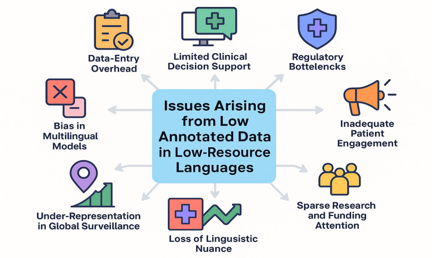
Custom Project: In-Context Learning and Chain-of-Thought Prompting based Efficient Data Annotation for Low-Resource Clinical Texts

Mohammad Junayed Hasan, Prajakta Shevakari, Swarali Mahimkar, Meghana Karnam

JOHNS HOPKINS UNIVERSITY

INTRODUCTION

- The digital healthcare revolution has generated massive volumes of unlabelled, freely available clinical text data on the internet and other digital resources.
- Specifically for low-resource languages, unlabelled, free data is available but lacks high-quality annotated datasets for downstream applications.
- Moreover, clinical texts are:
  - High-dimensional, complex, and unstructured.
  - Privacy-sensitive and expensive to annotate manually.
- As a result, existing NLP systems for tasks like information extraction and decision support struggle in low-resource settings.



- Large Language Models (LLMs) show promise for few-/zero-shot annotation, but:
  - Suffer from domain shift, hallucinations, and label inconsistencies.
  - Perform poorly on informal, code-switched clinical texts.
- **Goal:** Create an automatic, efficient, inexpensive, robust pipeline with reduced human effort and scalable methodology to annotate large-scale unlabelled low-resource data.
- **Our solution: A Prompt Engineering Workflow** combining:
  - Collection and analysis of small annotated dataset for low-resource language(s).
  - Domain-specific in-context learning for efficient and abundant data annotation.
  - Integration of Chain-of-thought (CoT) reasoning to improve annotation quality.

PROBLEM STATEMENT

- Let, U = {xi} denote a large unlabeled corpus of a low-resource language, where each xi is a clinical text sample. Moreover, let H = {xj, yj} denote the human-annotated version of the dataset, where yj is the gold-standard NER label sequence corresponding to xj.
- Our objective is to leverage a Large Language Model (LLM) to generate synthetic annotations y˜i for each xi ∈ U. This results in a final augmented dataset: D˜ ={(xi,y˜ i)}.
- Using D˜ , we train a downstream model gφ by minimizing a loss function.
- The final goal is to achieve: F1(gφ) ≈ F1(gφhuman) or F1(gφ) > F1(gφhuman), where gφhuman is a model trained solely on the human-annotated dataset H.

PROPOSED SOLUTION

- A systematic multi-stage pipeline with 3 stages: (i) Analysis of human-annotated data (ii) Prompt engineering through in-context learning, and (iii) Prompt engineering through in-context learning with Chain-of-Thought prompting.
- **Stage 1**: **Analysis of Human-Annotated Data**:
  - Analyse the distribution of entities in the human-annotated data
  - Analyse the consistency of annotation in the human-annotated data
  - Identify typical phraseology used for diseases, symptoms, medications, and procedures
- **Stage 2**: **Prompt Engineering with In-Context Learning (ICL)**
  - Design a prompt in the structure [Actor] [Task] [Instructions] [Format] [Constraints]
  - Integrate specific annotation guidelines that were given to the humans
  - Integrate few-shot examples of annotations as per human annotations
- **Stage 3**: **Prompt Engineering with ICL + Chain-of-Thought (CoT) Prompting**
  - Design a prompt in the structure [Actor] [Task] [Instructions] [Format] [Constraints]
  - Integrate specific annotation guidelines that were given to the humans
  - Integrate few-shot examples of annotations as per humans but with CoT reasoning
  - Integrate CoT constraints and formatting to the prompt
- **Stage 4**: **Downstream Training and Comparison**
  - Compare synthetic data with human-annotated data
  - Train encoder (e.g. BERT-based) models for downstream tasks (e.g. NER, MT etc.)
  - Compare downstream model performances (trained with synthetically annotated data versus with human-annotated data)

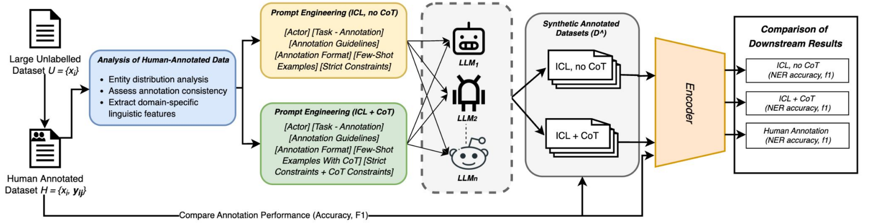


Figure: Overall system architecture of the proposed solution for efficient data annotation. The figure shows the various stages of the pipeline, from Stage 1 (Analysis of Human-Annotated Data), to Stages 2 and 3 (respectively prompt-engineering with ICL, and with CoT prompting), finally to Stage 4 (annotation quality and downstream NER performance comparison.

DATA

- **Bangla-HealthNER Corpus:** 31,783 samples (~144 000 sentences) drawn from a popular Bangladeshi medical forum; features informal Bengali with frequent English code-switching.
- **Entity Types:** Seven categories—symptom, health condition, medicine, specialist, age, dose, and procedure; entities make up approximately 23.5 % of all tokens.
- **Annotation Quality:** Inter-Annotator Agreement F1 = 88.6 %, Cohen's Kappa = 0.67, providing a reliable gold standard.
- **Train/Dev/Test Splits:** 80/10/10 stratified split to ensure each set maintains balanced distributions of the seven entity types.

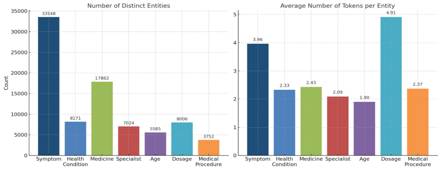


Figure: Distribution of entity types in the Bangla-HealthNER corpus. **Left:** Total number of distinct entity mentions per category. **Right:** Mean token span length for entities in each category.

EXPERIMENTS

Aspect	ICL Annotation	CoT Annotation	NER Training
Models	GPT-4.1, GPT-4.1-mini	GPT-4.1, GPT-4.1-mini	mBERT, BanglaBERT, BanglishBERT
Settings	3-shot few-shot; batch of 5; 1 s delay; strict JSON IOB2 schema	Same as ICL + multi-step CoT prompting	Linear-CRF head; AdamW (lr=2×10^{-5}); batch 32; early stopping on dev F1

- We applied two annotation pipelines—pure in-context learning (ICL) and multi-step chain-of-thought (CoT) prompting—using GPT-4.1 and GPT-4.1-mini to assign IOB2 labels to every sentence in the Bangla-HealthNER corpus, supplying concise annotation guidelines, three length-matched exemplars, and a strict JSON output schema.
- Annotation was executed in batches of five sentences with a one-second pause to satisfy rate limits; ICL runtimes were roughly 9 h for GPT-4.1 and 3.5 h for GPT-4.1-mini, while CoT increased compute to approximately three times those durations (~27 h and ~10.5 h, respectively).
- For downstream evaluation, we trained identical mBERT, BanglaBERT, and BanglishBERT architectures on the gold human annotations as well as on each of the two synthetic label sets.
- All models employed a linear-CRF output layer, the AdamW optimizer (learning rate = 2×10^{-5}), a batch size of 32, and early stopping based on development-set F1. We reported both token- and entity-level metrics (accuracy, precision, recall, F1) and assessed statistical significance via paired bootstrap resampling over 10 000 iterations ($p < 0.05$).

RESULTS

Annotator	Accuracy	Precision	Recall	F1
GPT-4.1 (ICL)	0.782	0.106	0.154	0.120
GPT-4.1 (ICL + CoT)	**0.804**	**0.238**	**0.313**	**0.259**
GPT-4.1-mini (ICL)	0.733	0.206	0.286	0.213
GPT-4.1-mini (ICL + CoT)	0.781	0.310	0.395	0.335

Table: Annotation performance of methods in comparison to human gold-labels.

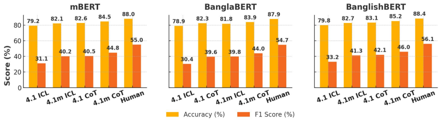


Figure: Downstream classification performance of encoders for different methods

- **Chain-of-Thought closes most of the gap:** Multi-step reasoning boosts annotation F1 by ~115% (GPT-4.1) and ~57% (GPT-4.1-mini) (Table 1), sharply reducing span hallucinations.
- **Downstream gains track annotation fidelity:** Every +1 pt in label-F1 yields ≈+0.35 pt in NER-F1 across all encoders, matching known error-propagation trends.
- **Mini-model + CoT hits 82% of human:** GPT-4.1-mini + CoT with BanglishBERT achieves 46.0 F1 vs. 56.1 gold—within 11.3 pts of human and zero manual-label cost.
- **Precision still limits performance:** CoT lifts both P & R, but false positives dominate the remaining gap—future work should target uncertainty filtering and span-calibration.
- **BanglishBERT excels regardless of noise:** Its mixed-code pretraining outperforms monolingual models on our corpus, independent of annotation quality.

CONCLUSIONS AND FUTURE WORKS

Conclusions

- **End-to-end, low-cost pipeline** for clinical NER in under-resourced languages.
- **CoT boosts annotation fidelity:** GPT-4.1-mini ↗ token-level F1 to 0.335 (×3 over naive ICL).
- **Strong downstream gains:** BanglishBERT on synthetic labels hits 82% of full-human F1—zero manual labour.
- **Label fidelity → NER performance:** Near-linear correlation means small annotation tweaks yield big task gains.
- **Broader impact:** CoT-enhanced LLM annotation is already a viable stand-in for manual labelling in Bengali—and paves the way for other low-resource domains.

Future Works

- **Human in the loop workflow:** Route low-confidence spans to experts.
- **Multi-lingual experiments:** Test on diverse low-resource languages.
- **Collect more data:** Crawl data from the internet, annotate, add it to training set.
- **Use other state-of-the-art models:** Claude / Gemini / DeepSeek / LLaMA etc.
- **Leverage machine translation:** English -> low-resource language -> annotate, train