Transforming Expert Knowledge into Scalable Ontology via Large Language Models

Ikkei Itoku
Amazon
New York, USA
itoku@amazon.com

David Theil
Amazon
Arlington, USA
dtheil@amazon.com

Evelyn Eichelsdoerfer Uehara
Amazon
Seattle, USA
eeichels@amazon.com

Sreyoshi Bhaduri
Amazon
New York, USA
drsre@amazon.com

Junnosuke Kuroda
Amazon
Seattle, USA
kurodaju@amazon.com

Toshi Yumoto
Amazon
Arlington, USA
futosy@amazon.com

Alex Gil
Amazon
New York, USA
gilalexg@amazon.com

Natalie Perez
Amazon
Honolulu, USA
natkper@amazon.com

Rajesh Cherukuri
Amazon
Seattle, USA
rccheruk@amazon.com

Naumaan Nayyar
Amazon
Seattle, USA
nayyarnn@amazon.com

***Abstract*—Having a unified, coherent taxonomy is essential for effective knowledge representation in domain-specific applications as diverse terminologies need to be mapped to underlying concepts. Traditional manual approaches to taxonomy alignment rely on expert review of concept pairs, but this becomes prohibitively expensive and time-consuming at scale, while subjective interpretations often lead to expert disagreements. Existing automated methods for taxonomy alignment have shown promise but face limitations in handling nuanced semantic relationships and maintaining consistency across different domains. These approaches often struggle with context-dependent concept mappings and lack transparent reasoning processes. We propose a novel framework that combines large language models (LLMs) with expert calibration and iterative prompt optimization to automate taxonomy alignment. Our method integrates expert-labeled examples, multi-stage prompt engineering, and human validation to guide LLMs in generating both taxonomy linkages and supporting rationales. In evaluating our framework on a domain-specific mapping task of concept essentiality, we achieved an F1-score of 0.97, substantially exceeding the human benchmark of 0.68. These results demonstrate the effectiveness of our approach in scaling taxonomy alignment while maintaining high-quality mappings and preserving expert oversight for ambiguous cases.**

I. Introduction

In today's data-driven world, organizations evolve and expand their operations across multiple domains and functions. Knowledge management is essential for organizations to scale efficiently, and ontologies and taxonomies are important tools within a knowledge management system ((1). Ontologies and taxonomies are often used to extract, organize, and structure knowledge across domains and fields such as healthcare, business or education to allow consistent classification, retrieval, and information integration ((2).

In general, taxonomies often represent hierarchical structures of concepts, while ontologies focus more on the structure and relationships of the network (3). As organizations seek to unify disparate classification systems, for instance, to integrate medical code consistency (4), better synthesize e-commerce inventories (5), or consolidate educational frameworks or objectives (6), the need for accurate, efficient, and scalable content alignment and mapping is critical. The siloed structure of most taxonomies—a media company, for instance, might use different content categorization systems across its print, digital, and video platforms—impedes cross-functional insights and knowledge reuse across organizations. As organizations seek to leverage their collective knowledge, integrating these disparate taxonomies into an unified ontology has become both critical and challenging.

Taxonomy alignment, the process of mapping between concept systems, traditionally requires time-consuming manual review by experts. Taxonomy alignment seeks to identify semantic correspondences (e.g., equivalences, subsumptions) between different ontologies or taxonomies (7). These semantic correspondences are critical for enabling interoperability between heterogeneous systems, facilitating data integration, and supporting knowledge-based applications.

Although early alignment efforts have historically relied on manual curation efforts, often by domain experts, they have produced relatively accurate matches, though at small scales (8). However, not surprisingly, as ontologies increased in size, complexity, and domain coverage, manual approaches have become prohibitively more expensive and time-consuming (9).

Conventional machine learning approaches require extensive labeled datasets yet struggle with the nuanced contextual reasoning these alignment tasks demand. In contrast, large language model (LLM)-based approaches, such as ours, work with limited labeled data and captures complex semantic relationships. Additionally, LLMs generate detailed rationales, which accelerate expert validation.

In this paper, we present an automated framework for taxonomy alignment that leverages LLMs to map concepts across taxonomies in different domains and functions. Building on a foundation of expert calibration, systematic prompt refinement, and human validation, our approach balances automation with expert judgment to achieve high-quality ontology mappings at scale. We validate our framework in a domain-specific case study where the goal is to determine whether one concept is essential to the definition or realization

of another.

II. Related Work

A. Historical Machine Learning (ML) Methods

To address the need for cheaper and less time-consuming efforts, semi-automated systems were created, such as PROMPT (8) and GLUE (10). PROMPT was designed to assist in ontology merging by suggesting potential mappings as well as conflicts; this allowed human users to confirm or revise the alignments, and while it successfully reduced manual overhead in certain contexts, it relied heavily on structural heuristics and domain-specific rules, which limited its generalizability across different ontological frameworks (8). GLUE advanced this by incorporating machine learning techniques to probabilistically match concepts based on a combination of syntactic, semantic, and instance-based similarity metrics (10); this technique brought the systems a step closer towards full automation. Ultimately, these early systems introduced rule-based or machine learning techniques to reduce expert labor, but both systems required extensive domain tuning and often struggled with heterogeneous structures and complex relationships.

Since the creation of semi-automated systems like PROMPT and GLUE, more recent research has focused on developing machine learning models that can generalize across domains and better capture semantic relationships. For instance, unsupervised and embedding-based approaches, like word2vec or GloVe, have been used to represent ontology terms in continuous vector spaces, enabling similarity computations based on distributional semantics (11)(12). A few years later, building upon the foundation set by unsupervised and embedding-based approaches, neural networks (GNNs) and knowledge graph embeddings were created to encode lexical and structural elements of ontologies for alignment tasks (13). Despite these advances, many models still require large labeled datasets for training and often lack interpretability—making it difficult for domain experts to understand or validate alignment decisions (14).

As the field reaches this inflection point between traditional methods and newer approaches, (15) offers a critical perspective on how ontologies might evolve in the era of large language models, highlighting both the opportunities and challenges at this intersection. This transition has prompted growing interest in the use and application of Large Language Models (LLMs) within the context of ontologies and taxonomies.

B. Emergence of LLMs for Semantic Tasks and Reasoning

Recent advances in LLMs have transformed the possibilities of semantic processing and automated reasoning, due to the models' capacities for deep natural language understanding, contextual reasoning, and flexible knowledge representation (16). These multifacted abilities makes LLMs well-suited for ontology-specific tasks such as entity mapping and axiom generation, where historical ML approaches have often fallen short (17; 18).

Early work by (19) demonstrated that language models can generate explanations for their reasoning, establishing a foundation for interpretable AI. Building on this, researchers developed various prompting techniques to elicit reasoning capabilities from LLMs. Chain-of-thought prompting (20) and other reasoning frameworks (21) showed promise in producing both structured reasoning and improved outputs, while (22) revealed that LLMs possess inherent zero-shot reasoning abilities that can be activated through carefully designed prompts. Beyond manual prompt engineering, automated optimization frameworks have emerged as systematic alternatives. AutoPrompt (23) pioneered gradient-based search for optimal prompts, while later approaches expanded these capabilities: APE (24) uses LLMs themselves as prompt engineers; OPRO (25) frames prompt optimization as a reinforcement learning problem; EvoPrompt (26) applies evolutionary algorithms; and MIPRO (27) employs Bayesian optimization for multi-stage prompt refinement. These approaches systematically generate and refine prompt instructions, often outperforming human-designed instructions for complex tasks. Despite these advances, LLM outputs may remain inconsistent if prompts are not meticulously designed (28). Recent research emphasizes that scaling beyond typical few-shot settings substantially improves performance. (29) demonstrate that many-shot prompting with tens or hundreds of examples can significantly boost in-context learning capabilities, especially for complex tasks requiring nuanced understanding of semantic relationships.

C. LLMs versus Traditional ML in Classification

The landscape of text classification reveals a spectrum of approaches with distinct trade-offs. Traditional machine learning methods such as Support Vector Machines (SVMs) and Naive Bayes classifiers represent the conventional end of this spectrum, relying on engineered features and performing well with sufficient annotated data (30; 31). Moving toward greater complexity, fine-tuned encoder models like BERT occupy a middle ground, leveraging contextual embeddings to capture linguistic patterns beyond the reach of traditional methods (32).

At the far end stand large language models, which, despite their computational demands, demonstrate superior capacity for understanding nuanced, context-dependent relationships crucial for tasks like ontology alignment (33). Recent research reinforces this advantage, showing that strategically prompted LLMs—especially those utilizing few- or many-shot demonstrations—consistently outperform both traditional algorithms and smaller neural models on complex classification challenges (29; 34). This progression illustrates not merely differences in technical approach but a fundamental evolution in how machines process and interpret textual information.

D. Related Work Summary and Study Significance

Ontologies and taxonomies are vital tools for organizing knowledge across domains such as healthcare, education, and commerce (1). As these systems scale in complexity, manual alignment methods have become too costly and inconsistent (8; 9).

Early semi-automated tools like PROMPT and GLUE reduced expert labor through rule-based and ML techniques but required extensive tuning and struggled with diverse ontologies (8; 10). Later models using embeddings and neural networks (e.g., word2vec, GloVe, GNNs) improved semantic understanding but often lacked transparency and required large labeled datasets (11; 12; 13; 14). However, LLMs offer a more flexible, data-efficient alternative, showing strong performance on ontology-related tasks with minimal supervision (17; 18). Prompt engineering and optimization techniques—such as few-shot learning, many-shot prompting, and frameworks like MIPRO, OPRO, and APE—further enhance accuracy and adaptability without extensive retraining (16; 20; 29; 27; 24; 25).

Building on these research insights, our approach integrates domain calibration with an LLM-based alignment pipeline, striking a balance between automation and expert validation. Our framework integrates systematic prompt refinement with minimal example-based guidance, enabling the model to produce both classification outcomes and rationales without requiring extensive labeled datasets.

We employ both manual optimization techniques and automated approaches like MIPRO to maximize performance while maintaining transparency in the reasoning process. Overall, our work addresses scalability and complexity challenges by leveraging LLMs to generate high-quality alignments while retaining expert oversight for domain-specific edge cases.

III. Methodology

A. Data Annotations

1) Initial Annotation Phase: We began with our initial annotation phase where a set of 973 concept pairs was independently labeled by four annotators. Each instance is represented as a pair of textual descriptions: (Concept A, Concept B). The annotation task assessed whether Concept A is essential for the realization of Concept B. The output was a binary label: *Required* (indicating Concept A is essential for Concept B's completion) or *Not Required* (indicating Concept A may be valuable but is not essential for Concept B's completion). Early observations revealed notable disagreements among annotators, with only 22% of concept pairs achieving unanimous agreement.

2) Calibration Sessions: To address low agreement among annotators, we conducted calibration sessions for the 759 non-unanimous pairs (78% of total). Initially, the annotators used a significance-based Likert scale with categories (*Critical*, *Significant*, *Beneficial*, *Marginal*, and *Irrelevant*) to rate each instance. However, the subjective interpretation of these terms led to varied assessments. During calibration discussions, we introduced an alternative frequency-based scale with categories (*Always*, *Usually*, *Often*, *Sometimes*, and *Not Necessary*). Under this scheme, only instances rated as *Always* were mapped to the label *Required*, while all other ratings were considered *Not Required*.

For example, "verbal communication" was determined to be *Always Necessary* (Required) for "mentoring" since it inherently involves spoken interaction, while "written communication" was classified as *Sometimes Necessary* (Not Required) as mentoring activities rarely depend solely on written exchanges.

3) Annotation Results: The calibration process established a ground truth dataset, resulting in 314 linkage rationales. The final distribution showed 34% of concepts marked as *Required* and 66% as *Not Required*. Comparing initial independent annotations against the calibrated ground truth across the full dataset of 973 samples yielded metrics of 0.69 for precision, recall, and accuracy, with a 0.68 F1-score (Table I), highlighting both the challenges in the annotation process and the importance of calibration. Unlike later model-based experiments that use data partitioning, these metrics represent the initial human benchmark on the entire collection. On average, reaching a consensus for each instance took approximately 4 minutes, with the overall calibration process spanning about 50 hours for the entire 973 samples.

Precision	Recall	F1 Score	Accuracy
0.690	0.693	0.682	0.691

TABLE I: Initial vs Calibrated Human Annotations (Full Dataset of 973 Samples)

B. Prompt Optimization

1) Manual Instruction Optimization (Zero-shot Prompting): With a calibrated ground truth dataset established, we focused on developing an LLM-based approach to scale this classification task. Building on the discovery that LLMs exhibit strong zero-shot reasoning abilities (22), our initial experiments focused on designing effective zero-shot instructions to guide model reasoning without exemplars. Transitioning from human annotation to an LLM-based system required thoughtful prompt engineering to guide the model toward decision patterns consistent with our calibration sessions. Our prompt optimization strategy evolved through several stages, beginning with basic zero-shot instructions and progressively incorporating insights from the calibration process.

We first manually refined the prompt by integrating detailed guidelines and reasoning processes derived from our calibration sessions. Specifically, we elaborated on the criteria for assessing concept necessity, provided illustrative examples to clarify ambiguous cases, and incorporated the documented rationales from annotators. These enhancements enriched the prompt with explicit context and decision-making cues, thereby improving the LLM's understanding and performance. (Detailed prompt formulations are available in Appendix A.)

2) Automated Instruction Optimization (Zero-shot Prompting): To build upon our manually refined prompt, we leveraged automated prompt optimization techniques. Drawing from the frameworks discussed in Section II-B, we employed the Multiprompt Instruction Proposal Optimizer (MIPRO) (27), for its effectiveness with structured reasoning tasks and ability to generate contextually relevant instructions.

MIPRO systematically enhances prompts in two phases: proposal generation and credit assignment. In the proposal generation phase, it creates diverse candidate prompts by

bootstrapping few-shot examples from the training data. It leverages multiple contextual sources—such as training dataset summaries, prompt summaries, previously bootstrapped examples, and randomly sampled generation tips—to produce new candidate instructions that capture the nuanced requirements of our task.

In the credit assignment phase, MIPRO applies Bayesian optimization to evaluate these candidate prompts. A Bayesian model estimates each prompt's performance using evaluation metrics (e.g., accuracy or exact match) on validation mini-batches. The optimizer then iteratively refines and selects the most effective combinations of candidate instructions and demonstrations in the form of example inputs and outputs, periodically validating on the full set.

We applied MIPRO to a dataset of 963 annotated samples, partitioned equally for training (instruction generation), development (credit assignment with exact match evaluation), and testing. Using zero-shot demonstrations with a mini-batch size of 25 and full validations every 10 mini-batch trials, this process generated optimized prompts.

3) Human-Generated Rationales (Few-shot Prompting): To further enhance the LLM's performance, we incorporated human-generated rationales into the few-shot prompts. During the calibration sessions described in Section III-A2, annotators resolved disagreements and documented 314 rationales that captured the reasoning behind each labeling decision (see Appendix B for an example of human-generated rationales). These rationales were integrated into the prompts alongside the optimized instructions from the previous section. We conducted experiments using both 3-shot and 10-shot demonstration settings, with each demonstration including a pair of textual descriptions, the calibrated binary classification, and the associated rationale.

4) LLM-Generated Rationales (Few-shot Prompting): Building on our experiments with human-generated rationales, we explored the use of LLM-generated rationales as an alternative. Recent research indicates that model-generated chain-of-thought rationales can surpass human-written explanations (35; 36). Our goal was to optimize these LLM-generated rationales for concept-linkage tasks and compare their effectiveness with human-generated ones. Using the same dataset split as in the previous section, we leveraged MIPRO to generate and refine candidate rationales. Specifically, MIPRO produced six candidate rationales per demonstration, and validation testing was used to select the most effective versions. We evaluated this approach using both 3-shot and 10-shot demonstration settings. (See Appendix B for examples of LLM-generated rationales.)

5) Many-Shot Demonstrations (Many-shot Prompting): While our few-shot demonstrations provided insights, recent work by (29) suggests that scaling the number of demonstrations can lead to performance improvements through in-context learning (ICL) (16**?**). Motivated by these findings, we extended our few-shot approach to the many-shot regime, hypothesizing that an increased number of examples would enhance LLM performance by exposing the model to diverse reasoning patterns and enabling pattern recognition across demonstrations. To this end, we created a demonstration pool by generating LLM rationales for 642 concept pairs from our training and development datasets. Each entry in the pool comprised three components: the pair of textual descriptions, the LLM-generated rationale, and the ground-truth binary classification.

For evaluation, we constructed prompts by randomly selecting between 50 and 300 demonstrations from this pool to assess how increasing the context window affects the model's ability to generate accurate rationales and classifications for new, unseen test samples.

IV. Experiments

A. Experimental Setup

Our experiments were conducted on a calibrated dataset of 973 annotated concept pairs, where each instance consists of two textual descriptions of concepts, along with a binary label (*Required*, *Not Required*) indicating whether one concept is essential to the definition or realization of the other. The dataset was partitioned into training, development, and test sets as described previously.

We evaluated our framework using several LLMs, including Anthropic's Claude 3.7 Sonnet v1, Claude 3.5 Sonnet v2, and Claude 3 Haiku v1. In our assessment of Claude 3.7 Sonnet, we examined both standard inference mode (without thinking) and enhanced thinking mode (with 10,000 reasoning tokens). Reasoning token budget optimization is explored in Section VI-B as future work. Additionally, we conducted prompt optimization experiments across zero-shot, few-shot, and many-shot demonstration settings to determine optimal prompting strategies.

Performance was evaluated by comparing the ground truth and generated labels (*Required*, *Not Required*) using precision, recall, and accuracy, with emphasis on the F1-score as our primary evaluation criterion. Human annotation performance established a benchmark F1-score of 0.68 for comparison. All experiments were executed in an offline environment where inference latency was not considered a critical constraint for evaluation purposes.

B. Results and Analysis

1) Prompt Optimization:

a) Manual Instruction Optimization (Zero-shot Prompting): Using a baseline one-sentence instruction, models showed varying F1 scores (Haiku3: 0.33, Sonnet3.5: 0.29, Sonnet3.7-standard: 0.62, Sonnet3.7-think: 0.76). By enhancing the prompt with guidelines and reasoning processes from our calibration sessions, we observed performance improvements across models (Haiku: 0.39, Sonnet3.5: 0.35, Sonnet3.7-standard: 0.71, Sonnet3.7-think: 0.85). As Figure 1 illustrates, the benefits of prompt optimization scale with model capacity, with larger models showing greater receptiveness to enhanced instructions. Notably, both Sonnet3.7-standard and Sonnet3.7-think exceeded human performance (Human: 0.68). See Appendices A and C for prompts and full results.

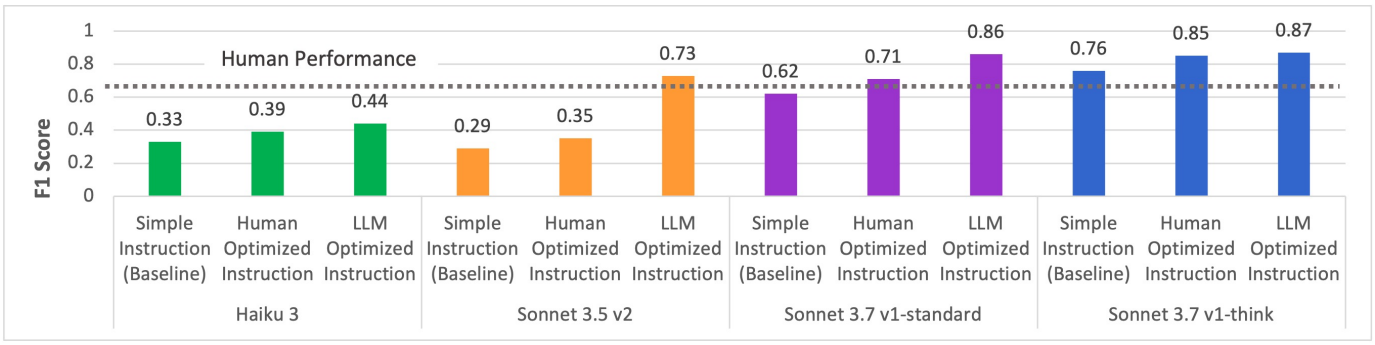

Fig. 1: Performance of Human vs LLM Optimized Instructions (Zero-shot)

b) Automated Instruction Optimization via MIPRO (Zero-shot Prompting): We applied the MIPRO framework to transform a single-sentence instruction into a structured prompt of approximately 40 sentences (see Appendix A for an example prompt). The enhanced prompt integrated explicit task guidelines, reasoning steps, and edge case handling. This optimization improved performance in F1 across all models (Haiku: 0.39 to 0.44, Sonnet3.5: 0.35 to 0.73, Sonnet3.7-standard: 0.71 to 0.86, and Sonnet 3.7-think: 0.85 to 0.87). As shown in Figure 1, Sonnet 3.5, and Sonnet 3.7-standard exceeded human benchmark (Human: 0.68) after optimization.

c) Human-Generated Rationales (Few-shot Prompting): Following instruction optimization, we investigated the efficacy of augmenting the instructions with human-generated rationales. While MIPRO provided methodological guidance, human rationales captured expert reasoning patterns documented during calibration sessions for each pair (see Section III-A3 for calibration details and Appendix B for an example of human-generated rationales). We examined the combination of LLM-optimized instructions with human-generated rationales. As illustrated in Figure 2, incorporating 10 human-generated rationales as demonstrations yielded performance improvements for Haiku 3 and Sonnet 3.5, but not for Sonnet 3.7. Specifically, Haiku 3's F1 score improved from 0.44 (LLM-optimized instruction only) to 0.52 (LLM-optimized instruction with human-generated rationales), while Sonnet 3.5 improved from 0.73 to 0.79.

These empirical results suggest a complementary relationship between optimized instructions and exemplar reasoning, where their combination produces performance gains exceeding those observed from either approach in isolation.

d) LLM-Generated Rationales (Few-shot Prompting): Building on experiments with human-generated rationales, we investigated whether LLM-generated rationales could serve as an alternative. As Figure 3 demonstrates, these rationales outperformed human-generated ones across most configurations. For most models in both 3-shot and 10-shot settings, LLM-generated rationales yielded superior results compared to human-generated rationales, with the exception being the 3-shot setting with Haiku 3. The largest improvements occurred with Sonnet 3.7, where both standard and think modes exhibited an F1 score increase of 0.17 when using LLM-generated rationales.

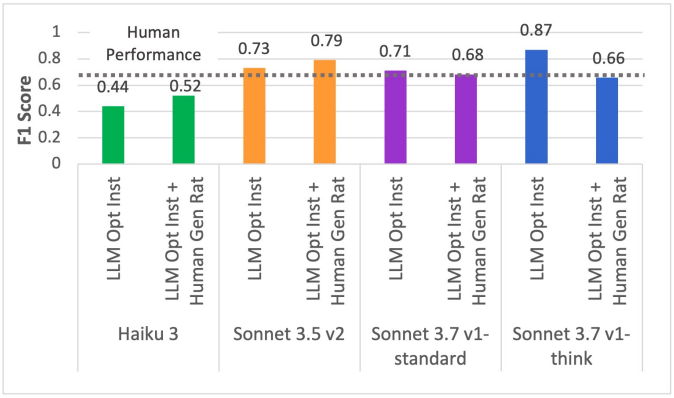

Fig. 2: Performance comparison between LLM-optimized instruction alone versus LLM-optimized instruction supplemented with human-generated rationales (10-shot prompting). Note: LLM Opt Inst = LLM-optimized Instruction, Human Gen Rat = Human-generated Rationales.

e) Many-Shot Demonstrations: Scaling the number of demonstrations up to 300 further boosted model performance across models. Figure 4 demonstrates that at 50 demonstrations, all models, including Haiku 3 (the smallest model tested), exceeded the human benchmark.

Performance improved as demonstrations increased from 3 to 300, with model-specific patterns. Haiku 3's F1 scores monotonically increased from 0.33 to 0.83, showing the gains throughout the scaling range. The Sonnet family demonstrated higher performance but with diminishing returns beyond 50-100 examples. Sonnet 3.7-standard reached 0.95 with 300 demonstrations, while Sonnet 3.7-think achieved 0.97 at 200 demonstrations. With 10,000 tokens already devoted to reasoning processes, the think variant reached its context limit at 200 demonstrations, preventing tests at 300 demonstrations—a constraint that motivates our exploration of adaptive budgeting in Section VI-B.

Results indicate that scaling benefits vary with model capacity. Smaller models continue to benefit from additional examples beyond where larger models plateau, suggesting n-shot prompting approaches should be tailored to specific model architectures. See Appendix D for full results.

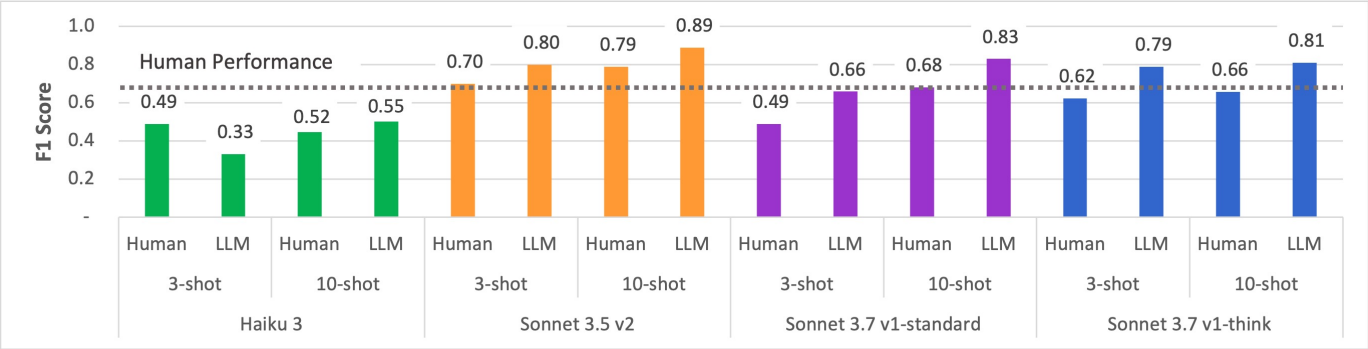

Fig. 3: Performance comparison between human-generated and LLM-generated rationales across different models (few-shot prompting)

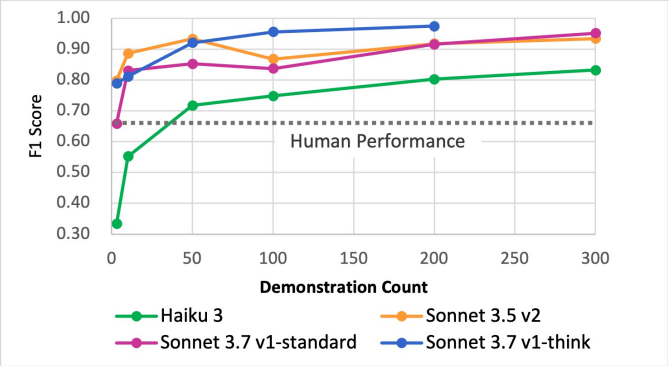

Fig. 4: Performance of Many-Shot Demonstrations

V. Discussion

A. Human vs. Automated Instruction Optimization

Our experiment comparing instruction optimization strategies revealed differences between manual and automated approaches. Through manual optimization, we expanded a one-line prompt to approximately 10 lines by incorporating guidelines and reasoning steps documented during calibration sessions, which yielded performance improvements across all models. Haiku3 and Sonnet 3.5 showed modest F1 improvements (from 0.33 to 0.39 and 0.29 to 0.35 respectively), while Sonnet 3.7 variants demonstrated more substantial gains (standard mode: from 0.62 to 0.71, think mode: from 0.76 to 0.85). These results confirm that extended instructions enhance model performance, though the improvement varies according to model capabilities.

In contrast, the MIPRO automated optimization technique transformed the original one-line prompt into a detailed set of instructions spanning approximately 40 lines. This approach systematically captured reasoning patterns and addressed edge cases that might not have been identified through manual refinement alone. With automated optimization, Sonnet 3.5 demonstrated the largest F1 improvement of 0.44, while Sonnet 3.7-standard improved by 0.24 and Sonnet 3.7-think by 0.02, highlighting varying responsiveness to instruction optimization across model architectures. Notably, after automated optimization, all Sonnet family models exceeded the human benchmark of 0.68. Results indicate that enhanced instruction detail contributes to performance gains, with effects contingent on model architecture and baseline capabilities. See Appendix A for examples of manual and automated optimized prompts.

B. Human-Generated vs. LLM-Generated Rationales

Incorporating rationales into optimized instructions consistently enhanced model performance across experiments. While human-generated rationales showed variable impact—Haiku 3 gained 18% in F1 score but Sonnet 3.7 v1-think declined by 24% in the 10-shot setting (Figure 2). Whereas LLM-generated rationales delivered superior and more reliable improvements (Figure 3). The performance advantage was substantial: Sonnet 3.7 v1-standard achieved a 34% improvement with automated optimization compared to only 15% with human-optimized prompts (see Appendix C for the full results).

The success of LLM-generated rationales stems from their systematic framework combining component-level assessment, necessity evaluation, and counterfactual reasoning—a structure particularly effective for resolving edge cases and ambiguous relationships. These results challenge conventional wisdom about human expertise as the upper bound for instructional quality, demonstrating that algorithmic reasoning patterns outperform human explanations in guiding complex classification decisions.

C. Zero-shot, Few-shot, vs. Many-shot Demonstrations

The progression from zero-shot to many-shot demonstration settings reveals a clear performance pattern. Zero-shot prompting with baseline instructions established a performance baseline, while few-shot demonstrations incorporating either human or LLM rationales yielded measurable improvements (Figure 4). The most substantial gains emerged in many-shot settings, where scaling to 300 examples enabled all tested models to surpass the human F1-score benchmark of 0.68. Notably, Sonnet 3.7 v1-think reached a score of 0.97, approaching near-perfect classification accuracy.

Exposure to diverse reasoning patterns through additional examples enhanced models' ability to resolve ambiguities and generalize to unseen cases. As the number of demonstrations increased, the performance gap between models narrowed,

indicating that lower-capacity models derive proportionally greater benefits from additional examples. While these results establish a new ceiling for concept linkage classification, they present a practical trade-off: many-shot approaches deliver superior performance but require longer prompts, increasing latency and inference costs. These findings suggest that n-shot learning strategies should be calibrated to balance model capabilities against computational constraints in deployment scenarios.

D. Human-in-the-Loop Validation and Operationalization Strategy

Our LLM approach significantly outperformed the human benchmark (F1 score of 0.97 compared to 0.68). This performance gap raises the question: When LLM predictions diverge from post-calibrated human ground truth annotations, which source is more reliable? We hypothesized that the LLM might be identifying oversights in the post-calibrated human annotations rather than making errors. This section evaluates this hypothesis through systematic analysis of discrepancies between LLM and human annotations. We also consider the implications for prioritizing expert review in real-world scenarios, where full manual verification becomes infeasible as taxonomies scale in size.

To investigate these discrepancies systematically, we analyzed instances where the LLM and post-calibrated ground truth disagreed. Table 5 presents the distribution of these disagreements. We identified 9 cases where the LLM classified concepts as *Required* ("Always Necessary") while human annotations marked them *Not Required*, and 7 cases where the LLM classified concepts as *Not Required* ("Usually Necessary") while human annotations marked them *Required*.

			Human Results	
			Required	Not Required
LLM Results	Required	Always Necessary	53	9
	Not Required	Usually Necessary	7	65
		Often Necessary	0	85
		Sometimes Necessary	0	75
		Not Necessary	0	30

Fig. 5: Confusion Matrix: LLM vs Human Results

Human annotators from the original calibration sessions reviewed these 16 cases alongside the LLM's detailed reasoning. This review process revealed that in all examined cases, the LLM classifications were superior—all 9 cases were confirmed to be false negatives and all 7 cases were false positives in the human annotations. These findings confirmed our hypothesis that the LLM had identified legitimate oversights in the human annotation process.

The inclusion of LLM-generated rationales improved review efficiency. Whereas the initial calibration sessions described in Section III-A2 required four-minute discussions to reach consensus, annotators reached immediate agreement after reading the LLM rationales, reducing the review time to about one minute per instance—primarily spent on reading.

These results have implications for ontology construction and maintenance. First, they demonstrate that even carefully calibrated human annotations can contain errors that LLMs with appropriate prompt engineering can detect. Second, they highlight the value of an iterative human-in-the-loop process where LLM outputs refine both model performance and training data quality.

For organizations maintaining ontologies at scale, our findings suggest an optimal operational strategy: combine automated LLM processing with targeted expert oversight. High-confidence LLM predictions can be accepted automatically, while ambiguous cases are flagged for expert review. This approach creates an efficient feedback loop that enhances both accuracy and scalability while deploying human expertise where it adds the most value.

E. Practical Application

Organizations develop taxonomies across functions that often evolve into silos impeding organizational effectiveness. In workforce management, these disconnected knowledge structures contribute to skills gaps and talent utilization challenges—with the World Economic Forum estimating over 100 million people across 18 economies are underutilizing their existing skills (37), thus hindering innovation and industry transformation.

To address this challenge, we developed a unified ontology by applying our LLM-based framework with custom SKOS extensions (38) that implement specialized mapping properties (*myskos:isRequiredFor* and *myskos:isNotRequiredFor*) to define prerequisite relationships between skills. By linking previously disparate taxonomies, we've created an integrated skills library that enables organizations to effectively upskill and reskill their workforce. This comprehensive solution powers talent management products while providing a knowledge base for LLM models that support learning initiatives, performance management, and talent assessment across organizations.

VI. Future Work

A. Dynamic Example Selection

Although many-shot demonstrations significantly improved performance, they introduced computational overhead by requiring models to process hundreds of examples per inference. Future work should explore dynamic example selection strategies that identify relevant demonstrations based on semantic similarity to the query. This approach may reduce the number of examples needed from hundreds to dozens while maintaining performance levels. Such optimization could decrease token consumption, latency, and inference costs, making LLM deployment more practical in resource-constrained environments. Moreover, the use of LLMs as judges could be explored to evaluate and arbitrate annotation disagreements, a method investigated by (39; 40), as a potential scalable alternative to traditional human-centric evaluation. This approach may offer efficiency gains, though careful evaluation would be needed to

ensure it can capture the subtle, context-specific nuances often found in qualitative data (41).

B. Adaptive Reasoning Token Budgeting

While our current framework utilizes fixed reasoning tokens of 10,000, recent research suggests that dynamically adjusting token budgets based on task complexity can enhance efficiency and performance. Han and Johnson (2024) introduced TALE, a token-budget-aware LLM reasoning framework that estimates optimal token allocations for varying problem types, reducing token consumption while maintaining accuracy (42). Similarly, Lee and Chen (2025) investigated the trade-off between reasoning length and accuracy, demonstrating that each task exhibits a 'token complexity'—the minimum number of tokens necessary for successful problem resolution (43). Incorporating adaptive token budgeting into our framework could optimize resource utilization and improve scalability, particularly for large-scale ontology alignment challenges.

VII. Conclusion

We proposed a novel framework leveraging large language models to automate ontology alignment tasks, significantly outperforming traditional manual approaches. By combining expert calibration, systematic prompt optimization, and human-in-the-loop validation, our method achieved classification performance well beyond human benchmarks while reducing the time and effort required for concept mapping. Experimental results demonstrated that automated prompt engineering techniques, LLM-generated rationales, and many-shot demonstrations all contributed to substantial performance improvements across model architectures.

Our findings reveal that LLM-generated rationales consistently outperform human-authored explanations. While human experts achieved only 22% unanimous agreement and an F1-score of 0.68, our LLM approach reached an impressive 0.97 F1-score. This performance advantage was substantial: Sonnet 3.7 v1-standard achieved a 34% improvement with automated optimization compared to only 15% with human-optimized prompts. The success of LLM-generated rationales stems from their systematic framework combining component-level assessment, necessity evaluation, and counterfactual reasoning.

The human-in-the-loop validation process further demonstrated the model's effectiveness. When reviewing disagreements between LLM and human annotations, experts confirmed that human annotators had overlooked linkages in all examined cases (9 false negatives and 7 false positives), highlighting the value of an iterative refinement process. These findings show that mapping relationships between taxonomies connects isolated systems, unifying fragmented information. The proposed framework strikes an effective balance between automation and expert oversight—allowing high-confidence cases to be processed automatically while flagging ambiguous instances for human review. This approach creates a scalable, consistent system that transforms taxonomy alignment from a labor-intensive process into an automated workflow with better quality, enabling organizations to build practical applications atop unified knowledge structures.

Acknowledgments

We extend our sincere gratitude to Jason VanDuine, Henry Zhu, and Hideo Kobayashi for their insightful feedback and support that strengthened this work.

References

[1] M. A. Osman, S. A. M. Noah, and S. Saad, "Ontology-based knowledge management tools for knowledge sharing in organization—a review," *IEEE access*, vol. 10, pp. 43 267–43 283, 2022.

[2] B. Abu-Salih, M. Al-Qurishi, M. Alweshah, M. Al-Smadi, R. Alfayez, and H. Saadeh, "Healthcare knowledge graph construction: State-of-the-art, open issues, and opportunities," *arXiv preprint arXiv:2207.03771*, 2022.

[3] M. M. Marques, A. J. Wright, E. Corker, M. Johnston, R. West, J. Hastings, L. Zhang, and S. Michie, "The behaviour change technique ontology: transforming the behaviour change technique taxonomy v1," *Wellcome open research*, vol. 8, p. 308, 2024.

[4] T. Syyrilä, S. Koskiniemi, E. Manias, and M. Härkänen, "Taxonomy development methods sciences–a systematic review," *International Journal of Medical Informatics*, p. 105438, 2024.

[5] J. Shi, J. Chen, H. Dong, I. Khan, L. Liang, Q. Zhou, Z. Wu, and I. Horrocks, "Subsumption prediction for e-commerce taxonomies," pp. 244–261, 2023.

[6] M. A. AlAfnan, "Taxonomy of educational objectives: Teaching, learning, and assessing in the information and artificial intelligence era," *Journal of Curriculum and Teaching*, vol. 13, no. 4, pp. 173–191, 2024.

[7] J. Euzenat, M.-E. Roşoiu, and C. Trojahn, *Ontology matching benchmarks: generation, stability, and discriminability*. Elsevier, 2013, vol. 21.

[8] N. F. Noy and M. A. Musen, "The prompt suite: interactive tools for ontology merging and mapping," *International journal of human-computer studies*, vol. 59, no. 6, pp. 983–1024, 2003.

[9] F. Mustafa and F. Dine, "Deep learning for knowledge representation: Automating semantic analysis and ontology construction," 2025.

[10] A. Doan, J. Madhavan, R. Dhamankar, P. Domingos, and A. Halevy, "Learning to match ontologies on the semantic web," *The VLDB journal*, vol. 12, pp. 303–319, 2003.

[11] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, 2013.

[12] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," pp. 1532–1543, 2014.

[13] Z. Ye, Y. J. Kumar, G. O. Sing, F. Song, and J. Wang, "A comprehensive survey of graph neural networks for

knowledge graphs," *IEEE Access*, vol. 10, pp. 75 729–75 741, 2022.

[14] X. Li, H. Xiong, X. Li, X. Wu, X. Zhang, J. Liu, J. Bian, and D. Dou, "Interpretable deep learning: Interpretation, interpretability, trustworthiness, and beyond," *Knowledge and Information Systems*, vol. 64, no. 12, pp. 3197–3234, 2022.

[15] F. Neuhaus, "Ontologies in the era of large language models–a perspective," *Applied ontology*, vol. 18, no. 4, pp. 399–407, 2023.

[16] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," vol. 33, 2020, pp. 1877–1901.

[17] H. Babaei Giglou, J. D'Souza, and S. Auer, "Llms4ol: Large language models for ontology learning," pp. 408–427, 2023.

[18] M. Val-Calvo, M. E. Aranguren, J. Mulero-Hernández, G. Almagro-Hernández, P. Deshmukh, J. A. Bernabé-Díaz, P. Espinoza-Arias, J. L. Sánchez-Fernández, J. Mueller, and J. T. Fernández-Breis, "Ontogenix: Leveraging large language models for enhanced ontology engineering from datasets," *Information Processing & Management*, vol. 62, no. 3, p. 104042, 2025.

[19] N. F. Rajani, B. McCann, C. Xiong, and R. Socher, "Explain yourself! leveraging language models for commonsense reasoning," 2019.

[20] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.

[21] A. Creswell, M. Shanahan, and I. Higgins, "Selection-inference: Exploiting large language models for interpretable logical reasoning," *arXiv preprint arXiv:2205.09712*, 2022.

[22] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," vol. 35, 2022, pp. 22 199–22 213.

[23] T. Shin, Y. Razeghi, R. L. Logan IV, E. Wallace, and S. Singh, "Autoprompt: Eliciting knowledge from language models with automatically generated prompts," *arXiv preprint arXiv:2010.15980*, 2020.

[24] Y. Zhou, A. I. Muresanu, Z. Han, K. Paster, S. Pitis, H. Chan, and J. Ba, "Large language models are human-level prompt engineers," 2022.

[25] C. Yang, X. Wang, Y. Lu, H. Liu, Q. V. Le, D. Zhou, and X. Chen, "Large language models as optimizers," *arXiv preprint arXiv:2309.03409*, 2023.

[26] Q. Guo12, R. Wang, J. Guo, B. Li23, K. Song, X. Tan, G. Liu, J. Bian, and Y. Yang, "Connecting large language models with evo-lutionary algorithms yields powerful prompt optimizers."

[27] K. Opsahl-Ong, M. J. Ryan, J. Purtell, D. Broman, C. Potts, M. Zaharia, and O. Khattab, "Optimizing instructions and demonstrations for multi-stage language model programs," *arXiv preprint arXiv:2406.11695*, 2024.

[28] H. Raj, V. Gupta, D. Rosati, and S. Majumdar, "Improving consistency in large language models through chain of guidance," *arXiv preprint arXiv:2502.15924*, 2025.

[29] Z. Wang, Y. Jiang, Y. Lu, P. He, W. Chen, Z. Wang, M. Zhou *et al.*, "In-context learning unlocked for diffusion models," *Advances in Neural Information Processing Systems*, vol. 36, pp. 8542–8562, 2023.

[30] F. Sebastiani, "Machine learning in automated text categorization," pp. 1–47, 2002.

[31] S. I. Wang and C. D. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2012, pp. 90–94.

[32] Y. Qiu and Y. Jin, "Chatgpt and finetuned bert: A comparative study for developing intelligent design support systems," *Intelligent systems with applications*, vol. 21, p. 200308, 2024.

[33] N. Chacko and V. Chacko, "Paradigm shift presented by large language models (llm) in deep learning," *Advances in Emerging Computing Technologies*, vol. 40, 2023.

[34] M. Suzgun, N. Scales, N. Schärli, S. Gehrmann, Y. Tay, H. W. Chung, A. Chowdhery, Q. V. Le, E. H. Chi, D. Zhou *et al.*, "Challenging big-bench tasks and whether chain-of-thought can solve them," *arXiv preprint arXiv:2210.09261*, 2022.

[35] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou, "Self-consistency improves chain of thought reasoning in language models," *arXiv preprint arXiv:2203.11171*, 2022.

[36] A. Madaan, S. Zhou, U. Alon, Y. Yang, and G. Neubig, "Language models of code are few-shot commonsense learners," 2022.

[37] A. Di Battista, S. Grayling, E. Hasselaar, T. Leopold, R. Li, M. Rayner, and S. Zahidi, "Future of jobs report 2023," in *World Economic Forum, Geneva, Switzerland. https://www. weforum. org/reports/the-future-of-jobs-report-2023*, 2023.

[38] A. Miles and S. Bechhofer, "Skos simple knowledge organization system reference," 2009.

[39] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing *et al.*, "Judging llm-as-a-judge with mt-bench and chatbot arena," vol. 36, 2023, pp. 46 595–46 623.

[40] R. Bedemariam, N. Perez, S. Bhaduri, S. Kapoor, A. Gil, E. Conjar, I. Itoku, D. Theil, A. Chadha, and N. Nayyar, "Potential and perils of large language models as judges of unstructured textual data," *arXiv preprint arXiv:2501.08167*, 2025.

[41] S. Bhaduri, S. Kapoor, A. Gil, A. Mittal, and R. Mulkar, "Reconciling methodological paradigms: Employing large language models as novice qualitative research assistants in talent management research," *arXiv preprint arXiv:2408.11043*, 2024.

[42] T. Han, Z. Wang, C. Fang, S. Zhao, S. Ma, and

Z. Chen, "Token-budget-aware llm reasoning," *arXiv preprint arXiv:2412.18547*, 2024.

[43] A. Lee, E. Che, and T. Peng, "How well do llms compress their own chain-of-thought? a token complexity approach," *arXiv preprint arXiv:2503.01141*, 2025.

APPENDIX A
HUMAN AND LLM OPTIMIZED INSTRUCTIONS

A comparison of three instruction approaches shows progression from a simple baseline instruction to a human-optimized version with structured evaluation steps, culminating in an LLM-optimized instruction that provides detailed guidance on competency analysis, reasoning steps, necessity ratings, and final determination criteria.

TABLE II: Comparison of Instruction Types

Simple Instruction	**Human Optimized Instruction**	**LLM Optimized Instruction**
Identify if a competency is ‘Required‘ or ‘Not Required‘ to complete the responsibility. Output just ‘Required‘ or ‘Not Required‘	As an ontological domain expert specializing in conceptual modeling, your role is to determine whether a specific concept is essential within a particular domain context. 1. Analyze the Relationship: Evaluate whether Concept A is essential to the definition or realization of Concept B. Provide your reasoning in the rationale section, ensuring your assessment strictly aligns with the conceptual definitions provided. 2. Rate Necessity: Using the rationale, rate the competency’s necessity for performing the responsibility according to the Likert scale below: - Always Necessary: Essential and required in all circumstances. - Usually Necessary: Very important, with rare exceptions. - Often Necessary: Frequently helpful but not critical. - Sometimes Necessary: Occasionally useful. - Not Necessary: Never required. 3. Determine Requirement: Based on your rationale and Likert scale analysis, identify whether the competency is Required or Not Required for fulfilling the responsibility. Output only ‘Required‘ or ‘Not Required‘.	You are an experienced ontological domain expert specializing in conceptual relationship analysis and knowledge mapping. Your task is to evaluate whether specific concepts are essential to other concepts by following this systematic approach: 1. First, carefully analyze: - Concept A and its definition - Concept B and its definition - How they fundamentally relate to each other 2. Provide detailed step-by-step reasoning that: - Breaks down the key components of Concept B - Examines how Concept A relates to these components - Considers if Concept B can be defined or realized without Concept A - Evaluates the impact of removing Concept A from the relationship 3. Rate the essentiality of Concept A to Concept B using this Likert scale: - Always Necessary: Essential and required in all circumstances - Usually Necessary: Very important with rare exceptions - Often Necessary: Frequently helpful but not critical - Sometimes Necessary: Occasionally useful - Not Necessary: Never required 4. Make a final determination: - Required: Concept B CANNOT be defined or realized without Concept A under ANY circumstances - Not Required: Concept A is helpful but not absolutely essential to Concept B Important: Focus on whether Concept B can exist or be meaningfully defined at all without Concept A, not just whether it would be fully developed. A concept should only be marked as "Required" if its absence makes the other concept impossible to define or realize, not just less complete. Provide your analysis, Likert rating, and final determination with clear supporting rationale.

APPENDIX B
HUMAN AND LLM GENERATED RATIONALES

This table presents a comparison between two types of rationales—Human-authored and LLM-generated—that evaluate whether 'Verbal Communication' is a required competency for effectively mentoring another team member.

TABLE III: Comparison of Human and LLM Rationales

Human Rationale	**LLM Rationale**
Mentoring another team member requires articulating expectations, sharing knowledge, and providing constructive feedback through effective verbal communication.	1. Let me analyze this case step by step: - Break down the key components of mentorship: - Knowledge transfer - Skill development - Performance feedback - Career guidance - Relationship building 2. Analyze the verbal communication requirements: - Explaining complex concepts clearly - Articulating expectations and goals - Delivering constructive feedback - Asking effective questions - Active listening and responding 3. Evaluate the connection: - Cannot transfer knowledge without clear verbal explanations - Feedback delivery requires nuanced verbal communication - Building mentor-mentee trust depends on verbal rapport - Guidance becomes ineffective without articulate expression - Misunderstandings multiply without verbal clarity 4. Consider if mentorship could succeed without verbal communication: - Written communication alone lacks immediate clarification - Non-verbal cues insufficient for complex knowledge transfer - Time-sensitive feedback requires verbal exchange - Relationship development hindered by lack of verbal interaction - Impact assessment: 5. Failure in verbal communication would result in: - Incomplete knowledge transfer - Misinterpreted feedback - Diminished mentee engagement - Ineffective skill development - Damaged mentor-mentee relationship Answer: Required

APPENDIX C
PERFORMANCE COMPARISON OF DIFFERENT INSTRUCTIONS

We evaluated various instruction strategies across multiple Anthropic models—Claude 3.7 Sonnet v1, Claude 3.5 Sonnet v2, and Claude 3 Haiku v1—comparing their performance using standard metrics: Precision, Recall, F1-score, and Accuracy. For Claude 3.7 Sonnet specifically, our analysis included both standard inference operation and enhanced thinking mode with 10,000 reasoning tokens. All metrics reported are weighted averages across classes, where weights are proportional to class frequencies in the dataset.

TABLE IV: Performance Comparison of Different Instructions (Weighted Average Metrics: Prec = Precision, Rec = Recall, Acc = Accuracy)

	Haiku 3 v1				**Sonnet 3.5 v2**				**Sonnet 3.7 v1-standard**				**Sonnet 3.7 v1-think**			
Experiment	**Prec**	**Rec**	**F1**	**Acc**	**Prec**	**Rec**	**F1**	**Acc**	**Prec**	**Rec**	**F1**	**Acc**	**Prec**	**Rec**	**F1**	**Acc**
Simple Instruction (Baseline)	0.76	0.42	0.33	0.42	0.78	0.19	0.29	0.19	0.47	0.94	0.62	0.60	0.67	0.88	0.76	0.78
Human Optimized Instruction	0.59	0.54	0.39	0.42	0.66	0.42	0.35	0.42	0.67	0.77	0.71	0.80	0.78	0.88	0.85	0.85
LLM Optimized Instruction	0.78	0.49	0.44	0.49	0.73	0.74	0.73	0.74	0.86	0.86	0.86	0.86	0.86	0.89	0.87	0.87

APPENDIX D
PERFORMANCE COMPARISON OF HUMAN AND LLM RATIONALES ACROSS DIFFERENT DEMONSTRATION SIZES

We compared the performance metrics (Precision, Recall, F1-score, and Accuracy) between human-generated and LLM-generated rationales across different demonstration sizes (3 to 300 examples) using four LLM models (Anthropic models—Claude 3.7 Sonnet v1, Claude 3.5 Sonnet v2, and Claude 3 Haiku v1). For Claude 3.7 Sonnet specifically, our analysis included both standard inference operation and enhanced thinking mode with 10,000 reasoning tokens. All metrics reported are weighted averages across classes, where weights are proportional to class frequencies in the dataset. Our results showed that LLM-generated rationales generally outperform human rationales and improve with more demonstrations, particularly for Claude Sonnet 3.7 v1.

TABLE V: Performance Comparison of Human and LLM Rationales Using Claude Haiku 3

	Precision		**Recall**		**F1-score**		**Accuracy**	
Demonstration Count	**Human**	**LLM**	**Human**	**LLM**	**Human**	**LLM**	**Human**	**LLM**
Few-shot (3)	0.75	0.82	0.53	0.38	0.49	0.33	0.53	0.38
Few-shot (10)	0.74	0.81	0.54	0.55	0.52	0.55	0.54	0.55
Many-shot (50)	0.60	0.80	0.59	0.70	0.59	0.72	0.59	0.70
Many-shot (100)	0.60	0.83	0.65	0.73	0.53	0.75	0.65	0.73
Many-shot (200)	-	0.84	-	0.79	-	0.80	-	0.79
Many-shot (300)	-	0.86	-	0.82	-	0.83	-	0.82

TABLE VI: Performance Comparison of Human and LLM Rationales Using Claude Sonnet 3.5 v2

	Precision		**Recall**		**F1-score**		**Accuracy**	
Demonstration Count	**Human**	**LLM**	**Human**	**LLM**	**Human**	**LLM**	**Human**	**LLM**
Few-shot (3)	0.74	0.88	0.69	0.79	0.70	0.80	0.69	0.79
Few-shot (10)	0.79	0.91	0.79	0.88	0.79	0.89	0.79	0.88
Many-shot (50)	0.77	0.94	0.78	0.93	0.77	0.93	0.78	0.93
Many-shot (100)	0.79	0.89	0.76	0.86	0.76	0.87	0.76	0.86
Many-shot (200)	-	0.92	-	0.92	-	0.92	-	0.92
Many-shot (300)	-	0.93	-	0.93	-	0.93	-	0.93

TABLE VII: Performance Comparison of Human and LLM Rationales Using Claude Sonnet 3.7 v1-standard

	Precision		**Recall**		**F1-score**		**Accuracy**	
Demonstration Count	**Human**	**LLM**	**Human**	**LLM**	**Human**	**LLM**	**Human**	**LLM**
Few-shot (3)	0.48	0.62	0.54	0.70	0.49	0.66	0.54	0.70
Few-shot (10)	0.76	0.83	0.62	0.83	0.68	0.83	0.62	0.83
Many-shot (50)	0.77	0.90	0.66	0.85	0.71	0.85	0.66	0.85
Many-shot (100)	0.81	0.89	0.73	0.83	0.77	0.84	0.73	0.83
Many-shot (200)	-	0.93	-	0.91	-	0.92	-	0.91
Many-shot (300)	-	0.96	-	0.95	-	0.95	-	0.95

TABLE VIII: Performance Comparison of Human and LLM Rationales Using Claude Sonnet 3.7 v1-think

	Precision		**Recall**		**F1-score**		**Accuracy**	
Demonstration Count	**Human**	**LLM**	**Human**	**LLM**	**Human**	**LLM**	**Human**	**LLM**
Few-shot (3)	0.72	0.81	0.65	0.81	0.62	0.79	0.65	0.80
Few-shot (10)	0.70	0.81	0.67	0.81	0.66	0.81	0.67	0.81
Many-shot (50)	0.70	0.93	0.67	0.92	0.66	0.92	0.67	0.92
Many-shot (100)	0.74	0.96	0.74	0.96	0.74	0.96	0.74	0.96
Many-shot (200)	-	0.97	-	0.96	-	0.97	-	0.97
Many-shot (300)	-	-	-		-		-	-