Directing Human Attention in Event Localization for Clinical Timeline Creation

Jason Zhao* JZHAO7@MIT.EDU
Department of Electrical Engineering & Computer Science
Massachussetts Institute of Technology
Cambridge, MA, USA

Monica Agrawal* MAGRAWAL@MIT.EDU
Department of Electrical Engineering & Computer Science
Massachussetts Institute of Technology
Cambridge, MA, USA

Pedram Razavi RAZAVIP@MSKCC.ORG
Department of Medicine
Memorial Sloan Kettering Cancer Center
New York, NY, USA

David Sontag DSONTAG@MIT.EDU
Department of Electrical Engineering & Computer Science
Massachussetts Institute of Technology
Cambridge, MA, USA

Abstract

Many variables useful for clinical research (e.g. patient disease state, treatment regimens) are trapped in free-text clinical notes. Structuring such variables for downstream use typically involves a tedious process in which domain experts manually search through long clinical timelines. Natural language processing systems present an opportunity for automating this workflow, but algorithms still have trouble accurately parsing the most complex patient cases, which may be best deferred to experts. In this work, we present a framework that automatically structures simple patient cases, but when required, iteratively requests human input, specifically a label for a single note in the patient's timeline that would decrease uncertainty in model output. Our method provides a lightweight way to leverage domain experts. We test our system on two tasks from a cohort of oncology patients: identification of the date of (i) metastasis onset and (ii) oral therapy start. Compared to standard search heuristics, we show we can reduce 80% of model errors with less than 15% of the manual annotation effort that may otherwise be required.

1. Introduction

Electronic Health Records (EHRs) contain a wealth of detailed patient information—such as past medical history, patient disease status, and treatment response—that can be leveraged for use cases ranging from improved decision support to cohort creation to retrospective

. * denotes equal contribution

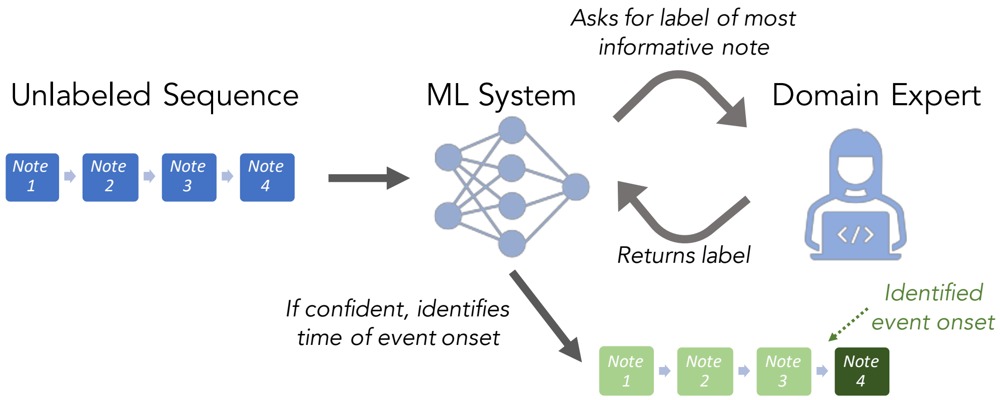


Figure 1: Diagram of our proposed approach to event identification in clinical timelines. Given an unlabeled sequence, our machine learning system can choose to directly extract the time of an event (e.g. metastatic recurrence), or it can query a domain expert for a single label (e.g. is the patient metastatic as of this note?). After receiving the label from the domain expert, the model can adjust its posterior and directly extract, or repeat the cycle as necessary.

research (Jensen et al., 2012). However, many variables crucial for downstream research are only captured within unstructured free-text clinical notes; as a result, extracting these variables often involves an arduous and expensive process of manual abstraction by domain experts (Allison et al., 2000). This manual chart review process can be particularly time-consuming when studying chronic diseases (e.g., cancer), in which patients have lengthy clinical timelines to sift through and structure (Berger et al., 2016). In this work, we consider extraction of temporal events from a patient's timeline, such as the date of metastatic recurrence. While understudied, the extraction of temporal elements enables researchers to ask key questions, such as whether treatments extend life span, and how disease progression trajectories differ across subpopulations (Banerjee et al., 2019).

The field of clinical information extraction aims to circumvent the arduous manual abstraction process via automatic systems that leverage natural language processing (NLP) and more recently, deep learning in particular (Carrell et al., 2014; Wang et al., 2018; Rajkomar et al., 2018). While these systems can improve efficiency, this efficiency often comes at the expense of reliability, and existing systems can make unpredictable mistakes. In high-risk settings such as healthcare, one solution to this trade-off is to complement machine learning models with expert human aid that can step in when models fail (Holzinger, 2016). In this work, we integrate humans into the loop by letting our extraction model, if needed, iteratively query an expert; this process is shown in Figure 1. By optimizing the queries solicited, we aim to reduce model errors while preserving most of the time and cost savings that automated NLP systems provide.

In Figure 2, we show the outcome of running our system over three patients who experience metastatic recurrence. The left two plots display two patients' clinical timelines; since pathology reports directly indicate a metastatic diagnosis, our system felt sufficiently

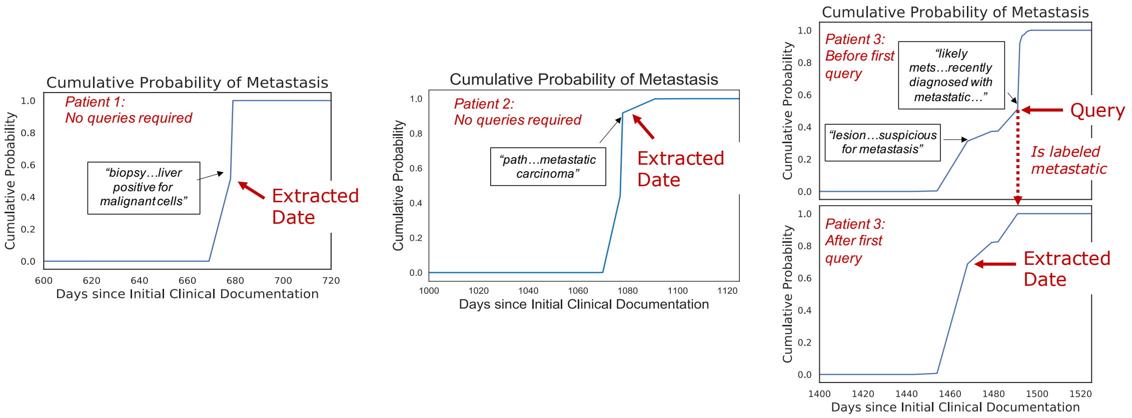


Figure 2: Plots of the model's cumulative probability over time that a given patient was metastatic. For patients 1 and 2 on the left, due to the sudden jump in cumulative probability, the model feels sufficiently confident to directly extract the date without eliciting any human input. For patient 3, on the right, before any queries are made, there is more ambiguity as to the date of metastatic recurrence. However, after a single query at the marked index, the model calculates a new posterior probability and is now sufficiently confident to extract.

confident to directly extract date of metastatic recurrence. The rightmost plot displays another patient's timeline where the system first queried a label to gain confidence.

Our framework for human-guided search helps regulate the efficiency-accuracy tradeoff for event identification from sequences, and we introduce the metric *Model-derived Query Utility* to choose the optimal query. Over a cohort of breast cancer patients, we empirically show our system's efficacy on identifying date of (i) metastatic recurrence and (ii) the start of a therapy regimen, two tasks that are crucial to leverage oncology real-world evidence.

Generalizable Insights about Machine Learning in the Context of Healthcare

In the field of clinical information extraction (and across most of machine learning and healthcare), existing works report the performance of a machine learning model unaided (Wang et al., 2018). However, due to the complexities in healthcare, such models remain imperfect, and their deployment is therefore subject to skepticism (Saria et al., 2018). On the other extreme, relying on manual effort alone is often not a scalable option for running large real-world evidence studies.

Rather than settle for an all-or-nothing approach, in this work, we advocate for a process in which a human-in-the-loop can step in when algorithms are uncertain, and we extend existing notions of rejection learning to temporal data. After demonstrating the utility of such an approach in boosting accuracy with minimal oversight, we hope more work is invested in considering how models can work in conjunction with domain experts, so that similar methods are considered viable options for model deployment looking forward.

2. Related Work

One classic way of controlling the accuracy-efficiency tradeoff is via rejection learning. In rejection learning, a system learns a classification model h and a rejection model r. The rejection model r may decide to either "reject" a data point x and incur a cost $c(x)$ (which can be viewed as asking the human expert to make the prediction), or decide to predict using h and incur a cost corresponding to misclassification error. Learning to defer (Madras et al., 2018; Geifman and El-Yaniv, 2019; Mozannar and Sontag, 2020) builds on the rejection learning framework by additionally allowing the system to adapt to different types of experts, where the cost of "rejection" also depends on the expert prediction m, i.e. $c(y, x, m)$. Unlike learning to defer, our formulation does not attempt to model different classes of experts, and instead assumes that experts are oracles for the labels. In contrast to this prior work however, we extend the notion by considering deferral specifically for just a single label in a sequence.

While sequences have not been studied in the context of the rejection learning framework, there have been sequence-based strategies in the active learning setting, in which the algorithm chooses full sequences to be labeled as additional training data (Settles and Craven, 2008). Classic approaches include uncertainty-based methods, that measure average or total label entropy, and disagreement-based methods; these approaches have been primarily studied over entire sequences, and where the labeling budget is based on the number of sequences, not factoring in variable time required per sequence. Tomanek and Hahn (2009) looked at per-instance labeling, instead of full sequences, for a token-labeling task; their method involved querying any record where the marginal probability of its likeliest label was under a given threshold. However, in our settings, labels in a patient timeline are highly correlated, so this solution may not be optimal. Fang et al. (2017) used a reinforcement learning approach for sequence selection, but assumed a large amount of available labels for validation, often impractical in healthcare settings. Furthermore, we note the underlying purpose of selecting labels in active learning is different than in rejection learning. In active learning, the goal is to find those sequences most informative for training a new model for a downstream evaluation, which is not necessarily equivalent to identifying outliers or incorrect sequences. Finally, our objective function is tailored to event identification, as compared to generic sequences.

Practically, there is great utility in efficient extraction of clinical fields from free-text notes. Clinical information extraction is an active subfield, mining diverse variables from comorbidities to treatment exposures to adverse events (Wang et al., 2018). Due to the clinical importance of recognizing metastasis to oncology cohort creation, there have been multiple studies showing that one can effectively extract metastatic status from a set of aggregated patient notes (Ling et al., 2019; Birnbaum et al., 2020). While such studies have been able to accurately identify metastatic status at a patient-level, they have not focused on the timing of metastatic recurrence, crucial clinically to assess outcomes using real-world evidence. As a bridge towards temporal precision, Banerjee et al. (2019) worked on identifying whether metastatic recurrence was present within a given quarter, by aggregating notes across 3-month time spans. Carrell et al. (2014) worked on a broader task to identify occurrences of *any* breast cancer recurrence (ipsilateral, regional, or metastatic), but discussed how their error rates have implications for the potential introduction of bias.

Our second task, the extraction of timing of oral cancer therapy start, has been explored with both rule-based and ML-forward implementations (Wang et al., 2019; Agrawal et al., 2018). However, in both tasks, no research has studied how to improve extraction by adding a human-in-the-loop. Here we work towards more-fine grained temporal accuracy and allow users to set their own tolerance for permissible errors.

3. Methods

In this section, we explain our human-guided search framework, consisting of the event identification task, our event extraction model, and our algorithm *Model-derived Query Utility* for choosing a query for a human-in-the-loop. Our framework iteratively decides between using the extraction model directly and querying an expert.

3.1. Preliminaries

Our dataset is composed of n sequences denoted by $(X_1, \cdots, X_n)$ where each sequence X_i is a set of a variable number of records t_i: $X_i = (X_i^1, \cdots, X_i^{t_i})$. We associate with each record X_i^j a timestamp which we store in the list T_i, i.e. record X_i^j occurs at timetstamp $T_i[j]$. Finally, denote y_i to to be the index at which the event of interest occurs for sequence i. From this dataset, we construct a set of latent sequence labels z_i^j for $j \in \{1, \cdots, t_i\}$, where

$$z_i^j = \begin{cases} 0 \text{ if } j < y_i \\ 1 \text{ if } j \geq y_i \end{cases} \tag{1}$$

This framework is sufficient for modeling a variety of clinical temporal extraction tasks. For example, in our first application, X_i represents the sequence of notes for patient i, y_i represents the index of metastatic recurrence for that patient, $T_i[y_i]$ represents the date of that recurrence, and $z_i^j = 0$ for the clinical notes before a patient's recurrence, and $z_i^j = 1$ for the clinical notes after.

3.2. Event Extraction Model

Given a sequence $X_i = (X_i^1, X_i^2, X_i^{t_i})$, we would like to extract the event index $y_i \in \{1, 2, \cdots, t_i\}$. We tackle this task by training a model q_θ to directly fit the distribution of y_i given the sequence X_i. In other words, $q_\theta\left(y_i = j | X\right)$ is the probability density function for the model's belief that the event occurs at index j. We parametrize q_θ using an LSTM (Hochreiter and Schmidhuber, 1997), a Recurrent Neural Network. Our extraction model takes as input the embeddings for each note, X_i, and passes it through a 1-layer bidirectional LSTM to obtain the hidden states h_i at every timestep. Finally, the h_i are passed through a fully-connected layer followed by a softmax to obtain the model probabilities $q_\theta\left(y_i = j | X\right)$ at every timestep i. Further details on embeddings are located in the experiments.

We then define $p_\theta\left(z_i^j = 1|X\right)$ to be the cumulative probability function that the event has occurred by index j for sequence i, formally defined below:

$$p_\theta\left(z_i^j = 1|X_i\right) = \sum_{0 \leq r <= j} q_\theta\left(y_i = r|X_i\right). \tag{2}$$

The extraction model then estimates the event index, which we call $\hat{y}$. We define $\hat{y}$ to be the index at the median of the cumulative event distribution, namely the j such that $p_\theta(z_i^j = 1|X) \geq 0.5$ and $p_\theta(z_i^{j-1} = 1|X) < 0.5$. Recall that the timestamps are stored in T_i, so that the estimated event timestamp is $T_i[\hat{y}]$.

3.3. Human-guided extraction

We now introduce a human expert who we assume can accurately label each record X_i^j with its label z_i^j to reduce ambiguity in our search space; the described process is illustrated in Figure 1.

In the case with a single event of interest, the feedback received from the expert can be sufficiently described by two variables: let a be the largest labeled index with a 0 label, and let b be the smallest labeled index with a 1 label. The index a is initialized at 0, and b is initialized at $t_i + 1$. We can now update our cumulative probability for the event occurrence in terms of this additional input, namely:

$$p_\theta\left(z_i^j = 1|X_i, a, b\right) = \frac{\sum_{a \leq r \leq j} q_\theta(y_i = r|X)}{\sum_{a \leq r \leq b} q_\theta(y_i = r|X)}. \tag{3}$$

We define $\hat{y}_i(a, b)$ to be the estimated index of the occurrence given the obtained bounds a and b. The objective of our system is to now iteratively select the query whose labeling would have the greatest effect on shifting this estimate, up until the estimate is sufficiently stable. If a note j is labeled, the bounds will update to some $(a^{'}, b^{'})$ —either (a, j) or (j, b) depending on the label of j—which has the potential to change our estimate from Equation 3. Our system chooses the index, which if labeled, would shift the estimate date by the largest number of days. We formalize this notion as *Model-derived Query Utility* $\mathbb{E}[\Delta_i^j]$, which is defined in terms of the current estimate $\hat{y}_i(a, b)$:

$$\begin{aligned}\mathbb{E}[\Delta_i^j] = {} & p_\theta\left(z_i^j = 0|X_i, a, b\right) \cdot |T_i\left[\hat{y}_i(a, b)\right] - T_i\left[\hat{y}_i(j, b)\right]| \\ & + p_\theta\left(z_i^j = 1|X_i, a, b\right) \cdot |T_i\left[\hat{y}_i(a, b)\right] - T_i\left[\hat{y}_i(a, j)\right]|\end{aligned} \tag{4}$$

Given this metric, at each iteration, we select:

$$\arg\max_{j \in [a,b]} \mathbb{E}[\Delta_i^j] \tag{5}$$

label the record at index j, and update $[a, b]$ to $[a^{'}, b^{'}]$ accordingly. We continue iteratively until $\mathbb{E}[\Delta_i^j] < L$ for a hyperparameter L that controls the accuracy-efficiency tradeoff. We note that if $\mathbb{E}[\Delta_i^j] < L$ at the first iteration, no queries at conducted at all.

We prove in Appendix B that in a zero-information scenario with a uniform distribution, this formulation collapses down to binary search, which has optimal time complexity for search. This is a greedy approach, but we also devised a reinforcement learning approach that learns a querying, which we compare to in Section 5 and describe in Appendix C.

4. Data

4.1. Cohort

We considered a retrospective cohort of breast cancer patients who presented to Memorial Sloan Kettering Cancer Center. All patient records were de-identified of PHI (e.g. names, dates), both within structured fields and unstructured clinical notes. This research was reviewed and determined to be IRB-exempt.

Over this cohort of breast cancer patients, we evaluate our system on two clinically important extraction tasks, each described in further detail below. All variables were extracted from notes by non-clinician abstractors who specialized in breast cancer clinical data collection. Quality control of labeled variables was ensured via scheduled auditing reports by an overseeing management team.

4.2. Metastatic Recurrence

Our first task is identification of the date of metastatic recurrence, defined as spread of the disease to distant organs or occurrence of unresectable locally advanced disease. The date of metastatic recurrence was abstracted as the date of pathologic confirmation, if available. When an initial metastatic biopsy was not performed, the date was extracted based on radiologic recurrence instead.

Existing literature has shown that simply identifying whether a patient has experienced a metastatic recurrence (without time localization) is relatively solvable for machine learning classifiers (Ling et al., 2019; Birnbaum et al., 2020). Therefore, for our training and evaluation, we restrict our cohort to only those patients who experienced metastatic recurrence.

In addition to a cohort of 476 patients where we have exact extracted date of metastasis, we additionally have a group of 379 patients with labels with less temporal specificity; the date of metastasis is approximated as the date of first line metastatic therapy. While not used for evaluation, we use this approximate cohort to train the extraction model. A consort diagram is available in Appendix A.

4.3. Therapy Start

The second evaluation task we consider is date of therapy start. In particular, we evaluate on three drugs taken orally: tamoxifen, letrozole, and palbociclib. Due to their method of administration, such drugs appear less consistently in structured data (e.g. compared to intravenous chemotherapy), and therapy regimens may be shifted from the original prescription time due to delays in insurance or pharmacy pickup.

For patients in our breast cohort, abstractors structured all drugs (oral and intravenous) taken for their breast cancer treatment, including those drugs prescribed for the patient's course at another institution. Therefore, we restrict our cohort for this task accordingly to those patient-drug pairs which are feasibly recoverable, given that our data set does not include scanned records from outside practices and all dates are de-identified within the notes themselves.

In our training cohort, we exclude examples in which the drug administration preceded the first record at the institution or the drug was not mentioned in any notes within one month of the noted start date. For our evaluation cohort, we further excluded examples in which the drug was not mentioned anywhere on the abstracted start date or in the two weeks following, and where there was no follow-up within 2 months; the purpose was to exclude patients who come in only for a second opinion or primarily for surgery.

The literature has shown high accuracy at the binary task of determining whether a patient has taken a certain drug (Agrawal et al., 2018). Therefore, for this task, we assume we are given a patient and a drug they took, and are asked to return the initial date of therapy start. Training was conducted over 8,843 patient-drug pairs, validation over 1890, and evaluation over 508 patient-drug pairs. Training and validation occurred over all drugs, whereas evaluation took place over just the three aforementioned oral drugs. A full consort diagram is present in Appendix A.

5. Experiments

In this section, we quantify the efficacy of our human-guided extraction framework on real-world extraction tasks. Label queries were solicited synthetically from “experts”, i.e. we make the framework assumption that domain experts can accurately conduct the extraction task and can return the true z_i^j given X_i^j.

5.1. Comparisons

First, we compare our human-guided extraction framework to model-only and human-only baselines. Additionally, within our framework, we consider other objective functions for selecting a query in addition to the *Model-derived Query Utility* method described in 3.3. To our knowledge, there is not previous work that has tackled this human-assisted sequential formulation, so besides the deferral of whole sequences, these objectives are also novel. Each is described below:

- In *Extraction Model Only*, we deploy the extraction model with no human input and estimate $\hat{y}_i$ directly.
- In *Vanilla Binary Search*, we do not use the extraction model and estimate how long it would take a human using binary search to pinpoint the timing of the clinical event.
- In *Whole Sequence Deferral*, we choose the P% of notes with highest label entropy (Settles and Craven, 2008) to undergo a full labeling (via binary search). As in the previous example P is tested over a variety of hyperparameter choices.
- With *Policy Model*, we follow the framework from Fang et al. (2017); we devise our own parametrization and define the reward for a given query as the number of days closer the estimate is after a query. We vary hyperparameter C, the cost for querying an expert which is reflected in the reward function. We define a policy model whose action space consists of querying for a label or directly predicting. Full details of our implementation of this method can be found in Appendix C.
- We additionally devised a new query method *Model-Augmented Binary Search*. In it, the domain expert is iteratively queried at the first record j for which the cumulative probability of the event occurrence $p_\theta(z_i^j)$ is at least 0.5. The model can terminate

its search early if the left and right bounds for the timing of the event drops below D days, where D is tuned as a hyperparameter.

5.2. Metastatic Recurrence

5.2.1. Task Setup

Next, we evaluated our system on the metastasis extraction task. We first trained our extraction model on a dataset of 693 patients, 379 with approximate labels of metastasis (based on date of first metastatic therapy), and 323 with gold extracted dates. The extraction model was first pre-trained on the gold patients, and then run on the full training cohort. 50 additional gold labels were reserved for validation of the extraction network, and 103 were used for testing the final system. For the reinforcement learning comparison, we use 192 of the patients allocated for training/validation of the extraction network for training and validation of the policy network instead.

Since documentation of some patients' metastasis occurred outside this cancer center, we conducted evaluation based on a shifted metastasis date–namely, the first note in our available records in which the patient was confirmed metastatic, and the closest possible we could get on our data set, given that dates in text have been de-identified. For this task, we considered notes across clinical oncology, pathology, and radiology. In our test set, patients had a median of 80 notes each.

5.2.2. Implementation Details

First, each record was encoded in a bag-of-words (BOW) fashion using n-grams ($1 \leq n \leq 3$) that occurred in at least 2% of notes. To generate a lower-dimensional embedding, we trained a LASSO regression to predict the latent z labels from individual notes, and then included only the 238 features (8% of the original total) with a nonzero weight in the LASSO regression in our final BOW embedding. These features included "mets", "to bone", and (stage) "IV". We note that in our preliminary experiments, we found that these BOW embeddings outperformed more complex note embeddings, generated via word2vec or convolutional neural networks.

We use these reduced BOW embeddings per note as input to our extraction model q_θ, in which the output dimension of the LSTM is size 64. We train our network for 20 epochs using a batch size of 8. We use the Adam optimizer (Kingma and Ba, 2014) with initial learning rate 0.01 and train using l2 regularization with a coefficient of 0.001. These hyperparameters were selected using the best-performing model on the validation set.

5.2.3. Results

The extraction model alone pinpoints the correct record indicating metastasis within 2 weeks 73% of the time, within a month 84% of the time, and within two months 88% of the time. If we examine the errors, they often arise when there is confusion in the original diagnosis, e.g. a lung or breast metastasis that may be a second primary, or from conflicting information in the original note, e.g. due to copy-forwarding. An example of the latter can be seen in Figure 2, where a note said both that the patient had "likely metastasis" and was

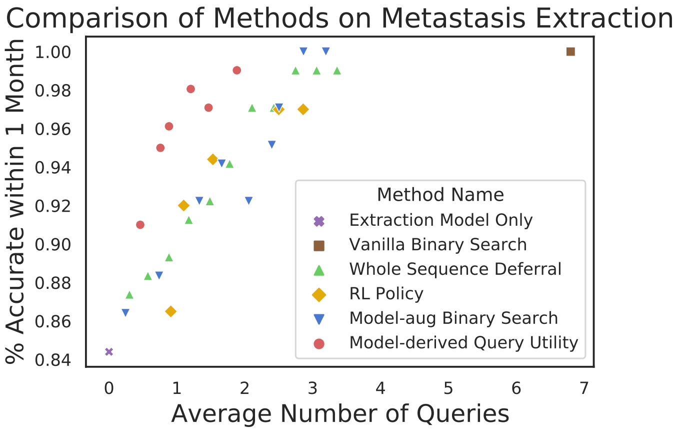
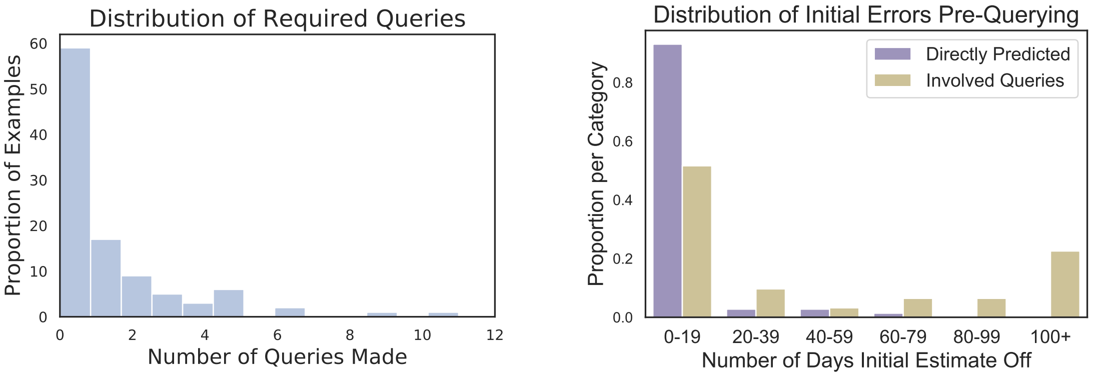


Figure 3: The above plot visualizes the trade-off between the average number of queries per patient, and the percentage of patients whose estimated date of metastatic recurrence fell within one month of the first metastatic note. At the bottom left, one can see the accuracy if the extraction model was used alone, and on the top right, the number of queries necessary if binary search was used to pinpoint each variable. We can observe that employment of *Model-derived Query Utility* provides the optimal trade-off, compared to the other methods.

"newly diagnosed metastatic", conflicting signals that a human is better suited at parsing. A fully manual binary search approach requires an average of 6.8 queries per patient.

Results on the metastasis task after adding a human-in-the-loop are in Figure 3; for methods with hyperparameters that tune the efficiency-accuracy tradeoff, outcomes are displayed over a variety of hyperparameters. The ideal case is to be in the top left corner (full accuracy with no queries required). We can see in the plot that the *Model-derived Query Utility* method is Pareto optimal on this dataset. Compared to the other methods, approximately one fewer query is needed per patient on average to achieve the same accuracy.

The left of Figure 4a) displays the distribution of required queries for the *Model-derived Query Utility* method, with the hyperparameter $L = 1$. Under this setting, 98% of metastases are correctly localized within a month and 99% are correctly localized within 2 months. For approximately 60% of patients, no queries are required at all, and for approximately another 20% of patients, only a single query is needed. On the right in Figure 4b), we can see the distribution of errors of the estimates $\hat{y}$ from the initial extraction model, before any querying. We split our distribution into the 60% of patients that were directly extracted and the 40% of patients that required further querying. For the directly extracted patients, we notice that a large majority have very close initial estimates; this is a desirable property, because our model can in fact achieve good performance on these patients without any queries. On the other hand, for patients which *Model-derived Query Utility* decided to undergo at least a single round of human-guided querying, initial predictions are far more



(a)

(b)

Figure 4: The left plot shows the distribution of number of queries solicited under the *Model-derived Query Utility* algorithm at L=1, in which the correct date was pinpointed within a month 98% of the time; 57% of patients required no queries at all. The right plot shows the distribution of initial errors of the extraction model, split into (i) the patients whose dates were directly extracted and (ii) the patients for whom queries were requested. We note that there were minimal errors on the set not queried.

erroneous, sometimes having over a 100 day difference, validating the importance of having a human-in-the-loop.

We examine cases in which our method required no queries while the comparison baselines required 3 to 4 queries. In one such example, there was large spacing between the radiologic evidence plus pathologic confirmation, and oncologist follow-up. The extraction model waited until the oncologist follow-up to become fully confident. The *Model-derived Query Utility* model was sufficiently confident that querying in this case was unnecessary, but due to the larger time gap, the other methods queried regardless.

5.3. Therapy Start

5.3.1. Task Setup

We first trained the extraction model on the 8,843 patient-drug pairs from breast cancer patients and validated on 1,890. For the RL policy model, we reduced the train and validation set size for the extraction model by 3022 and 483 pairs respectively to use in training and validating the policy network instead. The final system was tested on the 508 pairs in the evaluation set. Since a patient could take multiple drugs and therefore be in several pairs, train/validation/test sets were created to ensure no patient overlap between sets.

For this task, we only considered notes from clinical oncology; since we restrict to notes mentioning the drug, the patients in our evaluation cohort had only a median of 14 notes each. Evaluation is conducted on the basis of the ground truth abstracted date, even if it does not correspond to a note.

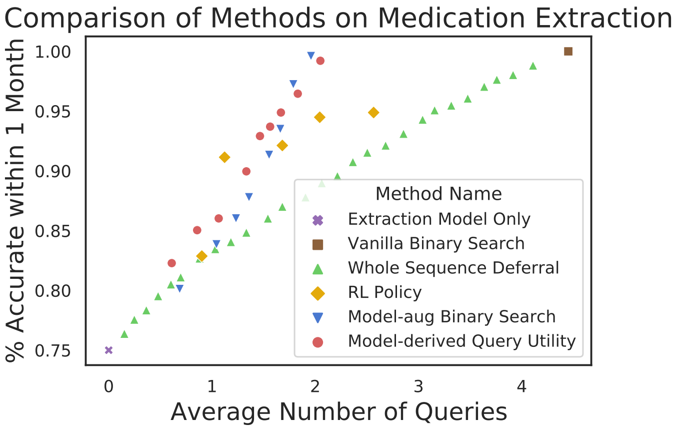


Figure 5: Results for the medication extraction task. The plot above shows the trade-off between the average number of queries per patient, and the percentage of patients that fall within one month of the gold standard therapy start dates. *Model-augmented binary search* and *Model-derived Query Utility* perform comparably and provide the most consistent performance across hyperparameter choices.

5.3.2. Implementation Details

Due to the large signal-to-noise ratio in records, we preprocess records based on whether they contain a mention of the drug; a mention of a drug is a string match of the brand name, generic name, or a common abbreviation (e.g. "palbo" for palbociclib). We tokenize into sentences and remove all sentences that do not mention the drug; notes with no mention of the drug were removed. Moreover, each mention of the target drug was replaced by a universal `CURR_DRUG_TOKEN`, and mentions of other common breast cancer medications were replaced by a universal `OTHER_DRUG_TOKEN` to allow for generalizability of features across drugs.

Then, similar to the previous task, each preprocessed record was encoded in a bag-of-words (BOW) fashion using n-grams ($1 \leq n \leq 3$), and thresholded for a vocabulary size of 650. Unlike the previous task, we omit the use of LASSO regression to generate a lower-dimensional embedding, based on results of initial experimentation.

Analogous to the previous task, we parametrize q_θ using an identical architecture, a bidirectional LSTM with output dimension 64, followed by a fully-connected layer. We train our model for 5 epochs using the Adam optimizer and an initial learning rate of 1e-3. We train using a batch size of 8, and we select the best model using early stoppage by taking the best-performing model on the validation set.

5.3.3. Results

The extraction model alone gets the correct date within one month 74% of the time and within two months 90% of the time. If we examine where errors are made, ambiguity in the underlying extraction model often arises from conflicting reports between prescriber's plans and patients' actions. For example, a note may indicate that the patient is "starting the drug" whereas the next note includes that the patient has "refused to switch treatment." In examining these cases, we do note poorer calibration of our extraction model, where the model tends to be overconfident; in the previous example, the cumulative probability was high once the doctor stated the patient's regimen had started, despite the downstream later evidence they had not yet begun. We also ran a binary search baseline with no model input, which required an average of 4.4 steps to finish at completion.

Results factoring in human input for start of oral therapy extraction are displayed in Figure 5. On this dataset, *Whole Sequence Deferral* based on label entropy performs by far the worst of all the methods here, nearing random performance. The RL policy model does relatively well at selection of the highest yield queries, but performance quickly tapers off. We find that our cumulative probability model-based approaches, *Model-derived Query Utility* and *Model-augmented Binary Search*, strongly outperformed other baselines and performed equivalently to one another. For example, we can achieve 90% accuracy within a month (a 60% error reduction) with only 30% of the queries a full search would require.

6. Discussion

Our results show that a small amount of human oversight is often sufficient to increase the reliability of one's model outputs, validating our human-guided framework for event extraction. Using the *Model-derived Query Utility* method, with fewer than an average of a single query per patient, dates of metastatic recurrence were correctly recovered within 2 months for 98% of patients in the test set, compared to 88% with the extraction model alone. This new accuracy is sufficiently high enough for most clinical research, while the initial model accuracy alone may have incurred worries about potential bias and noise trickling into the downstream applications. Moreover, this approach only requires 13% of the annotation effort that a binary search approach would have required. This indicates that there is great promise in using a joint extraction process with *Model-derived Query Utility* to manage the trade-off between effort and accuracy.

Compared to metastasis, the wins are less stark for extraction of medication therapy date. We partially attribute this to our pre-processing, which led to a shorter timeline length. In our pre-processing, we had already filtered out any notes that do not directly mention the drug, since (i) such notes are unlikely to contain the start date, and (ii) they would violate the assumption that a domain expert could tell the status of the drug regimen based on the note alone. However, such pre-filtering is not necessarily typical in clinical abstraction settings, so true time savings over a manual chart review may be larger than our results may indicate. We found that a further detriment to the medication extraction model was a more miscalibrated extraction model than the one for metastasis, in which the output probabilities did not fully reflect the true probability of misclassification error. This skewed the query utility downwards due to model overconfidence. Therefore, recalibration of extraction models may be a useful intermediate step, using existing off-the-shelf techniques (Guo et al., 2017).

Future Work Our existing framework can directly extend to jointly extract multiple events in a clinical timeline, assuming their relative ordering is known. For example, one may want to track monotonic disease staging across time, e.g. when cancer progresses from Stage n to $n+1$. q_θ would transition to a multivariate model, and $\mathbb{E}[\Delta_i^j]$ could be redefined as the sum of the expected date shifts for each new stage. Evaluation on such a dataset remains a direction for future study.

Another direction for future work is to increase the granularity of a note presented to the human labeler, by showing or highlighting just a specific subportion of a note. Due to practices like copy-forwarding, notes can become bloated; clinical oncology notes contain a median of over five hundred tokens in our dataset. Therefore, there is utility not just in localizing notes temporally, but also indicating which portion of the note to focus on. A model could learn to imitate what experts looked at in practice or learn to highlight in a fully unsupervised fashion.

Limitations As is always the case, we made modeling assumptions that while generally reasonable, may be simplifications of the messiness of real world clinical data. For example, our approach of labeling z_i^j hinges on disease stage being monotonically increasing; while metastatic breast cancer is uncurable, there are other diseases one might want to label with non-monotonic disease staging. Additionally, we assumed that at any given note, it would be possible to tell whether or not an event of interest had already occurred. This is often a sound conclusion, since oncology notes often contain a summary of the patient's disease and treatment course thus far, due to copy-forwarding and note bloat. In our drug start date experiment, we only included those notes that specifically mention the drug, to ensure this assumption upheld. However, that assumption may not necessarily hold true across clinical specialties or note writing styles.

Our final limitation is that our evaluation was run as a simulation of human-in-the-loop interaction, but not as an actual user study. A real user study would allow us to quantify the speed and the workflow of a labeler before and after use of our system. Further, a user study may reveal real-world preferences that may inform tweaks to the reward functions in our system. As a potential example, while manual chart review often involves jumping through a patient's timeline, it may be unnecessarily cognitively complex in this setting. Instead, it may be useful to incur a penalty if notes queried for labels are out-of-order.

7. Conclusions

We have introduced a framework for a human-in-the-loop system that regulates the efficiency-accuracy trade-off for event identification in clinical timelines. We have contributed a *Model-derived Query Utility* metric for query selection that consistently performs as well or better than other metrics across hyperparameter settings on two clinical event identification tasks: (i) metastatic recurrence and (ii) the start of an oral therapy regimen, two tasks that are important to oncological research. Further, we are the first to show that *rejection learning* can be used effectively on temporal, sequential data, which saves valuable domain expert annotation time in the clinical setting. Our framework can help enable institutions to leverage the real-world evidence in their unstructured EHR notes at scale, enabling cohort creation and retrospective clinical studies that would may otherwise have been prohibitively tedious or expensive to conduct.

Acknowledgments

This work was supported by Memorial Sloan Kettering Cancer Center, the National Institutes of Health (R01CA234361; P.R.), and the NCI Cancer Center Support Grant (CCSG, P30 CA08748; P.R.). The authors would like to thank Irene Chen, Hussein Mozannar, and the reviewers for their helpful comments on the manuscript, and thank Lior Gazit, Iker Huerga, and Alexander Grigorenko for their work facilitating the research collaboration.

References

Monica Agrawal, Griffin Adams, Nathan Nussbaum, and Benjamin Birnbaum. Tifti: A framework for extracting drug intervals from longitudinal clinic notes. *arXiv preprint arXiv:1811.12793*, 2018.

J. J. Allison, T. C. Wall, C. M. Spettell, J. Calhoun, C. A. Fargason, R. W. Kobylinski, R. Farmer, and C. Kiefe. The art and science of chart review. *The Joint Commission journal on quality improvement*, 2000. ISSN 10703241. doi: 10.1016/S1070-3241(00)26009-4.

Imon Banerjee, Selen Bozkurt, Jennifer Lee Caswell-Jin, Allison W. Kurian, and Daniel L. Rubin. Natural Language Processing Approaches to Detect the Timeline of Metastatic Recurrence of Breast Cancer. *JCO Clinical Cancer Informatics*, 2019. ISSN 2473-4276. doi: 10.1200/cci.19.00034.

Marc L. Berger, Melissa D. Curtis, Gregory Smith, James Harnett, and Amy P. Abernethy. Opportunities and challenges in leveraging electronic health record data in oncology, 5 2016. ISSN 17448301. URL https://www.futuremedicine.com/doi/abs/10.2217/fon-2015-0043.

Benjamin Birnbaum, Nathan Nussbaum, Katharina Seidl-Rathkopf, Monica Agrawal, Melissa Estevez, Evan Estola, Joshua Haimson, Lucy He, Peter Larson, and Paul Richardson. Model-assisted cohort selection with bias analysis for generating large-scale cohorts from the ehr for oncology research. *arXiv preprint arXiv:2001.09765*, 2020.

David S. Carrell, Scott Halgrim, Diem-Thy Tran, Diana S. M. Buist, Jessica Chubak, Wendy W. Chapman, and Guergana Savova. Using Natural Language Processing to Improve Efficiency of Manual Chart Abstraction in Research: The Case of Breast Cancer Recurrence. *American Journal of Epidemiology*, 179(6):749–758, 3 2014. ISSN 1476-6256. doi: 10.1093/aje/kwt441. URL https://academic.oup.com/aje/article-lookup/doi/10.1093/aje/kwt441.

Meng Fang, Yuan Li, and Trevor Cohn. Learning how to active learn: A deep reinforcement learning approach. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 595–605, 2017.

Yonatan Geifman and Ran El-Yaniv. SelectiveNet: A deep neural network with an integrated reject option. In *36th International Conference on Machine Learning, ICML 2019*, 2019. ISBN 9781510886988.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *34th International Conference on Machine Learning, ICML 2017*, 2017. ISBN 9781510855144.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9 (8):1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

Andreas Holzinger. Interactive machine learning for health informatics: when do we need the human-in-the-loop? *Brain Informatics*, 2016. ISSN 21984026. doi: 10.1007/s40708-016-0042-6.

Peter B. Jensen, Lars J. Jensen, and Soøren Brunak. Mining electronic health records: Towards better research applications and clinical care, 2012. ISSN 14710056.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Albee Y Ling, Allison W Kurian, Jennifer L Caswell-Jin, George W Sledge, Nigam H Shah, and Suzanne R Tamang. Using natural language processing to construct a metastatic breast cancer cohort from linked cancer registry and electronic medical records data. *JAMIA Open*, 2019. ISSN 2574-2531. doi: 10.1093/jamiaopen/ooz040.

David Madras, Toniann Pitassi, and Richard Zemel. Predict responsibly: improving fairness and accuracy by learning to defer. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 6150–6160, 2018.

Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert. In *International Conference on Machine Learning*, pages 7076–7087. PMLR, 2020.

Andrew Y. Ng and Michael I. Jordan. *Shaping and Policy Search in Reinforcement Learning*. PhD thesis, 2003. AAI3105322.

Alvin Rajkomar, Eyal Oren, Kai Chen, Andrew M. Dai, Nissan Hajaj, Michaela Hardt, Peter J. Liu, Xiaobing Liu, Jake Marcus, Mimi Sun, Patrik Sundberg, Hector Yee, Kun Zhang, Yi Zhang, Gerardo Flores, Gavin E. Duggan, Jamie Irvine, Quoc Le, Kurt Litsch, Alexander Mossin, Justin Tansuwan, De Wang, James Wexler, Jimbo Wilson, Dana Ludwig, Samuel L. Volchenboum, Katherine Chou, Michael Pearson, Srinivasan Madabushi, Nigam H. Shah, Atul J. Butte, Michael D. Howell, Claire Cui, Greg S. Corrado, and Jeffrey Dean. Scalable and accurate deep learning with electronic health records. *arXiv*, (January):1–10, 2018. ISSN 2398-6352. doi: 10.1038/s41746-018-0029-1. URL http://dx.doi.org/10.1038/s41746-018-0029-1.

Suchi Saria, Atul Butte, and Aziz Sheikh. Better medicine through machine learning: What's real, and what's artificial?, 2018. ISSN 15491676.

Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. In *EMNLP 2008 - 2008 Conference on Empirical Methods in Natural Language*

Processing, Proceedings of the Conference: A Meeting of SIGDAT, a Special Interest Group of the ACL, 2008. doi: 10.3115/1613715.1613855.

Katrin Tomanek and Udo Hahn. Semi-supervised active learning for sequence labeling. 2009. doi: 10.3115/1690219.1690291.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

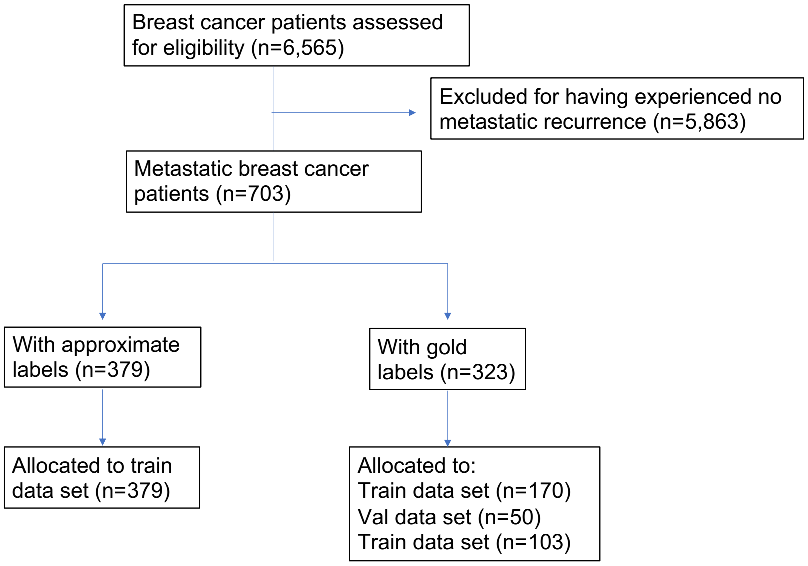
Liwei Wang, Jason Wampfler, Angela Dispenzieri, Hua Xu, and Hongfang Liu. Achievability to Extract Specific Date Information for Cancer Research. *Proceedings of AMIA Annual Symposium*, pages 893–902, 2019.

Yanshan Wang, Liwei Wang, Majid Rastegar-Mojarad, Sungrim Moon, Feichen Shen, Naveed Afzal, Sijia Liu, Yuqun Zeng, Saeed Mehrabi, Sunghwan Sohn, and Hongfang Liu. Clinical information extraction applications: A literature review, 2018. ISSN 15320464.

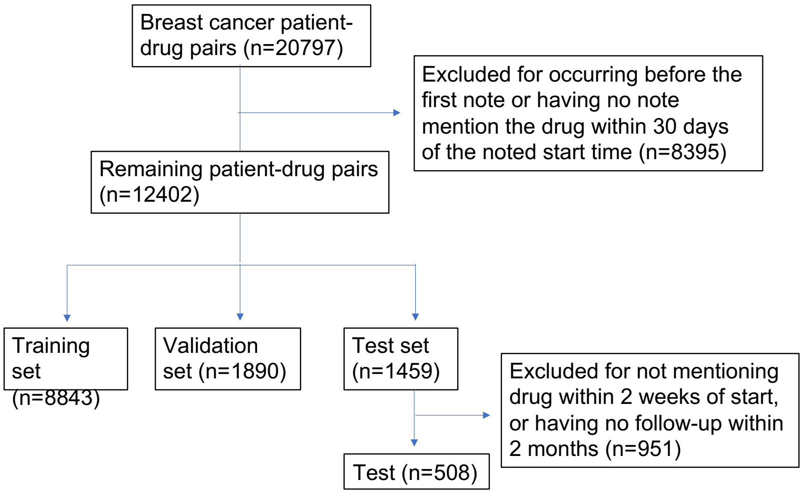
Ronald J. Willia. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning*, 1992. ISSN 15730565. doi: 10.1023/A:1022672621406.

Appendix A. Consort Diagrams

Below is the consort diagram for the metastatic extraction task.


Breast cancer patients assessed for eligibility (n=6,565)
Excluded for having experienced no metastatic recurrence (n=5,863)
Metastatic breast cancer patients (n=703)
With approximate labels (n=379)
With gold labels (n=323)
Allocated to train data set (n=379)
Allocated to:
Train data set (n=170)
Val data set (n=50)
Train data set (n=103)

Below is the consort diagram for the oral therapy extraction task.


Breast cancer patient-drug pairs (n=20797)
Excluded for occurring before the first note or having no note mention the drug within 30 days of the noted start time (n=8395)
Remaining patient-drug pairs (n=12402)
Training set (n=8843)
Validation set (n=1890)
Test set (n=1459)
Excluded for not mentioning drug within 2 weeks of start, or having no follow-up within 2 months (n=951)
Test (n=508)

Appendix B. Proof of Reduction to Binary Search

Theorem *The Model-derived Query Utility formulation collapses down to binary search in a zero-information scenario (uniform time and probability distribution).*

Proof Without a loss of generality, we can say that the endpoints for any given iteration are 0 and t, and that T[j]=j. By definition of the uniform distribution, the marginal probability at any given point j can be calculated by $p(y^j = 1|0, t) = j/t$. Then, since $p(y^{\frac{t}{2}} = 1|0, t) = 1/2$, we have that $\hat{y} = \frac{t}{2}$. Given our $\hat{y}$, we do casework to find the j that corresponds to $\arg\max_j E[\Delta^j]$. We let $\hat{z}$ be the latent labeling corresponding to $\hat{y}$.

Case 1: $\hat{z}^j = 1$

The case that $\hat{z}^j = 1$ occurs with probability j/t, making $(a^*, b^*) = (0, j)$ and $\hat{y}_[a^*, b^*] = j/2$.

Case 2: $\hat{z}^j = 0$

The case that $\hat{z}^j = 0$ occurs with probability $1 - j/t$, making $(a^*, b^*) = (j, t)$ and $\hat{y}_[a^*, b^*] = (t - j)/2$.

Plugging in, $E[\Delta^j] = (j/t)(t/2 - j/2) + (1 - j/t)(j/2)$. This quadratic has its maximum at $j = t/2$, the halfway point of the interval. This indicates that the point that maximizes *Model-derived Query Utility* over a uniform distribution is the midpoint, equivalent to binary search. ■

Appendix C. Reinforcement Learning Details

C.1. Decision-making agent

We view this formulation as a Markov Decision Process, where an agent views a state $s \in \mathcal{S}$ that encodes information about the past queries Q and the representations of the extraction model, f, conditional on the queries. The agent must choose an action $a \in \mathcal{A}$ that corresponds to either querying an index i, or making a final extraction.

The process terminates when the agent makes a prediction, or when the agent can verifiably determine the true labelling of the sequence. After termination, a reward is determined by the final accuracy of the extraction model, conditioned on the queried information.

C.1.1. State

At every state, we retain a history of past queries $Q = \{q_1, \cdots, q_m\}$. Our state s_i is composed of three components: a positional encoding of every index, the predicted marginals $p_\theta(y_i | \{q_1, \cdots, q_m\})$, and the final-layer hidden states of the prediction network at every index, h_i.

Positional Encoding Following (Vaswani et al., 2017), we use a sinusoidal positional encoding. Formally, for every index i, we define our positional embedding vector $\overrightarrow{p_i}$ of length 64 as

$$\overrightarrow{p_i}^{(j)} = \begin{cases} \sin(r^{\frac{j}{64}} \cdot i), \text{for } j \text{ is even} \\ \cos(r^{\frac{j-1}{32}} \cdot i), \text{for } j \text{ is odd} \end{cases}$$

where j represents the indices of $\overrightarrow{p_i}$, spanning from 0 to 63.

Predicted Marginals Using the predicted probabilities of our model $p_\theta(y = i | X, Q)$ at every index i, it is possible to induce marginals

$$\hat{z_i} = p_\theta(z_i = 1 | X, Q) = p_\theta(i \geq y | X, Q) = \sum_{i \geq y} p_\theta(y = i | X, Q)$$

Hidden States The third component of our state is the hidden states of the extraction model at every index, h_i. We concatenate all components to form our state, $s_i = [p_i, \hat{z_i}, 1 - \hat{z_i}, h_i]$.

C.1.2. Action

Our action space is $\mathcal{A} = \{q_1, \cdots, q_n, p\}$, where $a = q_i$ indicates the act of querying the oracle for the label of note i, and $a = p$ indicates the act of making a final prediction.

C.1.3. Reward

As training signal for the policy model, we may use a scalar reward which represents how well our extraction model performed after using all query information. However, using a delayed reward at the end of each example makes learning difficult. Instead, we advocate for reward shaping (Ng and Jordan, 2003), where intermediate rewards are provided to accelerate the learning process. Thus, we define the reward at a given state as

$$R(s_{i-1}, a) = \begin{cases} Acc(y, f_\theta(X|q_1, \cdots, q_i)) + Acc(y, f_\theta(X|q_1, \cdots, q_{i-1})) - C \text{ if } a \neq p \\ 0 \text{ if } a = p \end{cases}$$

where C is a hyperparameter for the cost of querying. In other words, the reward is the incremental improvement of querying, offset by a query cost. When the query cost outweighs improvement in accuracy, the model is incentivized to quit the process by making a prediction. There are many possibilities for parametrizing $Acc(y, \hat{y})$, but here we use $Acc(y, \hat{y}) = |y - \hat{y}|$.

C.2. Reinforcement learning

We use a reinforcement learning approach to learn a good policy for our agent. Formally, we define a policy network $\pi_\beta(s) = p_\beta(a|s), s \in \mathcal{S}, a \in \mathcal{A}$ that assigns probabilities to actions, given the current state of the agent.

We aim to find a set of values for β that maximizes the expected reward under the policy π_β. Thus, our objective is to maximize

$$J(\beta) = E_{(s_1,a_1,s_2,a_2,\cdots)}[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)]$$

where actions $a_i \sim \pi_\beta(s_i)$ are sampled from the policy and the next state $s_{i+1} \sim p(\cdot|s_i, a_i)$ is obtained via the MDP transition function. In this setting, γ is the discount factor of the MDP.

We optimize $J(\beta)$ by applying the policy gradient theorem and the REINFORCE algorithm Willia (1992). First, we sample a trajectory under the current policy π_β to obtain $\{s_1, a_1, s_2, a_2, \cdots, s_T, a_T\}$, where $a_i \sim \pi_\beta(s_i)$. In order to compute an estimate of the gradient of our objective, we first compute the cumulative reward at every step, $v_t = \sum_{j=t}^{T} \gamma^{t-j} R_j$. Our gradient then is

$$\nabla_\beta J(\beta) = \sum_{t=0}^{T} v_t \nabla_\beta \ln \pi_\beta(a_t|s_t)$$

C.2.1. Policy Architecture

We parametrize our policy model using a 1-layer Transformer encoder block as in (Vaswani et al., 2017), followed by a fully-connected layer. Because there are $n+1$ actions for an input of length n, we concatenate a trainable bias to the output of the fully-connected layer before passing it through a softmax function. In our experiments, we use a transformer with a hidden dimension of 64 and 16 attention heads.

C.2.2. Training Scheme

We train for 20 epochs using batched gradient ascent over sampled trajectories. In order to facilitate training, we use a warm-up scheme that linearly increases the hyperparameter

C from 0 to its desired value every epoch. We train using the Adam optimizer with initial learning rate 1e-3. All hyperparameters were selected using cross-validation over the validation set.