading to a more thorough understanding of the competencies within a docuemnt.

The granularity level in semantic matching was investigated within the context of another related NLP task, namely Machine Reading Comprehension (MRC)(Liu et al., 2022). MRC aims to develop systems capable of reading text, understanding its meaning, and answering questions automatically. This investigation focused on how different levels of granularity, such as paragraph-level versus sentence-level matching, impact the performance of semantic matching in MRC tasks. Similar to our findings, liu and al. has shown that matching at a coarser granularity, such as paragraphs, tends to yield more effective results compared to finer-grained approaches like sentence-level matching. This finding underscores the importance of selecting an appropriate level of granularity in semantic tasks to enhance comprehension and accuracy in processing textual information for tasks like MRC.

RQ3: Does fine-tuning the model improve competency-tagging performance?

Our third concern in this study focused on exploring the potential of fine-tuning Large Language Models (LLMs) to enhance the performance of semantic matching in competency tagging. To investigate this, we conducted two experiments involving the fine-tuning of the LLM using datasets comprising 2500 and 3500 instances, respectively.

A comparison of the performance of our approach in competency tagging using the original LLM, the fine-tuned version with 2500 instances, and the fine-tuned version with 3500 instances is shown in Tables 4 and 5.

Fine-tuning ameliorates performance significantly, achieving a recall@10 of 80.29% in tagging

	pre-doc		per-paragraph		per-sentence	
	Recall@10	MAP@10	Recall@10	MAP@10	Recall@10	MAP@10
Pre-trained llm	62.17%	34.17%	70.14%	48.09%	56.00%	29.95%
Fine-tuned llm with 2500 data	69.77%	**39.32%**	69.06%	47.01%	57.06%	33.84%
Fine-tuned llm with 3500 data	**69.83%**	38.58%	**70.51%**	**48.70%**	**57.80%**	**31.45%**

Table 5: Competency tagging results with pre-trained and fine-tuned all-miniLM-L6-v2 at different levels of granularity: document-level, paragraph-level, and sentence-level.

competencies across 164 pages. Similarly, fine-tuning enhances competency tagging in 35 modules across all granularity levels (per-page, per-sentence, and per-document), as shown in Table 5.

The data used for fine-tuning enhances the model's ability to effectively distinguish between closely related competencies, thereby improving its overall performance in competency tagging tasks. While several hundred well-labeled data samples are claimed to suffice for fine-tuning (Zhou et al., 2024), our observations indicate that larger dataset sizes yield better results. Parameter-efficient fine-tuning methods, such as Low-Rank Adaptation (LoRA) (Liu et al., 2022), provide a viable alternative to full fine-tuning, achieving a notable 6% improvement with minimal data. Additionally, facilitates cost-effective and timely fine-tuning processes, making advanced model training more accessible.

6 Conclusion

In this study, we proposed a novel method using Large Language Models (LLMs) and semantic matching for competency tagging. Experimental results demonstrated the effectiveness of our method. Additionally, we examined the impact of semantic search granularity and discovered that paragraph-level granularity produced the best results, enabling a comprehensive understanding of both the overall document context and specific details. Moreover, we found that fine-tuning a pretrained LLM on approximately 3,000 carefully curated examples using LoRA can significantly improve performance.

Limitations

One significant limitation of this study is the difficulty in accurately evaluating the performance of our model. Without comprehensive and representative datasets, it becomes challenging to ascertain the true capabilities and limitations of our method. Moreover, not evaluating our method using publicly available datasets for competency tagging inhibits our ability to compare results with state-of-the-art methods, which will be the focus of future work. Despite employing fine-tuning techniques like LoRA (Low-Rank Adaptation) to enhance generalization, our model's ability to adapt to out-of-domain scenarios may still be restricted. This limitation becomes evident when the model encounters entirely new domains or tasks not covered in the training data. Therefore, the necessity for extensive and diverse datasets during the fine-tuning phase becomes paramount to improving the model's adaptability to novel contexts.

References

Saunders Danielle, Lerner Matt, Plagge Tom, Gee Matt, et al. 2020. Skill competency data translation and analysis. *NASWA Workforce Technology*.

Michel C Desmarais. 2012. Mapping question items to skills with non-negative matrix factorization. *ACM Sigkdd Explorations Newsletter*, 13(2):30–36.

Eric S Holmboe, Jonathan Sherbino, Donlin M Long, Susan R Swing, Jason R Frank, and International CBME Collaborators. 2010. The role of assessment in competency-based medical education. *Medical teacher*, 32(8):676–682.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Mario Karlovčec, Mariheida Córdova-Sánchez, and Zachary A Pardos. 2012. Knowledge component suggestion for untagged content in an intelligent tutoring system. In *Intelligent Tutoring Systems: 11th International Conference, ITS 2012, Chania, Crete, Greece, June 14-18, 2012. Proceedings 11*, pages 195–200. Springer.

Mosh Levy, Alon Jacoby, and Yoav Goldberg. 2024. Same task, more tokens: the impact of input length on the reasoning performance of large language models. *arXiv preprint arXiv:2402.14848*.

Zhi Li, Zachary A Pardos, and Cheng Ren. 2024. Aligning open educational resources to new taxonomies: How ai technologies can help and in which scenarios. *Computers & Education*, 216:105027.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965.

Steven Moore, Huy A Nguyen, and John Stamper. 2022. Leveraging students to generate skill tags that inform learning analytics. In *Proceedings of the 16th International Conference of the Learning Sciences-ICLS 2022, pp. 791-798*. International Society of the Learning Sciences.

Zachary A Pardos and Anant Dadu. 2017. Imputing kcs with representations of problem content and context. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, pages 148–155.

Thanaporn Patikorn, David Deisadze, Leo Grande, Ziyang Yu, and Neil Heffernan. 2019. Generalizability of methods for imputing mathematical skills needed to solve problems from texts. In *Artificial Intelligence in Education: 20th International Conference, AIED 2019, Chicago, IL, USA, June 25-29, 2019, Proceedings, Part I 20*, pages 396–405. Springer.

Cheng Ren, Zachary Pardos, and Zhi Li. 2024. Human-ai collaboration increases skill tagging speed but degrades accuracy. *arXiv preprint arXiv:2403.02259*.

Xavier Roegiers. 2016. A conceptual framework for competencies assessment. in-progress reflection no. 4 on" current and critical issues in the curriculum and learning". *UNESCO International Bureau of Education*.

Jia Tracy Shen, Michiharu Yamashita, Ethan Prihar, Neil Heffernan, Xintao Wu, Ben Graff, and Dongwon Lee. 2021a. Mathbert: A pre-trained language model for general nlp tasks in mathematics education. *arXiv preprint arXiv:2106.07340*.

Jia Tracy Shen, Michiharu Yamashita, Ethan Prihar, Neil Heffernan, Xintao Wu, Sean McGrew, and Dongwon Lee. 2021b. Classifying math knowledge components via task-adaptive pre-trained bert. In *Artificial Intelligence in Education: 22nd International Conference, AIED 2021, Utrecht, The Netherlands, June 14–18, 2021, Proceedings, Part I 22*, pages 408–419. Springer.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Meng Zhao, Faizan Javed, Ferosh Jacob, and Matt McNair. 2015. Skill: A system for skill identification and normalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, pages 4012–4017.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2024. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36.