AI-Driven Extraction and Standardization of Medical Entities from Hospital Discharge Summaries Using NLP

Daniel Collis
Mike Cottrell College of Business
University of North Georgia
Dahlonega, GA USA

***Abstract*—This project revolves around developing and constructing an AI-driven program that retrieves structured clinical data out of free-text hospital discharge summaries. It should be capable of identifying significant medical entities (diagnoses, medications, procedures, etc.) and normalizing them with UMLS (Unified Medical Language System) codes. This would enhance data interchangeability among providers and allow consistent reporting between Electronic Health Record (EHR) systems. The parser system uses natural language processing (NLP) techniques via the spaCy and SciSpaCy libraries, served through a FastAPI backend and a React-based frontend. The system's outputs can be validated for accuracy and potential utility in real-world clinical workflows.**

Keywords—natural language processing (NLP), hospital discharge summaries, clinical named entity recognition (NER), UMLS mapping, medical concept normalization, SciSpaCy, spaCy, electronic health records (EHR), AI in healthcare

I. Introduction

Healthcare providers of the digital age face an overwhelming challenge: having too much data. Electronic health records have streamlined many aspects of the medical field, but they've also exacerbated data overload. As patients go in and out every day, hospitals pile up enormous amounts of patient information, and a large portion of said information is stored in unstructured, messy formats such as free-text discharge summaries. The importance of these documents cannot be understated, but neither can their inability to be shared and read efficiently across different healthcare providers. This presents a problem, as these documents contain crucial information such as diagnoses, medications, follow-up instructions, etc., but since they're written in an inconsistent format, they can be incredibly difficult to work with systematically.

To further illustrate the complications posed by these documents – when a patient moves between providers or healthcare systems, important information can often be overlooked. These negative effects often snowball, leading to clinical decision support systems not being able to easily parse through narrative text to flag potential drug interactions or treatment conflicts. Additionally, researchers struggle to extract standardized data from the pools of discharge summaries at their disposal. What we need is a method to transform said data into a clean format so that computers (and healthcare providers) can actually interpret it in a meaningful way.

Recent advances, however, offer a promising solution in artificial intelligence (AI) and natural language processing (NLP), consisting of reliable platforms and tools for reconstructing clinical text into clean, machine-readable data. Named Entity Recognition (NER) and entity linking techniques – more specifically the techniques fueled by NLP frameworks like spaCy and SciSpaCy – are reliable approaches for extracting and normalizing medical terms. What these techniques do is connect extracted terminology and link it to its corresponding Unified Medical Language System (UMLS) Concept Unique Identifier (CUI). This is a huge step towards creating a more standardized approach to medical data sharing.

In this research project, the design and development of an AI-driven discharge summary parser leveraging NLP to extract and standardize key medical information from unstructured clinical text will be detailed, alongside an evaluation of the parser. The technologies used includes a FastAPI backend in tandem with a React-based frontend. Integrated with these platforms are multiple pretrained models from SciSpaCy that can identify medical entities and map them to UMLS CUIs. This system serves to demonstrate the feasibility of automated information extraction, while laying the groundwork for more efficient clinical workflows and data exchange mechanisms. The project will evaluate the performance of this system and the models that are being used within it to determine how usable it would be in a realistic clinical setting. The metrics that are being analyzed include precision, recall, and F1 score on a curated test set.

II. Literature Review and Related Work

In the healthcare domain, extracting clinical data from unstructured text has become an important but cumbersome challenge, and the adoption of electronic health records has only made this process more difficult. A key source of this unstructured text is discharge summaries, which often lack structured formatting, making them difficult to process and share across healthcare systems. This gap, however, can be addressed with recent advancements in artificial intelligence and natural language processing; researchers have explored many different methods – including relation extraction, entity link, NLP, etc. – to identify and standardize vital clinical concepts such as diagnoses, medications, and procedures. Using different tools such as SciSpaCy and LLMs such as GPT-4 have proved successful in pushing the boundaries of automated clinical NLP, with studies indicating improvements in both the accuracy and interoperability of extracted data. This literature review analyzes how these technologies have been used in the past and improved over time, while examining how they've been used for discharge summary parsing. This has laid the foundation for this project's focus on entity extraction and UMLS-based normalization using spaCy and SciSpaCy.

Extracting structured data from unstructured clinical documents has been streamlined in recent years using Artificial Intelligence (AI) and Natural Language Processing (NLP), especially in reference to hospital discharge summaries. A recent study developed an automated model designed to extract important information from hospital discharge summaries, demonstrating the potential for AI in enhancing clinical documentation processes [1].

When considering these technologies for clinical practices, NLP, named entity recognition (NER), and relation extraction are pivotal for structuring and understanding clinical narratives. A recent review explores these concepts in a clinical context, highlighting the advancements and evolution of methodologies and their applications in processing free text [2].

The accuracy and thoroughness of discharge summaries are crucial for patient health and care continuity. A study explores whether or not LLM-generated discharge summaries could be compared to those drafted by the physicians themselves, emphasizing how important it is to maintain accurate information transfer in clinical settings [3].

Furthermore, a study explored the application of machine learning techniques and how they can be used to extract clinical entities from discharge summaries. Various approaches were evaluated to identify medical problems, tests, and treatments within clinical narratives [4].

SciSpaCy, a Python package for processing biomedical text, has been central in connecting entities from discharge text to the Unified Medical Language System (UMLS), basically converting any loose writing into standardized codes that are easily parsable to other healthcare destinations. However, researchers have explored possible drawbacks of this service, as properly translating this text is not only a difficult process, but an extremely important and fragile one, as well [5].

Lastly, the potential of large language models in regard to automating the generation of inpatient discharge summaries has been investigated in a study that analyzed how effective GPT-4o could be in producing accurate and efficient clinical documentation [6].

III. Design and Methodology

In this project, a web-based, AI-driven tool is the focus. The purpose of this tool is to parse unstructured discharge summaries and extract structured clinical data – specifically diagnoses and medications – and link them to their corresponding UMLS CUI. The architecture involves using a FastAPI backend for processing and a React frontend for user interaction, with the core NLP abilities being handled by spaCy and SciSpaCy. The goal of the system is to support interoperability among healthcare providers by transforming free-text clinical documents into something more structured and machine-readable.

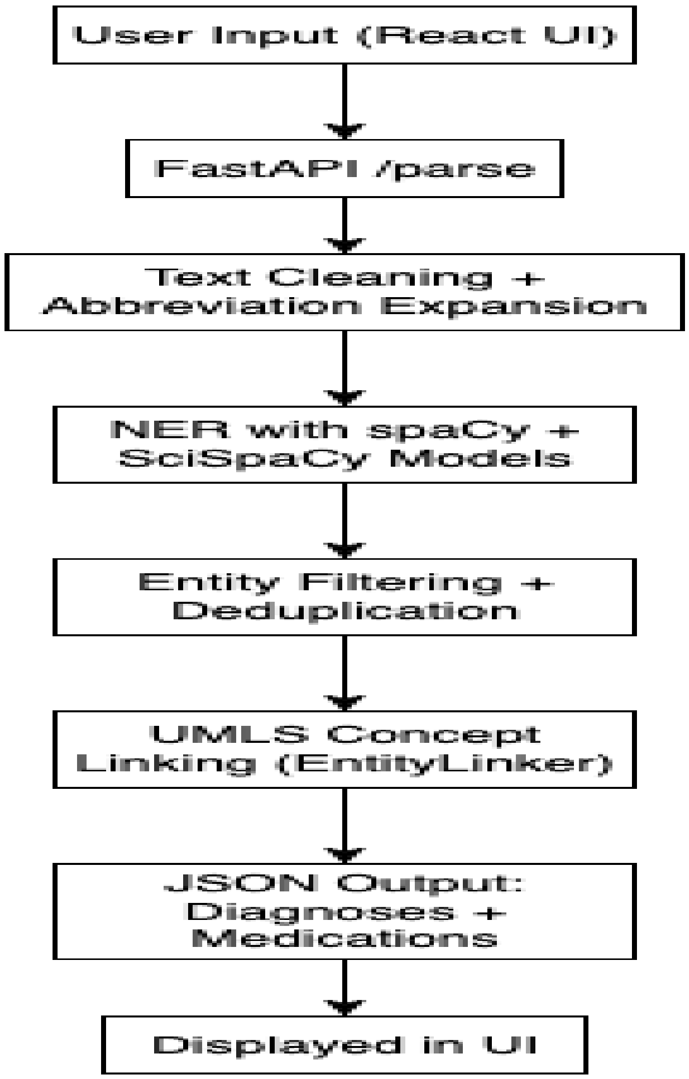

User Input (React UI)
FastAPI /parse
Text Cleaning + Abbreviation Expansion
NER with spaCy + SciSpaCy Models
Entity Filtering + Deduplication
UMLS Concept Linking (EntityLinker)
JSON Output: Diagnoses + Medications
Displayed in UI

Figure 1: System Architecture Design

A. *Data Acquisition and Preprocessing*

Discharge summaries for testing were created by using LLMs such as ChatGPT and Claude to mimic real-world discharge summaries, then manually going through those to edit and curate them, prioritizing variability and human language. The evaluation dataset includes 15 synthetic or de-identified summaries.

The preprocessing pipeline includes:

- Text cleaning: Removal of extraneous whitespace, special characters, and inconsistent formatting using utility functions (utils.py).
- Abbreviation expansion: Implemented with SciSpaCy's *AbbreviationDetector* to resolve shorthand (e.g., "HTN" to "Hypertension").
- Case normalization: Ensures consistent token recognition across summaries.

B. *Entity Extraction and Normalization*

At the core of the pipeline is the *parser.py* module. It leverages pretrained biomedical NLP models:

- *en_ner_bc5cdr_md*: Specializes in disease and chemical/drug detection.
- *en_core_sci_md*: Provides domain-specific tokenization and embeddings.

The pipeline performs NER (named entity recognition), which is responsible for extracting spans that represent clinical entities, then it uses entity filtering to deduplicate overlapping mentions and filters redundant results. More specifically, it ensures that a disease will not show up twice in the results just because it is written twice in the discharge summary. Finally, through SciSpaCy's *EntityLinker*, extracted entities are matched to their corresponding CUIs using a context-aware ranking system.

C. *Backend Architecture*

Built using FastAPI, the backend exposes two endpoints: GET /ping, which is a simple health check, and POST /parse, which accepts raw discharge summaries and returns a JSON payload of extracted entities and UMLS links.

The parser executes the following pipeline:

1. Clean input text
2. Extract entities using spaCy
3. Link entities to UMLS using SciSpaCy's *EntityLinker*
4. Format response as JSON

All API responses omit logging for data security, and sensitive medical identifiers are never stored nor retained.

D. *Frontend Integration*

The frontend, written in React + Vite, is a simple interface, providing a text input box where users are intended to paste discharge summaries, a parse button to trigger the pipeline once everything is pasted, and a display for the JSON output to view the parsed results.

Additionally, the frontend gracefully handles errors, alerting users if parsing fails.

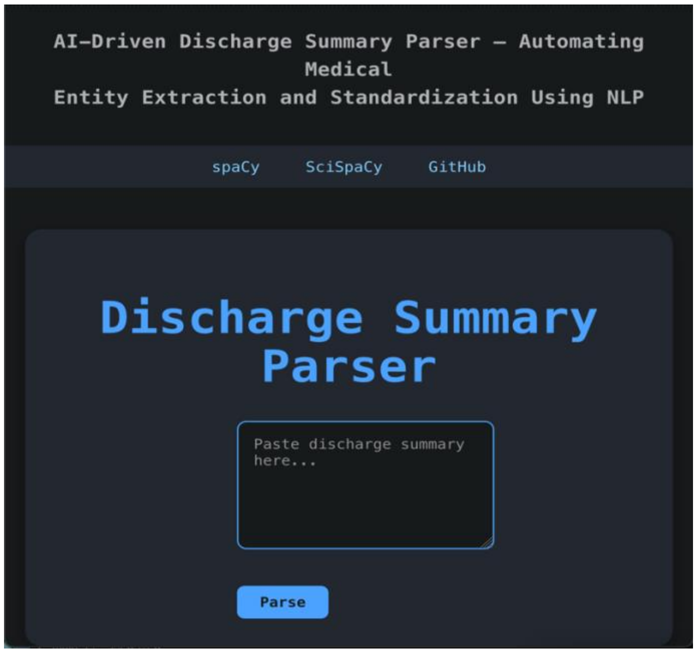

AI-Driven Discharge Summary Parser – Automating Medical
Entity Extraction and Standardization Using NLP
spaCy
SciSpaCy
GitHub
Discharge Summary Parser
Paste discharge summary here...
Parse

Figure 2: UI Design

E. *Evaluation Framework*

To measure the accuracy of the parser, a custom evaluation module (*evaluate_parser.py*) was implemented. It is responsible for loading the manually annotated evaluation set (*evaluation_set_new.json*), running each summary through the parser, then comparing the predicted outputs to the actual outputs, matching by both UMLS CUI and UMLS name.

After this, it calculates:

- Precision = TP / (TP + FP)
- Recall = TP / (TP + FN)
- F1 Score = 2 * (Precision * Recall) / (Precision + Recall)

Metrics are computed per entity type (diagnoses, medications) and aggregated for overall performance.

These metrics are standard in information extraction and provide a balance between sensitivity (recall) and accuracy (precision), especially important in clinical NLP applications.

F. *Observational and Manual Evaluation Notes*

While the evaluation was robust, some challenges and limitations emerged.

Firstly, and most obviously, the test case discharge summaries were created using widely applicable LLM models, and they were manually fine-tuned by the author. Both the LLM and I have no medical experience and cannot confidently curate real-world discharge summaries for testing. This could lead to overly simplistic summaries that do not come close enough to legitimate examples, however, the ability of the model to extract diagnoses and medications and link them to their UMLS CUI counterparts can still be effectively demonstrated, even with self-made discharge summaries.

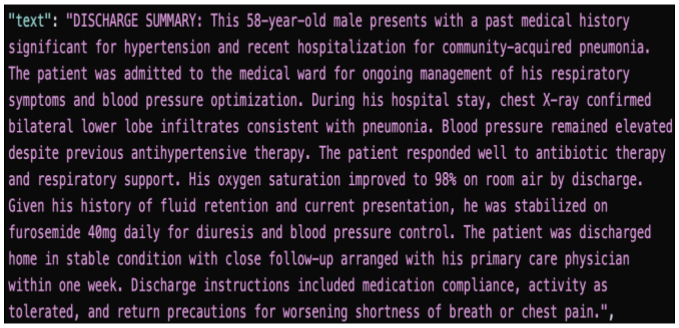
Another limitation that can be contributed to my lack of knowledge about discharge summaries is that I am unsure if users of this system would want prospective diseases listed in the output or not. For example, if a discharge summary says something like, "Patient scheduled follow-up appointment to ensure there is no development of bronchitis" then it is unclear whether or not bronchitis would be useful to have present in the output. Is it better for doctors to know everything they can, or would it only waste time and effort to have them thinking about something that the patient wasn't officially diagnosed with?

Going through each discharge summary manually also presents the challenge of knowing what the model should extract in the first place. In testing, the issue was seldom with the UMLS CUI mapping – that aspect was almost perfect; the most uncertainty came with not knowing when something should be included in the output. Similar to the last challenge, it is difficult to know if trivial things like "pain" or "sore throat" should be included in the output, if the source of said pain or sore throat was already accounted for in a different diagnosis, such as "strep throat". If real doctors could be consulted to review and edit the existing test summaries, then the evaluation would be more indicative of how this tool would perform in practical situations. Having them review things like the language of the summaries and the predicted diagnoses and medications that should be identified would lead to better results.

Lastly, while there are guardrails in place to prevent duplicate entries from appearing in the output, there is nothing to stop the tool from providing two of the same disease with different entries in UMLS in the output. To be more specific, the model could identify something like "community-acquired pneumonia" but then also identify just "pneumonia", and since these have separate entries in UMLS, the tool would output both of them as separate diseases. This could be seen as more of a problem with UMLS than it is with this project, as it's technically doing what it was designed to do.

IV. Results and Reflection

The parser was evaluated using a test set of 15 manually annotated example discharge summaries, each containing a variety of diagnoses and medications alongside their corresponding UMLS Concept Unique Identifiers.



```
"text": "DISCHARGE SUMMARY: This 58-year-old male presents with a past medical history
significant for hypertension and recent hospitalization for community-acquired pneumonia.
The patient was admitted to the medical ward for ongoing management of his respiratory
symptoms and blood pressure optimization. During his hospital stay, chest X-ray confirmed
bilateral lower lobe infiltrates consistent with pneumonia. Blood pressure remained elevated
despite previous antihypertensive therapy. The patient responded well to antibiotic therapy
and respiratory support. His oxygen saturation improved to 98% on room air by discharge.
Given his history of fluid retention and current presentation, he was stabilized on
furosemide 40mg daily for diuresis and blood pressure control. The patient was discharged
home in stable condition with close follow-up arranged with his primary care physician
within one week. Discharge instructions included medication compliance, activity as
tolerated, and return precautions for worsening shortness of breath or chest pain.",
```

Figure 3: Example Discharge Summary

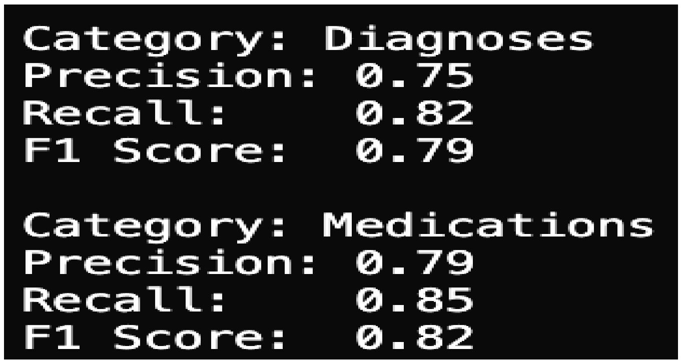
This set was used as a gold standard for what the parser should produce in its output, and the accuracy to this standard was measured using precision, recall, and F1 score. The evaluation was performed using a custom script, *evaluate_parser.py*, which calculates these three metrics for each sample and category (diagnoses and medications). The script then aggregates these results to produce average scores.

A. *Overview of Parser Evaluation*

The parser achieved the following average scores across the evaluation set:

Category	**Precision**	**Recall**	**F1 Score**
Diagnoses	0.75	0.82	0.79
Medications	0.79	0.85	0.82

Figure 4: Evaluation Metrics (Table)



```
Category: Diagnoses
Precision: 0.75
Recall:    0.82
F1 Score:  0.79

Category: Medications
Precision: 0.79
Recall:    0.85
F1 Score:  0.82
```

Figure 5: Evaluation Metrics (Console)

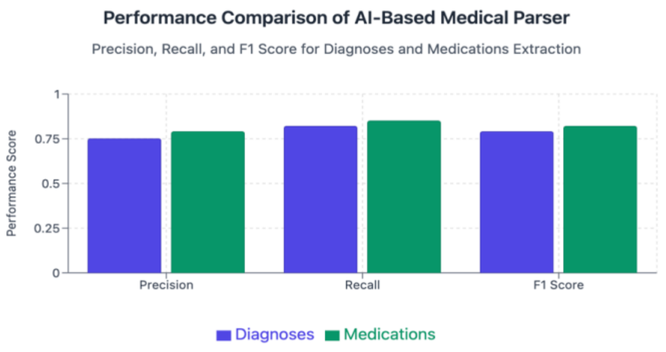
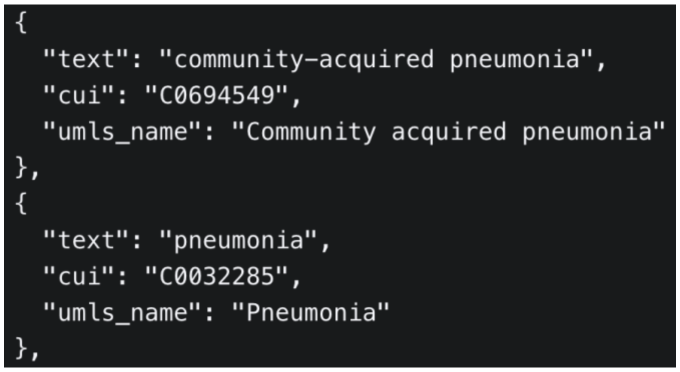


Figure 6: Performance Metrics (Visual)

These scores are indicative of relative success in both categories, with medication extraction slightly outperforming that of the diagnoses. The high F1 scores indicate a good balance of identifying relevant terms while avoiding incorrect classifications.

As for the discrepancy in diagnosis vs medication extraction, this can most likely be attributed to the small sample size, however there are some characteristics of the discharge summaries that may make it easier to identify drugs and chemicals as opposed to diseases. Most notably, there is less variance when it comes to medications. These medicines are just drugs, and if drugs are mentioned in the discharge summary, then the parser will add it to the list of medications. With diagnoses, there can be many different types of the same disease. A prevalent example of this is in sample 1: the discharge summary says the patient was diagnosed with "community acquired pneumonia", which the model correctly identifies, but when the summary mentions pneumonia again, the model then adds "pneumonia" to the list of diagnoses in addition. Quirks like this can make it difficult to A) correctly predict how the model will identify terms and B) evaluate the model's success.



```
{
  "text": "community-acquired pneumonia",
  "cui": "C0694549",
  "umls_name": "Community acquired pneumonia"
},
{
  "text": "pneumonia",
  "cui": "C0032285",
  "umls_name": "Pneumonia"
},
```

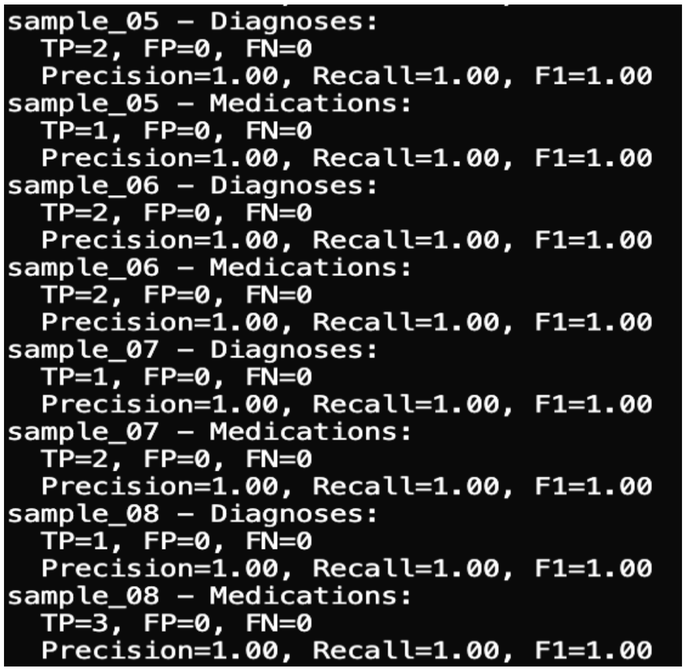
Figure 7: Duplicate Output Example

This could also be seen as a weakness of the model, however. Since it's eager to add any drug to the list of outputs, this could lead to false positives if the medications listed are not directly being prescribed to the patient. An example of this is in sample 4 where glucose is mentioned as an indicator of a disease, but the model added it to the list of medications anyway.

The combination of a smaller sample size in conjunction with the straight forwardness of medicine identification is what led to this slight discrepancy, most likely.

B. *High-Performing Cases*

Near-perfect performance was observed in a select amount of test cases. From sample 5, for instance, the model perfectly extracted the medication ("pantoprazole") and diagnoses ("abdominal pain" and "pancreatitis"). Similarly, samples 6-8 experienced flawless performance, where the parser consistently identified all expected entities without any extraneous results.



```
sample_05 - Diagnoses:
  TP=2, FP=0, FN=0
  Precision=1.00, Recall=1.00, F1=1.00
sample_05 - Medications:
  TP=1, FP=0, FN=0
  Precision=1.00, Recall=1.00, F1=1.00
sample_06 - Diagnoses:
  TP=2, FP=0, FN=0
  Precision=1.00, Recall=1.00, F1=1.00
sample_06 - Medications:
  TP=2, FP=0, FN=0
  Precision=1.00, Recall=1.00, F1=1.00
sample_07 - Diagnoses:
  TP=1, FP=0, FN=0
  Precision=1.00, Recall=1.00, F1=1.00
sample_07 - Medications:
  TP=2, FP=0, FN=0
  Precision=1.00, Recall=1.00, F1=1.00
sample_08 - Diagnoses:
  TP=1, FP=0, FN=0
  Precision=1.00, Recall=1.00, F1=1.00
sample_08 - Medications:
  TP=3, FP=0, FN=0
  Precision=1.00, Recall=1.00, F1=1.00
```

Figure 8: Streak of Good Scores

These samples are not as complex as samples 1-4, indicating that the model excels when analyzing clearly worded, unambiguous summaries that limit repetition. The model also performs well when diagnoses and medications are explicitly listed and follow predictable syntactic structures. Phrases such as "diagnosed with…" or "found to have…" make it easier for the model to know when a disease should be identified. Likewise, words like "prescribed" or "received" indicate to the parser that a medication should be present.

C. *Challenging Cases and Error Analysis*

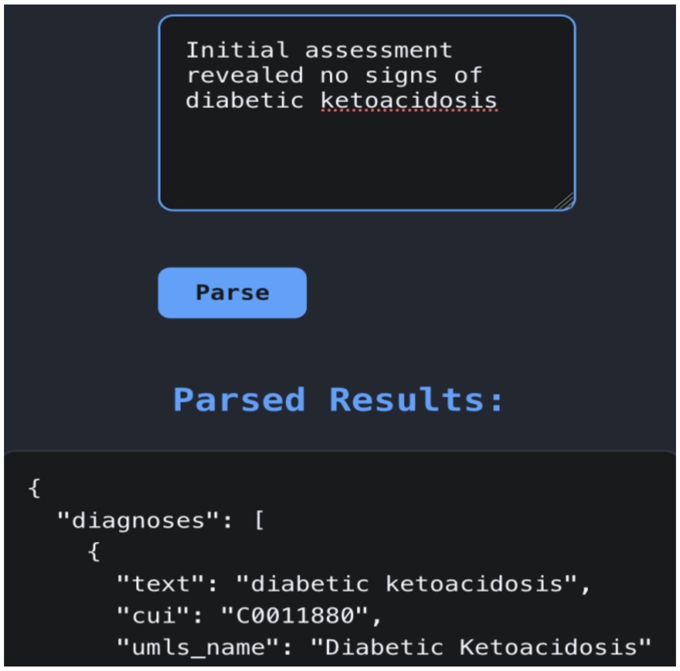
Although the model showed strong overall performance, there were still some test cases that highlighted its important limitations.

a. Over-Prediction/False Positives (Low Precision)

Sample 2 does a good job of showcasing a few distinct limitations of this project. In this example, the model achieved a low precision of 0.50 for diagnoses. Out of the eight diagnoses it predicted, only 4 were correct. Among the incorrect extractions were generic patient observations such as "complaints of chest discomfort" and "diabetic diet". These are more consistent with descriptions of symptoms as opposed to illnesses. This depicts a crucial aspect of this project, which is the disconnect between what *I* think the model should predict vs what the gold standard should actually be**.** In any case, this shows how the model can sometimes hallucinate and over-predict, leading to inaccurate extraction.

Additionally, in sample 2, the aforementioned "complaints of chest discomfort" was, predictably, not mapped to a CUI nor UMLS name. I am unsure of why this was identified as a diagnosis as it is more in line with a symptom as previously mentioned.

Another challenge for the model is to separate what the patient *was* diagnosed with from what they *were not* diagnosed with. In sample 4, the summary explicitly states that the patient does not have "diabetic ketoacidosis", but the model lists it as a diagnosis anyways, leading to additional hallucination and over-predicting. This specific error illustrates the model's difficulty in analyzing context and using it to make informed decisions. Instead, the model operates on simply finding the keywords and extracting them, regardless of their place in the summary.



```
Initial assessment
revealed no signs of
diabetic ketoacidosis
```

```
{
  "diagnoses": [
    {
      "text": "diabetic ketoacidosis",
      "cui": "C0011880",
      "umls_name": "Diabetic Ketoacidosis"
```

Figure 9: False Positive Example

b. Under-Prediction/False Negatives (Low Recall)

There were a couple examples of the parser missing a classification or two that it shouldn't have. For instance, in sample 10, while the parser achieved a perfect precision of 1.00 for diagnoses, the recall was only 0.50, indicating that there was one missed diagnosis out of the two.

Similarly, this low recall pattern can be observed in sample 3, where expected conditions such as "reactive airway disease" and "bronchodilation" were not identified. This could be another case of bronchodilation being a symptom rather than a disease, but that wouldn't explain why the model did not classify it as a disease, since it's proven to do that with symptoms in other examples. As for the failure to retrieve "reactive airway disease", the model instead just caught "airway disease". This is an example of the model, seemingly arbitrarily, choosing when to be specific and when not to be. This will be discussed in further detail in the next section.

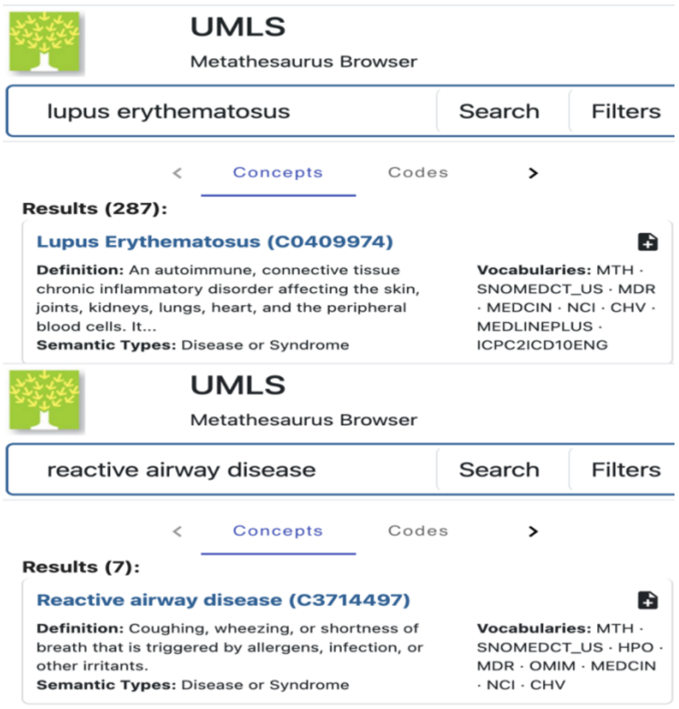
D. *Entity Specificity and Inconsistency*

A consistent and difficult challenge observed was inconsistency in entity specificity. As previously mentioned, the model has trouble deciding when to provide the full name of the disease vs when to use the base version of it. This provided further complications when writing test cases, as it highlighted the unpredictability of the model.

The model identifying "reactive airway disease" as just "airway disease" is one example of the model being less specific than it should have been. This can also be observed in sample 7. In this instance, the discharge summary clearly states that the patient has a "migraine with aura". However, the model only used "migraine" in its disease specifications.

These instances are especially baffling, since the model also performed to a high level of specificity in other cases. For example, sample 9 diagnoses the patient with "systemic lupus erythematosus", which is a specific form of lupus erythematosus. I anticipated the model to use the broader form of the disease, but it included every description available, in this case.

It should also be noted that for each of these inconsistencies, UMLS CUIs exist for the counter. Meaning, in examples of the model removing specificity, UMLS classifies the more specific term as its own entry. "Reactive airway disease" is a separate UMLS entry from "airway disease" with a different CUI and description; it is unknown why the model would skip over this. In other cases of the model identifying the specific term, there were also existing UMLS entries for more vague terms. There exists a separate entry for "lupus erythematosus" from its systemic version.


UMLS

Metathesaurus Browser

lupus erythematosus

Search Filters

< Concepts Codes >

Results (287):

Lupus Erythematosus (C0409974)

Definition: An autoimmune, connective tissue chronic inflammatory disorder affecting the skin, joints, kidneys, lungs, heart, and the peripheral blood cells. It...

Semantic Types: Disease or Syndrome

Vocabularies: MTH · SNOMEDCT_US · MDR · MEDCIN · NCI · CHV · MEDLINEPLUS · ICPC2ICD10ENG

UMLS

Metathesaurus Browser

reactive airway disease

Search Filters

< Concepts Codes >

Results (7):

Reactive airway disease (C3714497)

Definition: Coughing, wheezing, or shortness of breath that is triggered by allergens, infection, or other irritants.

Semantic Types: Disease or Syndrome

Vocabularies: MTH · SNOMEDCT_US · HPO · MDR · OMIM · MEDCIN · NCI · CHV

Figure 10: Existing UMLS Entries Showcase

Figure

E. *Reflection on Evaluation Methodology*

The major concern with the evaluation method is the manual drafting and annotating of the "gold standard" expected outputs. With these summaries coming from someone with a non-medical background, it can be argued that there is unneeded variance in the model's accuracy due to my lack of knowledge in this field. This variance could come in the form of predicting entities that should not be, not predicting entities that should be, or just writing faulty and unrealistic discharge summaries.

Another uncertainty comes in the form of borderline cases such as "pain", "fatigue", or "shortness of breath". Should these examples be included as diseases, or should they be considered symptoms and left for a different section?

A more robust evaluation would involve a clinical professional annotating the gold standard and drafting the example summaries. However, this project still demonstrates the viability of AI parsing technology, even if it's just as a proof of concept. Even with non-expert input, this project sets the foundation for future, more rigorous validation. Additionally, the uncertainty in expected outputs is not prevalent in every sample, and when it is, it usually only brings into question one or two of the outputs. Most summaries include simple disease names and medications that take nothing but common sense to identify. Because of this, the evaluation appears largely valid, despite certain limitations.

F. *Key Takeaways*

The parser demonstrates strong recall, which is valuable in medical contexts where information-rich documents are often preferred. This tendency to include more information is reflected in the slightly lower precision that was shown, which suggests the need for improvements in contextual understanding, such as adding negation detection or relation extraction. Medication detection appears more consistent than diagnoses, likely because medication names are more standardized and less context dependent. In a real-world setting, this project would benefit from enhanced context modeling, additional fine-tuning, and expert-in-the-loop validation.

G. *Final Thoughts*

These results confirm the feasibility of using AI and NLP to extract structured medical data from discharge summaries. F1 scores are promising, as they exceed 0.80 across both entity types, which demonstrates the parser's capability in supporting clinical workflows and contributing to data standardization, easing the process of healthcare provider communication.

However, edge cases, ambiguous phrasing, and clinical nuances are pain points of this system and must be considered in further development. Model fine-tuning, expert-reviewed training data, and the addition of context-aware components will significantly enhance the parser's dependability and clinical utility.

V. Conclusion

This project has shown that pragmatically using AI-driven natural language processing to extract and normalize critical medical information from unstructured hospital discharge summaries is feasible and that it is technically sound. Using the realization of spaCy and SciSpaCy, the system is able to identify clinical entities with high efficiency and normalize them based on their UMLS counterparts, which is an important step towards improving interoperability and unhampered communication in healthcare data systems.

During performance evaluation, the parser demonstrated strong medication extraction capabilities; however, it also revealed limitations such as entity specificity, contextual understanding, and uncertainty of data. These illuminate areas in which the project needs improvement in the future, those areas being negation detection, better context modeling, and utilizing expert-reviewed datasets.

Although the system was tested with artificial summaries and without external clinical supervision, appropriate diagnoses and medications were retrieved with F1 values greater than 0.80. This indicates feasibility regardless of real-world deployment being contingent upon expert-in-the-loop validation, larger datasets, and domain-specific fine-tuning.

Ultimately, the proof of concept presented by this research is important, and it bolsters the ability of AI and NLP to bridge the gap between raw clinical text and interoperable medical data. Through continued refinement and feedback from clinical professionals, such tools as this parser could one day be instrumental to optimizing health information exchange, reducing provider burden, and enhancing continuity of care between medical providers.

VI. References

[1] R. M. Siepmann et al., "An Automated Information Extraction Model for Unstructured Discharge Letters Using Large Language Models and GPT-4," Healthcare Analytics, Jan. 2025, p. 100378. [Online]. Available: https://doi.org/10.1016/j.health.2024.100378

[2] D. F. Navarro et al., "Clinical Named Entity Recognition and Relation Extraction Using Natural Language Processing of Medical Free Text: A Systematic Review," International Journal of Medical Informatics, vol. 177, Sep. 2023, pp. 105122–105122. [Online]. Available: https://doi.org/10.1016/j.ijmedinf.2023.105122

[3] C. Y. K. Williams et al., "Physician- and Large Language Model–Generated Hospital Discharge Summaries," JAMA Internal Medicine, May 5, 2025. [Online]. Available: https://doi.org/10.1001/jamainternmed.2025.0821

[4] M. Jiang et al., "A Study of Machine-Learning-Based Approaches to Extract Clinical Entities and Their Assertions from Discharge Summaries," Journal of the American Medical Informatics Association, vol. 18, no. 5, Sep. 2011, pp. 601–606. [Online]. Available: https://doi.org/10.1136/amiajnl-2011-000163

[5] M. Neumann et al., "ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing," Proc. 18th BioNLP Workshop and Shared Task, 2019, pp. 319–327. [Online]. Available: https://doi.org/10.18653/v1/W19-5034

[6] T. Osborne et al., "Towards Inpatient Discharge Summary Automation via Large Language Models: A Multidimensional Evaluation with a HIPAA-Compliant Instance of GPT-4o and Clinical Expert Assessment," MedRxiv, Apr. 4, 2025. [Online]. Available: https://doi.org/10.1101/2025.04.03.25325204