Towards Scalable Domain-Specific Document Annotation:
A Semantic Archetype-Driven Framework

Moritz Steigerwald

October 16th 2025, Final Presentation Master's Thesis

Chair of Software Engineering for Business Information Systems (sebis)
Department of Computer Science
School of Computation, Information and Technology (CIT)
Technical University of Munich (TUM)
wwwmatthes.in.tum.de

Outline

Introduction & Motivation

Research Questions

Methodology

Results

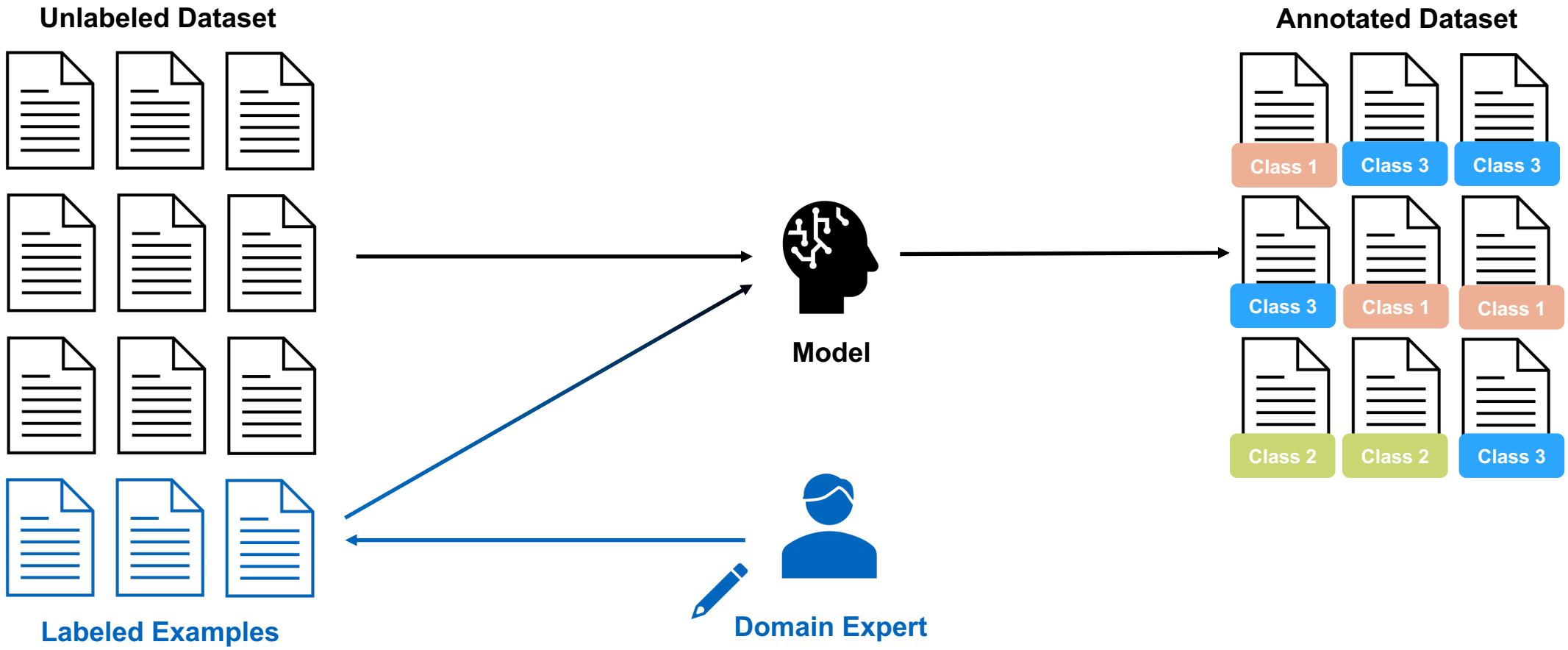
Challenges and Future Work

Conclusion

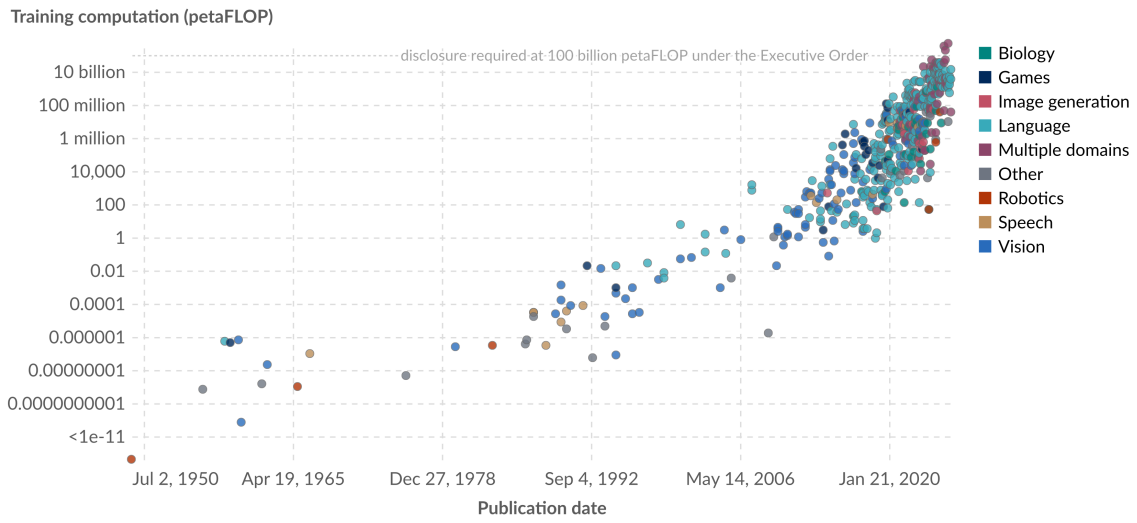
Project Context

CreateData4AI: Leveraging Domain Knowledge and Context Rules to Transform Large-Scale Unstructured Text Corpora into Structured and Annotated Datasets

Introduction & Motivation



Motivation: “Labels are the bottleneck”



Training compute for notable AI models has grown exponentially for over a decade (data: Epoch AI via Our World in Data).

Current Limitations:

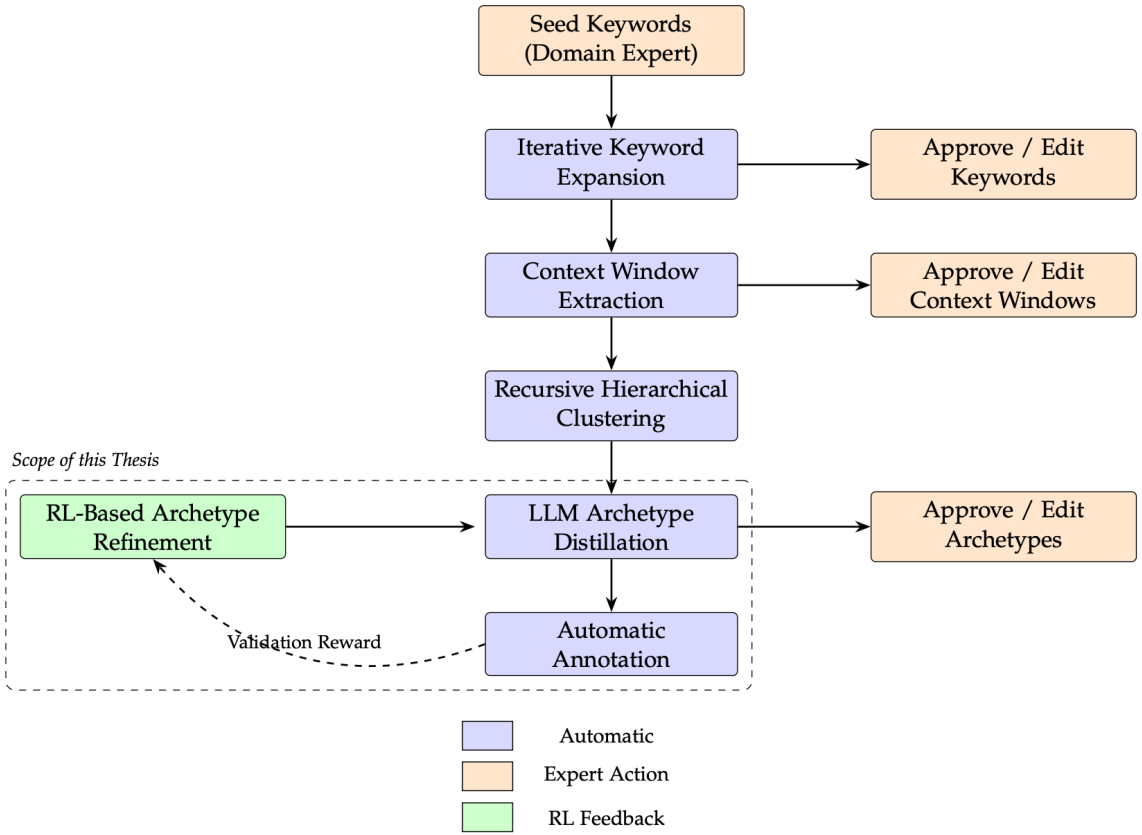
Compute scales exponentially → largest training runs doubled every ~3.4 months since 2012

Human labeling doesn’t → it’s the deployment bottleneck: slow, costly, hard to parallelize.

Quality is inconsistent → domain tasks show high inter/intra-annotator variance; experts are scarce.

LLM auto-labeling isn’t free → prone to hallucinations; eval still needs ground truth; API cost/latency.

The CD4AI Pipeline and Thesis Scope



We have a partially annotated corpus* with:

1. **Keywords**,
2. **Context Windows**,
3. **Semantic Archetypes.**

But the rest of the corpus is still unlabeled!

1. We **need** a **robust classifier** to
   (a) use these archetypes and
   (b) systematically annotate all remaining data.

2. We also need to **evaluate** how well this system performs

* This entails Distilled Archetypes, (expanded) keywords, a few gold examples and the remaining documents

Research Questions

RQ1 How can semantic archetypes be leveraged to classify and annotate domain-specific documents?

RQ2 Can a reward-based feedback system boost both classification accuracy and archetype quality?

RQ3 How does this archetype-driven framework compare to supervised models and zero-shot LLMs in terms of accuracy and resource utilization across different domains?

Methodology

Input: Partially Labeled (Data) Sets* & Archetypes

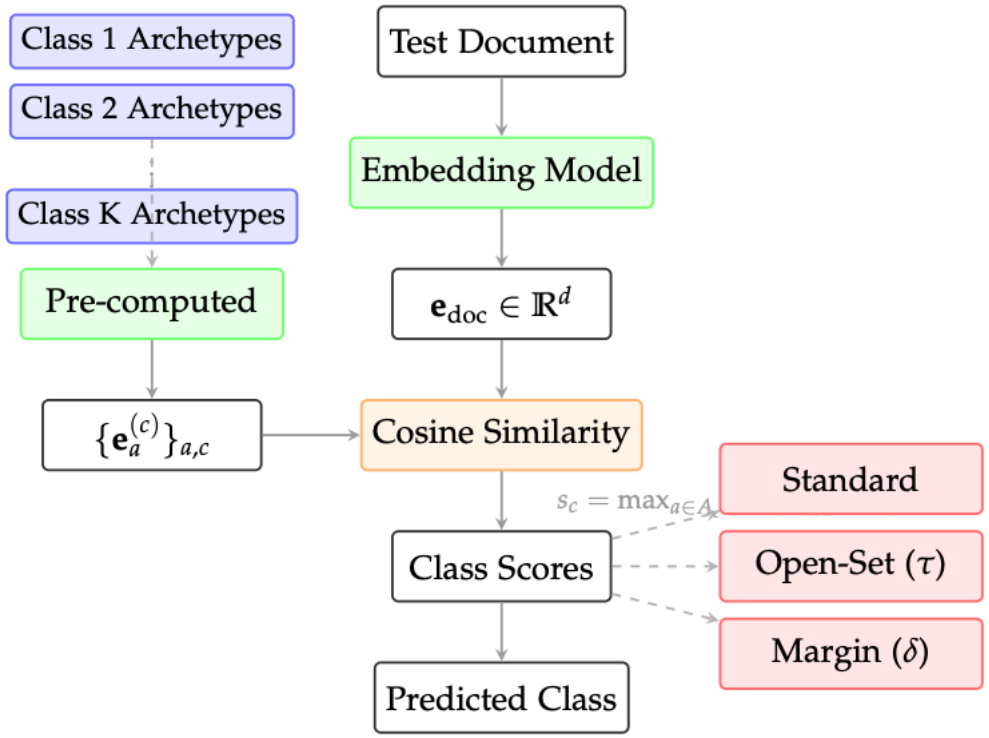
- from Domain Experts
- Used to guide our classifier(s)

For Building a Classifier 4 Methods are used:

1. Embedding Based Similarity
2. Contrastive Learning for Domain-Specific Embeddings
3. Weak Supervision via Progressive Pseudo-Labeling
4. GRPO – Reinforcement Learning for Archetype Selection

* This entails (expanded) keywords, a few gold examples and the remaining documents

Methodology - Embedding Based Similarity

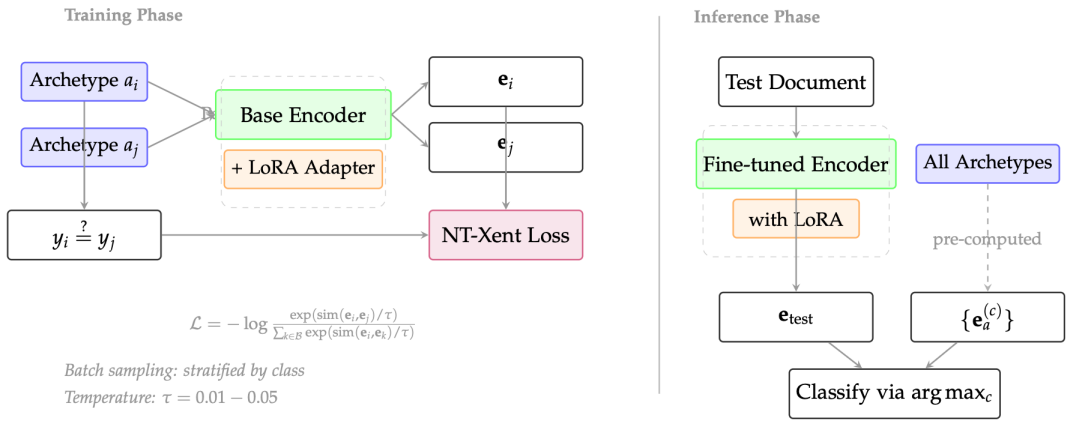


What it does: Classifies a document by comparing its embedding to pre-computed archetype embeddings and taking the best-matching class; supports open-set threshold (τ) and margin (δ) decisions.

Intuition: If archetypes are good prototypes, the correct class should be closest in embedding space.

Why it matters: Gives a training-free, fast interpretable baseline that plugs directly into existing pipeline

Methodology – Contrastive Learning

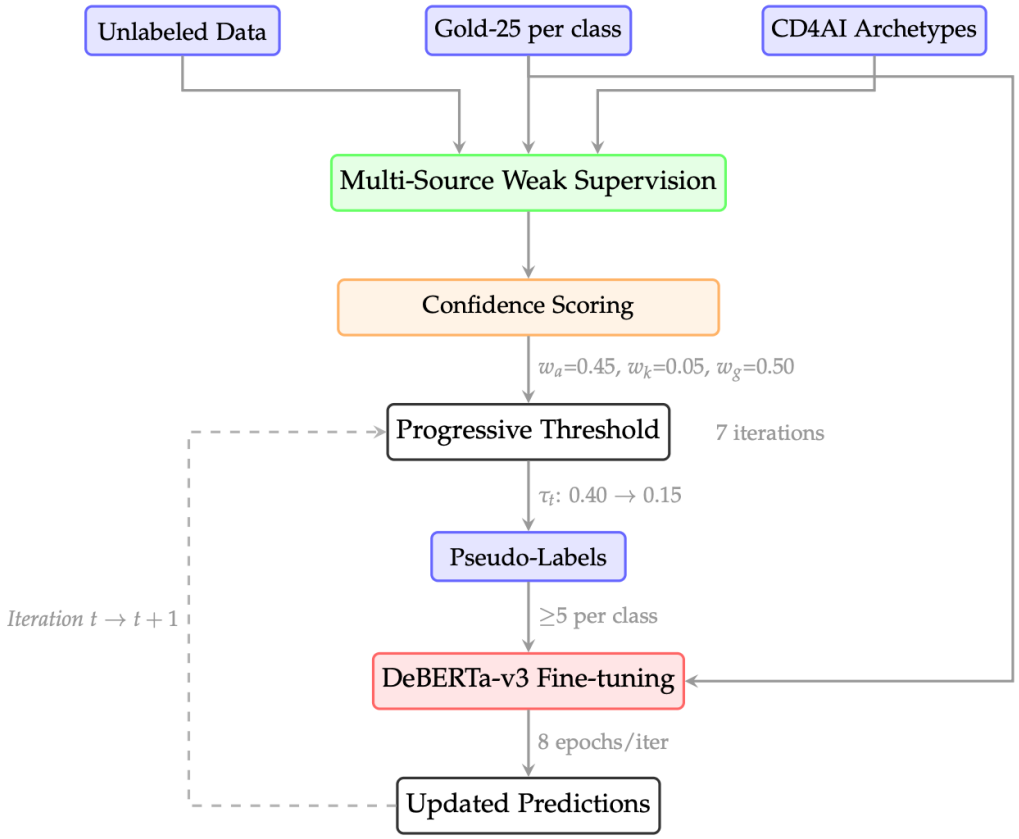


What it does: Fine-tune the encoder with a contrastive loss/NT-Xent Loss so same-class (doc/archetype) pairs pull together and different classes push apart.

Intuition: Generic embeddings capture “semantic similarity,” not our class boundaries; contrastive tuning reshapes the space for classification.

Why it matters: Produces domain-adapted representations while keeping inference simple (still nearest-archetype)

Methodology – Weak Supervision via Progressive Pseudo-Labeling

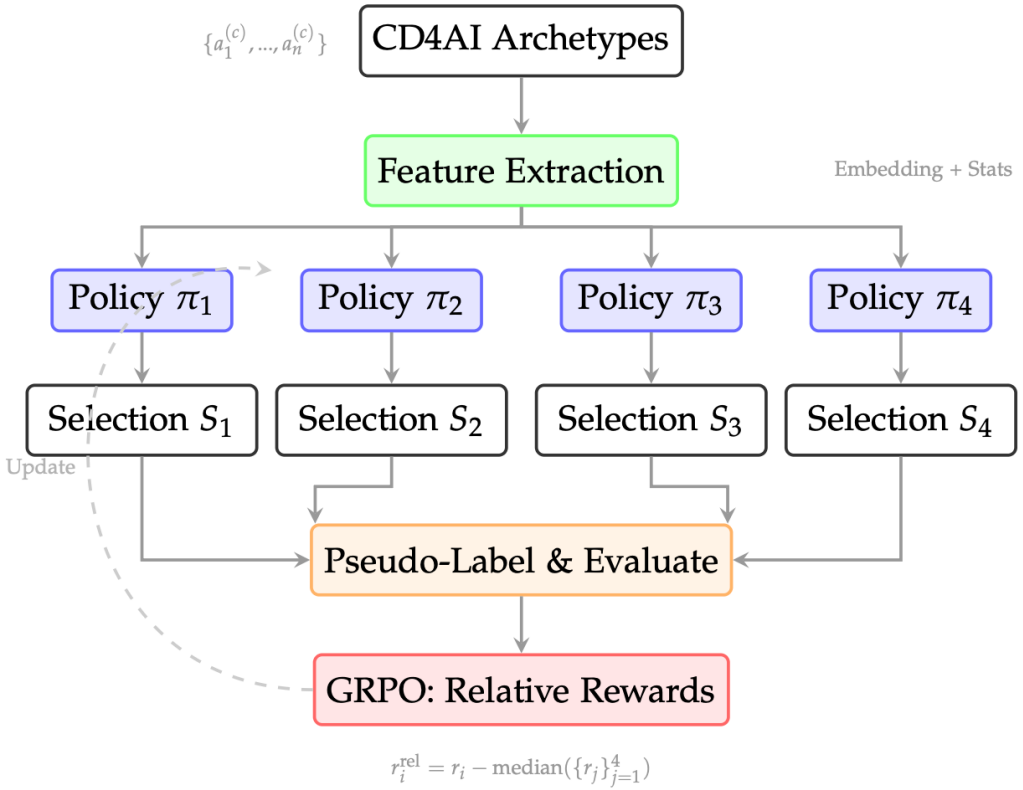


What it does: Combine three weak signals, archetype similarity, keyword matches, and similarity to a few gold examples, into a confidence score; add only high-confidence pseudo-labels each round and decay the threshold over iterations.

Intuition: Start from the most reliable hints and gradually grow the labeled set (curriculum), while down-weighting uncertain labels to avoid confirmation bias.

Why it matters: Turns a tiny seed (25/class) plus archetypes into a much larger, still-clean training set

Methodology – GRPO: Reinforcement Learning for Archetype Selection



What it does: Train multiple small policies that include/exclude archetypes; reward each by downstream validation performance relative to peers; keep the best subset.

Intuition: CD4AI over-produces archetypes; some are noisy or redundant, learn to keep the ones that actually help classification.

Why it matters: Improves supervision quality without regenerating archetypes or retraining the distillation LLM (computationally efficient way to denoise the pool)

Baselines

- **LLM prompting**
  - zero/few shot prompting with fixed prompts and examples drawn from training data
  - 4 groups:
    - High Performance proprietary: `gpt-4o, claude-3.7-sonnet`
    - High Performance open-weight: `Qwen2.5-72B-Instruct, Mixtral-8x22B-Instruct-v0.1`
    - Efficient mid-size (~7–8B) : `Llama-3.1-8B-Instruct, Mistral-7B-Instruct-v0.3`
    - Small / Tiny (~1–4B): `Phi-3.5-mini-instruct, Qwen2.5-1.5B-Instruct`

- **Supervised Encoders**
  - 3 fine tuned base encoders: `RoBERTa-base, ELECTRA-base, DeBERTa-v3-base`
  - 3 Data Regimes:
    - Archetype Only
    - Gold-25/cls
    - Full-train (upper bound)

- **Weak supervision – Label Propagation**
  - Graph/similarity propagation from the small gold set over unlabeled data (Weak Supervision SOTA)

Evaluation

Datasets

Dataset	Classes	Orig.		Our splits		
		Train	Test	Train	Dev	Test
20 Newsgroups	20	11 314	7 532	4 000	1 000	1 000
AG News	4	120 000	7 600	12 000	3 000	1 000
BBC News	5	1 225	1 000	1 225	300	700
DBpedia-14	14	560 000	70 000	14 000	3 500	1 000
arXiv	10	100 000	–	15 000	3 000	1 000

Primary Classification Metrics: **Accuracy** and **Macro-F1** (plus **Macro-Precision**/**Recall** for detail).

Efficiency metrics: **Throughput (samples/s)** and **end-to-end latency** measured on stated hardware (local models) or as request timings for API LLMs.

Hardware

Component	Specification
GPU	1 × NVIDIA RTX 5000 Ada
CPU	14 vCPU
System memory	62 GB RAM
Platform	Runpod VM instance
Precision	Mixed precision (automatic, where available)
Random seed	42 (training, data splits, and sampling)

Note: Due to hardware constraints, the two largest open-weight models: Qwen2.5-72B-Instruct (72B parameters) were accessed through OpenRouter

Results – Baselines

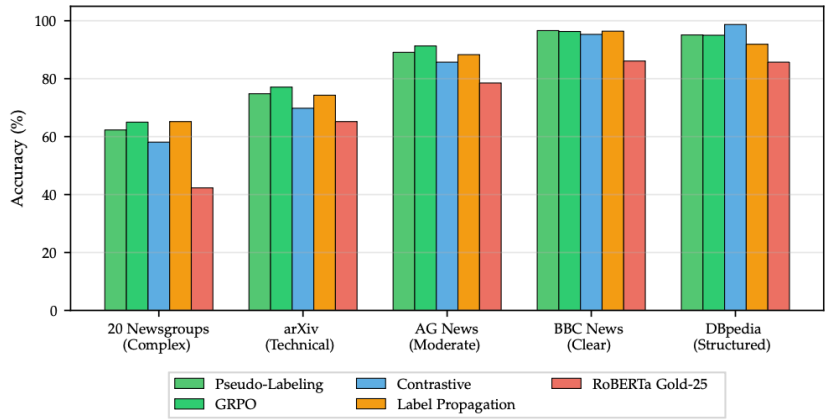
Method	Avg. Acc. (%)	Labels Required	Throughput (samples/s)	Latency (ms)	Cost (Relative)	Deploy (Ease)
Realistic Scenarios (25 labels/class)						
Label Propagation	**83.1**	25/class	6.9	191.7	Low	Medium
LLM-5shot (GPT-4o)	82.8	5 total	18.5	61.7	High	Easy
LLM-0shot (GPT-4o)	80.9	**0**	20.2	47.5	High	Easy
RoBERTa Gold-25	71.6	25/class	**400.0**	**2.5**	**Low**	**Easy**
Archetype (RoBERTa)	48.9	**0**	11.3	88.6	**Low**	Medium
Unrealistic Reference† (Full supervision)						
RoBERTa Full-train	86.4	200-3000†	400.0	2.5	Low	Easy

†: References an "Unrealistic" Scenario trained on the full training regime modelling a fully supervised training scenario.
Unrealistic meaning a scenario needing more then the available CD4AI inputs

Design targets for our methods: **≥80% accuracy** and **≥50 samples/s** (beat LP/LLM limits while staying fast).

Results – Comparison

Method	Type	Avg. Acc. (%)	Avg. F1 (%)	Throughput (samples/s)	Labels (per class)	Training (minutes)	Deploy (Ease)
Methods Meeting Design Targets (>80% acc, >50 samples/s)							
GRPO-Pseudo	CD4AI+	**84.9**	**84.6**	191.3	25	42-270	Medium
Pseudo-Labeling	CD4AI	83.5	83.2	275.6	25	7-79	Medium
Contrastive (Qwen3)	CD4AI	81.5	81.4	**1,601.6**	25	2-71	Easy
High-Accuracy But Slow Methods							
Label Propagation	Baseline	83.1	82.4	6.9	25	10-60	Medium
LLM-5shot (GPT-4o)	Baseline	82.8	82.6	18.5	**5 total**	**None**	**Easy**
LLM-0shot (GPT-4o)	Baseline	80.9	80.7	20.2	**0**	**None**	**Easy**
Fast But Lower-Accuracy Methods							
RoBERTa Gold-25	Baseline	71.6	71.2	400.0	25	0.5-2	**Easy**
Contrastive (MPNet)	CD4AI	80.3	80.2	999.4	25	2-20	Easy
Embedding (Qwen3)	CD4AI	41.5	40.8	988.1	25	**None**	**Easy**
Archetype (RoBERTa)	Baseline	48.9	43.6	11.3	**0**	5-15	Medium
Reference (Unrealistic)†							
RoBERTa Full-train	Baseline	86.4	86.2	400.0	200-3000	15-45	Easy

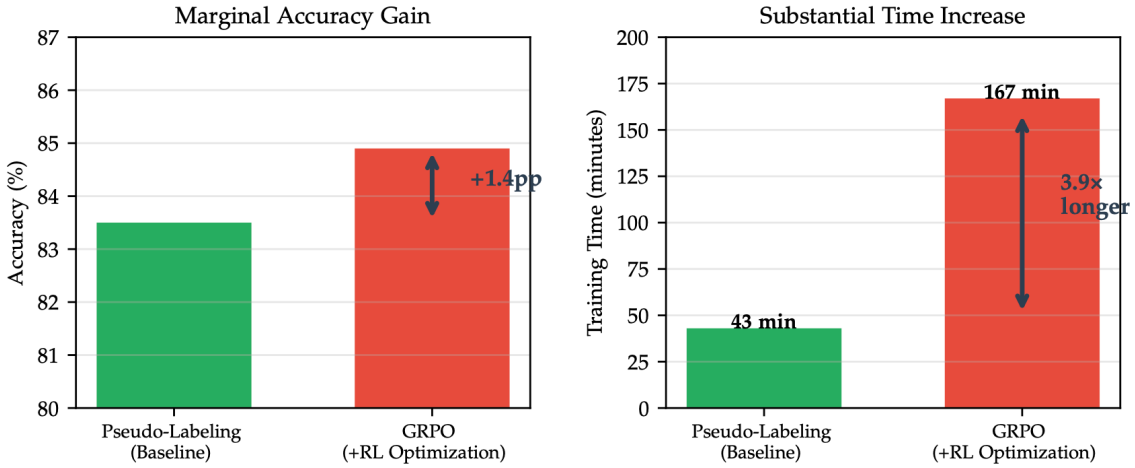


- **All three methods clear the bar** (quality + speed)
- **Pseudo-labeling** = balanced default
- **Contrastive** = fastest path
- **GRPO** helps on messy domains (pruning noisy/redundant archetypes matters most where classes overlap.

CD4AI: The CD4AI Pipeline extended as planned with a Pipeline Component for Classification/Annotation
CD4AI+: The Pipeline extended with a feedback mechanism to preselect suitable Archtypes at an earlier stage
Baseline: The Baseline Methods we evaluate our proposed approaches agains

Results – GRPO Caveats

Method	Avg. Acc. (%)	Improvement (pp)	Training Time	Throughput (samples/s)	Cost Ratio
Pseudo-Labeling	83.5	baseline	7-79 min	**275.6**	**1×**
GRPO Selection	**84.9**	+1.4	0.7-4.5 hr	191.3	3-6×
Dataset-specific improvements over pseudo-labeling:					
High-complexity (20NG, arXiv)		+2.5 avg			
Low-complexity (BBC, DBpedia)		-0.2 avg			



Key Takeaways

- **Buys very little gain** over plain pseudo-labeling on average, gains are diminishing.
- **Costs a lot more compute**: training time grows several-fold and inference gets slower.
- **Can be worse on some datasets** (esp. clear/structured ones): simpler methods match or beat GRPO there
- **Probable root cause**: the bottleneck is archetype quality, not selection; more RL doesn't fix noisy archetypes.

Answering the RQs

RQ1 How can semantic archetypes be leveraged to classify and annotate domain-specific documents?

- **Yes**, with a small gold seed: **archetypes become effective supervision** when paired with our four methods (embedding baseline → contrastive → progressive pseudo-labels → optional RL).
- **Best used as weak supervision**, not alone: combining archetype similarity + a few gold examples produces reliable pseudo-labels.

RQ2 Can a reward-based feedback system boost both classification accuracy and archetype quality?

- **Helps a bit**, mainly on harder datasets (20NG, arXiv), but **returns are diminishing**.
- **Compute trade-off is real**: **training** becomes much **heavier**; **throughput drops** vs. the simpler pipeline.

RQ3 How does this archetype-driven framework compare to supervised models and zero-shot LLMs in terms of accuracy and resource utilization across different domains?

- **Better accuracy-speed trade-off** than label propagation (too slow) and LLM prompting (good quality but slow/costly); **closer to full-train** than small supervised baselines.
- **Design targets met**: practical accuracy at practical throughput on the same splits/hardware.

Limitations & Future Work

Limitations

- **Overall quality is bottlenecked** by the **quality**/coverage of generated **archetypes**; weak seeds propagate noise.
- Several **heuristic choices** (e.g., 25 labels/class, GRPO policy selection biases, Label Budget) were not systematically optimized.
- **Evaluation is English-only** and **single-label**; behavior on multilingual and multi-label tasks remains untested.
- **Throughput is high at inference**, but full pipeline (distillation + iterations + RL) adds **compute/ops complexity**.
- **Upstream CD4AI stages** (keyword expansion, context windows, clustering, distillation) were **tuned before a classifier** existed; may be suboptimal now.

Future Work

- **Move RL "upstream"**: optimize archetype generation (prompting/policies) with validation-based rewards, **not only selection.**
- **End-to-end retuning of CD4AI stages** to maximize downstream classifier performance (jointly revisit expansion, windowing, clustering, distillation).
- **Multilingual and multi-label extensions**; stress-test on morphologically rich and low-resource languages and overlapping-topic corpora.
- Test **CD4AI Pipeline** with **HITL** for stages (Context Windows, Clusters etc.)

Conclusion

- **Completed CD4AI end-to-end** by adding automatic annotation from archetypes plus an optional RL feedback loop.
- **Introduced four complementary methods**: nearest-archetype baseline, contrastive adaptation, progressive pseudo-labeling, and GRPO selection.
- **Delivered a label-efficient, interpretable, and high-throughput pipeline** that works with ~25 labels per class across diverse datasets.
- **Established clear usage guidance**: Contrastive for speed, Pseudo-labeling as the balanced default, GRPO only when small extra accuracy on hard domains justifies compute.
- Showed that **expert-guided archetypes** can **stand in** for large labeled sets while retaining transparency in decisions.
- **Mapped the next steps**: focus on improving archetype generation, end-to-end retuning, and extending to multilingual and multi-label scenarios.

Moritz Steigerwald
Master Information Systems

Technical University of Munich (TUM)
TUM School of CIT
Department of Computer Science (CS)
Chair of Software Engineering for Business Information Systems (sebis)

Boltzmannstraße 3
85748 Garching bei München

+49.89.289.

wwwmatthes.in.tum.de