COMPARATIVE ANALYSIS OF NLP TECHNIQUES FOR AUTOMATED MATCHING OF MEDICAL INTAKE FORMS TO THE FHIR DATA SCHEMA: EMBEDDING SIMILARITY AND LANGUAGE MODELS

By

Amrish T. Pipalia

CAPSTONE

Presented to the Department of Medical Informatics & Clinical Epidemiology and the Oregon Health & Science University
School of Medicine
in partial fulfillment of
the requirements for the degree of

Master of Science

September 2024

Table of Contents

ABSTRACT

This study explores the application of contemporary Natural Language Processing (NLP) methods to automate the matching of medical intake form fields to the Fast Healthcare Interoperability Resources (FHIR) data schema. We evaluated three text embedding models, three small parameter size large language models (LLMs), and three large parameter size LLMs on a set of ten manually coded medical intake forms. Results indicated that LLMs significantly outperformed vector embedding search, with smaller LLMs performing comparably to larger models. The highest F1 scores for FHIR resource and element matching ranged from 0.63 to 0.80.

Anticipated challenges for achieving higher performance and operational feasibility include the complexity of intake forms and the deeply hierarchical structure of the FHIR schema with limited representations. Despite these limitations, semi-automated or human-in-the-loop implementations could prove viable. Future research directions include investigating different prompting techniques to enhance overall performance, exploring multimodal LLMs for visual form understanding, employing hierarchical matching methods for deeper schema matches, and auto-generating FHIR queries to assess retrieval capabilities.

As healthcare continues its digital transformation, efficient, accurate, and adaptable matching tools are essential for interoperability. This study represents a step towards leveraging AI to tackle the challenge of healthcare data interoperability, with potential benefits not only for reducing patient data entry burden, but also for improving patient care through better data exchange, reducing administrative burden and facilitating health research

INTRODUCTION

The complexity of matching one schema to another has long been a challenge in information systems, particularly in healthcare where data interoperability is crucial for patient care and research. Patients often express frustration at repeatedly filling out similar forms across different healthcare providers, highlighting a patient-centered motivation for improved data integration. This project aims to address this challenge by evaluating modern natural language processing (NLP) methods for the automated matching of medical intake form fields to the Fast Healthcare Interoperability Resources (FHIR) data schema.

The evolution of health data standards has been a journey spanning several decades, marked by efforts to standardize and improve the exchange of healthcare information. Early standards such as Health Level Seven (HL7) version 2, introduced in the 1980s, paved the way for more sophisticated approaches. The development of HL7 version 3 and the Clinical Document Architecture (CDA) in the early 2000s represented significant advancements, but also revealed the need for a more flexible and implementable standard. This realization led to the creation of FHIR, which combines the best features of previous standards with modern web technologies. (1)

FHIR has emerged as a pivotal standard in healthcare interoperability, offering a flexible, extensible framework for exchanging healthcare data. Its importance is underscored by its rapid adoption across the healthcare industry. Barker et al. (2) conducted a survey in 2022 showing that at least 50% of digital health companies are making extensive use of FHIR in the EHR integrations, increasing to 89% for companies using only standards-based application programming interfaces (APIs). This widespread

adoption reflects the growing recognition of FHIR's potential to enhance data sharing and improve patient care.

Despite its promise, the implementation of FHIR continues to face persistent challenges. (3,4) These include the complexity of matching legacy systems to FHIR concepts, varying levels of FHIR adoption across healthcare organizations, and the need for ongoing maintenance and updates to keep pace with evolving healthcare needs. Additionally, the lack of standardized terminologies and value sets across different healthcare systems poses a significant barrier to seamless data exchange.

Medical intake forms represent a particularly challenging but poignant area of data interoperability. They are ubiquitous in new patient visits and are a critical juncture where patient data is collected and potentially integrated into electronic health records. These forms typically encompass registration processes and gather comprehensive history and physical data. A literature search found no descriptive statistics or terminologies or taxonomies to describe medical intake form data in a standardized way. Due to their usual use on physical paper and the lack of a standard, they represent unstructured healthcare data. The only related study I found used the Delphi technique to achieve consensus on integrated patient history intake questions, but I found no work on this topic since then. (5)

To address the challenge of integrating diverse data sources, several related concepts have been used in the field of data integration. Schema matching refers to the process of identifying correspondences between elements of two data schemas. (6) Schema matching, on the other hand, focuses on the data transformations necessary for final data exchange after matching is complete. (6) Entity matching focuses on

identifying records that refer to the same real-world entity across different data sources. (7) Ontology matching seeks to establish correspondences between concepts in different ontologies. (8) These and other interrelated concepts have been used in the literature to describe the general challenge of matching one dataset to another. Schema matching is the primary focus of this study.

The history of automated schema matching methods reveals an evolution from simple string-matching techniques to sophisticated machine learning approaches. Early methods relied on lexical similarity and structural matching, which proved effective for simple schemas but struggled with complex, heterogeneous data sources. (9,10) The advent of complex statistical techniques and machine learning techniques, including supervised and unsupervised learning algorithms, marked a significant advancement in the field. (11) These methods can be highly effective. (12) However, they often require extensive expert domain knowledge or training data, making them less capable at adapting to novel data schemas, or "schema-agnostic" matching.

Due to recent advances in natural language processing (NLP), particularly vector embedding representations of text and pre-trained large language models (LLMs), schema-agnostic and other related agnostic matching have become a possibility. (13) Recent research demonstrates the power of pre-trained language models (LMs) for automating schema and entity matching, even without task-specific training data. Caulfield et al. (14) introduced the Structured Prompt Interrogation and Recursive Extraction of Semantics (SPIRES) method uses LMs to populate knowledge bases by recursively extracting information according to user-defined schemas. Their F-scores ranged from the 30s to 40s using a GPT-3.5-turbo. Zhang et al. (15) proposed a Learned

Schema Mapper (LSM) using the BERT pre-trained language model for matching customer schemas to various industry standards. They showed top-1 prediction accuracies of 0.65 to 1.00 depending on the industry standard and significant improvements for top-3 predictions. They further combined the method with a human feedback process and showed that, together, the approach could save up to 81% of labeling costs compared to manual labeling. Teong, Soon, and Su (7) also developed an approach based on BERT for entity matching. They achieved F1 scores ranging from .83 to 1.00 on a range of datasets. These approaches demonstrate the potential viability of language models for schema-matching in rapidly evolving domains and data-scarce situations.

In the realm of healthcare, my literature review found that schema matching research has focused on ontology matching using syntactic and semantic similarity search using non-transformer-based embedding models. (8,16–18) One intriguing study by Kiourtis et al. (16) algorithmically created ontologies from structured healthcare data, mapped it with semantic relationships, and then matched it with a semantically mapped FHIR schema. They achieved a remarkable 0.93-0.95 F1 score for schema matching. However, these results were possible because they leveraged the meaningful coded relationships that already exist in structured healthcare data. This approach would be unsuccessful with unstructured healthcare data.

Unstructured healthcare data would be more amenable to NLP methods with a wide breadth of general language understanding, such as recently developed transformer-based text embedding models and LLMs. To the best of my knowledge, transformer-based text embedding models or large language models (LLMs) have not yet been systematically evaluated for schema matching against standardized healthcare data

schemas in the existing scientific literature. This study aims to address this gap by evaluating the effectiveness of state-of-the-art NLP methods, including large language models, for matching paper medical intake forms to the FHIR data schema.

This research focuses on three distinct NLP methods: similarity search with transformer-based text embedding models, small LLMs with 7-8 billion parameters, and large LLMs with 70 billion or more parameters. By comparing their performance, we aim to identify the most effective approach for this specific use case and provide valuable insights for future developments in healthcare informatics. I also hope to contribute to the broader goal of reducing documentation burden in healthcare and improving the patient experience. The potential for automated schema matching to streamline data integration could significantly reduce the need for repetitive data entry, alleviate provider burnout, and enhance the overall efficiency of healthcare delivery.

METHODS

Medical Intake Form Coding

The first step of the study was collecting medical intake forms from local clinics and expanding to forms available online from health clinic websites when local options were exhausted. Forms were manually coded onto spreadsheets with multiple columns partitioning the text for each field: form title, subject heading, instructions, field name, field sub-name, and input type/options. The last section included any multiple-choice options provided for that field on the form.

Forms were labeled in multiple ways for analysis:

1. Form sections: Chief Concern, Registration, Medical History, Family History, Surgical History, Social History, OB/GYN History, Allergies, Medications, Health Maintenance, Review of Systems, Pharmacy, and Other.

2. *Importance*: Fields were labeled as *important* if the information was consistent over time, typically captured during routine visits, and helpful for patient autofill. Some examples of *unimportant* fields are Review of Systems questions, as they only relate to the visit at hand, and some Social History questions such as number of alcohol drinks per week, as these may change quickly over time

3. Match: Fields were labeled as a *match* or a *mismatch* to the FHIR schema. (19) Some fields did not have any reasonable FHIR matches, even accounting for LOINC code use via the Observation resource.

4. Multiple representation: If the field was a match, then it would be further labeled as having multiple valid representations or not. This meant that the field matched multiple possible elements in FHIR or multiple codes in the terminology linked to by the FHIR element (e.g., LOINC), and that no one match was clearly preferred over another.

5. FHIR schema match: Fields with a schema match were labeled with the most appropriate FHIR concept. FHIR concepts are structured primarily as *resources* that each contain multiple *elements* (e.g. the `Patient` resource has `name` and `birthDate`). Thus, fields were matched with a resource and the minimum elements necessary to represent the information requested by the field. For example, retrieving current medications requires retrieving of the

`effectivePeriod` element to ensure the medication is still active in addition to the `medication` element to obtain the name. This has been termed a *complex* match, whereas a 1:1 match may be called an *elementary* match. (13)

Several choices were made in choosing FHIR schema matches for the study. First, it is important to note that while schema matches were made for the two primary layers of the FHIR schema, deeper layers exist. Analysis of FHIR elements found almost all to be *non-terminal* (99%, see Results)—data types that contain multiple other nested elements in one or more hierarchies. Finer-grained schema matching, however, could essentially use multiple iterations of the approach in this study and should be a focus of future work.

For the `Observation` resource, certain choices had to be made in response to multiple valid representations. While many form fields have relevant LOINC codes, there may be multiple appropriate LOINC codes to match to, each with a potentially different nominal datatype. LOINC codes with Boolean values are also coded as a nominal datatype instead of a true Boolean, and there are multiple LOINC answer lists that operate as Boolean. Furthermore, official recommendations from FHIR documentation discourage using valueBoolean. Thus, any expected Boolean responses from a LOINC term referenced via the `Observation` resource were given an element label of `valueCodeableConcept`.

Decisions also had to be made when one form field needed information from another field to be properly referenced. For example, if a current medication list is requested with name and dosage, the dosage needs to be associated with the name for retrieval. However, the converse is equally valid: the name can be retrieved from the

dosage. In both cases, though, the `effectivePeriod` element should be checked to ensure only current medications or dosages are being retrieved. In these situations, I chose to match both fields to the common required element but did not match the mutually retrievable element. In contrast, a set of fields asking for the date of onset of specific health conditions is not retrieving all health conditions. Thus, matching to the condition under question is a prerequisite to matching to a valid onset date because there are onset dates for conditions not being asked about.

Finally, efforts were made to ensure consistent form coding between forms. Previous forms were repeatedly updated as new forms introduced uncertainty in labeling, such as in assigning sexual orientation under Social History or placing it in Other or a new category. Automated checks were written to ensure that all resource and element labels were exact matches to set of relevant FHIR data (see below). However, no additional reviewers were involved to help validate more subjective aspects of labeling.

FHIR resource and element extraction

In line with the goal of validating my approach on a simpler classification problem than the complete task, I used only set of 15 FHIR resources that medical intake form fields matched to (see Results) as the scope of targets for matching. Descriptions of these FHIR resources and their elements were derived from official FHIR documentation for version 5.0.0. The official FHIR documentation offers multiple forms of representation of the FHIR data standard, such as the JSON schema and the Structure Definition. After some pre-testing on a subset of intake form data, the JSON schema was chosen due to its greater performance.

Each resource and element was extracted alongside its narrative description provided in the JSON schema under the "description" property (Appendix B). Additionally, the following elements common to most resources were excluded for various reasons: `id, meta, implicitRules, language, text, contained, extension, modifierExtension, resourceType`. Most serve only administrative purposes unrelated to clinical workflows. `Text` is unstructured, and thus an inappropriate match. `Extension` is not a guaranteed match as extensions are not a requirement of the standard, so these were excluded as well.

Vector text embedding process

After pre-testing a wide range of text embedding models, I chose three text embedding models due to their efficacy, popularity, and to capture both open-source and closed-source types:

1. bert-large-cased: based on the original BERT (Bidirectional Encoder Representations from Transformers) architecture, this open-source model by Google uses a bidirectional transformer and is primarily trained for question-answering and text classification, not specifically for text embeddings. (20)
2. gte-large-en-v1.5: an open-source model by Alibaba with a bidirectional transformer-based architecture (based on BERT) optimized for general text embedding. (21)
3. text-embedding-3-large: a commercial model by OpenAI with a unidirectional transformer-based architecture optimized for text embeddings. (22)

Open-source text embedding models were obtained from HuggingFace repositories. The commercial model was accessed via the OpenAI API. In addition, the common basic search algorithms BM25-Okapi and its BM25L variant were tested on a limited dataset for comparison but were not included in the final evaluation due to very poor performance.

Each resource and resource along with their description was passed to the embedding model and the resulting vector embedding was stored into a vector database (ChromaDB). Form fields were converted to single strings. Different permutations of form field text were tested, and the most effective description excluded the form title and included the rest in concatenated form. This was then passed to the embedding model and stored as mentioned. Document and query prompts were trialed as recommended for better retrieval performance in many embedding model instructions, but using no prompt consistently outperformed using either recommended or custom prompts. Separate vector database collections were made for the `resources`, each set of `elements` corresponding to each `resource`, and the form fields.

The resource collection was then queried with each form field embedding. The top 7 results were recorded based on cosine distance. A result count of 7 was chosen because by random chance out of a total of 15 resources in the embedding collection has a near-50% chance of containing the correct resource label (see Results for number of unique target resource matches). The element set collection corresponding to the top resource match (i.e., the resource with the lowest cosine distance) was then queried with the same form field embedding. No other similarity search metrics were tested.

LLM-inferencing process

Preliminary testing as done on a wide range of models. This included very small, locally-hosted open-source models such as Qwen2-1.5B (1.5-billion-parameter), Gemma2-2B, and orca-mini-3B. However, these showed very poor performance and were often unable to provide consistent enough output to allow scripted analysis. Other models in the 7-12B, 70B, and large commercial (e.g. 400B+) range were tested on a limited dataset. The highest-performing open-source models and a representative commercial model were chosen for analysis.

The three families of state-of-the-art instruction-tuned models were chosen:

1. Alibaba's Qwen2 (7B and 72B)
2. Meta's Llama-3.1 (8B and 70B)
3. OpenAI's GPT4o (mini and standard versions)

Qwen2 7-billion parameter (7B) and 72B models were run on a serverless endpoint hosted by Alibaba Cloud's AI platform. Llama-3.1-8B and -70B were run on a serverless endpoint hosted by Azure AI Studio. GPT-4o-mini (2024-07-18) and GPT-4o (2024-08-06) were accessed via OpenAI's API.

For the models accessed via commercial clouds, stable version endpoints were utilized where available, but only OpenAI offered this. All models' temperature parameter was set to 0 for the study to promote deterministic responses. Top P, if provided as a parameter, was not adjusted from default values. Specific seed values were not set. Safety filter levels, if available, were set as low as possible.

For inferencing, simple prompts were used (see Appendix B). A wide range of prompting strategies were considered, but the variability that complex prompts would introduce outweighed their benefits for evaluation and generalizability. The user prompt briefly gave a context for the task, provided the resources or elements and their descriptions to choose from, and provided a form field with basic instructions. For element predictions, the model was instructed to choose all the elements required for answering the form field. The system prompt provided directions for formatting the response in a JSON format and provided the JSON schema in the Qwen2 and Llama 3.1-8B models. The Llama 3.1-70B and OpenAI APIs offered a parameter-based JSON schema output method. Certain model's outputs required post-processing to obtain valid JSON objects. No knowledge of FHIR schema matching was injected into the prompt—other than the list of resources and elements provided—to allow for better evaluation of the model's underlying language skills and knowledge.

As in the vector embedding search process, the model was first asked to match the form field to a resource. Then, depending on the answer, a separate prompt gave it a set of that resource's elements to match to. If the resource prediction was incorrect, the element prediction prompt for that form field was skipped. Although providing the model with the true resource label's elements in those cases would have provided additional element matching data for analysis, the practical implications of those data are uncertain if the model cannot first identify the appropriate resource. This strategy also cut down on inferencing costs and time.

Analysis

Form field data and labels were combined with prediction data and cleaned. I paid particular attention to a few variables that would have significantly influence statistical analysis. Except for form descriptive statistics, all form fields labeled as a schema mismatch were excluded from analysis.

The element prediction lists were cleaned in two specific ways. To remove extraneous but not incorrect labeling, label formats were modified. For example, some models often included the resource name as a prediction (e.g., "condition" for the `Condition` resource), the full element name reference (e.g., "Condition.code" for the `code` element), or subcomponents of elements (e.g. "reaction.substance" for the `reaction` element in the `AllergyIntolerance` resource). These labeling issues were corrected, leaving just the intended element. This decision was made in part because the issues don't disqualify the predictions as potentially usable labels and because they were simple to programmatically correct, as might be done in an operational deployment. The other cleaning step was to remove any status elements (`active`, `status`, `clinicalStatus`, `verificationStatus`) from element predictions unless they were included in the true label. Almost all fields require querying a status element for proper retrieval, but this part of a query would be the same for almost all queries and thus could be set up programmatically (e.g., retrieve `Patient.[x]` if `Patient.active` is True).

Descriptive statistics for forms were calculated. Due to the small number of forms in the study, median and interquartile range is reported for central tendency and spread, respectively. For each model, I calculated recall, precision, and F1 for resources and elements separately. To calculate element metrics for embedding models, their element

predictions were truncated to the number of element labels in the true set for each form field.

For resources, multi-class weighted averages of these metrics were used due to the significance of certain labels and due to class imbalances. Weighted averages were calculated by taking the metrics (e.g. recall) calculated for each class of labels, weighting them by their relative frequency compared to other classes. Because element classification problem could involve multiple labels and because each set of labels was insignificant without their resource classification, multi-label micro-averaged metrics were used. Micro-averaging treats each instance of a label as an individual binary classification problem without regards to the label's name. The calculation then proceeds as usual; for example, for recall, the sum of true positives divided by the sum of true positives and false negatives. Recall, precision, and F1 were further stratified by true resource label, form section, field match / mismatch, and multiple representation.

For certain chart depictions, a combined F1 score was used. For this metric, the unweighted mean of the resource and element F1 scores were calculated. The unweighted mean was used because both resource and element classification are equally important and differences in weighting would be artificially related to the experimental setup—excluding element prediction if the resource was misclassified.

Due to the propensity of LLMs to provide answers out of context of the provided options and to hallucinate, error metrics were calculated. Resource, element, and total error rates were calculated. These included any reason for an error, including legitimate FHIR label predictions outside of the given options, syntax differences, and

hallucinations. True hallucination rates were also calculated, excluding the other sources of error.

Embedding model results underwent further analysis. Prediction cosine distances were converted to cosine similarities and used to calculate the receiver-operating-characteristic (ROC) curve and its area under the curve (AUC). Youden's index was used to identify the optimal threshold. This allowed assessment of the potential strength of cosine similarity as a classifier for a successful label prediction. Additionally, to evaluate how many predictions are required to capture the true label, I calculated resource and all-element recall by number of top-*k* predictions. Notably, however, using all-element recall, will make this metric appear worse than the multi-class element F1 scores.

Finally, primary metrics were recalculated using only form fields labeled as *important*. This allowed interpretation of performance specifically for auto-filling medical intake forms. However, the *importance* label does not represent fields that would be useful for exporting completed form field information into a FHIR data structure.

With regards to software tools used, all processing and analysis was done in Python (version 3.9.6) using the following open-source packages: NumPy (23), pandas (24), matplotlib (25), seaborn (26), scikit-learn (27), Rank-BM25 (28), transformers (29), and chroma (30). Some visualization was done using Microsoft Excel for Mac. ChatGPT-4o and -4o-mini were used to provide guidance on form coding and writing python code. No recommendations were used without further referencing FHIR or python package documentation and performing manual testing of code. Claude-3.5-Sonnet was used to help revise text for writing this paper. No generated text was used without manual review and editing.

RESULTS

Characteristics of Medical Intake Forms and FHIR Schema Matches

The dataset consisted of 10 medical intake forms with a median of 138 fields per form (IQR: 83-178) (Table 1). Of these fields, only 3% (median) were mismatches to FHIR resources. A majority (59%, median) of the fields were classified as *important* for patient autofill purposes. Notably, 44% (median) of the fields had multiple representations, mostly due to LOINC code use in the `Observation` resource (97%). The dataset exhibited a high rate of *complex* element matches (69%, median) and *non-terminal* element matches (99%, median), indicating the complexity of the schema matching task. The bulk of form fields

Table 3. Intake form characteristics

	Median [IQR]
Forms (n)	10
Fields	138 [83-178]
Mismatch	3% [1-3]
Important	59% [49-70]
Multiple representation	44% [21-52]
Complex element match	69% [61-80]
Non-terminal element	99% [98-99]

Table 1. Form section distribution

Form Section	Median (%)	IQR
Allergies	2%	1 - 2
Chief Concern	1%	1 - 1
Family History	8%	6 - 18
Health Maintenance	10%	9 - 13
Medical History	9%	6 - 20
Medications	3%	2 - 3
OB/GYN History	1%	1 - 7
Other	1%	1 - 2
Pharmacy	2%	1 - 2
Registration	7%	5 - 18
Review of Systems	33%	33 - 46
Social History	23%	16 - 26
Surgical History	2%	1 - 2

Table 2. FHIR resource match distribution

Resource	Median (%)	IQR
AllergyIntolerance	2%	1 - 2
CareTeam	2%	1 - 3
Condition	4%	2 - 11
Coverage	8%	5 - 9
DiagnosticReport	6%	5 - 7
DocumentReference	1%	1 - 1
Encounter	2%	1 - 2
FamilyMemberHistory	8%	5 - 18
Immunization	4%	3 - 8
MedicationStatement	3%	2 - 3
Observation	44%	31 - 64
Patient	5%	3 - 11
Procedure	2%	1 - 3
RelatedPerson	7%	7 - 7
ServiceRequest	1%	1 - 1

in the intake forms were from Review of Systems and Social History questions with median proportions of 33% and 23%, respectively (Table 2). There was notable variability in the sizes of the Registration and Family History sections, ranging from about 5 to 18%.

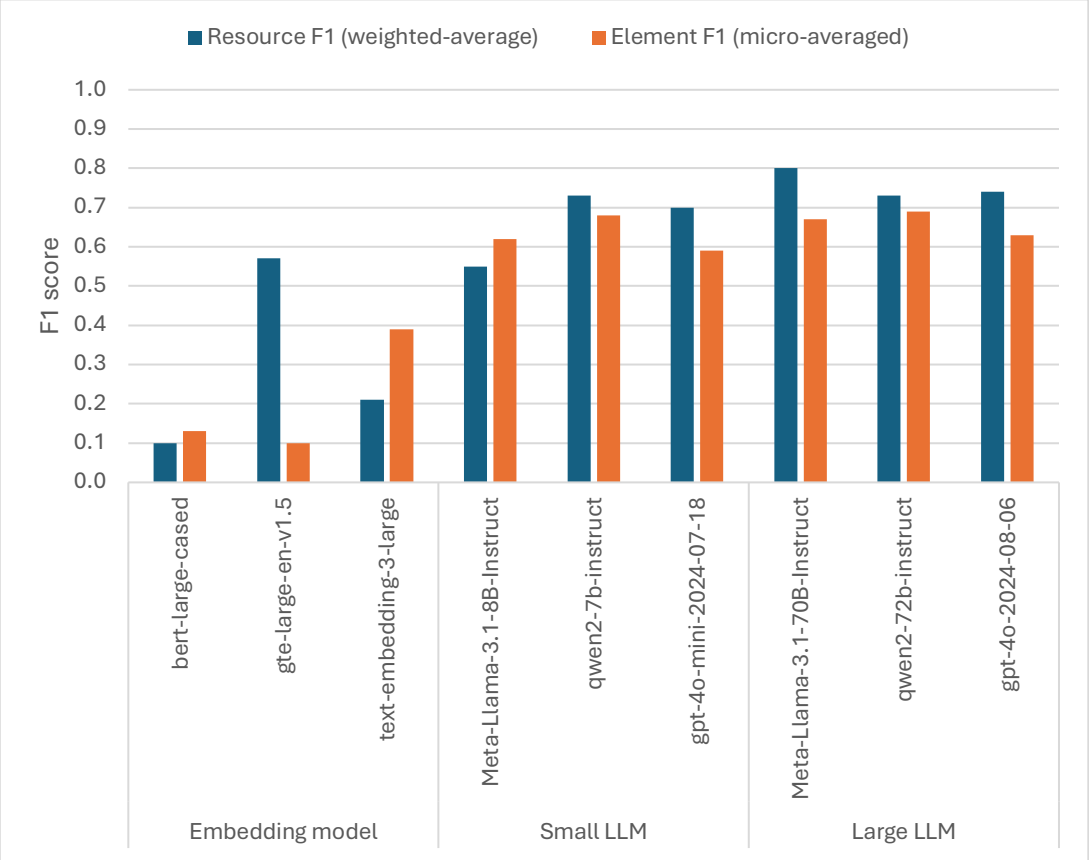
Unsurprisingly, the FHIR resource matches had a similar distribution (Table 3). The Review of Systems and Social History sections matched almost exclusively to the `Observation` resource, making it the most predominant at a median of 44%. Although the `Coverage` resource was relatively high compared to the others, this was reflective of a select couple of forms that included many questions on insurance coverage. Form fields matched to a total of 15 unique resources. Although a count of 10 forms still left some variability, the content represented in each form was largely saturated. Schema matches from the first 2 forms captured almost all the resources matched to in the rest of the forms. Coverage was added in the 4th form, and ServiceRequest in the 10th.

Overall Performance Trends

Across all models, the weighted-average F1 scores for resource prediction ranged from 0.10 to 0.80 (Figure 1). Embedding models were highly variable in performance, with gte-large-en-v1.5 having a relatively high resource F1 of 0.57, but a low element F1 at 0.10; in contrast, text-embedding-3-large had only a resource F1 of 0.21, but a higher element F1 at 0.39. Small LLMs easily outperformed embedding models, particularly in element matching. Except for Qwen2-7b-Instruct, which performed on par with much larger LLMs, large LLMs generally outperformed small LLMs in both resource and

element prediction. Meta-Llama-3.1-70B achieved the highest resource F1 score of 0.80, while Qwen2-72b-Instruct achieved the highest element F1 score at 0.69, although Llama

Figure 1. Resource and element F1 scores by model



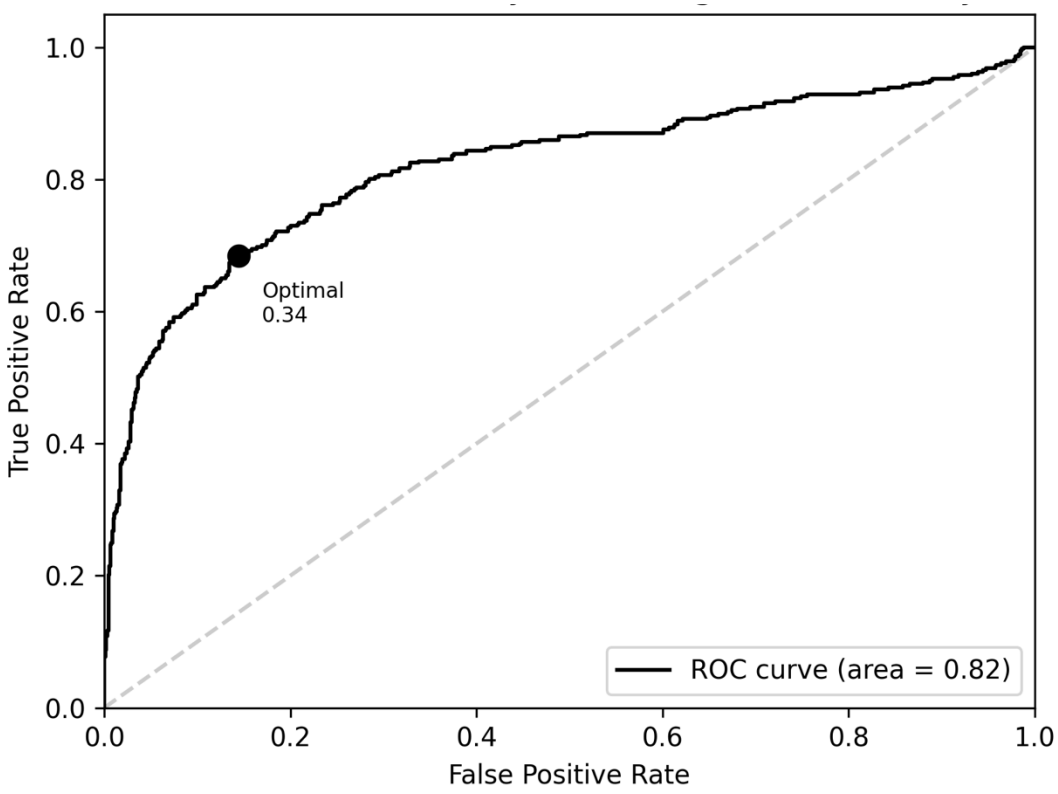
was not far behind at 0.67.

Embedding model cosine similarity metrics

The text-embedding-3-large model scored the highest on using cosine similarity as a classification predictor. Figure 2 shows the model's receiver operating characteristic (ROC) curve for resource prediction using cosine similarity. The area under the curve (AUC) of 0.82 suggests that cosine similarity could be a usable predictor of a correct resource match. The optimal threshold for classification, as determined by Youden's

index, is 0.34. The gte-large-en-v1.5 and bert-large-cased had poor AUCs of 0.51 and 0.44.

Figure 3. ROC curve for resource classification using embedding cosine similarity



Figures 3 and 4 illustrate the recall performance for increasing numbers of top-*k* similarity search classifications for the text-embedding-3-large model. Resource recall starts higher than random chance and improves at a faster rate than random chance as more top predictions are considered, reaching 0.93 for the top 7 out of a total of 15 possible resource

Figure 2. Top-k similarity search resource match prediction

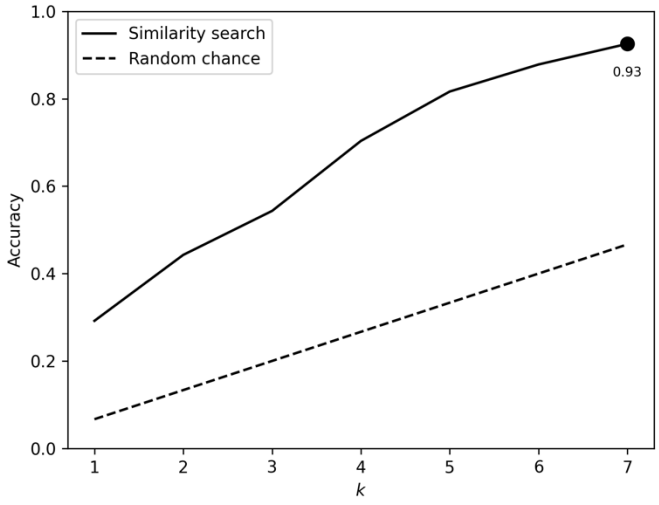
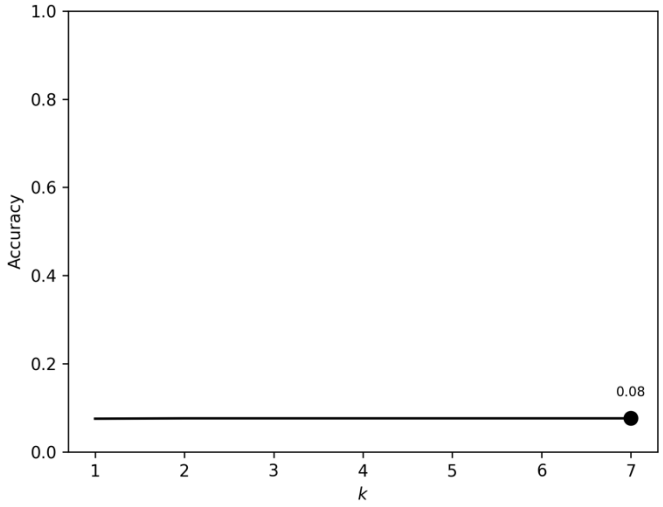


Figure 4. Top-k similarity search element match prediction

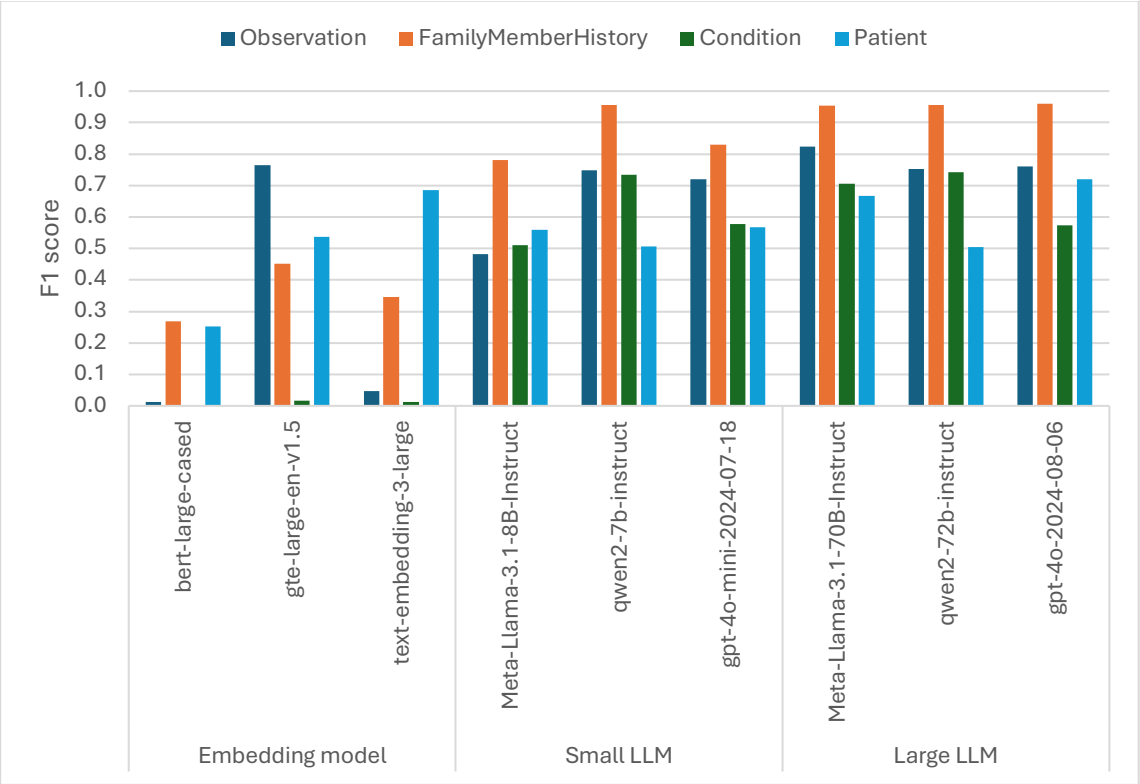


predictions. However, element recall remains very low (0.08) even when considering the top 7 predictions despite this model's relatively higher element F1 score. This is in part due to the way recall was calculated for top-*k*, requiring recall of all elements f or successful classification. In comparison, gte-large-en-v1.5 sc ored similarly for both resources and elements, but bert-large-ca sed did poorly on both, only achieving 0.42 resource recall at top 7.

Performance Across Different Resources

The four most frequent FHIR resources in the aggregated dataset were `Observation` (652, 50.4%), `FamilyMemberHistory` (134, 10.4%), `Condition` (122, 9.4%) and `Patient` (111, 8.6%) with a sharp drop-off thereafter for Immunization (45, 3.5%). Performance varied significantly across the top 4 resources (Figure 5). For `Observation`, F1 scores ranged from 0.01 to 0.82, with large LLMs performing best. `FamilyMemberHistory` had consistently higher F1 scores (0.27 to 0.96) across all model categories, with large LLMs achieving near-perfect performance. `Condition` showed variable performance (F1 scores 0.0 to 0.74), with large LLMs showing the most consistent results. The `Patient` resource had moderate to high performance (F1 scores 0.26 to 0.72), with large LLMs again demonstrating superiority.

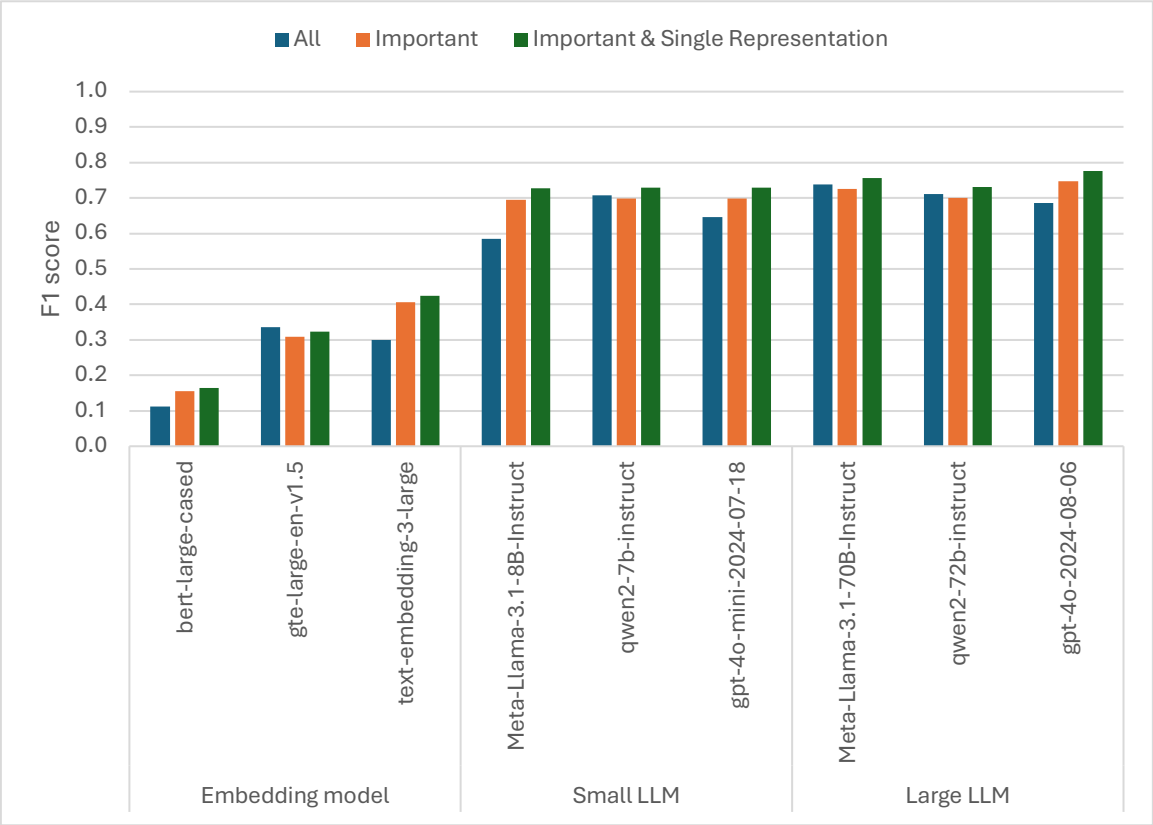
Figure 5. Combined F1 scores for top 4 most frequent resources by model



Performance by Field Characteristics

When considering only *important* fields, the F1 scores improved for most models, more for embedding models and small LLMs (Figure 6). Further restricting the analysis to *important* and single-representation fields resulted in slight performance gains across all model categories.

Figure 6. Combined F1 scores for all fields versus only 'important' versus both 'important' and single representation



Error Analysis and Other Statistics

Using Qwen2-72B-Instruct as a representative large LLM, the very low error rates were observed (Table 4). Other models had similarly low error rates. Llama had significant variability in label naming convention, but most were caught by element label cleaning.

Table 4. Miscellaneous statistics for Qwen2-72B-Instruct

Prompt tokens	2574202
Completion tokens	23220
Status element miss rate	0.68
Resource error rate	0.00
Element error rate	0.01
Total error rate	0.00
Resource hallucination rate	0.00
Element hallucination rate	0.01
Total hallucination rate	0.00
Perfect match rate	0.34

Running the 10 forms and all their fields with the prompts amounted to about 2.5 to 2.9 million input tokens depending on the model. Qwen2-72B had a status element miss rate of 0.68, and the other models were all within the 0.60 to 0.70 range as well. Llama-3.1-70B-Instruct highest perfect match rate at 0.55.

Confusion Matrix Analysis

Confusion matrices for resource classification provide insight into the failure modes of various classification errors. Embedding models frequently misclassified `Observation` as other resources, particularly `Patient` and `Condition` (Appendix A, Figure 7). They also used `FamilyMemberHistory`, `Patient`, and `MedicationStatement` classifications erroneously. At the level of form fields, medical history and social history questions were often confused with `FamilyMemberHistory` by these models.

Small LLMs showed markedly improved discrimination but still exhibited some confusion between similar resources, particularly `Condition` versus `Observation` and `Patient` versus `Observation` (Appendix A, Figure 8). Large LLMs demonstrated the clearest diagonal pattern, although only slightly better than small LLMs (Appendix A, Figure 9).

Importantly, the confusion matrices also show that the LLMs make resource classifications that were outside of the prompted set of label options. For example, `Organization` was labeled instead of `CareTeam` for capturing a patient's preferred pharmacy. This indicates that they were relying too heavily on their training data for

some classifications. Although hallucination was rare, one OB/GYN history question was labeled to a Pregnancy resource, a resource that does not exist in the FHIR standard.

DISCUSSION

The results of this study provide valuable insights into the potential of modern natural language processing (NLP) methods for automating the matching of medical intake form fields to the FHIR data schema. This discussion will explore the implications of these findings, their context within the broader field of healthcare informatics, and the challenges and opportunities they present for future research and implementation.

Performance of LLMs

One of the more striking findings of this study is the comparable performance of relatively small LLMs such as Alibaba's Qwen2-7B and Meta's Llama-3.1-8B to their larger counterparts, and their notably superior performance compared to embedding models. This is particularly promising for the future practical feasibility of deploying these models in a privacy-compliant manner on locally-hosted hardware for healthcare data schema matching. The ability to achieve high-quality results with smaller models could significantly reduce the computational resources required, making implementation more accessible to a wider range of healthcare organizations.

However, it's important to note that while the performance scores obtained in this study are encouraging, they do not yet appear high enough for immediate operational use without further refinement. The best-performing models achieved F1 scores of 0.80 for resource prediction and 0.69 for element prediction, which, while impressive, still leave room for improvement in a field where accuracy is paramount. This suggests that

additional strategies may be necessary to bridge the gap between current performance and operational requirements.

One potential avenue for improvement lies in prompt engineering strategies with FHIR-specific knowledge injection. By incorporating domain-specific knowledge into the prompts used to query the models, it may be possible to significantly enhance their performance. This approach could involve providing more context about FHIR resources and their relationships or including examples of correct matches for similar fields. While this would reduce the generalizability of the approach to some extent, it could still provide meaningful levels of partial automation, potentially offering a balance between performance and flexibility.

Moreover, the rapid pace of progress in artificial intelligence suggests that model performance is likely to improve in the near future. As new architectures and training techniques are developed, we can expect to see even more capable models.

Comparison of LLM Models

The study's comparison of different LLM models revealed interesting nuances in their behavior. For instance, the Llama models, while performing well overall, did not follow instructions as reliably as some other models, often providing resource names and element subcomponents in element predictions. This highlights the importance of model-specific considerations in deployment scenarios.

Interestingly, the issue with Llama's output format was addressable through a simple algorithmic post-processing step. This demonstrates the potential for combining LLM outputs with rule-based systems to enhance overall performance and reliability. It

also suggests that model-specific prompt engineering could potentially preclude such issues altogether, further improving the efficiency of the matching process.

The variability in model behavior underscores the need for careful evaluation and selection when choosing an LLM for a specific task. While overall performance metrics are important, factors such as instruction adherence, output consistency, and the ease of post-processing can significantly impact the practical utility of a model in real-world applications.

Challenges and Limitations of LLM Approaches

While LLMs showed promising results in this study, several challenges and limitations were identified that warrant further consideration:

1. Overreliance on Training Knowledge: LLMs may sometimes rely too heavily on their training data, which can be problematic when dealing with rapidly evolving standards like FHIR. If a model's training data doesn't include the most up-to-date FHIR specification, it may make outdated or incorrect matches. This highlights the need for regular model updates or fine-tuning to keep pace with changes in healthcare data standards.

2. Hallucinations: Although rare in this study, LLMs can sometimes generate plausible-sounding but incorrect information, a phenomenon known as hallucination. Since the set of valid labels is limited and mostly unchanging, simple validation mechanisms could catch these errors. However, this requires creating custom algorithms that would in turn incur upkeep costs.

3. Instruction Adherence: Some models, particularly the Llama variants, had difficulty consistently following instructions regarding the format of their outputs. This necessitated additional post-processing steps to clean up the results. While this issue was manageable in the context of this study, it underscores the importance of clear, model-specific prompting and potentially the need for output validation and correction mechanisms in real-world applications.

4. Inappropriate Use of Extensions: Models often suggested the use of FHIR extensions for matching certain fields. While extensions are a valid part of the FHIR standard, their overuse can lead to interoperability issues. Ideally, matches should prioritize standard resources and elements where possible, reserving extensions for truly unique or organization-specific data points. Prompt engineering can help steer models away from extension elements. In this study, I removed extension elements from the list of elements to choose from, and this precluded most of them. More direct prompting, however, may work better.

5. Lack of Quantifiable Confidence: Unlike embedding models, which provide a similarity score that can be used as a proxy for confidence, most LLMs do not inherently provide a quantifiable measure of confidence in their predictions. This lack of a built-in confidence metric can make it challenging to implement partial automation strategies that rely on human review for low-confidence matches.

6. Performance Trade-offs: While LLMs generally outperformed embedding models in terms of accuracy, they are typically slower and more computationally expensive to run. This trade-off between performance and efficiency needs to be carefully considered in the context of specific use cases and available resources.

Embedding Model Performance and Potential

While embedding models generally underperformed compared to LLMs in this study, they still offer certain advantages that warrant further exploration. The study found that embedding models struggled with longer text inputs and repetitive text (such as form titles), suggesting that careful preprocessing and feature selection could potentially improve their performance.

The receiver operating characteristic (ROC) curve analysis for the text-embedding-3-large model, which achieved an area under the curve (AUC) of 0.82 for resource prediction, suggests that cosine similarity could be a usable predictor of correct resource match. This finding is particularly interesting as it points to the potential for using embedding models as a fast, initial filtering step in a multi-stage matching process.

The poor performance of embedding models on element matching was poor by comparison. The text-embedding-3-large model had a noticeably higher element F1 score than the others. Some of this may have been due to precision for certain elements. The top-*k* element classification results were also very poor. However, this metric was harsher because it did not consider partial element classification. Overall, the poor performance on elements could be due to the more nuanced distinctions required for element-level matching, which may not be well-captured by embedding models trained on standard corpora.

Future research could explore ways to leverage embedding model strengths while mitigating the weaknesses:

1. Two-stage Matching: Embedding models could be used for rapid initial resource prediction, followed by LLM-based element matching for the top-*k* resource candidates. This approach could reduce the computational load on the more expensive LLM component.
2. Confidence Thresholding for Human Review: The cosine similarity scores from embedding models could be used to implement a confidence threshold, above which matches are automatically accepted, and below which they are flagged for human review or passed to a more sophisticated model.

Challenges Inherent to Medical Intake Forms

The study also highlighted several challenges that are inherent to the specific use case of medical intake forms. These forms, while generally consistent in content, exhibit wide variation in format and structure. This variability introduces complexities that any automated matching system must navigate:

1. Format Diversity: Medical intake forms often employ nested selections, unusual table layouts, and sparsely marked spaces for information entry. This diversity can make interpretation difficult for both humans and machines, requiring a flexible approach to form field identification and categorization.
2. Granularity Variations: The same information may be requested at different levels of detail across forms. For example, one form might ask broadly about "any major health problems in your family," while another might request detailed information about "the name, age, and all conditions for each of your first-degree relatives."

This variation in granularity poses challenges for consistent matches to FHIR resources and elements.

3. Multi-atomic Fields: Many fields on intake forms ask for more than one atomic piece of information. For instance, a field might request a medication name, its dosage, and its frequency. In FHIR, these would typically be represented as separate elements. Splitting such fields for matching purposes often requires referencing another field's entry for a correct match, adding complexity to the matching process.

4. Contextual Interpretation: The meaning of certain fields may depend on their context within the form. For example, a field labeled "Name" under an "Emergency Contact" section has a different semantic meaning than a similar field under a "Patient Information" section. Accurate matching requires understanding this context.

These inherent ambiguities in medical intake forms provide a compelling argument for the use of LLMs in the matching process. Multi-modal LLMs, which can process both text and visual information, may be particularly well-suited to understanding form fields at different levels of abstraction. Their ability to consider context and draw on broad knowledge bases could help them navigate these ambiguities more effectively than traditional rule-based matching engines.

FHIR-Specific Challenges

The study also revealed several challenges specific to matching to the FHIR data schema, which contribute to the complexity of the task:

1. Schema flexibility versus specificity: Almost all (97%) of the matches with multiple valid representations were in the `Observation` resource due multiple representations in the LOINC code set. While the flexibility provided by the `Observation` resource is beneficial for accommodating diverse use cases, it introduces ambiguity in the matching process and makes automation more challenging.

2. Boolean Value Representation: Despite the existence of a `valueBoolean` element in FHIR, official guidance discourages its use in favor of coded concepts. This is because many Boolean values are derived from coded lists (e.g., in LOINC) and many yes/no questions include additional options like "don't know" or "unknown". This nuance requires a deeper understanding of FHIR best practices beyond simple structural matching.

3. *Complex* matches: Many form fields require information from multiple FHIR elements for a complete representation. For instance, retrieving a patient's current medications requires not only the medication name but also checking the `effectivePeriod` to ensure the medication is still active. This one-to-many type of relationship between form fields and FHIR elements adds complexity to the matching process.

4. *Non-Terminal* Elements: The study noted that 99% (median) of matches were to *non-terminal* elements, meaning they are complex data types containing multiple sub-elements. This multi-level structure of FHIR resources adds depth to the matching task, requiring decisions not just about which resource and top-level

element to match with, but also which specific sub-elements are relevant. Techniques like the recursive SPIRES method may help to address this. (14)

5. Terminology Binding: Many FHIR elements are bound to specific terminology systems like LOINC or SNOMED CT. Accurate matching often requires not just identifying the correct FHIR element but also selecting the appropriate code from the bound terminology. This adds another layer of complexity to the matching process.

6. Dynamic Standards: FHIR is an evolving standard, with new versions introducing changes to resources and elements. This dynamism, while necessary for the standard's improvement, poses challenges for maintaining accurate and up-to-date matching over time.

7. Use of Extensions: While FHIR extensions provide flexibility for representing non-standard data, their use can lead to interoperability issues if not carefully managed. Deciding when to use standard elements versus when to employ extensions is not always straightforward and can impact the broader utility of the matched data.

These FHIR-specific challenges highlight the need for sophisticated matching approaches that go beyond simple string matching or structural alignment. Effective FHIR schema matching appears to require a deep understanding of the standard's intricacies, best practices, and the clinical context of the data being matched.

Limitations of the Study

While this study provides valuable insights into the potential of NLP methods for FHIR schema matching, it's important to acknowledge its limitations:

1. Limited Sample Size: The study analyzed only 10 medical intake forms. Although the content and FHIR matches appeared saturated, a larger sample size would provide more robust evidence of the generalizability of the findings across diverse form types and healthcare settings.

2. Lack of Secondary Coding: The absence of a secondary coder to validate the form coding and FHIR matches introduces the potential for bias or inconsistencies in the ground truth data. Future studies would benefit from employing multiple coders and assessing inter-rater reliability to ensure the validity of the manual matches.

3. Model Currency: The rapid pace of development in the field of AI means that the specific models evaluated in this study may be outdated within 6-12 months. While the general trends and comparative performance are likely to remain relevant, absolute performance metrics may improve with newer models.

4. ROC Curve Calculation: The receiver operating characteristic (ROC) curve was not optimally calculated for multi-class classification problems. A one-vs-rest calculation method would have been more appropriate and might have provided more accurate insights into the discriminative power of the cosine similarity metric.

5. LLM Consistency: I did not collect data on the consistency of LLM outputs across multiple runs. Given the potential for variation in LLM outputs even with temperature set to zero, information on output stability would be valuable.

6. Limited Exploration of Prompt Engineering: While I intentionally used simple prompts to evaluate the base capabilities of the models, model performance might have significantly benefitted from more sophisticated prompt engineering techniques.

7. Focus on English-language Forms: The study focused on English-language intake forms. The performance of these NLP methods on forms in other languages or multilingual settings remains an open question.

Addressing these limitations in future research will provide a more comprehensive and robust understanding of the potential for modern NLP methods in automating FHIR matching for medical intake forms.

Future Directions

The findings of this study, along with its identified limitations and challenges, point to several promising directions for future research and development in the field of automated FHIR matching:

1. Augmented Prompts: Developing more sophisticated prompts that incorporate FHIR-specific knowledge could potentially address some of the observed issues, such as the inappropriate use of certain elements, such as `valueBoolean` in the `Observation` resource.

2. Multimodal LLM Evaluation: Given the visual nature of many medical intake forms, evaluating the performance of multimodal LLMs that can process both text and images could yield interesting results. These models might be better equipped to interpret the structure and context of form fields, potentially improving matching accuracy.

3. Query Generation: Generating FHIR queries based on the matched elements could bridge the gap between schema matching and mapping (i.e., practical data retrieval). This would involve translating the matched FHIR resources and elements into executable queries and understanding the logical relations between them in order retrieve the right data from a FHIR server.

4. End-to-End Testing: Conducting end-to-end testing with a test FHIR server would provide a litmus test for practical viability of these methods.

5. Hierarchical Matching: Testing an iterative process, such as the SPIRES method, (14) for deeper matching into sub-elements until a primitive type is reached could provide more complete FHIR representations. This would address the challenge of *non-terminal* element matching identified in the study.

SUMMARY AND CONCLUSION

This study provides valuable insights into the potential of modern NLP methods, particularly large language models, for automating the matching of medical intake form fields to the FHIR data schema. The results demonstrate that even relatively small LLMs can perform comparably to larger models and significantly outperform vector embedding approaches in this complex task.

The challenges identified, both in terms of the inherent complexities of medical intake forms and the intricacies of the FHIR standard, underscore the need for more sophisticated and recursive matching solutions. While the current performance of these models may not yet be sufficient for fully automated operational use, the results are promising and suggest that with further refinement, NLP-based approaches could play a significant role in streamlining healthcare data integration.

The limitations of the study provide clear directions for future research, including expanding the dataset, exploring more sophisticated prompt engineering techniques, and evaluating the consistency and reliability of model outputs. Additionally, the identified future directions, such as multimodal LLM evaluation, query generation and end-to-end testing, and recursive hierarchical matching approaches offer exciting possibilities for advancing the field.

As healthcare continues to digitize and the need for interoperable data grows, the development of efficient, accurate, and adaptable matching tools becomes increasingly crucial. This study represents an important step towards leveraging the power of artificial intelligence to address the long-standing challenge of healthcare data interoperability. By continuing to explore and refine these NLP-based approaches, we can work towards a future where healthcare data flows seamlessly between systems, ultimately improving patient care, reducing administrative burden, and facilitating more effective health research.

REFERENCES

1. Imler TD, Vreeman DJ, Kannry J, Imler TD, Vreeman DJ, Kannry J. Healthcare Data Standards and Exchange. Clinical Informatics Study Guide [Internet]. 2016

[cited 2024 Sep 17];233–53. Available from: https://link.springer.com/chapter/10.1007/978-3-319-22753-5_11

2. Barker W, Maisel N, Strawley CE, Israelit GK, Adler-Milstein J, Rosner B. A national survey of digital health company experiences with electronic health record application programming interfaces. Journal of the American Medical Informatics Association [Internet]. 2024 Apr 3 [cited 2024 Sep 17];31(4):866–74. Available from: https://dx.doi.org/10.1093/jamia/ocae006

3. Setyawan R, Hidayanto AN, Sensuse DI, Kautsarina, Suryono RR, Abilowo K. Data Integration and Interoperability Problems of HL7 FHIR Implementation and Potential Solutions: A Systematic Literature Review. Proceedings - International Conference on Informatics and Computational Sciences. 2021;2021-November:292–8.

4. Ayaz M, Pasha MF, Alzahrani MY, Budiarto R, Stiawan D. The Fast Health Interoperability Resources (FHIR) Standard: Systematic Literature Review of Implementations, Applications, Challenges and Opportunities. JMIR Med Inform [Internet]. 2021 Jul 30 [cited 2024 Sep 17];9(7):e21929. Available from: http://www.ncbi.nlm.nih.gov/pubmed/34328424

5. Lindahl MG. Development of an integrated patient history intake tool: a Delphi study [Internet]. 2003 [cited 2024 Mar 12]. Available from: https://ir.library.oregonstate.edu/concern/graduate_thesis_or_dissertations/9880vt86j

6. Bellahsene Z, Bonifati A, Duchateau F, Velegrakis Y, Bellahsene Z, Bonifati A, et al. On Evaluating Schema Matching and Mapping. Schema Matching and Mapping [Internet]. 2011 [cited 2024 Sep 19];253–91. Available from: https://link.springer.com/chapter/10.1007/978-3-642-16518-4_9

7. Teong KS, Soon LK, Su TT. Schema-Agnostic Entity Matching using Pre-trained Language Models. International Conference on Information and Knowledge Management, Proceedings [Internet]. 2020 Oct 19 [cited 2024 Sep 17];2241–4. Available from: https://dl.acm.org/doi/10.1145/3340531.3412131

8. Kolyvakis P, Kalousis A, Smith B, Kiritsis D. Biomedical ontology alignment: An approach based on representation learning. J Biomed Semantics [Internet]. 2018 Aug 15 [cited 2024 Sep 19];9(1):1–20. Available from: https://link.springer.com/articles/10.1186/s13326-018-0187-8

9. Mecca G, Papotti P, Santoro D. A short history of schema mapping systems. In: Proceedings of the 20th Italian Symposium on Advanced Database Systems, SEBD 2012. 2012.

10. Modern Approaches to Schema Matching - DataMade [Internet]. [cited 2024 Sep 17]. Available from: https://datamade.us/blog/schema-matching/

11. Doan A, Domingos P, Halevy AY. Reconciling schemas of disparate data sources. In: Proceedings of the 2001 ACM SIGMOD international conference on Management of data [Internet]. New York, NY, USA: ACM; 2001 [cited 2024 Sep 17]. p. 509–20. Available from: https://dl.acm.org/doi/10.1145/375663.375731

12. Asif-Ur-Rahman M, Hossain BA, Bewong M, Islam MZ, Zhao Y, Groves J, et al. A semi-automated hybrid schema matching framework for vegetation data integration. Expert Syst Appl [Internet]. 2023 Nov 1 [cited 2024 Sep 17];229:120405. Available from: https://doi.org/10.1016/j.eswa.2023.120405

13. Sheetrit E, Brief M, Mishaeli M, Elisha O. ReMatch: Retrieval Enhanced Schema Matching with LLMs. 2024 Mar 3 [cited 2024 Sep 19]; Available from: https://arxiv.org/abs/2403.01567v1

14. Caufield JH, Hegde H, Emonet V, Harris NL, Joachimiak MP, Matentzoglu N, et al. Structured Prompt Interrogation and Recursive Extraction of Semantics (SPIRES): a method for populating knowledge bases using zero-shot learning. Bioinformatics [Internet]. 2024 Apr 5 [cited 2024 Mar 17];40(3). Available from: /pmc/articles/PMC10924283/

15. Zhang Y, Floratou A, Cahoon J, Krishnan S, Müller AC, Banda D, et al. Schema Matching using Pre-Trained Language Models. Proc Int Conf Data Eng. 2023;2023-April:1558–71.

16. Kiourtis A, Mavrogiorgou A, Menychtas A, Maglogiannis I, Kyriazis D. Structurally Mapping Healthcare Data to HL7 FHIR through Ontology Alignment. J Med Syst [Internet]. 2019 Mar 1 [cited 2024 Sep 19];43(3). Available from: https://pubmed.ncbi.nlm.nih.gov/30721349/

17. Rodrigues JM, Robinson D, Della Mea V, Campbell J, Rector A, Schulz S, et al. Semantic Alignment between ICD-11 and SNOMED CT. Stud Health Technol Inform [Internet]. 2015 [cited 2024 Sep 19];216:790–4. Available from: https://ebooks.iospress.nl/doi/10.3233/978-1-61499-564-7-790

18. Kiourtis A, Mavrogiorgou A, Kyriazis D. A Semantic Similarity Evaluation for Healthcare Ontologies Matching to HL7 FHIR Resources. Stud Health Technol Inform [Internet]. 2020 Jun 16 [cited 2024 Sep 19];270:13–7. Available from: https://pubmed.ncbi.nlm.nih.gov/32570337/

19. Kent W. Solving domain mismatch and schema mismatch problems with an object-oriented database programming language. In: Proceedings of the 17th International Conference on Very Large Data Bases. 1991. p. 147–60.

20. Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. CoRR [Internet]. 2018;abs/1810.04805. Available from: http://arxiv.org/abs/1810.04805

21. Zhang X, Zhang Y, Long D, Xie W, Dai Z, Tang J, et al. mGTE: Generalized Long-Context Text Representation and Reranking Models for Multilingual Text Retrieval. [cited 2024 Sep 18]; Available from: https://hf.co/Alibaba-NLP/gte-multilingual-base

22. New embedding models and API updates | OpenAI [Internet]. [cited 2024 Sep 18]. Available from: https://openai.com/index/new-embedding-models-and-api-updates/

23. Harris CR, Millman KJ, der Walt SJ van, Gommers R, Virtanen P, David Cournapeau, et al. Array programming with NumPy. Nature [Internet]. 2020 Sep;585(7825):357–62. Available from: https://doi.org/10.1038/s41586-020-2649-2

24. The pandas development team. pandas-dev/pandas: Pandas (v2.2.2) [Internet]. Zenodo; 2024. Available from: https://doi.org/10.5281/zenodo.10957263

25. Hunter JD. Matplotlib: A 2D graphics environment. Comput Sci Eng. 2007;9(3):90–5.

26. Waskom ML. seaborn: statistical data visualization. J Open Source Softw [Internet]. 2021;6(60):3021. Available from: https://doi.org/10.21105/joss.03021

27. Pedregosa F, Varoquaux G, Gramfort A, Michel V. and Thirion B, Grisel O, Blondel M, et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research. 2011;12:2825–30.

28. Brown D. Rank-BM25: A Collection of BM25 Algorithms in Python [Internet]. Zenodo; 2020. Available from: https://doi.org/10.5281/zenodo.4520057

29. Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi A, et al. Transformers: State-of-the-Art Natural Language Processing. In Association for Computational Linguistics; 2020. p. 38–45. Available from: https://www.aclweb.org/anthology/2020.emnlp-demos.6

30. chroma-core/chroma: the AI-native open-source embedding database [Internet]. [cited 2024 Sep 17]. Available from: https://github.com/chroma-core/chroma

APPENDIX A

Figure 7. Confusion matrix for resource classification by embedding models

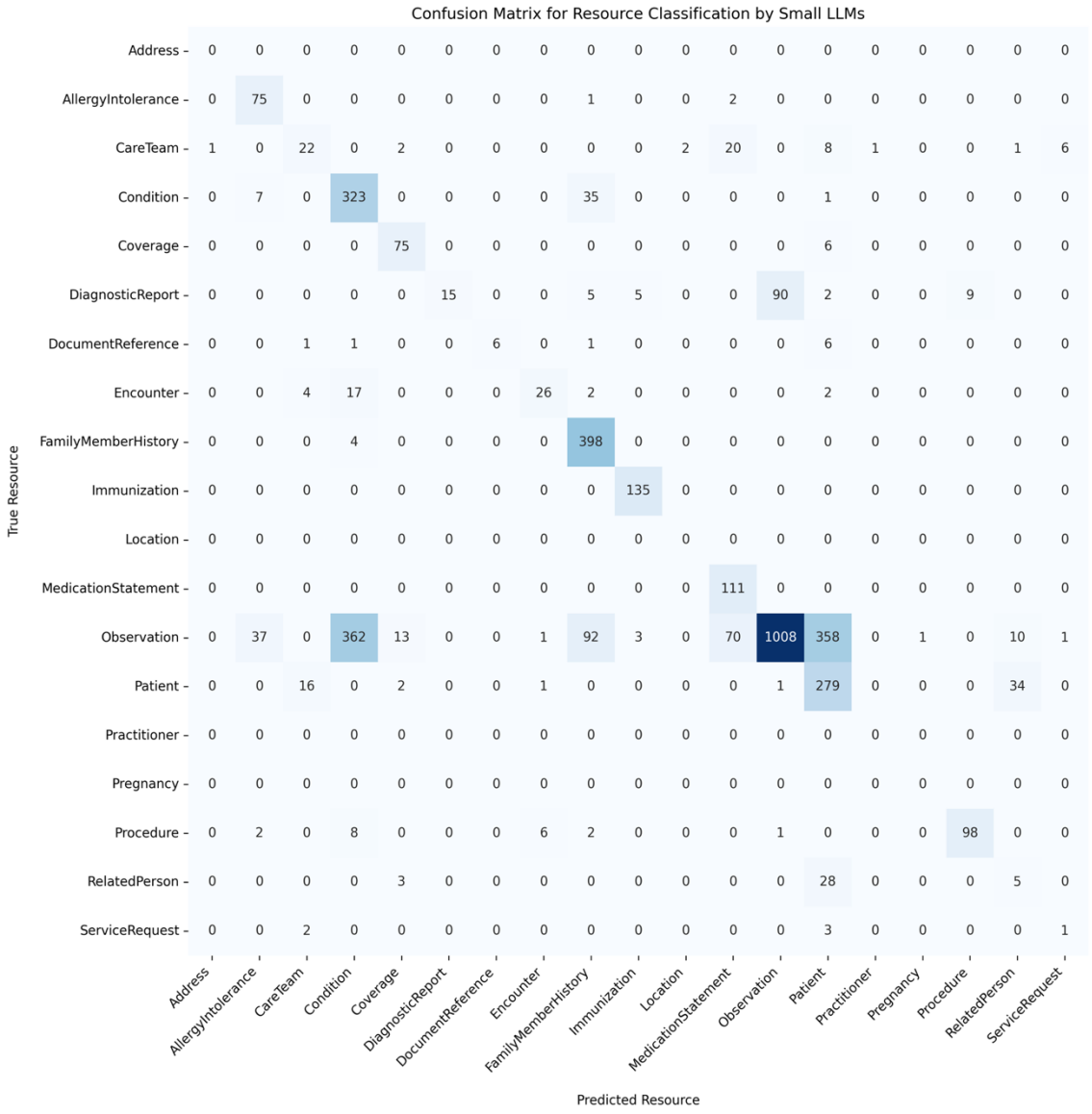


Figure 8. Confusion matrix for resource classification by small LLMs

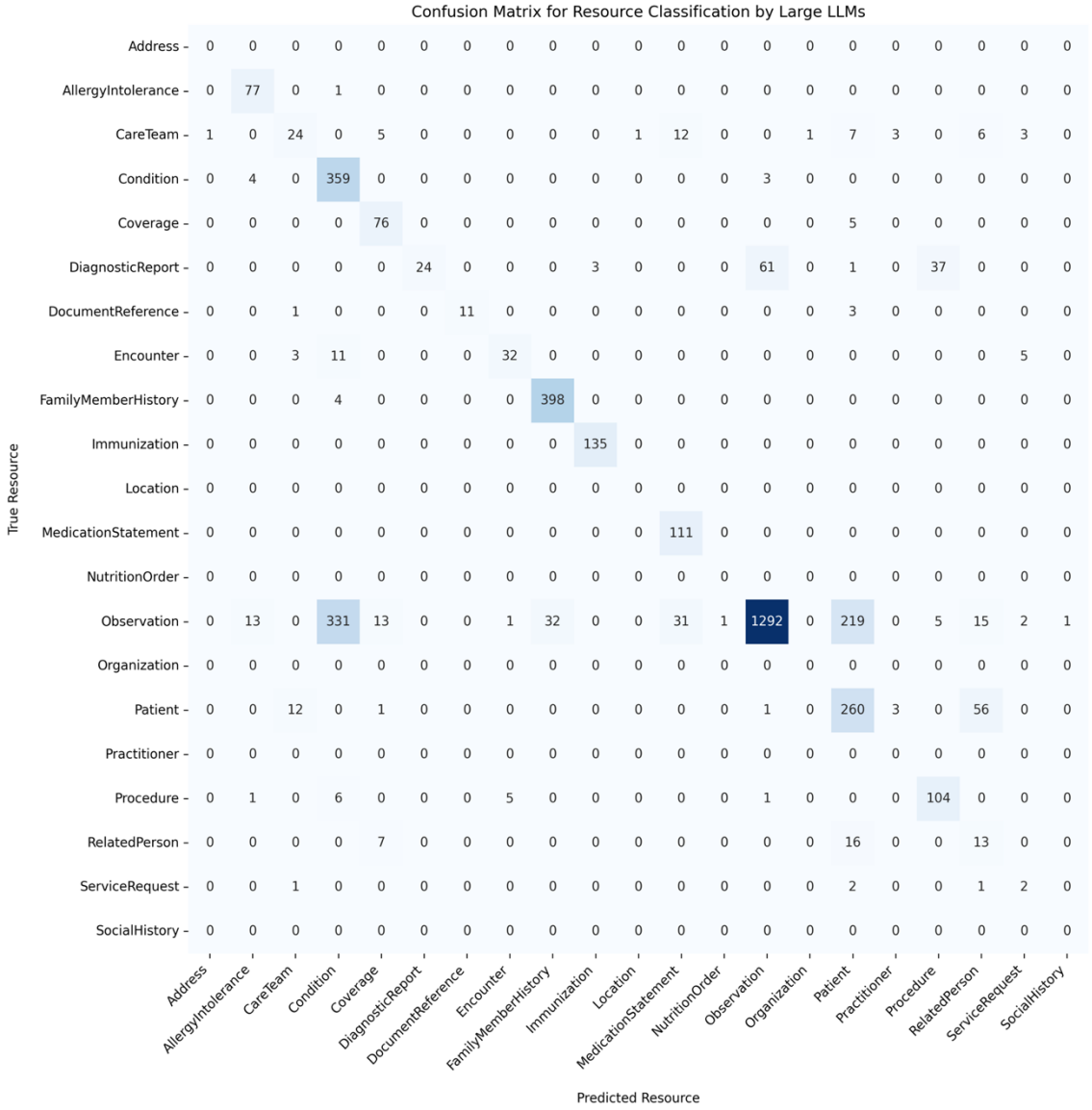


Figure 9. Confusion matrix for resource classification by large LLMs

Appendix B

Resource classification prompt

"These are select FHIR resources and their descriptions from the HL7 FHIR data exchange standard:

{FHIR resources and their descriptions}

Match the following medical intake form field with the FHIR resource that would most likely contain the information requested by the field."

Element classification prompt

"These are the FHIR element names and descriptions contained within the {Resource_match} FHIR resource:

{FHIR elements and their descriptions}

Match the following medical intake form field with the minimum necessary FHIR element(s) that would be required to complete the form field."

System prompts

"Provide an answer only in the specified JSON schema below. Resource_match should have the name of the FHIR resource matched to the field.

{JSON schema}"

"Provide an answer only in the specified JSON schema below. Element_matches should be a list names of one or more FHIR elements from within the matched resource that match to the field.

{JSON schema}"

JSON schemas

```
{
        "type": "object",
        "properties": {"Resource_match": {"type": "string"}},
        "additionalProperties": False,
        "required": ["Resource_match"],
}
```

```
{
        "type": "object",
        "properties": {
                "Element_matches": {"type": "array", "items": {"type": "string"}}
        },
        "additionalProperties": False,
        "required": ["Element_matches"],
}
```