Evaluating Tools for Medical Terminology Extraction

Alex Hassett Jasir Nawar Mamadou Ndom

1 Problem Statement

Main idea. We will run an evaluation project that compares different tools for **medical terminology extraction** (clinical named entity recognition, NER). The goal is to measure how well these tools find and label spans like anatomy, findings, and diagnoses in radiology reports. The practical question is: **do you need a lot of compute to get good NER results, or can smaller models do the job well enough**.3

We will compare **three models** to start. Two will be **encoder models** that we **fine-tune** on the dataset: **BioClinical-ModernBERT-base** and **ModernBERT-base**. One will be a **decoder model** that we only **prompt** for inference, without any fine-tuning: **Llama 4**. If time allows, we may add **Deepseek** as a fourth system. All models will be run on the same data and scored with the same code.

Our main dataset will be **RaTE-NER** (radiology NER). It is recent, radiology-focused, and has labeled entities needed for NER. We will keep the setup simple and consistent. We will decide on one **span matching policy** for scoring (**strict exact match** or **lenient overlap**) and use that for all models.

We will report **micro F1** as the **primary metric**. We will also report **precision** and **recall**. Micro F1 fits this task because both types of mistakes matter in clinical notes: flagging a term that is not there and missing a term that is present. Micro F1 is also commonly used for evaluations on the NER task, and using common metrics also makes it easier to compare our results to other papers.

We will **fine-tune only the encoder models**. For fairness, we will use the same training budget and a similar **hyperparameter search space** for both encoders. We will use **five-fold cross validation (CV)** on the training and development pool to choose hyperparameters, and we will keep a **held-out test set** that we use only once at the end. For the decoder model, we will design a simple prompt that returns entities in a machine readable format and keep the decoding settings fixed.

A major part of our project is **logging compute use**. We will track **GPU/CPU/system power (watts)**, **GPU memory (VRAM)**, **system RAM**, and **throughput/latency**. We will collect this during both training and inference. The project is framed as a quality comparison and a **resource** comparison, so readers can see the tradeoff between F1 and cost.

2 Team and Roles

Alex — coding and theory. Fine-tuning and cross validation for encoders. Writes the **resource logger** and integrates it with Weights and Biases.

Mamadou — writing and coding. Manages the proposal and paper, organizes tables and figures, and keeps the work aligned with the course guidelines.

Jasir — software development and evaluation. Sets up the **data pipeline** and **unified scorer**, and runs decoder style models with prompts on CPU or GPU. Oversees end to end evaluation runs.

3 Project Plan

3.1 Data and Splits

Dataset: **RaTE-NER** (radiology NER). We will convert outputs from all systems into a single internal format for scoring. We will keep one **test set** that is never used during training or hyperparameter selection. For the rest of the data, we will use **five-fold cross validation** to pick hyperparameters. We will do an **80/20 split**, the 80 percent will be used for CV and the 20 percent will be the held-out test set. Since we are doing k-fold cross-validation, one fold will be used for validation during training and the remaining k-1 folds will be used for training. We were planning to do 5-fold, so that means during training, each iteration will use 1 of the 5-folds as validation, so 20 percent of the training data will be used for validation.

3.2 Systems

Encoders to fine-tune:

- **BioClinical-ModernBERT-base**
- **ModernBERT-base**

Decoder model to prompt (no fine-tuning):

- **Llama 4**

Stretch goal: add **Deepseek** as an extra comparator if time and resources allow.

3.3 Training and Tuning

We will keep the encoder procedures as similar as possible. Same optimizer type, similar ranges for **learning rate**, **batch size**, and **number of epochs**, and the same **early stopping** rule. **Cross validation** will pick the settings that do best on validation F1. We will fix **random seeds**. After we pick hyperparameters, we will retrain on the full train plus dev pool, and then do a **single evaluation on the test set**.

3.4 Prompted Inference

We will use a simple prompt for **Llama 4** that asks for labeled spans in a machine readable format. We will use the same **prompt and decoding settings** for all runs, and we will parse the outputs into the same schema used by the encoders so the scorer treats them the same way.

3.5 Scoring and Span Policy

We will decide on one **span policy** and keep it fixed for all models. By default we will choose **strict exact span matching**. The scorer will compute **precision**, **recall**, and **micro F1**. We will focus on **micro F1** in the proposal. For the final paper we may add **per entity F1**.

4 Evaluation Plan

Primary metric: micro F1. We will also report precision and recall. We will not design new metrics because we want our results to line up with standard practice.

We will keep the **test set separate** from any cross validation. Cross validation rotates the train and dev parts but never uses the test set. We will record **loss**, **learning rate**, and other training curves to check for overfitting. The decoder model runs will also be logged and scored with the same code.

5 Error Analysis

We will look at **confusion matrix** patterns to see if there are many false positives or false negatives and whether **boundary mistakes** are common. We will save example sentences with the gold and predicted spans to show typical errors. If time allows for the final paper, we will add **per entity F1** plots to see which categories are hard.

6 Compute and Resource Logging

We will log the following during training and inference: **GPU, CPU, and system power (watts)**, **GPU memory (VRAM)**, **system RAM**, and **throughput or latency**. This will be part of our main comparison so readers can see **cost next to quality**. We will keep logging **consistent** across models and runs.

7 Collaboration Plan

Jasir will set up the **data pipeline** and the **unified scorer**. **Alex** will handle **fine-tuning**, **cross validation**, and the **resource logger**. **Mamadou** will prepare the **proposal and paper**, organize results, and keep the group on schedule. We will use **GitHub**. The workflow will be split so we can work in **parallel** on data, training and tuning, evaluation, and writing.

If we use the **Greene** cluster and need access to **A100** or **H100**, Alex will handle the requests and job scripts. We will keep **configurations and random seeds** in version control for reproducibility.

8 Questions To Resolve Early

If we use **cross validation**, what is the exact split. We plan on **five-fold CV** on the train and dev pool and a **separate test set**. We may compare performance with CV to a single split to show why CV helps when data is limited. We will benchmark all models on the same dataset with the same procedures. The only differences should be **model specific hyperparameters** that come from the same search space.

9 Related Work (ACL Anthology)

RaTEScore / RaTE-NER (EMNLP 2024) . This paper introduces radiology-focused evaluation resources and the RaTE-NER dataset used for entity extraction. Relation to our project: it is the basis for our data choice and defines realistic radiology entities and span annotations. Role in our project: we will use RaTE-NER as the primary dataset and follow its labeling scheme and recommended evaluation setup so that our results are comparable to published work.

NLP Power: Efficiency vs Accuracy (Workshop 2022) . This paper motivates reporting efficiency together with accuracy, including measurements of speed, memory, and energy. Relation to our project: it justifies why we log GPU/CPU/system power, VRAM, RAM, and throughput alongside F1. Role in our project: it guides the design of our resource logger and our reporting tables so we present F1 next to cost metrics in a consistent way.

Medical Spoken NER (NAACL Industry 2025) . This paper reports medical NER results with clear precision, recall, and F1 tables across multiple models. Relation to our project: it provides an example of straightforward NER evaluation and table formats that emphasize micro F1 while still listing precision and recall. Role in our project: we will mirror this reporting style for our radiology text experiments so the results are easy to read and compare.

Zero-shot Clinical NER with LLMs (NAACL Long 2025) . This paper shows that prompted large language models can perform clinical NER without fine-tuning. Relation to our project: it supports our choice to include a prompted decoder model (Llama 4) as a baseline next to fine-tuned encoders. Role in our project: it informs our prompt design and the decision to keep decoding settings fixed and deterministic for fair comparison.

scispaCy (BioNLP 2019) . This paper presents robust biomedical NLP pipelines and common evaluation practices for entity extraction. Relation to our project: it reinforces standard span-based scoring with precision, recall, and F1, and highlights simple, reproducible baselines. Role in our project: it anchors our evaluation conventions (one span policy, a unified scorer, and clear metric definitions) and helps motivate using strong but efficient encoder baselines.

References

[1] Zhao, Weiming, et al. 2024. RaTEScore: A Metric for Radiology Report Generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP 2024)*. Available at: `https://aclanthology.org/2024.emnlp-main.836/`.

[2] Ang, Phyllis; Dhingra, Bhuwan; and Wills, Lisa Wu. 2022. Characterizing the Efficiency vs. Accuracy Trade-off for Long-Context NLP Models. In *Proceedings of NLP Power! The First Workshop on Efficient Benchmarking in NLP*. Available at: `https://aclanthology.org/2022.nlppower-1.12/`.

[3] Le-Duc, Khai; Thulke, David; Tran, Hung-Phong; Vo-Dang, Long; Nguyen, Khai-Nguyen; Hy, Truong-Son; and Schlüter, Ralf. 2025. Medical Spoken Named Entity Recognition. In *Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Industry Track)*. Available at: `https://aclanthology.org/2025.naacl-industry.59/`.

[4] Averly, Reza, and Xia Ning. 2025. Entity Decomposition with Filtering: A Zero-Shot Clinical Named Entity Recognition Framework. In *Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Long Papers)*. Available at: `https://aclanthology.org/2025.naacl-long.150/`.

[5] Neumann, Mark; King, Daniel; Beltagy, Iz; and Ammar, Waleed. 2019. ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*. Available at: `https://aclanthology.org/W19-5034/`.