Large Language Models are Few-Shot Clinical Information Extractors

Monica Agrawal  
MIT CSAIL  
magrawal@mit.edu

Stefan Hegselmann  
University of Münster  
stefan.hegselmann@uni-muenster.de

Hunter Lang  
MIT CSAIL  
hjl@mit.edu

Yoon Kim  
MIT CSAIL  
yoonkim@mit.edu

David Sontag  
MIT CSAIL  
dsontag@mit.edu

Abstract

A long-running goal of the clinical NLP community is the extraction of important variables trapped in clinical notes. However, roadblocks have included dataset shift from the general domain and a lack of public clinical corpora and annotations. In this work, we show that large language models, such as InstructGPT (Ouyang et al., 2022), perform well at zero- and few-shot information extraction from clinical text despite not being trained specifically for the clinical domain. Whereas text classification and generation performance have already been studied extensively in such models, here we additionally demonstrate how to leverage them to tackle a diverse set of NLP tasks which require more structured outputs, including span identification, token-level sequence classification, and relation extraction. Further, due to the dearth of available data to evaluate these systems, we introduce new datasets for benchmarking few-shot clinical information extraction based on a manual re-annotation of the CASI dataset (Moon et al., 2014) for new tasks\(^1\). On the clinical extraction tasks we studied, the GPT-3 systems significantly outperform existing zero- and few-shot baselines.

1 Introduction

Clinical text contains a large amount of valuable information that is not captured by the structured data fields in electronic health records (Zweigenbaum et al., 2007; Wang et al., 2018). However, there are significant challenges to clinical information extraction. Because clinical text contains irregularities such as ambiguous jargon and nonstandard phrasal structure, most off-the-shelf NLP tools perform poorly, and clinical text annotation requires domain expertise (Zheng et al., 2011). Further, due to the sensitive nature of clinical text, public corpora are rare and restrictively licensed. As a result, clinical NLP datasets tend to be small and splintered across institutions (Xia and Yetisgen-Yldiz, 2012).

In this work, we benchmark how large language models (LLMs) such as GPT-3 (Brown et al., 2020; Ouyang et al., 2022) perform at clinical NLP tasks. This takes the form of three contributions:

- We introduce three new annotated datasets for

\(^1\)https://huggingface.co/datasets/mitclinicalml/clinical-ie
benchmarked few-shot clinical information extraction methods, as many shared clinical corpora (Murphy et al., 2010; Henry et al., 2020; Johnson et al., 2016) have data use agreements that prevent their use with LLM APIs such as OpenAI’s. The datasets were generated by re-annotating the dataset from Moon et al. (2014) for new tasks.

- We show that GPT-3 performs well in clinical NLP over a set of diverse tasks (see Table 1), despite not being trained specifically for the domain. By replacing the complex hand-curated domain knowledge with the natural-language output of an LLM, the engineering effort required to solve a particular extraction task can be greatly reduced.

- While LLMs have been primarily evaluated at classification and generation tasks, our tasks involve a greater variety of expected output structures, such as relation extraction (see last three rows of Table 1). We therefore introduce guided prompt design to steer the LLM towards an easy-to-structure output and resolvers to map from the LLM outputs to the structured label space; see Figure 1.

## 2 Related Work

### 2.1 Prompt-Based Learning

In prompt-based learning (also known as in-context learning), a pretrained language model is adapted to different tasks via priming on natural language prompts—pieces of text that are combined with an input and then fed to the language model to produce an output for that task.

This paradigm has been successful for few-shot and zero-shot learning at many general-domain tasks (Brown et al., 2020; Liu et al., 2021; Wei et al., 2021; Sanh et al., 2021). More recently, large language models such as T0 and InstructGPT have re-configured their training objectives to explicitly encourage the model to perform well at such prompts (Sanh et al., 2021; Ouyang et al., 2022).

While prompt-based learning can be extended straightforwardly to classification tasks (e.g., multiple choice), more complex tasks require creativity in their implementation (Mishra et al., 2021). For example, coreference resolution is often re-framed as classification, asking which of two antecedents a pronoun refers to (Sanh et al., 2021) or whether a candidate antecedent is correct (Yang et al., 2022). This approach requires a list of antecedent candidates, which requires an additional component (e.g. a noun phrase generator) or many—potentially expensive—queries. Span classification and named entity recognition have been similarly reframed. For example, given a candidate entity \( X \) and full model access, the entity type can be predicted via an argmax over the possible types \( Y \) of the probability of statements like “\( X \) is a \( Y \) entity” (Cui et al., 2021). Alternatively, if only a single entity is being queried for a given input, prompting can be as simple as “What is the location?” (Liu et al., 2022a); however, clinical NLP often concerns itself with

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>Example Text</th>
<th>Answer</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clinical sense disambiguation</td>
<td>Given a note and an abbreviation, expand the abbreviation (classification)</td>
<td>[...] was sent to IR for thrombolysis. Post IR, ultrasound showed that […]</td>
<td>Interventional radiology</td>
<td>41 acronyms from 18,164 notes from CASI (Moon et al., 2014) and 8912 notes from MIMIC (Adams et al., 2020)</td>
</tr>
<tr>
<td>Biomedical evidence extraction</td>
<td>Given an abstract, list interventions (multi-span identification/generation)</td>
<td>[...] paliperidone extended-release tablets and […] with risperidone […]</td>
<td>-paliperidone extended-release tablets -risperidone</td>
<td>187 abstracts (token-level) and 20 newly annotated abstracts (arm identification) from EBM-NLP (Nye et al., 2018)</td>
</tr>
<tr>
<td>Coreference resolution</td>
<td>Given a note and a pronoun, identify the antecedent (span identification)</td>
<td>[...] Did develop some tremors, however. These were well managed […]</td>
<td>some tremors</td>
<td>105 newly annotated examples from CASI (Moon et al., 2014) with one pronoun-antecedent pair each</td>
</tr>
<tr>
<td>Medication status extraction</td>
<td>Given a note, extract medications and their status, e.g. active (NER + classification)</td>
<td>[...] have recommended Citrucel […] discontinue the Colace. […]</td>
<td>-Citrucel: active -Colace: discontinued</td>
<td>105 newly annotated examples from CASI (Moon et al., 2014) with 340 medication-status pairs</td>
</tr>
<tr>
<td>Medication attribute extraction</td>
<td>Given a note, extract medications and 5 attributes, e.g. dosage, reason (NER + relation extraction)</td>
<td>[...] she was taking 325 mg of aspirin per day for three years for a TIA. […]</td>
<td>aspirin: {dose: 325 mg, freq: per day, duration: three years, reason: TIA}</td>
<td>105 newly annotated examples from CASI (Moon et al., 2014) with 313 medications and 533 attributes</td>
</tr>
</tbody>
</table>

Table 1: Overview of the five tasks studied in this work and the datasets that were used.
extraction of multiple concepts. To extract multiple spans simultaneously, Li et al. (2019b) and Li et al. (2019a) use techniques from machine reading comprehension, relying on access to the underlying model and labeled data for training the extraction layer. While InstructGPT (Ouyang et al., 2022) has ~ 2% or ≤ 1k extraction examples in its training, the LLM output is never converted to a structured form, and extraction examples are only evaluated qualitatively for improvement over other models. That is, only results for classification and generation tasks are quantified.

2.2 Pretrained LMs for Clinical NLP

Clinical text differs significantly from text typically utilized in general NLP, both in syntax and vocabulary (Wu et al., 2020). As a result, the clinical NLP subcommunity often trains domain-specific models on clinical corpora following advances in language modeling from the broader NLP community. For example, clinical neural word embeddings were trained following word2vec (Mikolov et al., 2013; Wu et al., 2015; Roberts, 2016). More recently, following BERT, many clinical and biomedical variations swiftly followed including ClinicalBERT, SciBERT, BioBERT, and PubMedBERT (Devlin et al., 2018; Alsentzer et al., 2019; Ammar et al., 2018; Lee et al., 2020; Gu et al., 2021). However, in several applications, researchers observed the performance gains to be marginal to none over classical methods such as logistic regression (Chen et al., 2020; Krishna et al., 2021). Additionally, previous work has so far been unable to achieve competitive results on biomedical NLP tasks using domain-agnostic LLMs like GPT-3 (Moradi et al., 2021; Gutiérrez et al., 2022).

3 Methods

3.1 Predicting Structured Outputs with LLMs

In this work, we assume only query access to a large language model (i.e., no gradients, no log probabilities).

Suppose we have a set of n examples $\{(x_i, a_i)\}_{i=1}^{n}$, where $x_i$ is the input text as a string, $a_i$ is (optional) side information as a string (e.g., which acronym to disambiguate). The outputs $y_i \in O$ are unobserved (i.e., to be predicted). The output space $O$ is defined per task. For example, for a binary sequence labeling task, if we let $|x_i|$ be the number of tokens in $x_i$, $O$ is $\{0, 1\}^{|x_i|}$.

Prompt-based learning requires the specification of a prompt template to be applied on the input. In this work, we handcraft our prompt templates using a set of 5 validation examples per task. Let $p_j(x, a)$ be the result of filling prompt template $j$ with inputs $x$ and $a$, and further let $f(p_j(x, a)) \in \Sigma^*$ be the string output by an LLM on input $p_j(x, a)$. The next step involves mapping the LLM generation from $\Sigma^*$ to the structured label space $O$. For example, in classification, the verbalizer defines a mapping between the LLM output space $\Sigma^*$ and the discrete set of labels $O = \{1, \ldots, L\}$ using a dictionary of token/label pairs (Schick and Schütze, 2021). However, for our structured tasks of interest, the label space $O$ is more complex, and more complicated functions are needed to map to an element of $O$. We define the resolver $R$ as a function $R(x, a, f(p_1(x, a)))$ that maps the combined input and LLM output to the task-specific output space $O$. For example, suppose the output space $O$ is a list of strings. Then the resolver needs to turn each output $f(p_j(x, a))$ into a list (perhaps by choosing spans of text from inside of $f(p_j(x, a))$).

For example, for medication extraction we might have:

$x = \text{"switched Advil for Tylenol"}$, $a = \text{"N/A"}$.
$p_1(x, a) = \text{"Note: switched Advil for Tylenol."}$

Task: List medications.

$f(p_1(x, a)) = \text{"Tylenol and Advil"}$

$R(x, a, f(p_1(x, a))) = \{\text{"Tylenol"}, \text{"Advil"}\}$

We refer to the output from the resolver as Resolved GPT-3, or GPT-3 + R, for short. Throughout, when comparing resolvers, we place in parentheses the lines of code (LOC) in the resolver, as a proxy for complexity (defined as human effort, not runtime). The required complexity of the resolver depends largely on the cleanliness of the prompt output, and by extension the prompt itself. We introduce guided prompt design to simplify the resolver required for complex output. As seen in Figure 1, this consists of (i) a one-shot example with an output in the desired structured format (which could be incorrect content-wise), and (ii) guiding the model to use the same format. Specific constructions are found in Sections 6 and 7.

3.2 Dataset Annotation

In the short-term, research on clinical extraction via prompting may rely on sending data to external APIs. Since data use agreements on many existing annotated clinical datasets prohibit such activity, there is a dearth of benchmarks for the community.
to build on. The de-identified Clinical Acronym Sense Inventory (CASI) dataset is therefore a valuable resource, as it is “publicly available to support the research of the greater NLP and biomedical and health informatics community” (Moon et al., 2014). CASI contains snippets of clinical notes across specialties in four University of Minnesota-affiliated hospitals. While CASI was originally annotated for acronym disambiguation, we created three new annotated datasets from existing snippets of the CASI dataset. Annotation was performed by two of the authors who have background in both clinical NLP and medicine. For each task, a set of examples was jointly annotated to establish an annotation schema, each annotator then independently labeled the same set of 105 examples using PRApCER software (Levy et al., 2021), and the two sets were then merged via joint manual adjudication.

In the following sections, we show how to build simple resolvers for five clinical NLP tasks. We find that resolvers for guided prompts are much easier to write than resolvers for un-guided prompts. The implicit structured imposed by the prompt guidance means that resolvers for a guided prompt can be less than 10 LOC. On the tasks below, we find that GPT-3 + R matches or exceeds strong few-shot, zero-shot, and even supervised baselines.

4 Clinical Sense Disambiguation

Overview. Clinical notes are rife with overloaded jargon and abbreviations. Pt can mean patient, prothrombin time, physical therapy, or posterior tibial (Weeber et al., 2001; Shilo and Shilo, 2018). This ambiguity impacts the utility of notes for patients, clinicians, and algorithms (Kuhn, 2007; Mowery et al., 2016). In this section, we first evaluate clinical sense disambiguation on the CASI dataset directly and then transfer a model distilled via weak supervision to another dataset.

Dataset 1. The Clinical Acronym Sense Inventory dataset consists of 500 text examples for each of 75 acronyms (Moon et al., 2014). Due to noise in the dataset (e.g. duplications), it is common to filter to a subset of the dataset; we follow the filtering from Adams et al. (2020), leading to a subset of 18,164 examples and 41 acronyms for evaluation. Similar to other works, we treat the task as multiple-choice.

Dataset 2. We additionally use a reverse substitution dataset (Adams et al., 2020) generated over the MIMIC-III Critical Care Database (Johnson et al., 2016). In reverse substitution, labeled data is generated from unlabeled text by replacing expansions (e.g. physical therapy) with their acronyms (PT) and using the original expansion as the label. We evaluate on their 8912 test examples over the same 41 acronyms as the CASI subset. Since we cannot query GPT-3 on this dataset, we distill and transfer a model trained on the outputs from Dataset 1.

Prompting + Resolver. We used GPT-3 edit (using engine text-davinci-edit-001) with greedy decoding (temperature = 0). For each example, we provided the full clinical snippet and appended the single instruction Expand the abbreviation:{abbr}. Since we did not provide the LLM with the answer choices, the form of the output string could still differ slightly from all the candidate answers (e.g. editing “RA” to “right atria” when “right atrium” was expected). In the resolver, we choose the answer choice with the highest contiguous character overlap with the LLM generated output.

Model Distillation via Weak Supervision. Direct deployment of large language models can be difficult due to model size and data privacy. To remedy these issues, we follow several recent works (Lang et al., 2022a; Smith et al., 2022; Wang et al., 2021) and show that we can instead view the LLM + resolver system as a labeler rather than as a classifier, and that this can even boost performance. In particular, we use outputs of this system on CASI as weak supervision (e.g., Ratner et al., 2017) to train a smaller, task-specific model. Here we fine-tune PubMedBERT (Gu et al., 2021) and follow Lang et al. (2022a); details and hyperparameters are found in the appendix.

Baselines. We compare the performance of our approach to other zero-shot language modeling methods: (i) Latent Meaning Cells (LMC), a deep latent variable model from Adams et al. (2020) which is pre-trained on millions of notes from MIMIC, (ii) ELMo pre-trained on the same dataset (Peters et al., 2018), and (iii) Clinical BioBERT (Alsentzer et al., 2019). Numbers for these three baselines are taken from Adams et al. (2020); for all three, they choose the answer choice with the most similar representation to the contextual representation of the acronym. We also show performance for random guessing and choosing the most common answer choice per acronym (since the expansions of many acronyms follow a long-tailed distribution).

Evaluation. Accuracy and macro F1 are calculated per acronym and averaged over all acronyms (see
## 5 Biomedical Evidence Extraction

### Task Overview.
Evidence-based medicine (EBM) involves synthesizing findings from across clinical research studies, but the current rapid clip of research makes it nearly impossible to keep up with all studies (Sackett, 1997; Bastian et al., 2010). Therefore, automated approaches for parsing clinical abstracts could aid the adoption of EBM (Verbeke et al., 2012; Nye et al., 2018). Here, we focus on extracting interventions and controls (which we will refer to just as Intervention), where the underlying goal is to identify the distinct arms of a clinical trial (Nye et al., 2018). Token-level classification is often used as a proxy for this goal, but distilling identified spans into distinct interventions is non-trivial and often requires significant domain knowledge. Prior work on arm identification has attempted to use coreference resolution (Ferracane et al., 2016) and to identify pairs of spans with redundant information (Nye et al., 2018).

### Dataset.
We assess intervention identification from the angles of (i) the token classification proxy task and (ii) the underlying task of arm identification. For (i), we use the token-level annotations provided in version 2 of the dataset from Nye et al. (2018) and evaluate on the 187 test abstracts provided. The average Cohen’s $\kappa$ was only 0.59 on this set. For (ii), the two annotators from Section 3.2 manually derived a list of the intervention-control arms for 20 abstracts in the test set, with perfect agreement.

### Prompting + Resolvers.
We use a single prompt with InstructGPT (engine `text-davinci-002`) and greedy decoding. The resolver for the token-labeling task removes noisy tokens (stop words) from the LLM output, maps remaining tokens in the output to the original input and labels those as 1, and merges fractured spans. The full process can be found in Appendix C.2. For the arm identification task, resolving simply involved splitting the output string on new lines.

### Comparison.
We compare to supervised approaches that train on the 4800 labeled training examples from Nye et al. (2018). PubMedBERT with an additional classification layer (LSTM or CRF) achieves close to state-of-the-art performance on the full task (Gu et al., 2021). Since prior works report numbers combined over multiple classes, we re-run training on only the Intervention label using PubMedBERT-CRF. We also include the best supervised baseline from Nye et al. (2018), an LSTM-CRF over word and character-level embeddings.

### Token-level results (Proxy Task).
We first evaluate sequence labeling precision at the token-level (F1 in Table 3). Resolved GPT-3 performs re-

---

### Table 2: Clinical sense disambiguation.
Accuracy and macro F1 for zero-shot language modeling approaches on a subset of the Clinical Acronym Sense Inventory (CASI) data set (Moon et al., 2014) and the MIMIC Reverse substitution dataset (Adams et al., 2020). GPT-3 is not run on MIMIC due to the data use agreement. To evaluate on MIMIC we distill GPT-3 + R into a smaller model by treating the outputs as weak supervision and following Lang et al. (2022b) “+ distillation”, then evaluate the smaller model on MIMIC as well.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>CASI Acc.</th>
<th>CASI Macro F1</th>
<th>MIMIC Accuracy</th>
<th>MIMIC Macro F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>0.31</td>
<td>0.23</td>
<td>0.32</td>
<td>0.28</td>
</tr>
<tr>
<td>Most Common</td>
<td>0.79</td>
<td>0.28</td>
<td>0.51</td>
<td>0.23</td>
</tr>
<tr>
<td>BERT (from Adams et al. (2020))</td>
<td>0.42</td>
<td>0.23</td>
<td>0.40</td>
<td>0.33</td>
</tr>
<tr>
<td>ELMo (from Adams et al. (2020))</td>
<td>0.55</td>
<td>0.38</td>
<td>0.58</td>
<td>0.53</td>
</tr>
<tr>
<td>LMC (from Adams et al. (2020))</td>
<td>0.71</td>
<td>0.51</td>
<td>0.74</td>
<td>0.69</td>
</tr>
<tr>
<td>GPT-3 edit + R: 0-shot</td>
<td>0.86</td>
<td>0.69</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>GPT-3 edit + R: 0-shot + distillation</td>
<td><strong>0.90</strong></td>
<td><strong>0.76</strong></td>
<td><strong>0.78</strong></td>
<td><strong>0.69</strong></td>
</tr>
</tbody>
</table>

Left of Table 2). On CASI, GPT-3 edit + R already clearly outperforms the LMC model on both metrics, and the addition of weak supervision with PubMedBERT further boosts this performance. On the MIMIC Reverse Substitution dataset, despite being transferred to a new domain, our weakly-supervised PubMedBERT model performs similarly to LMC (Adams et al., 2020), which was pre-trained specifically on the MIMIC distribution. This indicates we can use GPT-3 edit + R to label a public dataset, distill its labels into a smaller task-specific model, and then transfer that model to a private dataset to obtain competitive performance. Since the CASI dataset is publicly accessible, one possible caveat is that the dataset could have been in the language model’s training data; to investigate further (see Section C.1), we prompt the LLM on acronyms not in the original annotations.
spectably compared to supervised deep baselines, but underperforms on these token-level metrics. Many error modes occur due to particularities of the schema, e.g. including extra details (like dosing schedule or route of administration) and only including an acronym or its expansion, but not both. A clarifying example can be found in Section C.2.

Arm Identification Results. To measure arm identification accuracy, we evaluated whether the number of arms was accurate and manually checked whether the main differentiator of each intervention arm was captured, similar to Ferracane et al. (2016). For the PubMedBERT baseline, in order to distill the identified spans to a list of arms, we assume (i) oracle splitting of spans into arms (given a span describing multiple arms, we can correctly split the span) and (ii) near-oracle coreference resolution (given multiple spans describing the same arm, we can correctly merge). Resolved GPT-3 successfully identified the correct number and content of the arms in 17 of the 20 examples. The three examples it missed were also missed by PubMedBERT. Assuming oracle splitting and coreference (a nontrivial task), PubMedBERT would still have issues with 10 further examples. Details of the evaluation and error modes are in Section C.2.

6 Coreference Resolution

Task Overview. Coreference resolution involves grouping noun phrases that refer to the same underlying entity (e.g. a person, a medical concept), and it is considered particularly important for clinically accurate information retrieval and summarization (Zheng et al., 2011). For example, when surfacing past medical history, it is critical to correctly parse pronouns to understand whether the history describes the patient or a family member.

Dataset Description. In clinical NLP, coreference resolution has been largely evaluated on the 2011 i2b2/VA challenge, which consists of thousands of coreference chains (Uzuner et al., 2012). Due to i2b2’s data use agreement, the two annotators annotated a new dataset using CASI snippets, with 5 coreference pairs for prompt design and 100 pairs for evaluation (Moon et al., 2014). We prioritized difficult examples by focusing on pronoun coreference, where the input is a pronoun, the output its antecedent, and no tokens overlap between the two. More details are in Section B.2.

Prompting and Resolvers. We used the 5 examples for prompt design with InstructGPT (engine text-davinci-002) and greedy decoding (temperature = 0). We use a guided 1-shot prompt, where we provide an example input and begin a formatted response: “{pronoun} refers to”. For 1-shot, we experiment with both correct (the true noun phrase) and incorrect answers (a random incorrect noun phrase preceding the pronoun) in the example input to tease apart the effect of the example answer versus the example formatting. To clarify that effect, we average over results from 5 different 1-shot examples. We also compare to an unguided zero-shot prompt, which simply appends “What does {pronoun} ... refer to?” to the input. The zero-shot resolver involves mapping tokens back to the input due to potential paraphrases; the one-shot resolver involves only the removal of a single quotation mark, making the guided prompt easier to resolve. Section A.3 contains more detail on the prompts.

Comparison. We compare to deep end-to-end coreference resolution, as it has been shown to perform well (Lee et al., 2017). In particular, we compare to the longdoc model from (Toshniwal et al., 2020), which trained on multiple coreference datasets in order to generalize to new settings.

Results.

We evaluated via macro unigram recall (% of label’s unigrams in the resolved output) and unigram

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Token-level F1</th>
<th>Abstract-level Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>PubMedBERT-CRF (sup)</td>
<td>0.69</td>
<td>0.35</td>
</tr>
<tr>
<td>LSTM-CRF (sup)</td>
<td>0.65</td>
<td></td>
</tr>
<tr>
<td>GPT-3 + R: 0-shot</td>
<td>0.61</td>
<td>0.85</td>
</tr>
</tbody>
</table>

Table 3: Biomedical Evidence Extraction. Test F1 scores on the binary token-level sequence labeling problem for Intervention identification (Nye et al., 2020), and abstract-level accuracy at arm identification. The supervised baselines were trained on 4,800 abstracts.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Recall</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toshniwal et al. (2020, 2021)</td>
<td>0.73</td>
<td>0.60</td>
</tr>
<tr>
<td>GPT-3 + R (50 LOC): 0-shot</td>
<td>0.78</td>
<td>0.58</td>
</tr>
<tr>
<td>GPT-3 + R (1 LOC): 1-shot (incorrect)</td>
<td>0.76</td>
<td>0.78</td>
</tr>
<tr>
<td>GPT-3 + R (1 LOC): 1-shot (correct)</td>
<td>0.75</td>
<td>0.77</td>
</tr>
</tbody>
</table>

Table 4: Coreference Resolution. Macro unigram recall and unigram precision of methods on our newly annotated task using CASI (Moon et al., 2014). The end-to-end baseline was trained on three non-clinical coreference resolution datasets and transferred to this new setting. 1-shot results are averaged over 5 prompts.
1-shot example was irrelevant to the performance, without guidance, the LLM may output e.g., “The long- (Qi et al., 2020) for these metrics. While the recognition of clinical concept mentions (prob-
lems, treatments, etc.), their modifiers (e.g., nega-
tion), and relations (e.g., dosage) is a fundamental building block in clinical NLP (Jiang et al., 2011).
Here we examine the extraction of medication con-
cepts with two different schemas.

7 Medication Extraction
The recognition of clinical concept mentions (pro-
blems, treatments, etc.), their modifiers (e.g., nega-
tion), and relations (e.g., dosage) is a fundamental building block in clinical NLP (Jiang et al., 2011).
Here we examine the extraction of medication con-
cepts with two different schemas.

7.1 Recognition + Status Classification
Here we extract a list of medications and label each with a status modifier: active, discontinued, or neither (e.g. allergy or proposed medication).

Dataset description. We created new annotations for medication and status on top of CASI Moon et al. (2014). The examples were enriched for changeover in treatment. For 105 randomly selected snippets, the annotators extracted all medications mentioned and classified its status with one of the 3 labels. Further details are in Appendix B.3. Unlike in Section 7.2, all mentions corresponding to the same medication are collapsed.

Prompting and Resolver. We again used 5 examples for prompt design with InstructGPT (en-
gine text-davinci-002) and greedy decoding. Our prompt asked the model to simultaneously output the list of medications and the status of each. We evaluate the prompt in an unguided zero-shot manner and in a guided one-shot manner. Further, to clarify the effect of the 1-shot example on modifier accuracy, we examine how status classification performance changes if we (i) artificially augment the 1-shot example so all three status classes are observed, and (ii) whether the statuses need to be correct, or just present. We averaged over 5 different 1-shot inputs to clarify these effects; each 1-shot example contained between 3 and 8 medications. We describe the resolvers for the zero- and one-shot cases in detail in Section C.4; the former involved several regular expressions, and the latter required only a few short lines.

Comparison. We used a rule-based method as a medication extraction baseline, since historically they perform well (Sohn et al., 2014). To this end, we leveraged the Python library ScispaCy with the en_core_sci_sm package for entity recognition (Neumann et al., 2019, details in Appendix C.4).² For medication status classification, we compare to T-Few (Liu et al., 2022b), a few shot LLM method fine-tuned on a set of additional snippets we labeled from the same distribution (20 snippets containing 60 medication statuses). This method predicts the status, given the token indices for each medication.

Results. Table 5 shows micro recall and precision for medication extraction on our self-annotated dataset.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Recall</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>ScispaCy (Neumann et al., 2019)</td>
<td>0.73</td>
<td>0.67</td>
</tr>
<tr>
<td>GPT-3 + R (32 LOC) (0-Shot)</td>
<td>0.87</td>
<td>0.83</td>
</tr>
<tr>
<td>GPT-3 + R (8 LOC) (1-Shot)</td>
<td>0.90</td>
<td>0.91</td>
</tr>
</tbody>
</table>

Table 5: Medication extraction. Micro recall and precision for medication extraction on our self-annotated dataset.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Conditional Accuracy</th>
<th>Conditional Macro F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>T-Few (20-shot)</td>
<td>0.86</td>
<td>0.57</td>
</tr>
<tr>
<td>GPT-3 + R (32 LOC) (0-Shot)</td>
<td>0.85</td>
<td>0.69</td>
</tr>
<tr>
<td>GPT-3 + R (8 LOC) (1-shot)</td>
<td>0.89</td>
<td>0.62</td>
</tr>
<tr>
<td>GPT-3 + R (8 LOC) (1-shot) + added classes</td>
<td>0.88</td>
<td>0.71</td>
</tr>
<tr>
<td>GPT-3 + R (8 LOC) (1-shot) with shuffled classes</td>
<td>0.88</td>
<td>0.66</td>
</tr>
</tbody>
</table>

Table 6: Medication status classification. Conditional accuracy and macro F1-score for Identification of medication status active, discontinued, and neither.

precision (% of unigrams in the resolved output in the label) (Table 4). We tokenized using Stanza (Qi et al., 2020) for these metrics. While the long-doc baseline trained on thousands of non-clinical coreference examples performed considerably well already, it is outperformed by Resolved GPT-3. We found the 1-shot example mostly constrains the LLM output to quoting (rather than paraphrasing); without guidance, the LLM may output e.g., “The antecedent is unclear.” Further, the accuracy of the 1-shot example was irrelevant to the performance, an observation previously reported in the classification setting, now seen for span extraction (Min et al., 2022).

2We do not use a supervised baseline trained on the i2b2 2009 challenge data (as in Section 7.2) because their schema purposefully excluded medications in the Neither category.
Table 7: Medication attribute extraction. F1 scores on our newly annotated medication extraction dataset. The baselines are trained using supervised learning on i2b2 (Uzuner et al., 2010), then transferred to the test domain. Relation Extraction additionally requires the model to match modifiers (dosage, route, etc.) to the medication span. Baseline end-to-end relation extraction performance suffers due to errors cascading from the extraction step.

<table>
<thead>
<tr>
<th>Subtask</th>
<th>Algorithm</th>
<th>Medication</th>
<th>Dosage</th>
<th>Route</th>
<th>Frequency</th>
<th>Reason</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Token-level</td>
<td>PubMedBERT + CRF (Sup.)</td>
<td>0.73</td>
<td>0.78</td>
<td>0.71</td>
<td>0.41</td>
<td>0.22</td>
<td>0.30</td>
</tr>
<tr>
<td></td>
<td>GPT-3 + R: 1-shot</td>
<td>0.75</td>
<td>0.82</td>
<td>0.81</td>
<td>0.87</td>
<td>0.21</td>
<td>0.25</td>
</tr>
<tr>
<td>Phrase-level</td>
<td>PubMedBERT + CRF (Sup.)</td>
<td>*</td>
<td>0.67</td>
<td>0.65</td>
<td>0.36</td>
<td>0.19</td>
<td>0.21</td>
</tr>
<tr>
<td></td>
<td>GPT-3 + R: 1-shot</td>
<td>*</td>
<td>0.80</td>
<td>0.63</td>
<td>0.60</td>
<td>0.34</td>
<td>0.16</td>
</tr>
</tbody>
</table>

loaded abbreviations and missing vendor-specific drug names. Table 6 shows conditional accuracy on medication status classification. For an apples-to-apples comparison, we conditioned on the subset of medications found by all GPT-3 methods (241/340) and evaluated T-few on that subset as well. We find that if the rarer Neither class wasn’t demonstrated in the 1-shot example, it was unlikely to be output, depressing the F1 score; including all classes in the 1-shot prompt appears more important than necessarily having the correct labels.

7.2 Recognition + Relation Extraction

Dataset description. The annotators created a second new dataset for medication extraction from the snippets from Moon et al. (2014). The annotators closely followed the schema from the 2009 i2b2 medication challenge (Uzuner et al., 2010), with small deviations explained in Appendix B.4. For 105 randomly selected snippets, the annotators labeled mentions of medications, dosages, routes, frequencies, reasons, and durations, if available, and their correspondences. We examine the task from three different framings: a token-level annotation task, a phrase-level annotation task, and end-to-end relation extraction. For the example phrase “Tylenol twice daily”, the desired output for the tasks would be: [Med, Frequency, Frequency], [B-Med, B-Frequency, I-Frequency], and Medication: “Tylenol”, Frequency: "twice daily"., respectively.

Prompting and Resolver. We again used 5 examples for prompt design with InstructGPT (engine text-davinci-002) and greedy decoding (temperature = 0). We use a different guided 1-shot prompt (containing 7 entities each) for each of the three framings outlined above; these can be found in Appendix A. The resolvers for all were short.

Comparison. For token and phrase-level classification, we used a PubMedBERT model topped with a CRF layer. For end-to-end relation extraction, we first used the token-level baseline to extract entity spans, then used the technique from Shi and Lin (2019) to classify whether each pair of entities was related. We then postprocessed these pairwise outputs to match modifiers to their medications. For all the three tasks, since we followed the 2009 i2b2 medication extraction annotation guidelines, we fine-tuned the baselines with labeled data from i2b2 (10 fully annotated notes with 154 medication mentions, which we postprocess into smaller annotated chunks) and directly evaluated them on our datasets. (Uzuner et al., 2010). Appendix C.5 contains more detail for the baselines and evaluation.

8 Conclusion

In this work, we introduced new annotated datasets to show that (i) large language models have great promise at diverse clinical extraction tasks and (ii) we can guide generations to map to complex output spaces with only light post-processing. We also demonstrated how weak supervision over the system’s outputs can be used to train smaller, task-specific models that are more deployable. The scope of clinical NLP extends past what we studied here, and important next steps involve experiment-
ing with LLMs such as OPT (Zhang et al., 2022) for which we can run inference locally, enabling evaluation on existing benchmarks and fine-tuning. Another important direction involves leveraging the outputs from several prompts (e.g. 1-shot prompts with different examples) to learn to determine when GPT-3 is uncertain; this increased reliability will be vital given the high-stakes in clinical information extraction. Taken as a whole, our work indicates a new paradigm for clinical information extraction—one that can scale to the lofty goals of clinical NLP.

Limitations

While large language models show great promise at clinical information extraction, there are clear limitations to their use. First, it is still difficult to guide a LLM to match an exact schema—clinical annotation guidelines are often multiple pages. We found that even when the Resolved GPT-3 outputs were impressive qualitatively, they did not always match at the token-level. For example, in tagging durations, one Resolved GPT-3 output was “X weeks” instead of “for X weeks”. While this particular omission is trivial, it highlights the difficulty of communicating nuanced guidelines.

Second, we found a bias in GPT-3 towards outputting a non-trivial answer even where none exists. For example, for medication extraction the prompt we ended up using was, “Create a bulleted list of which medications are mentioned and whether they are active, discontinued, or neither.” However, prior to this we had experimented with two separate prompts: “Create a bulleted list of active medications, if any.” and “Create a bulleted list of discontinued medications, if any.” If there was one active and one discontinued medication, the respective LLM outputs would be correct. However, if there were two active medications and none discontinued, the LLM primed with the discontinuation prompt tended to try to find an output and usually resorted to listing one or more active medications. Therefore, it is important to craft prompts or tasks that avoid this pitfall. For example, this could be achieved via (i) chaining multiple prompts, e.g., first asking if a certain entity type exists in the input, before asking for a list (Li et al., 2019b; Wu et al., 2022) or (ii) using an output structure like the sequence tagging approach.

Finally, because of the data use restrictions on most existing clinical datasets, which prohibit publicly sharing the data (e.g., to the GPT-3 APIs), all tasks except for biomedical evidence extraction were derived from the publicly-available CASI dataset (Moon et al., 2014). While we show the promise of transferring to a new setting in Section 4, it would be ideal to have been able to directly evaluate on multiple hospital systems at multiple points throughout time. Clinical text in CASI was drawn from notes from several hospitals and a diverse set of specialties, but is by no means representative of all clinical text. For example, the CASI paper states that the notes were “primarily verbally dictated and transcribed,” but this practice is not universal. Further, as is unfortunately common in clinical NLP, we only tested in English, leaving testing the ability of LLMs to operate in different languages to future work (Névéol et al., 2018).

Ethics Statement

The datasets introduced in this paper involved only new annotations on top of existing, publicly available clinical text. Dataset annotation was conducted by two authors of the paper, and therefore there are no associated concerns, e.g. regarding compensation. As discussed in limitations, we believe these new annotated datasets serve as a starting point for the evaluation of LLMs on clinical text, but we concede that conclusions about specific performance cannot be ported to other languages, hospital systems, or temporal settings (as clinical text is quite subject to dataset shift).

If large language models were to be integrated into clinical extraction pipelines, as presented in this paper, there are large potential benefits. Clinical text is being created at a scale far too large for manual annotation, and as a result, cohorts for clinical study are largely small and hand-curated. Automatic structuring of clinical variables would help catalyze research that may be prohibitively expensive otherwise—allowing for study of rarer or less funded diseases as well as the analysis of real-world evidence for subpopulations that may not be observed in clinical trials. However, due to the high-stakes setting, it is imperative that the performance of such a system is evaluated in the same environment it will be used in, and that the performance numbers are stratified by cohorts of note (e.g. racial, socioeconomic, patient comorbidities, disease stage, site of care, author’s clinical role and seniority); such variables were not available in the data we used here.

In this work, we accessed the GPT-3 model using
the OpenAI API alone. However, we acknowledge that even the inference cost is still nontrivial (see Appendix D). We presented in Section 4 a paradigm of using weak supervision to distill a much smaller model, using pseudolabels learned from GPT-3, and we encourage such work to mitigate the environmental impact of deployment.
Acknowledgements

MA was supported by a Takeda Fellowship, the MIT Deshpande Center, and the MLA@CSAIL Initiatives which includes companies Arrow, EY, Cisco, Capital One, Ahold Delhaize, SAP, and BT. HL was supported by NSF AiTF award CCF-1723344, and SH by the German Academic Exchange Service. MA and DS were also partially supported by Memorial Sloan Kettering Cancer Center. Thanks to NVIDIA Corporation for their donation of two NVIDIA A100 GPUs, and to OpenAI for providing quota to access their models. Finally, thanks to Rebecca Boiarsky for feedback on drafts of this paper.

References


Hilda Bastian, Paul Glasziou, and Iain Chalmers. 2010. Seventy-five trials and eleven systematic reviews a day: how will we ever keep up? PLoS medicine, 7(9):e1000326.


Danielle L Mowery, Brett R South, Lee Cisternsen, Jianwei Leng, Laura-Maria Peltonen, Sanna


International Conference on Very Large Data Bases, volume 11, page 269. NIH Public Access.


A Prompts and Sample GPT-3 Outputs

We present examples for each task alongside their corresponding prompts to illustrate different prompting strategies used for each task.

A.1 Clinical Sense Disambiguation

For clinical sense disambiguation we used GPT-3 edit mode which edits the input text rather than continuing it.

Example 1

**GPT-3 edit input:**


**GPT-3 edit prompt:**

Expand the abbreviation: CTA

**GPT-3 edit output:**

GENERAL: Patient is sedated on vent. HEENT: Normocephalic, atraumatic. Pupils are sluggish but reactive and equal bilaterally. NECK: Supple. No lymphadenopathy. No JVD. CARDIAC: Regular rate and rhythm. No murmurs. LUNGS: Clear to auscultation, intubated. ABDOMEN: Obese, nontender, positive bowel sounds. EXTREMITIES: Positive pulses, positive edema. SKIN: Chronic changes pretibial area bilaterally.

Example 2

**GPT-3 edit input:**

2. Severe hypertension. 3. Severe mitral regurgitation. 4. Osteoporosis. PROCEDURES: 1. Coronary angiography and hemodynamic evaluation with right heart catheterization. Right heart catheterization shows right atrial pressure of 1, right ventricular pressure of 47/11. PA pressure of 48/16, with a pulmonary capillary wedge pressure of 29, with a large B-wave confirmed with a wedge saturation of 95

**GPT-3 edit prompt:**

Expand the abbreviation: PA

**GPT-3 edit output:**

2. Severe hypertension. 3. Severe mitral regurgitation. 4. Osteoporosis. PROCEDURES: 1. Coronary angiography and hemodynamic evaluation with right heart catheterization. Right heart catheterization shows right atrial pressure of 1, right ventricular pressure of 47/11, pulmonary artery pressure of 48/16, with a pulmonary capillary wedge pressure of 29, with a large B-wave confirmed with a wedge saturation of 95%.
A.2 Biomedical Evidence Extraction

For biomedical evidence extraction we used GPT-3 Davinci. The generated output is shown in blue.

Example 1

GPT-3 Davinci:

Assessment of acute bronchodilator effects from specific airway resistance changes in stable COPD patients.

BACKGROUND In COPD patients, reversibility is currently evaluated from the changes of forced expiratory volume at 1s (ΔFEV1) and forced vital capacity (ΔFVC). By lowering peripheral airway smooth muscle tone, bronchodilators should decrease dynamic hyperinflation, gas trapping, and possibly dyspnea at rest. Hence, we hypothesize that specific airway resistance changes (ΔsRAW) should better characterize the acute response to bronchodilators.

METHODS On two days, 60 COPD patients underwent dyspnea evaluation (VAS score) and pulmonary function testing at baseline and one hour after placebo or 300 μg indacaterol administration.

RESULTS Spirographic and ΔsRAW-based criteria identified as responders 24 and 45 patients, respectively. ΔsRAW correlated with changes of intrathoracic gas volume (ΔITGV) (r=0.61; p<0.001), residual volume (ΔRV) (r=0.60; p<0.001), ΔFVC (r=0.44; p<0.001), and ΔVAS (r=0.73; p<0.001), while ΔFEV1 correlated only with ΔFVC (r=0.34; p=0.008). Significant differences in terms of ΔITGV (p=0.002), ΔRV (p=0.023), and ΔVAS (p<0.001) occurred only if patients were stratified according to ΔsRAW.

Create a bulleted list of the arms in this trial.

- Placebo
- Indacaterol (300 μg)

Example 2

GPT-3 Davinci:

Serum biochemical characteristics of Beluga, Huso huso (L.), in response to blood sampling after clove powder solution exposure.

In order to investigate the effect of anesthesia on serum parameters, Beluga, Huso huso (L.) were blood-sampled immediately without anesthesia (control) or subjected to following anesthesia procedure: 40, 120, and 240 s exposure to 3,000, 700, and 500 mg l⁻¹ clove solution, respectively. Blood samples were collected after these periods, when fish were immobile and reached stage 4 anesthesia. Results showed that cortisol and glucose levels were significantly high in 700 and 500 but not 3,000 mg l⁻¹ group compared to control. Serum lactate levels were significantly high in 500 mg l⁻¹ group compared to control group. Lactate levels were not significantly differed between control, 3,000, and 700 mg l⁻¹ groups. There were no significant differences in serum levels of cholesterol, total protein, lactate dehydrogenase, aspartate aminotransferase, alanine aminotransferase, Na⁺, Cl⁻, K⁺, and Ca²⁺. Results suggest that rapid anesthesia with higher dose is better than slow anesthesia with lower dose for blood sampling in Beluga.

Create a bulleted list of the arms in this trial.

- Control
- 3,000 mg l⁻¹ clove solution
- 700 mg l⁻¹ clove solution
- 500 mg l⁻¹ clove solution
A.3 Coreference Resolution

Example 1: Zero-shot unguided

GPT-3 Davinci:

 [...] Her current regimen for her MS is Rebif Monday, Wednesday, and Friday and 1 gram of methylprednisolone p.o. every month. This had been working previously; however, she feels that her symptoms return before her next dose of methylprednisolone is due.

What does "This" in the sentence "This had been working previously; however, she feels that her symptoms return before her next dose of methylprednisolone is due." refer to?

The sentence is unclear, but it seems to be referring to the Rebif regimen.

Example 1: One-shot guided (incorrect)

 [...] ASSESSMENT & PLAN: The patient has no physical evidence of progression of disease. It is somewhat worrisome that her CEA is up, but will be keep watching that to see if it is just within a normal swing for her. [...] 

What does "that" in the sentence "It is somewhat worrisome that her CEA is up, but will be keep watching that to see if it is just within a normal swing for her." refer to?

"that" refers to "progression of disease"

 [...] Her current regimen for her MS is Rebif Monday, Wednesday, and Friday and 1 gram of methylprednisolone p.o. every month. This had been working previously; however, she feels that her symptoms return before her next dose of methylprednisolone is due.

What does "This" in the sentence "This had been working previously; however, she feels that her symptoms return before her next dose of methylprednisolone is due." refer to?

"This" refers to "her current regimen for her MS"

Example 1: One-shot guided (correct)

 [...] ASSESSMENT & PLAN: The patient has no physical evidence of progression of disease. It is somewhat worrisome that her CEA is up, but will be keep watching that to see if it is just within a normal swing for her. [...] 

What does "that" in the sentence "It is somewhat worrisome that her CEA is up, but will be keep watching that to see if it is just within a normal swing for her." refer to?

"that" refers to "her CEA"

 [...] Her current regimen for her MS is Rebif Monday, Wednesday, and Friday and 1 gram of methylprednisolone p.o. every month. This had been working previously; however, she feels that her symptoms return before her next dose of methylprednisolone is due.

What does "This" in the sentence "This had been working previously; however, she feels that her symptoms return before her next dose of methylprednisolone is due." refer to?

"This" refers to "her current regimen for her MS"
A.4 Medication Status Extraction

Example 1: Zero-shot guided

- "Kadian" (active)
- "Dilaudid" (discontinued)
- "Levaquin" (active)

Example 1: One-shot guided (incorrect)

- "Ativan" (discontinued)
- "Inderal LA" (active)
- "Celexa" (neither)
- "Lexapro" (active)
- "Esmolol" (active)

Example 1: One-shot guided (correct)

- "Ativan" (active)
- "Inderal LA" (active)
- "Lexapro" (active)
- "Celexa" (neither)
A.5 Medication Attribute Extraction

Example 1: Token-level

<One-shot-example>
Label the tokens: medication, dosage, route, frequency, duration, reason, or none. Ignore allergies.
<One-shot-labels>

Input: 8. Albuterol 2 puffs every 4-6 hours as needed. HOSPITAL COURSE: This is an 80-year-old female who was hospitalized about 2 months ago for chronic obstructive pulmonary disease exacerbation. At that time she was put on prednisone and antibiotics and seemed to get better. However, she was put on Augmentin ES and continued to have difficulty tasting food and felt that food tasted very salty. She had no appetite and she has continued to lose weight over the last 2 months.

Label the tokens: medication, dosage, route, frequency, duration, reason, or none. Ignore allergies.

- 8: none
- .: none
- Albuterol: medication
- 2: dosage
- puffs: dosage
- every: frequency
- 4-6: frequency
- hours: frequency
- as: none
- needed: none
- .: none
- HOSPITAL: none
- COURSE: none
- This: none
- is: none
- an: none
- 80-year-old: none
- female: none
- who: none
- was: none
- hospitalized: none
- about: duration
- 2: duration
- months: duration
- ago: duration
- for: reason
- chronic: reason
- obstructive: reason
- pulmonary: reason
- disease: reason
- exacerbation: reason
- .: none
- At: none
- that: none
- time: none
- she: none
- was: none
- put: none
- on: none
- prednisone: medication

Example 1: Phrase-level

<One-shot-example>
Label the tokens: medication, dosage, route, frequency, duration, reason, or none. Chunk the same entity together. Ignore allergies.
<One-shot-labels>

Input: 8. Albuterol 2 puffs every 4-6 hours as needed. HOSPITAL COURSE: This is an 80-year-old female who was hospitalized about 2 months ago for chronic obstructive pulmonary disease exacerbation. At that time she was put on prednisone and antibiotics and seemed to get better. However, she was put on Augmentin ES and continued to have difficulty tasting food and felt that food tasted very salty. She had no appetite and she has continued to lose weight over the last 2 months.

Label the tokens: medication, dosage, route, frequency, duration, reason, or none. Chunk the same entity together. Ignore allergies.

- 8: none
- .: none
- Albuterol: medication
- 2 puffs: dosage
- every 4-6 hours: frequency
- as needed: duration
- .: none
- HOSPITAL COURSE: none
- This: none
- is: none
- an: none
- 80-year-old: none
- female: none
- who: none
- was: none
- hospitalized: none
- about: duration
- 2 months: duration
- ago: none
- for: none
- chronic: none
- obstructive: none
- pulmonary: none
- disease: reason
- exacerbation: none
- .: none
- At: none
- that: none
- time: none
- she: none
- was: none
- put: none
- on: none
- prednisone: medication
[...]
Example 1: Relation Extraction

Label medications, ignoring allergies. Include dosage, route, frequency, duration, reason, if available.

Input: 8. Albuterol 2 puffs every 4-6 hours as needed. HOSPITAL COURSE: This is an 80-year-old female who was hospitalized about 2 months ago for chronic obstructive pulmonary disease exacerbation. At that time she was put on prednisone and antibiotics and seemed to get better. However, she was put on Augmentin ES and continued to have difficulty tasting food and felt that food tasted very salty. She had no appetite and she has continued to lose weight over the last 2 months.

Label medications, ignoring allergies. Include dosage, route, frequency, duration, reason, if available.

- medication: "Albuterol", dosage: "2 puffs", frequency: "every 4-6 hours", duration: "as needed"
- medication: "Prednisone", duration: "2 months"
- medication: "Antibiotics", duration: "2 months"
- medication: "Augmentin ES", duration: "2 months"
B Annotation Process

A screenshot of the annotation process can be seen in Figure 2.

B.1 Biomedical Evidence Extraction

For arm identification 20 out of 187 abstracts from the test set of Nye et al. (2018) were randomly selected. Both annotators separately identified the study arms in each abstract. They included all characteristics of a arm that were necessary for differentiation (Ferracane et al., 2016). For example, we would not require the route of administration for a drug (e.g. “oral” in oral X) unless another arm contained the same drug in a different formal (e.g. X nasal spray). There was full consensus between annotators for the identified numbers of arms. A single abstract was replaced due to its ambiguity.

B.2 Coreference Resolution

Annotators labeled 105 snippets from the CASI dataset with pronouns and their corresponding noun phrase antecedent (Moon et al., 2014). The antecedent was annotated as the entire noun phrase (barring any dependent clauses); in cases where two antecedents were available, both were labeled. For the purposes of evaluation, we chose the antecedent with the highest overlap to each model’s output. To ensure nontrivial examples, the annotators excluded all examples of personal pronouns (e.g. “he”, “she”) if another person (and possible antecedent) had not yet been mentioned in the snippet.

B.3 Medication Status Extraction

We wanted to create a dataset of challenging examples containing a changeover in treatment. From a sample, only ~5% of CASI snippets contained such examples. To increase the density of these examples, speeding up annotation, clinical notes were filtered with the following search terms: discont, adverse, side effect, switch, and dosage, leading to 1445 snippets. We excluded snippets that were purely medication lists, requiring at least some narrative part to be present. For 105 randomly selected snippets, the annotators first extracted all medications. Guidelines excluded medication categories (e.g. “ACE-inhibitor”) if they referred to more specific drug names mentioned elsewhere (even if partially cut off in the snippet). For instance, only the antibiotic Levaquin was labeled in: “It is probably reasonable to treat with antibiotics [...] I would agree with Levaquin alone [...]”. Guidelines also excluded electrolytes and intravenous fluids as well as route and dosage information. In a second step, medication were assigned to one of three categories: active, discontinued, and neither. Discontinued medications also contain medications that are temporarily on hold. The category neither was assigned to all remaining medications (e.g. allergies, potential medications).

B.4 Medication Attribute Extraction

For medication attribute extraction, we also labeled 105 examples from CASI (Moon et al., 2014). Annotation guideline were adopted from the 2009 i2b2 medication extraction challenge (Uzuner et al., 2010) with slight modifications. We allowed medication attributes to have multiple spans. Also, we grouped together different names of the same drug (e.g. “Tylenol” and “Tylenol PM”) for the purpose of relation extraction. After annotation of the data, we create three versions of the dataset: token-level, phrase-level, and relation-level. For the first, we split all word in the example and assigned them their respective label or none if they were not part of a label (see token-level example in A.5. For phrase-level, we kept consecutive words with the same label grouped together as phrases (see phrase-level example in A.5. The relation level just contained the extracted medication and their attributes (see relation extraction example in A.5. We note that medication lists were downsampled in the creation of the dataset, since the 2009 i2b2 challenge had found performance on narrative text was far lower than on medication lists.
Figure 2: Platform used for annotation of the three new labeled datasets, shown for the coreference resolution annotation task.
C Additional Experimental details

Across all datasets, similar to Honovich et al. (2022), we assume the latest generation of models on the OpenAI API is the set of InstructGPT models from Ouyang et al. (2022).

C.1 Clinical Sense Disambiguation

How do we know CASI is not in the LLM training set? Since the CASI dataset is publicly accessible from the Internet and on Github, one potential pitfall is that the dataset may have been in the language models’ training data. While this is also true of other common NLP benchmarks, we attempted to confirm results were not merely an artifact. To do so, we annotated 50 distinct acronyms that occurred in sentences in the CASI dataset that were not included in the original annotations. While this set of acronyms is easier (e.g., they many only have a single clinical expansion), this allows us to check that GPT-3 is not simply pattern matching to potential past training data. In the set of 50, we find GPT-3 edit correctly expanded 47 (94%). In 2 of these cases, the acronym was in fact a typo (SMIV instead of SIMV, AVG instead of ABG), and the correct expansion was given regardless. Of the 3 that were incorrect, one was in fact incorrect, one was of unspecified meaning to the annotator, and one had 2/3 of the words correct in the expansion.

Resolver Details

Weak Supervision For weak supervision, we only consider the 97% of the dataset where the overlap with an answer choice was at least 5 characters as candidates for pseudolabels. Following prior work (Lang et al., 2022a,b), we additionally used a technique called the cut statistic to select a high-quality subset of the weakly labeled data to reduce the noise in the training process. We selected a subset of size 75% to decrease noise while still choosing a large enough set to ensure all acronyms were seen during training. We fine-tuned a PubMedBERT (Gu et al., 2021) model, a BERT variant that was pretrained on biomedical abstracts and full-text articles from PubMed, using learning rate 1e-5, weight decay 0.01, the AdamW optimizer (Loshchilov and Hutter, 2018), and batch size 4, using the BERTForMultipleChoice functionality in HuggingFace Transformers (Wolf et al., 2019).

C.2 Biomedical Evidence Extraction

Baseline Training Details We trained for 10,000 steps using the AdamW optimizer, learning rate 2e-5, batch size 32, and weight decay 1e-6, inheriting these hyperparameters from Zhang et al. (2021). These were the best-performing hyperparameters across the set reported in Zhang et al. (2021, Table 10, “BERT-CRF”).

Resolver Details To evaluate on the original token-level labels we tokenize the GPT-3 output and remove bullet points, numbers, stop words, and the words “treatment”, “control”, and “group” which GPT-3 often appended for clarification (e.g. “- Placebo (Control group)”). Then, any token in the input that is found in the remaining GPT-3 output is labeled with a 1, and others with a 0. Since our procedure may have interrupted valid spans, we fill in any 0’s between 1’s as well as acronyms within parentheses. These steps transform the LLM output strings \( l_i \) to a binary labeling of the full input.

Example of Token-level Error Modes As an example describing token-level error modes of GPT-3, consider the output, the resolved output, and the gold label for a study with two arms below.

GPT-3 output

\[
\text{Inhaled fluticasone reduces [...] double-blind, placebo-controlled study [...] inhaled fluticasone [...] or placebo. Large-scale [...] of inhaled steroid therapy on [...]}
\]

Gold-label (token-level):

\[
\text{Inhaled fluticasone reduces [...] double-blind, placebo-controlled study [...] inhaled fluticasone [...] or placebo. Large-scale [...] of inhaled steroid therapy on [...]}
\]

GPT-3 correctly identifies both study arms. However, the resolved output, which simply labels the token sequence of the identified arms in the original input, disagrees with the gold labels for several tokens. For example, the output includes the route, “inhaled”, which isn’t kept in the annotation schema, dinging precision. Further, the output excludes “placebo-controlled” (given “placebo” is included), dinging recall. Therefore, despite qualitatively capturing the arms of this trial, there was a middling F1-score of 0.70 for this example. This serves to underline why token-level metrics can be misleading as to true performance towards the
underlying goal.

Oracle Details We assumed oracle splitting and oracle coreference resolution in order to distill the token-level labels to a list for the PubMed-BERT baselines. As an example of oracle splitting, PubMedBERT assigned a 1 to the span “40, 120, and 240 s exposure to 3,000, 700, and 500mg l\(^1\) clove solution;” this span in fact contains three different arms, and we assume it can be perfectly split, since the required information is theoretically present in the identified span. As an example of oracle coreference resolution, consider this example with two arms: "capecitabine and oxaliplatin plus radiotherapy (Cap-Oxa-CRT) and concurrent capecitabine and radiotherapy (Cap-CRT)."

Analysis of Error Modes for Arm Identification Resolved GPT-3 successfully identified the correct number and content of the arms in 17 of the 20 examples. The three examples it missed were also missed by PubMedBERT. In one case with two arms, both methods included a procedure as a separate third arm; in reality, the procedure occurred for both arms and was not the intervention itself. In the second case, the prompt output did not elaborate on the treatment group sufficiently, and in the final case, it fully misparsed. Assuming the oracle splitting and coreference, PubMedBERT would still have issues with 10 further examples: two again included a common procedure as a third arm, four were missing control arms, one was missing a treatment arm, two arms required further domain knowledge to consolidate (e.g., that Ramipril is an ACE inhibitory therapy), and another required properly consolidating a therapy with no overlapping tokens.

C.3 Coreference Resolution

Baseline Details We benchmark using a transformer-based model trained jointly on three large coreference datasets (Toshniwal et al., 2021) that can be found on the HuggingFace model hub (shtoshni/longformer_coreference_joint).

Resolvers The resolver for the 0-shot unguided prompt was 50 LOC, or 973 tokens in the Codex tokenizer. In contrast, the 1-shot guided prompt required only stripping a final quotation mark, period, or space, which required 20 tokens per the Codex tokenizer.

C.4 Medication + Status Extraction

Resolver details For an unguided prompt, to map the GPT-3 output string to a list of medication strings, the first step is to break the output string up into substrings by parsing the “bulleted list” output by GPT-3, which we do with regular expressions. The output strings for this prompt followed several different formats, making this step slightly more involved than in previous cases. The two basic formats were a newline-separated list and a comma-separated list of medication names. The modifiers were also expressed in different ways: some outputs were `{Medication}: {Status}`, while others were `{Medication} ({Status})`. A few examples instead grouped the medications by status, so the output was `Active: {medication1}, {medication2}, Discontinued: {medication3}`. Examples of these outputs can be found in Appendix A.4. Despite this variation, we output a list by simply replacing newlines with commas to reduce to the comma-separated case, and then applying two regular expressions to extract the medication names and modifiers from the list.

The previous steps turn the LLM output strings into lists of strings. The next step in the resolver is to denoise the individual strings in each list by first stripping dosage and route information (e.g., “10 mg” or “patch”) and then performing input-consistency checking by removing tokens that do not appear in the input. Finally, strings that, after the prior denoising steps, only consist of stop words or primarily consist of punctuation and whitespace, are removed from the prediction lists. This required 32 lines of code, and 946 tokens in a byte-pair encoding. In contrast, with a 1-shot prompt, output could be simply split on the bullets, and the status extracted from parentheses, requiring 8 lines of code and 165 tokens in a byte-pair encoding.

Medication Extraction Baseline For normalization, all entities were linked to the UMLS via the default string overlap functionality of ScispaCy (Bodenreider, 2004). We filtered the resulting UMLS
concepts by their semantic types and only kept concepts of the types Antibiotic, Clinical Drug, Pharmacologic Substance, and Vitamin. Finally, the baseline predictions are run through the same denoising steps as the GPT-3 predictions to ensure a fair comparison.

**Status Classification: T-Few** We use T-Few (Liu et al., 2022b) for medication status classification using 20 additional annotated examples as the few-shot training set. We used a single prompt:

```
In the clinical note below, what is the status of the medication Albuterol?
Albuterol 2 puffs every 4-6 hours as needed.
```

For the answer choices, we used Discontinued, Active, and Neither. We did not use IA\(^{(3)}\) pre-training, but otherwise directly followed the T-Few recipe (i.e., we used the default values for all hyperparameters including batch size, learning rate, number of steps, length normalization, etc.). We used the T0-11B model.

**C.5 Medication + Relation Extraction**

**Resolver** The resolver for the first two tasks iterates over the lines in the GPT-3 output and grabs both the text span and the label; the text span is mapped to tokenized space, and all labels not in the label space (e.g. “Instructions”) are mapped to None. For phrase-level labeling, a single additional step is conducted to map the labels to BIO format. For the relation extraction task, the resolver additionally assumes all entities mentioned in a line correspond to the medication on that line.

**Sequence Tagging baseline** We model extraction and labeling of medication + modifier (dosage, frequency, route, duration, reason) as a sequence tagging task. We use the B/I/O encoding for the label space, adding tags to the B and I labels indicating the type of entity. For training data, we split the 10 notes from the 2009 i2b2 challenge into shorter contexts using an off-the-shelf sentence segmenter, and merged split contexts of less than 30 tokens into the previous context. This results in 176 training contexts for the PubMedBERT + CRF model. As with Biomedical Evidence Extraction, we search for hyperparameters over the search space reported in Zhang et al. (2021, Table 10, “BERT-CRF”). The final model is chosen based on validation F1 score on a randomly selected validation set of 10% of the training data (i.e., 18 contexts).

**Relation Extraction Baseline** We use the model from Shi and Lin (2019) for relation extraction on top of PubMedBERT. For training data, we again use the 2009 i2b2 challenge set, but since the goal is to associate modifiers with individual medications, we split up the 10 long notes into rolling chunks around each medication mention. For each ground-truth medication entity, we create a context including the 30 tokens before and after that entity. We extended these windows to be on an 0 label so that entities are not split across contexts. We use a binary label space, since each modifier type (dosage, route, etc.) determines the relation type: the relevant task is to classify whether each pair of (medication, modifier) entities in a span is associated. We create one positive sample for each truly related (medication, modifier) pair. For each context, we add a negative sample for each (medication, modifier) pair that is not related. This results in 1416 examples (many of which have largely overlapping context, but a different pair of entities) for training the relation extraction model.
<table>
<thead>
<tr>
<th>Task</th>
<th>Cost/Token</th>
<th>Tokens/Example</th>
<th># of Examples</th>
<th>Experimental Settings</th>
<th>Estimated Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clinical sense disambiguation</td>
<td>$0 (free in edit beta mode)</td>
<td>100</td>
<td>105</td>
<td>1</td>
<td>$0</td>
</tr>
<tr>
<td>Biomedical evidence extraction</td>
<td>$0.000006</td>
<td>300</td>
<td>187</td>
<td>1</td>
<td>$6</td>
</tr>
<tr>
<td>Coreference resolution</td>
<td>$0.000006</td>
<td>300</td>
<td>105</td>
<td>11</td>
<td>$21</td>
</tr>
<tr>
<td>Medication status extraction</td>
<td>$0.000006</td>
<td>300</td>
<td>105</td>
<td>16</td>
<td>$30</td>
</tr>
<tr>
<td>Medication attribute extraction</td>
<td>$0.000006</td>
<td>600</td>
<td>105</td>
<td>3</td>
<td>$12</td>
</tr>
</tbody>
</table>

Table 8: Estimate of cost of running the experiments included in this work

D Experimental Cost

At time of experimentation the cost of experiments included in this work were under $100. A breakdown of the upper bound of API costs can be found in the table below and is based on OpenAI API pricing in spring 2022. All estimates of tokens/example are rough upper bounds; some experimental settings were cheaper.