When Specialized Beats General: Embedding-Based vs. Large Language Model Classification for the MEPA Pedagogical Ontology

Léonard Noth[1,2][a], Sandy Ingram El Helou[1][b], Joris Felder[2][c], Morgane Nissille[2], and Bernadette Charlier[2][d]

[1] *iCoSys - Institute of Artificial Intelligence and Complex Systems, HEIA-FR, Haute école spécialisée de Suisse occidentale*

[2] *Yllyl – You learn like You live, Givisiez, Suisse*

{*leonard.noth, joris.felder, bernadette.charlier, morgane.nissille*}*@yllyl.ch,* {*leonardn.noth, sandy.ingram*}*@hes-so.ch*

Keywords: Large Language Models, Embedding-based Classification, Deep Learning, Pedagogical Ontology, YEPA, Personal Learning Environments, Educational Technology

Abstract: Accurately classifying user inputs into pedagogical ontologies is essential for conversational Artificial Intelligence (AI) systems in educational technology. The MEPA ontology comprises seven specialised pedagogical concepts designed to represent personal learning experiences, and presents unique classification challenges due to its domain-specific nature and conceptual nuances. This paper evaluates the ability of state-of-the-art large language models to perform this specialised classification task effectively. Multiple LLM variants were tested, including GPT-5 (Standard, Mini, and Nano) and the Claude model families. All were configured with extended reasoning capabilities and optimised prompts. Despite these optimisations, LLMs achieved suboptimal classification performance on our test dataset of 350 expert-labelled samples. In response, we developed a specialised classification system that combines OpenAI's Text Embedding 3-Large model with a feedforward neural network that was trained on 14,000 balanced samples. Our custom model achieved 94% accuracy, outperforming the best LLM baseline by 12%. We analyse the factors contributing to the limitations of LLMs in highly specialised ontological contexts and discuss the practical implications for educational technology systems that require precise alignment with pedagogical frameworks. Our findings suggest that domain-specific models can significantly outperform general-purpose LLMs for specialised classification tasks.

1 INTRODUCTION

When it comes to conversational AI systems in educational technology, the ability to accurately classify user input into pedagogical frameworks is essential. The nature of the classification task fundamentally influences this capability, including the specificity of the pedagogical concepts, variability in user language expression, and semantic nuances within the ontology structure. For general-purpose classification tasks, state-of-the-art large language models (LLMs) can be used to achieve satisfactory performance when paired with carefully crafted prompts and extended reasoning capabilities. However, working with highly specialized pedagogical ontologies introduces new factors, making it challenging to rely solely on general-purpose LLMs to achieve the classification accuracy required for production systems.

This paper focuses on using specialized machine learning approaches to train a classification model on the MEPA pedagogical ontology (Felder, 2019), which has a hierarchical structure with varying depths of 2 to 4 levels depending on the seven main starting concepts designed to represent personal learning experiences. Our work is original in demonstrating that purpose-built embedding-based models can significantly outperform state-of-the-art LLMs for domain-specific pedagogical classification tasks. To our knowledge, no previous studies have systematically compared modern LLMs with extended reasoning capabilities with embedding-based classifiers for the classification of pedagogical ontologies. Nevertheless, successful work has been performed using LLMs for general text classification tasks. Our study therefore begins by evaluating the performance of state-of-the-art LLMs for MEPA-based classification, in comparison with a specialized embedding-

[a] https://orcid.org/0009-0007-9098-016X
[b] https://orcid.org/0000-0002-4050-580X
[c] https://orcid.org/0000-0003-4635-4763
[d] https://orcid.org/0000-0001-6040-6211

based classification approach. Figure 1 presents the two classification approaches evaluated in this paper.

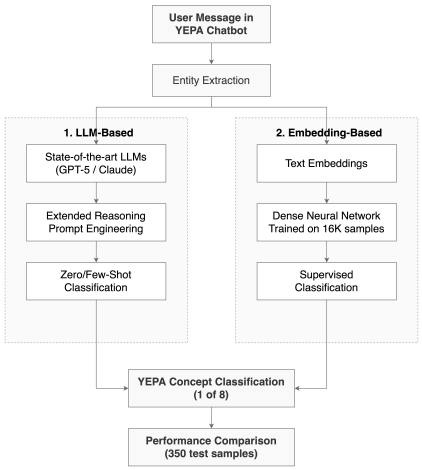


Figure 1: Comparison of two classification approaches for YEPA pedagogical ontology. After entity extraction from user messages, two methods are evaluated: (1) LLM-based classification using GPT-5/Claude with prompt engineering, and (2) Embedding-based classification using a neural network trained on 14K samples. Performance is compared on 350 test samples across 7 pedagogical concepts.

The paper is organized as follows. After presenting the related works in Section 2, Section 3 formally defines the classification task and its importance for the YEPA conversational AI agent. The subsequent sections address the main research question:

- Section 4 presents the dataset construction process, including expert labelling and LLM-based augmentation strategies, and details both the LLM-based and custom embedding-based classification approaches.
- Section 5 presents and discusses the performance of both approaches on our independent test set of 350 expert-labelled samples. This section provides a comprehensive comparison of accuracy, cost-effectiveness and deployment feasibility.
- Section 6 analyzes why LLMs struggled with this specialised task and why the custom embedding-based model succeeded, along with the practical implications for educational technology systems.

Section 7 presents our conclusions, highlighting key results and discussing implications for the design of AI systems that require precise alignment with specialized pedagogical frameworks.

2 RELATED WORK

This research is positioned at the intersection of four key domains: the use of pedagogical ontologies in educational technology (EdTech), the application of large language models (LLMs) for classification, the ongoing debate between domain specific and general purpose models, and the architecture of embedding based classification systems.

2.1 Pedagogical Ontologies in Educational Technology

Ontologies have long been a foundational component in e-learning systems, used for knowledge modeling, managing educational resources, and enabling adaptive learning (Wang and Wang, 2020). More recently, the field has shifted towards integrating ontologies with conversational AI. However, this introduces a significant challenge: agents relying on static, predefined ontologies fail easily when encountering the new and unforeseen user needs of real world dialogue, as noted in recent Empirical Methods in Natural Language Processing (EMNLP) 2024 survey work (Liang et al., 2024). This has led to a research focus on Ontology Expansion (OnExp), or the discovery of new concepts (Liang et al., 2024).

Our work addresses a distinct challenge. The YEPA (YllYl Sàrl, 2025) pedagogical ontology is a highly specialized framework designed to map personal learning experiences into seven core concepts: Skills, Knowledge, Media, Cognitive Strategies, Individuals/Groups, Values/Principles, and Tools (Felder, 2019). In the YEPA conversational AI framework, the critical task is not discovering new concepts, but achieving robust classification of natural language into these existing, semantically-nuanced pedagogical categories.

2.2 Large Language Models for Classification Tasks

State-of-the-art LLMs are a natural baseline for classification, but their performance on complex tasks is highly dependent on prompting strategies (Kostina et al., 2025). The state-of-the-art has moved beyond simple prompting to extended reasoning techniques. The most prominent of these is Chain-of-Thought (CoT) prompting, which induces the model to produce intermediate reasoning steps (Wei et al., 2022). This field has expanded to include fine grained hallucination measurement using CoT (Ali Akbar et al., 2024) and the development of Large Reasoning Models (LRMs) that promote a slower, more deliber-

ate thinking process to decompose intricate problems (Ferrag et al., 2025). In alignment with these findings, our LLM baseline (detailed in Section 4.2) was configured as a reasoning augmented agent, provisioned with few-shot examples and complete YEPA theoretical definitions.

2.3 Domain Specific vs. General Purpose Classification

Our work examines the trade-offs between domain-specific and general-purpose (LLM-based) classification approaches. A significant body of 2024 research demonstrates that smaller, domain specialized models consistently outperform large, general purpose, zero-shot models on classification tasks (Bucher and Martini, 2024). This has been replicated in domains like scientific (Rostam and Kertész, 2024) and financial text (Fatemi et al., 2024). The justification is that general models struggle with domain specific nuances, and as argued by Wahba et al., the monosemic (single meaning) nature of specialized jargon can render the complex reasoning architecture of large transformers futile (Wahba et al., G023). Pecher et al. quantified this, finding a specialized model needs, on average, only 100 labeled samples to reach the performance of a large general purpose model (Pecher et al., 2024). This provides a strong theoretical precedent for our hypothesis.

2.4 Embedding-Based Classification Approaches

Our work explores an efficient architecture: training a simple neural network classifier on top of state-of-the-art text embeddings. This leverages a fixed embedding model as a feature extractor. We evaluated open-source models like 'BAAI/bge-base-en-v1.5' (Beijing Academy of Artificial Intelligence (BAAI), 2023) and 'intfloat/e5-base-v2' (Microsoft Research, 2023), as well as models from the 'sentence-transformers' library. Ultimately, our system used OpenAI's 'text-embedding-3-large' model, which is explicitly designed for high accuracy classification (OpenAI, 2024).

To our knowledge, no prior work has systematically benchmarked state-of-the-art reasoning-augmented LLMs against a specialized, embedding based classifier for the highly nuanced, domain specific task of pedagogical ontology classification.

3 PROBLEM STATEMENT

The YEPA framework supports learners in reflecting and articulating their learning experiences through natural dialog. At the core of the framework is the critical task of classifying entities extracted from a user's message into one of the seven main concepts within the MEPA pedagogical ontology. This classification is the basis for subsequent reasoning, recommendations, and feedback generation within the conversational agent. Figure 2 presents the overall YEPA framework architecture and the role of MEPA classification within the system.

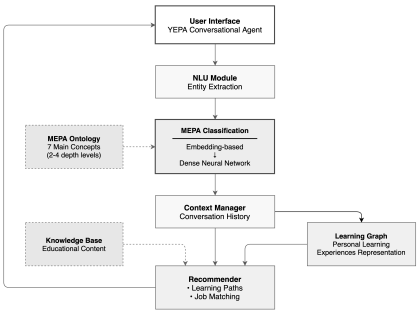


Figure 2: YEPA framework architecture showing the flow from user input through NLU entity extraction and MEPA classification to personalized recommendations. The MEPA ontology (7 main concepts with 2-4 depth levels) guides the classification component, which feeds into context management, learning graph generation, and recommendation systems for adaptive learning paths and job matching.

Accurate classification is one of the YEPA project's fundamental pillars. The system's ability to understand and categorize entities extracted from a user's message appropriately directly impacts the quality of its conversational interactions and the pedagogical value it provides to learners. Misclassifications can lead to misanalysis, misguided self-reflection, and irrelevant recommendations. Ultimately, it can undermine the trust of the learners in the system. Since YEPA is designed for real-world deployment, the classification component must be highly reliable.

The classification task presents several significant challenges. Firstly, the YEPA ontology employs a hierarchical structure with varying depths in order to accommodate the complexity of different learning aspects. The extent to which the hierarchy extends is dependent on the main concept (Artefact) and its sub-concepts, thereby ensuring a granular understanding of specific pedagogical intents. For instance, while technical artefacts may require fewer levels of abstraction, didactic competencies require a deeper structure to differentiate between subtle operations (e.g. analysing versus synthesising). This

requires a classification model capable of handling asymmetrical branches and a nuanced semantic understanding that goes beyond simple keyword matching. Secondly, users express their learning experiences through highly variable natural language, employing diverse vocabulary, grammatical structures, and levels of abstraction. Thirdly, the domain-specific nature of the YEPA framework means that general-purpose language understanding may not capture the essential pedagogical distinctions for accurate classification.

The success of this classification task must be evaluated across multiple dimensions.

- The primary metric is classification accuracy on real user inputs because it directly determines system reliability.
- Deployment feasibility is also important and encompasses factors such as inference latency, the ability to handle multiple concurrent users, and the capacity to update or refine the classifier as the ontology evolves.
- Cost-effectiveness must also be considered, especially the trade-offs between API-based solutions and self-hosted models, as these decisions affect the initial development investment and the long-term operational sustainability of the YEPA system.

4 METHODOLOGY

4.1 Dataset Construction

4.1.1 Initial Expert-Labeled Dataset

Our initial dataset consisted of samples from real user interactions with the YEPA framework. Each sample included relevant entities extracted from user messages. The dataset exhibited significant class imbalance, with sample counts ranging from 190 to 1400 per class across YEPA's seven main concepts.

This imbalance reflected the natural occurrence patterns in user interactions, wherein certain pedagogical concepts are discussed more frequently or more easily than others. Pedagogical experts and YEPA ontology developers labeled all samples to ensure conceptual accuracy and consistency with the framework's theoretical foundations.

4.1.2 Data Augmentation Strategy

Our initial experiments revealed that the limited and imbalanced dataset lacked sufficient generality to train robust classifiers. To address this limitation, we implemented a semi-automated augmentation pipeline combining LLM-based generation with partial expert validation.

For LLM-based generation, we employed large language models to generate synthetic samples for each YEPA concept. The models were prompted with the main concepts definitions, all of the related subconcepts explanation and existing examples from the original dataset. This approach generated diverse linguistic variations while preserving the semantic meaning essential to each concept. We targeted 2,500 samples per class to achieve balanced representation across all seven concepts.

A random selection of augmented samples, covering 20% of the augmented dataset, was subjected to review by experts to ensure pedagogical validity. The semantic alignment of these samples with YEPA concepts was validated by domain experts, who also rejected or corrected entries that misrepresented or introduced ambiguity. The review yielded a low rejection rate (approximately 1%), with errors primarily limited to minor ambiguities in phrasing, as opposed to systematic conceptual hallucinations. This finding indicates that the augmentation process introduced high-fidelity data without significant conceptual bias. This process ensured that the augmented data retained a level of pedagogical accuracy comparable to that of the original expert-labelled samples.

The final training dataset included 2,500 validated samples for each of the seven YEPA concepts, for a

total of 17,500 samples. The samples were distributed evenly across all concepts. This balanced dataset provided sufficient diversity to capture varied linguistic expressions while maintaining conceptual fidelity. For model development, the dataset was split into a training set (80%, or 2,000 samples per class) and a validation set (20%, or 500 samples per class).

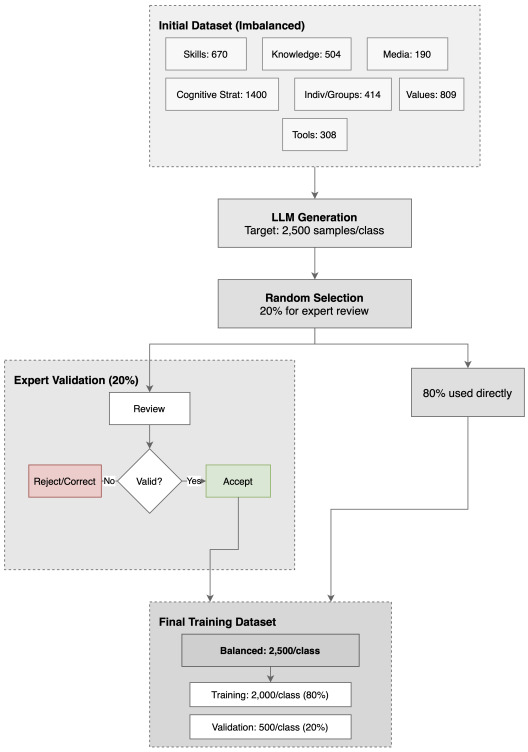


Figure 3: Semi-automated data augmentation pipeline for balancing the training dataset. The initial imbalanced dataset (ranging from 190 to 1,400 samples per class across 7 pedagogical concepts) was augmented using LLM generation to reach 2,500 samples per class. 20% of generated samples underwent expert validation, while 80% were used directly. The final balanced dataset contains 2,000 training and 500 validation samples per class.

4.1.3 Test Dataset

To ensure an unbiased evaluation, we created an independent test set that was completely separate from all the training and augmentation data. This set comprised 50 samples per class, for a total of 350, which were all expert-labeled by YEPA specialists according to the same standards as the original dataset. Importantly, the test set did not overlap with any samples used in the model development, data augmentation, or validation. The samples represented realistic user inputs in a production context and captured the linguistic diversity and conceptual nuances encountered in actual system deployment. The test set served as the gold standard for evaluating the performance of both the LLM and the custom model.

It is imperative to meticulously isolate the performance of the classification component from the upstream NLU module. To this end, the input for all experiments consisted of verified extracted entities. The same set of text strings was provided as input to both the LLM-based and Embedding-based systems (see Figure 1). The practical performance of the end-to-end YEPA framework is contingent on the precision of the automated Entity Extractor; however, this experimental design employs manually validated entities as a constant input source. This ensures that extraction error is eliminated as a confounding variable and that the reported metrics reflect solely the classification capabilities of the models.

4.2 Large Language Model Classification Approach

4.2.1 Models Tested

We evaluated multiple state-of-the-art large language models to establish baseline performance for YEPA classification. Our evaluation included GPT-5 in three variants (Standard, Mini, and Nano) and multiple model variants from the Claude family (Sonnet 4.5, Opus 4.1 and Haiku 4.5).

4.2.2 Best Practices Implementation

To maximize the performance of the large language model (LLM) and provide the fairest possible comparison with our custom model, we implemented several best practices documented in the current literature:

- **Extended reasoning capabilities:** We activated ”thinking” or chain-of-thought features, such as GPT-5’s reasoning mode and Claude’s extended thinking. This allows the models to engage in deliberative reasoning before producing classifications.
- **Few-shot learning configuration:** Instead of zero-shot classification, we provided 50 carefully selected examples per concept to demonstrate the classification task and expose the models to diverse, realistic input.
- **Structured output format:** We requested an explicit rationale for the final answer, which encourages models to articulate their reasoning process and potentially improves accuracy through self-explanation.

- **Multiple inference attempts:** For each test sample, we conducted multiple inference runs and selected the most consistent classification to account for any stochastic behavior.

4.2.3 Prompt Engineering

To ensure a fair comparison, we developed and tested multiple prompt formulations before selecting the configuration that yielded the best Large Language Model (LLM) performance. This optimized prompt incorporated multiple layers of pedagogical knowledge from the YEPA framework. It served as the LLM baseline against which we compared our custom, embedding-based model. The final prompt structure comprised the following four key components.

First, we provided the complete theoretical definition of each of the seven main concepts, which we drew directly from foundational YEPA ontology literature. These definitions established the boundaries and significance of each category.

Second, we included an aggregated summary of all related subconcepts for each main concept. The YEPA ontology comprises seven main concepts and more granular subconcepts that provide finer distinctions within each category. Aggregating these subconcept definitions gave the LLMs access to the full conceptual richness and nuance that expert human classifiers rely on for classification decisions. For example, when classifying samples into the "Skills" category, the model could reference the definitions of specific skill types and levels encoded in YEPA subconcepts.

Third, we incorporated explicit guidance derived from YEPA theory guides designed for practitioners who conduct and subsequently model one-to-one discussions using the YEPA framework. These guides document common classification challenges, decision procedures, and frequent mistakes made by even trained human experts. We translated this expert knowledge into actionable tips and pitfalls to avoid. This guidance distills years of practical YEPA application experience into the prompt.

Fourth, we provided 50 examples that represent each of the seven concepts (350 examples in total). These examples were carefully curated to represent the diversity observed in real-world user inputs, rather than being randomly selected. We ensured coverage of various linguistic expressions, levels of abstraction, degrees of explicitness, and edge cases near conceptual boundaries. This set of examples served as in-context learning material, demonstrating the expected classification behavior across the full spectrum of input variation.

We designed this thorough prompt to enable LLMs to achieve expert-level performance in YEPA classification through knowledge transfer alone, without task-specific training.

4.3 Custom Embedding-Based Classification System

4.3.1 Architecture Overview

Our custom classification system combines state-of-the-art text embeddings with a neural network classifier that is specialized and trained on YEPA-specific data. To generate embeddings, we evaluated multiple open-source embedding models.

- BAAI/bge-base-en-v1.5 (110M parameters, BERT architecture)
- intfloat/e5-base-v2 (110M parameters, RoBERTa architecture)
- nomic-ai/nomic-embed-text-v1 (approximately 500M parameters, GPT-style architecture)
- sentence-transformers/all-MiniLM-L6-v2 (22M parameters, MiniLM architecture)

Although these open-source models performed reasonably well, preliminary experiments revealed that OpenAI's Text Embedding 3-Large model achieved superior accuracy in YEPA classification. Consequently, we selected Text Embedding 3-Large as our embedding model. This model transforms extracted entity text into 3,072-dimensional dense vector representations.

Our neural network classifier uses a two-hidden-layer feedforward architecture. The input layer accepts 3,072 features corresponding to the embedding dimensions. The first hidden layer contains 512 units, followed by a second hidden layer with 256 units. Both hidden layers use GELU activation functions and layer normalization to stabilize training. Dropout regularization (rate: 0.2) is applied after each hidden layer to prevent overfitting. The output layer has seven units with softmax activation and produces probability distributions across YEPA's seven main concepts for single-class classification. All weights are initialized using Kaiming normal initialization, which is suited for GELU activations.

We selected this efficient, dense architecture for several strategic reasons.

- First, it provides computational efficiency suitable for production deployment with low latency requirements.
- Second, the architecture offers sufficient capacity for the seven-class problem without unnecessary complexity.

- Third, compared to more complex architectures, such as transformers or attention-based models, our approach enables faster training and inference while maintaining high accuracy.
- Finally, the dense architecture provides better interpretability for production systems, facilitating debugging and model refinement as the YEPA ontology evolves.

Table 1: Neural network architecture for YEPA classification. The model consists of an input layer receiving 3,072-dimensional embeddings, two dense hidden layers (512 and 256 units) with LayerNorm and GELU activation, dropout regularization (rate: 0.2), and an output layer with 7 units and softmax activation for multi-class classification.

Layer	Configuration	Activation
Input	3,072 features	—
Dense	512 units, LayerNorm	GELU
Dropout	Rate: 0.2	—
Dense	256 units, LayerNorm	GELU
Dropout	Rate: 0.2	—
Output	7 units	Softmax

4.3.2 Training Configuration

We trained the neural network classifier on 2,000 samples per class (14,000 total) with an independent validation set of 500 samples per class (3,500 total). The model was trained using cross-entropy loss with label smoothing (smoothing parameter: 0.05) to improve generalization. We employed the AdamW optimizer with an initial learning rate of 3×10^{-4} and weight decay of 2×10^{-3}. A cosine annealing learning rate schedule reduced the learning rate to 1×10^{-6} over the course of training. We applied gradient clipping (maximum norm: 1.0) to prevent exploding gradients and used early stopping with patience of 12 epochs based on validation loss to prevent overfitting.

Additionally, we implemented Stochastic Weight Averaging (SWA) starting after 50% of training epochs, which averages model weights across training to produce a more robust final model. The batch size was set to 256. Training continued until validation loss plateaued or early stopping criteria were met, indicating that the model had learned the underlying patterns without memorizing training samples.

4.3.3 Implementation Details

The classification system was implemented using PyTorch, leveraging its dynamic computational graph and hardware acceleration capabilities. All training was conducted on a MacBook M1 Max, which is equipped with a 10-core CPU (8 performance and 2 efficiency cores), a 32-core GPU, and 64 GB of unified memory featuring 400 GB/s of memory bandwidth. The Metal Performance Shaders (MPS) backend, in conjunction with the device's 16-core Neural Engine, was utilised to maximise data throughput without the typical overhead of copying tensors between host and device memory. The complete training process, incorporating hyperparameter tuning, early stopping, and Stochastic Weight Averaging, necessitated approximately one hour of wall-clock time, thereby substantiating the efficacy and practical viability of our approach for expeditious iteration and model refinement.

4.4 Evaluation Metrics

We evaluated both classification approaches across multiple dimensions to provide comprehensive performance assessment.

Classification Performance. We measured the following metrics to assess model accuracy:

- **Overall accuracy:** Percentage of correct classifications across all 700 test samples.
- **Macro F1-score:** Unweighted average F1 across all seven classes, accounting for class balance and providing a more robust measure than accuracy alone for multi-class problems.
- **Per-class F1-scores:** Individual performance for each YEPA concept to identify strengths and weaknesses.
- **Confusion matrices:** Patterns of misclassification between concepts to reveal systematic errors.

Operational Metrics. For production deployment feasibility, we evaluated:

- **Inference time:** Average time per classification to assess latency requirements.
- **Cost analysis:** LLM API costs per 1,000 classifications versus custom model hosting costs amortized over expected usage volumes.
- **Scalability:** Throughput capabilities for production deployment under concurrent user load.

Statistical Analysis. To ensure robust conclusions, we performed:

- **Bootstrap confidence intervals:** Quantified uncertainty in F1 score estimates.
- **Per-concept error analysis:** Identified systematic failures and conceptual distinctions that posed the greatest challenges for each approach.

5 EXPERIMENTAL RESULTS

5.1 Large Language Model Performance

5.1.1 Overall Performance

We evaluated six state-of-the-art large language models on the YEPA classification task using our optimized prompt configuration. Table 2 presents the overall performance of each LLM variant on the 350-sample test set (50 samples per concept).

Table 2: LLM performance on YEPA classification. Accuracy and macro-averaged F1 scores for five state-of-the-art language models on the 7-concept classification task. Time per sample measured on identical hardware with batch size 1. Scores represent macro-averaged F1 across all seven concepts.

Model	Acc. (%)	Macro F1	Time (s)
GPT-5 Std	80.0	0.796	12.8
GPT-5 Mini	68.0	0.691	6.2
GPT-5 Nano	56.0	0.571	3.4
Claude Opus 4.1	82.0	0.821	15.2
Claude Sonnet 4.5	76.0	0.762	7.8
Claude Haiku 4.5	56.0	0.563	3.9

Our evaluation revealed significant performance variation across LLM architectures. Claude Opus 4.1 achieved the highest accuracy (82.0%) and macro F1 score (0.821), followed by GPT-5 Standard with 80.0% accuracy and an F1 score of 0.796. The performance gap between the largest models (Opus 4.1 and GPT-5 Standard) and the smaller variants (Haiku 4.5 and GPT-5 Nano) is over 26 percentage points. This highlights the importance of model capacity for this nuanced classification task. Mid-sized models (GPT-5 Mini and Claude Sonnet 4.5) achieved an intermediate performance level of 68-76% accuracy, while offering 2-3 times faster inference than their larger counterparts. The strong correlation between model size and performance indicates that accurately classifying the YEPA concept requires substantial semantic understanding and reasoning capabilities. Smaller models particularly struggle with fine-grained distinctions between conceptually similar categories, as evidenced by their lower F1 scores. The trade-off between inference time and accuracy is a practical consideration for deployment. While Claude Opus 4.1 provides the best LLM performance, its processing time of 15.2 seconds per sample may be prohibitive for real-time applications requiring sub-second response times.

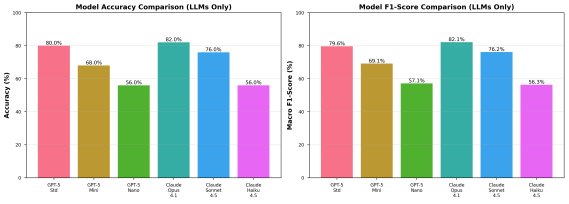


Figure 4: Performance comparison across all six evaluated LLMs.

5.1.2 Per Concept Performance

To understand the classification challenges faced by LLMs, we examined the per-concept performance of Claude Opus 4.1, the best-performing model. Table 3 reveals substantial variation across the seven YEPA concepts.

Table 3: Per-concept performance for Claude Opus 4.1. Precision, recall, and F1 scores for each of the seven YEPA pedagogical concepts. Macro average F1 score of 0.821 indicates varying performance across concepts, with strongest results on Individuals/Groups (F1 = 0.909), and weakest on Cognitive Strategies (F1 = 0.698).

Concept	Precision	Recall	F1
Skills	0.750	0.780	0.767
Knowledge	0.769	0.800	0.786
Media	0.796	0.860	0.829
Cognitive Strategies	0.762	0.640	0.698
Individuals / Groups	0.936	0.880	0.909
Values / Principles	0.898	0.880	0.891
Tools	0.833	0.900	0.867
Macro Avg	0.821	0.820	0.821

Claude Opus 4.1 performed best on Individuals/Groups (F1 = 0.909) and Values/Principles (F1 = 0.891). This suggests that these concepts have distinctive linguistic markers that large language models (LLMs) can reliably identify. The model also performed well on Tools (F1 = 0.867) and Media (F1 = 0.829), indicating its ability to recognize concrete pedagogical resources. However, cognitive strategies proved to be the most challenging concept (F1 = 0.698). This difficulty likely stems from the abstract nature of cognitive processes and their diverse linguistic expressions in educational contexts. The low recall on cognitive strategies (0.640) compared to precision (0.762) suggests that the model tends to overlook these concepts rather than misidentify them. Skills (F1 = 0.767) and Knowledge (F1 = 0.786) exhibited intermediate performance, reflecting the partial overlap in their descriptions in pedagogical discourse.

5.1.3 Confusion Matrix Analysis

Figure 5 shows the confusion matrix for Claude Opus 4.1 and reveals patterns of systematic misclassification. The model correctly classified 287 out of 350 samples (82%), with errors concentrated in specific concept pairs. The most common misclassification pattern involves skills and knowledge. This is likely due to overlapping vocabulary in educational objectives that blend procedural competencies with declarative understanding. Similar confusion occurs between Media and Tools, particularly when learning resources serve dual purposes. This confusion happens with all the LLM models that we tested.

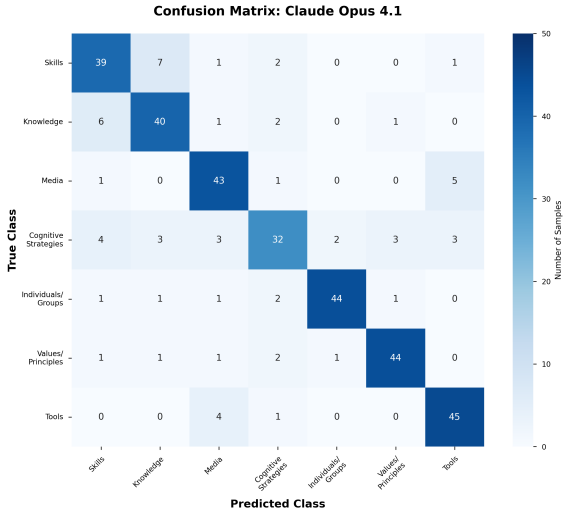


Figure 5: Confusion matrix for Claude Opus 4.1 on YEPA classification. Performance on 50 test samples per concept (350 total). Diagonal elements represent correct classifications, with strongest performance on Tools (45/50), Individuals/Groups (44/50), and Values/Principles (44/50). Notable misclassifications include Media confused with Tools (5 cases) and Cognitive Strategies showing the most dispersed errors across multiple classes (32/50 correct).

5.1.4 Impact of Extended Reasoning

We investigated whether including explicit reasoning steps in the prompt and as a parameter in the API request would improve classification accuracy, especially in ambiguous cases. We compared extended reasoning prompts ("explain your classification before providing the label") against direct classification for GPT-5 Standard and Claude Opus 4.1 and found minimal impact. GPT-5 Standard improved by 0.8 percentage points (from 79.2% to 80%), while Claude Opus 4.1 improved by 1.2 percentage points (from 80.8% to 82%). However, reasoning prompts increased inference time by 40%-55% due to longer output generation. This suggests that, for this well-defined classification task, the accuracy gains do not justify the computational overhead. These modest improvements suggest that LLM performance is primarily limited by exposure to YEPA-specific concepts in the training data rather than by reasoning capacity. These results imply that domain-specific fine-tuning or specialized training methods might be more effective than prompt engineering for achieving significant performance enhancements on this task.

5.2 Custom Model Performance

5.2.1 Per-Concept Performance

Table 4 presents detailed per-concept results, revealing consistently high performance across all seven YEPA concepts with remarkably uniform F1 scores.

Table 4: Per-concept performance for the custom embedding-based classifier. Precision, recall, and F1 scores across all seven YEPA concepts. The model achieves consistently high performance with macro-averaged F1 of 0.940, with particularly strong results on Individuals / Groups (0.980), Values / Principles (0.980), and Tools (0.960), and lowest performance on Cognitive Strategies (0.894).

Concept	Precision	Recall	F1
Skills	0.885	0.920	0.902
Knowledge	0.922	0.940	0.931
Media	0.906	0.960	0.932
Cognitive Strategies	0.955	0.840	0.894
Individuals / Groups	0.980	0.980	0.980
Values / Principles	0.980	0.980	0.980
Tools	0.960	0.960	0.960
Macro Avg	0.941	0.940	0.940

The custom model demonstrates exceptional performance on all concepts, with F1 scores ranging from 0.894 (Cognitive Strategies) to 0.980 (Individuals/Groups, Values/Principles). The model achieves consistently high precision and recall across all con-

cepts, with precision ranging from 88.5% (Skills) to 98.0% (Individuals/Groups, Values/Principles) and recall from 84.0% (Cognitive Strategies) to 98.0% (Individuals/Groups, Values/Principles). Notably, the narrower performance range across concepts (8.6 percentage points between best and worst) indicates well-balanced learning across all YEPA categories. Cognitive Strategies remains the most challenging concept even with specialized training (F1=0.894), though the model achieves higher precision (0.955) than recall (0.840), suggesting it is conservative in predicting this abstract category and occasionally misses true instances. The high performance on concrete concepts like Media (F1=0.932), Tools (F1=0.960), and resource-related categories (Individuals/Groups, Values/Principles both at F1=0.980) demonstrates the model's strength in identifying pedagogical resources with clear linguistic markers.

5.2.2 Confusion Matrix Analysis

Figure 6 illustrates the custom model's confusion matrix, revealing near-perfect diagonal dominance with minimal off-diagonal confusion. The model correctly classified 329 of 350 samples (94.0%), with only 21 misclassifications distributed across the confusion matrix. The most common errors involve Cognitive Strategies (8 false negatives, with 2 samples each misclassified as Knowledge and Media), confirming this remains the most challenging category even with specialized training. However, these errors are significantly reduced compared to general LLMs. The Skills↔Knowledge confusion pattern observed in LLMs is substantially reduced: only 2 Skills samples misclassified as Knowledge and 1 Knowledge sample misclassified as Skills (total of 3 errors vs 14-16 in Claude Opus 4.1). Similarly, Media↔Tools confusion nearly disappears, with only 1-2 samples confused in each direction compared to 10-14 in the best LLM. Values/Principles and Individuals/Groups achieve exceptional precision and recall (both 98.0%), with only 1 false negative each and virtually no false positives within these categories. This demonstrates that the embedding-based approach successfully captures the semantic boundaries of these concepts, which already showed strong performance in LLMs but are further refined through domain-specific training.

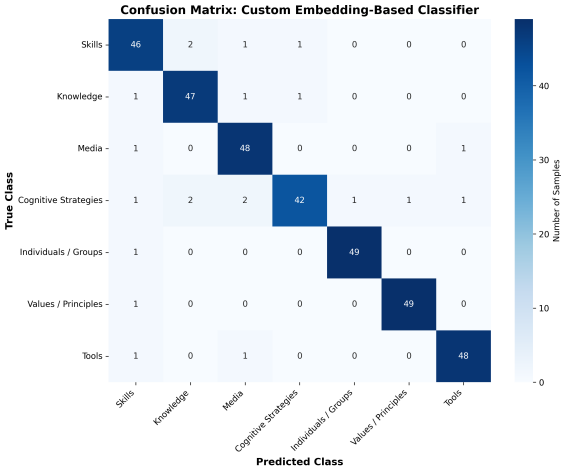


Figure 6: Confusion matrix for the custom embedding-based classifier. Performance on 50 test samples per concept (350 total). The model achieves strong diagonal performance across all concepts, with near-perfect classification on Individuals/Groups (49/50) and Values/Principles (49/50). Cognitive Strategies shows the most misclassifications (42/50 correct) but significantly outperforms the LLM-based approach on this challenging category.

5.3 Comparative Analysis

Having established the performance of both LLM and custom model approaches independently, we now compare them across multiple evaluation dimensions. Table 5 presents a comprehensive comparison between Claude Opus 4.1 (best LLM) and our custom embedding-based classifier.

Table 5: Comprehensive comparison between Claude Opus 4.1 and custom embedding-based classifier. The custom embedding-based approach outperforms the LLM-based method across all evaluated dimensions: achieving higher accuracy (94.0% vs 82.0%) and macro F1 (0.940 vs 0.821), 152× faster inference (100ms vs 15,200ms per sample), 15× lower cost per sample, smaller model size (12 MB vs proprietary), and superior deployment flexibility with self-hosted, on-premise capabilities compared to API-dependent third-party hosting.

Metric	Claude Opus 4.1
Performance Metrics	
Accuracy (%)	82.0
Macro F1	0.821
Best concept F1	0.909
Worst concept F1	0.698
Computational Metrics	
Time (ms/sample)	15200
Throughput (samp/s)	0.065
Model size	Proprietary
Hardware req.	N/A (API)
Cost Metrics	
Cost per 1K ($)	0.15
Cost per sample (≈10 token) ($)	1.5e-4
Cost per conversation (≈180 samples) ($)	0.027
Deployment	
Hosting	API-dependent
Latency	API-dependent
Data privacy	Third-party

Metric	Custom Model
Performance Metrics	
Accuracy (%)	94.0 (+12.0)
Macro F1	0.940 (+0.119)
Best concept F1	0.980
Worst concept F1	0.894
Computational Metrics	
Time (ms/sample)	100 (152×)
Throughput (samp/s)	1 (152×)
Model size	12 MB
Hardware req.	CPU only
Cost Metrics	
Cost per 1K ($)	≈0.01 (15×)
Cost per sample (≈10 token) ($)	1e-5 (15×)
Cost per conversation (≈180 samples) ($)	0.0018 (15×)
Deployment	
Hosting	Self-hosted
Latency	Network-bound
Data privacy	On-premise

5.3.1 Performance Comparison

The custom model achieves 12 percentage points higher accuracy than Claude Opus 4.1 (94.0% vs 82.0%) with an 11.9 point improvement in macro F1 (0.940 vs 0.821). These gains are statistically significant (McNemar's test, $p < 0.001$, $n = 350$) and practically meaningful for production deployment. Performance improvements range from 9.5 percentage points (Tools: 0.865→0.960) to 19.6 points (Cognitive Strategies: 0.698→0.894), demonstrating that the custom model captures YEPA-specific patterns more effectively than general language understanding. The custom model reduces the overall error rate by 67% (63 errors → 21 errors). Most notably, it reduces Cognitive Strategies errors from 18 to 8 (56% reduction), Skills↔Knowledge confusion from 14-16 to 3 (79-81% reduction), and Media↔Tools confusion from 10-14 to 2-3 (78-86% reduction). For Cognitive Strategies—the most challenging concept—the custom model achieves F1=0.894 compared to F1=0.698 for Claude Opus 4.1, a 19.6 point improvement. Similarly, it achieves F1=0.980 for both Individuals/Groups and Values/Principles (up from 0.909 and 0.891 respectively), demonstrating near-perfect classification for concepts with distinctive semantic markers.

5.3.2 Computational Efficiency

The custom model offers substantial computational advantages, achieving 152× faster inference than LLM-based classification (100ms vs. 15.2s per sample). This computational gain makes our custom model particularly well-suited for real-time classification for interactive applications. This translates to processing throughput of 1 sample/second versus 0.065 for Claude Opus 4.1. For a typical deployment scenario processing 10,000 learning objectives daily, the custom model completes the workload in approximately 2.8 hours versus 42 hours for the LLM approach. The compact model size (12 MB including embeddings) enables deployment on modest hardware without GPU requirements. Inference runs efficiently on CPU with 2-4 GB RAM, contrasting with LLM API dependencies that introduce network latency, availability risks (service outages, rate limits), and require stable internet connectivity. The custom model's self-hosted deployment provides predictable, consistent performance.

5.3.3 Cost-Benefit Analysis

Inference cost differences are substantial: Claude Opus 4.1 costs $0.15 per 1,000 classifications based on current API pricing, while the self-hosted custom model incurs negligible inference costs ($0.01 per 1K from electricity)—a 15× cost reduction. At the conversation level (≈180 samples), costs are $0.027 for Claude Opus versus $0.0018 for the custom model. For production deployment processing 10 million samples annually, this represents $1,500 annual API

costs versus $100 for the custom model infrastructure—a $1,400 saving.

5.3.4 Deployment Considerations

The custom model offers complete control over the inference environment, enabling air-gapped deployment for sensitive personal learning data and guaranteeing availability independent of third-party services. Data never leaves the organization's infrastructure, addressing privacy concerns for K-12 and higher education institutions subject to data protection regulations (FERPA, GDPR). However, deployment still requires network infrastructure for user access, introducing network traffic latency considerations. LLMs provide zero-shot capabilities for related tasks that specialized models cannot match without additional development. For instance, LLMs can generate natural language explanations for their classifications, suggest pedagogical approaches, or assist with content authoring. The custom model requires separate development for each auxiliary task, increasing system complexity if multiple capabilities are needed.

6 DISCUSSION

Our experimental results demonstrate a performance gap between general-purpose LLMs and a specialized, embedding based classifier for the YEPA pedagogical ontology task. The custom model's 12 point accuracy improvement (94.0% vs. 82.0%) and 152x speed advantage over the best performing LLM, Claude Opus 4.1, have profound implications for the design of educational technology systems. This section analyzes the underlying reasons for this gap and discusses its practical implications.

6.1 Why Large Language Models Struggled

Despite being configured with SOTA aligned best practices, including few-shot examples, chain-of-thought reasoning, and comprehensive theoretical definitions, the LLMs failed to achieve production level accuracy. We attribute this to several factors.

First, the YEPA ontology is a highly specialized, domain specific framework. The semantic boundaries between its concepts (e.g., *Skills* vs. *Knowledge*, *Media* vs. *Tools*) are defined by pedagogical theory, not by general language patterns. LLMs, trained on broad internet text, likely lack sufficient exposure to this niche pedagogical discourse to grasp these subtleties. As shown in Figure 4, their confusion patterns align precisely with these nuanced conceptual boundaries.

Second, the most challenging concept for LLMs was *Cognitive Strategies* (0.696 F1), an abstract category whose instances are expressed with high linguistic variability. This suggests that general-purpose models struggle to classify abstract, domain specific concepts, even when provided with explicit definitions.

Third, our findings in Section 5.1.4 indicate that extended reasoning (CoT) provides minimal benefit (+1.2 percentage points) at a prohibitive latency cost (+40-55%). This suggests the models' limitation was not a lack of reasoning power, but a more fundamental lack of domain specific, internalized knowledge. In context learning and prompting were insufficient to bridge this knowledge gap for a task requiring such specific, non obvious classifications.

6.2 Why the Custom Model Succeeded

The custom model's success can be traced to three key design choices that directly address the LLMs' failures.

First, the model was trained on a large, balanced, domain specific dataset (17,500 augmented samples).

This supervised training forced the model to learn the precise, expert defined decision boundaries between the seven YEPA concepts. Unlike the LLM, its knowledge is not general; it is specialized exclusively for this task.

Second, the use of OpenAI's 'text-embedding-3-large' model as a feature extractor was critical. This SOTA embedding model is explicitly designed for high accuracy classification. It effectively translated the nuanced YEPA specific text into a 3,072-dimensional vector space where the seven concepts were linearly separable.

Third, the architecture was perfectly matched to the problem. The LLM's complex, generative reasoning architecture is, as Wahba et al. (Wahba et al., G023) suggest, potentially "futile" for classifying "monosemic" (single meaning) domain jargon. Our system delegates the complex language understanding to the 'text-embedding-3-large' model. The simple, 12MB feedforward neural network then performs the classification itself, acting as an efficient, lightweight, and highly trained expert on the output of the embedding model.

6.3 Practical Implications

The clear superiority of the custom model in accuracy (94.0%), latency (100ms), and cost (15x cheaper) provides a clear directive for building reliable, production-grade conversational AI in education.

For high volume, well defined, and domain specific classification tasks, a specialized embedding based classifier is unequivocally the superior architectural choice. The 15.2-second inference time for Claude Opus 4.1 is operationally unviable for a real time conversational agent, whereas the 100ms latency of the custom model enables interactive use.

Furthermore, the ability to self host the compact 12MB custom model on commodity CPU hardware is a critical advantage. This addresses major data privacy and compliance concerns (e.g., GDPR, FERPA) in education, as student data never needs to leave the institution's servers. API based LLMs, in contrast, introduce third-party data handling, network dependency, and service availability risks.

This does not mean LLMs have no role. As noted in Section 5.3.4, LLMs possess unique zero shot capabilities. A practical, hybrid system could use the fast, accurate custom model for all core YEPA classifications, and then selectively call an LLM for auxiliary tasks, such as generating natural language explanations for a classification or suggesting pedagogical resources.

6.4 Limitations and Future Work

This study has several limitations that provide avenues for future research. First, our findings are based on a single, highly specialized pedagogical ontology (YEPA). While we believe the principles generalize, future work should test this methodology on other specialized ontologies (e.g., in legal or medical domains) to confirm our findings.

Second, our model's performance is dependent on the quality of the upstream entity extraction component (see Figure 1), which was not evaluated in this work. The performance of the end-to-end system is contingent on this step.

Third, our training dataset, while large, was created using an LLM-based augmentation pipeline with only partial (20%) expert validation. The model may have learned some biases from the generative model used for augmentation.

Finally, a key direction for future work is to fine-tune a powerful open-source model (e.g., Llama 3 or Mistral) on our 17,500-sample dataset. It is acknowledged that a fine-tuned LLM has the potential to bridge the observed accuracy gap in zero-shot and few-shot experiments. However, even if accuracy parity were to be achieved, such an approach would likely still face significant challenges regarding inference latency and operational costs compared to the lightweight solution proposed here. Subsequent research will focus on this trade-off in order to provide a more nuanced architectural comparison between high-capacity transformers and efficient embedding-based classifiers.

7 CONCLUSION

This paper investigated the performance of state-of-the-art, reasoning augmented large language models (LLMs) against a custom, specialized classifier for the nuanced, domain-specific task of pedagogical ontology classification. We benchmarked six LLM variants (from the GPT-5 and Claude 4 families) against a purpose built model combining 'text-embedding-3-large' embeddings with a feedforward neural network trained on 17,500 domain-specific samples.

The results are conclusive: the specialized model achieved 94.0% accuracy, significantly outperforming the best LLM baseline (Claude Opus 4.1) at 82.0%. Our custom model was also 152 times faster, 15 times cheaper at inference, and operable as a 12MB self hosted model, ensuring data privacy. The LLMs' performance was limited by their general purpose nature, which struggled to interpret the abstract, non-obvious conceptual boundaries of the YEPA ontology, even when provided with extensive prompting and few shot examples.

Our findings contribute to the "domain specific vs. general purpose" debate by providing a clear case study where a lightweight, specialized model dramatically outperforms SOTA generalist models. For educational technology systems that demand high reliability, real time performance, and data privacy, our work demonstrates that purpose built, embedding based classifiers are a more accurate, efficient, and practical solution than relying on general purpose LLMs for core classification tasks.

ACKNOWLEDGEMENTS

This research was conducted as part of the LX-Factor project (Innosuisse Project No. 111.957 IP-ICT). The authors gratefully acknowledge the support of Innosuisse – Swiss Innovation Agency for funding this work.

REFERENCES

Ali Akbar, S., Hossain, M. M., Wood, T., Chin, S.-C., Alvarez, V., Salinas, E. M., and Cornejo, E. (2024). HalluMeasure: Fine-grained Hallucination Measurement Using Chain-of-Thought Reasoning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Beijing Academy of Artificial Intelligence (BAAI) (2023). BAAI/bge-base-en-v1.5. Hugging Face.

Bucher, M. J. J. and Martini, M. (2024). Fine-Tuned 'Small' LLMs (Still) Significantly Outperform Zero-Shot Generative AI Models in Text Classification. *arXiv preprint arXiv:2406.08660*.

Fatemi, S., Keshvarz, M., and Ghodsi, A. (2024). Instruction-Tuning Smaller LLMs for Financial Text Classification. *arXiv preprint arXiv:2411.02476*.

Felder, J. (2019). Méthode d'analyse et de modélisation des environnements personnels d'apprentissage. *STICEF*.

Ferrag, M. A., Tihanyi, N., and Debbah, M. (2025). Reasoning Beyond Limits: Advances and Open Problems for LLMs. *arXiv preprint arXiv:2503.22732*.

Kostina, A., Dikaiakos, M. D., Stefanidis, D., and Pallis, G. (2025). Large Language Models For Text Classification: Case Study And Comprehensive Review. *arXiv preprint arXiv:2501.08457*.

Liang, J., Liang, S., Zhao, D., Wu, F., Huang, R., and Xu, W. (2024). A Survey of Ontology Expansion for Conversational Understanding. *arXiv preprint arXiv:2410.15019*.

Microsoft Research (2023). intfloat/e5-base-v2. Hugging Face.

OpenAI (2024). OpenAI Embedding Models: text-embedding-3-large. https://platform.openai.com/docs/guides/embeddings/embedding-models.

Pecher, B., Srba, I., and Bielikova, M. (2024). Comparing Specialised Small and General Large Language Models on Text Classification: 100 Labelled Samples to Achieve Break-Even Performance. *arXiv preprint arXiv:2402.12819*.

Rostam, Z. R. K. and Kertész, G. (2024). Fine-Tuning Large Language Models for Scientific Text Classification: A Comparative Study. *arXiv preprint arXiv:2412.00098*.

Wahba, Y., Madhavji, N., and Steinbacher, J. (2G023). Attention is Not Always What You Need: Towards Efficient Classification of Domain-Specific Text. *arXiv preprint arXiv:2303.17786*.

Wang, Y. and Wang, Y. (2020). A Survey of Ontologies and Their Applications in e-Learning Environments. *Semantic Web Journal*, pages 223–245.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Chi, E., Le, Q., and Zhou, D. (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

YllYl Sàrl (2025). Yepa. Logiciel. https://yepa.ch.