Semi-Automated Data Curation from Biomedical Literature

Protiva Rahman, Ph.D., Daniel Fabbri, Ph.D.
Vanderbilt University Medical Center, Nashville, TN

Abstract

Data curation is a bottleneck for many informatics pipelines. A specific example of this is aggregating data from preclinical studies to identify novel genetic pathways for atherosclerosis in humans. This requires extracting data from published mouse studies such as the perturbed gene and its impact on lesion sizes and plaque inflammation, which is non-trivial. Curation efforts are resource-heavy, with curators manually extracting data from hundreds of publications. In this work, we describe the development of a semi-automated curation tool to accelerate data extraction. We use natural language processing (NLP) methods to auto-populate a web-based form which is then reviewed by a curator. We conducted a controlled user study to evaluate the curation tool. Our NLP model has a 70% accuracy on categorical fields and our curation tool accelerates task completion time by 49% compared to manual curation.

Introduction

Informatics research often relies on data present in unstructured text such as biomedical literature. As a result, the first step of many projects is curating data from free-text documents. This is often a tedious and time-consuming task where curators manually fill out structured data fields, i.e., curation forms. An example of this is curating data for the Preclinical Science Integration and Translation (PRESCIANT)[1] method. PRESCIANT aims to extract information from biomedical literature on animal models of human diseases and translate the findings to humans. There are multiple publications on animal models for a particular disease, e.g., over 6000 for atherosclerotic vascular disease (AVD), but these findings have not been integrated. Integrating them involves the extraction of various details from the literature. For AVD, this includes number of weeks on diet, the affected gene, and the change in lesion size. There is thus a need to accelerate data curation by automatically extracting and populating form fields from biomedical literature.

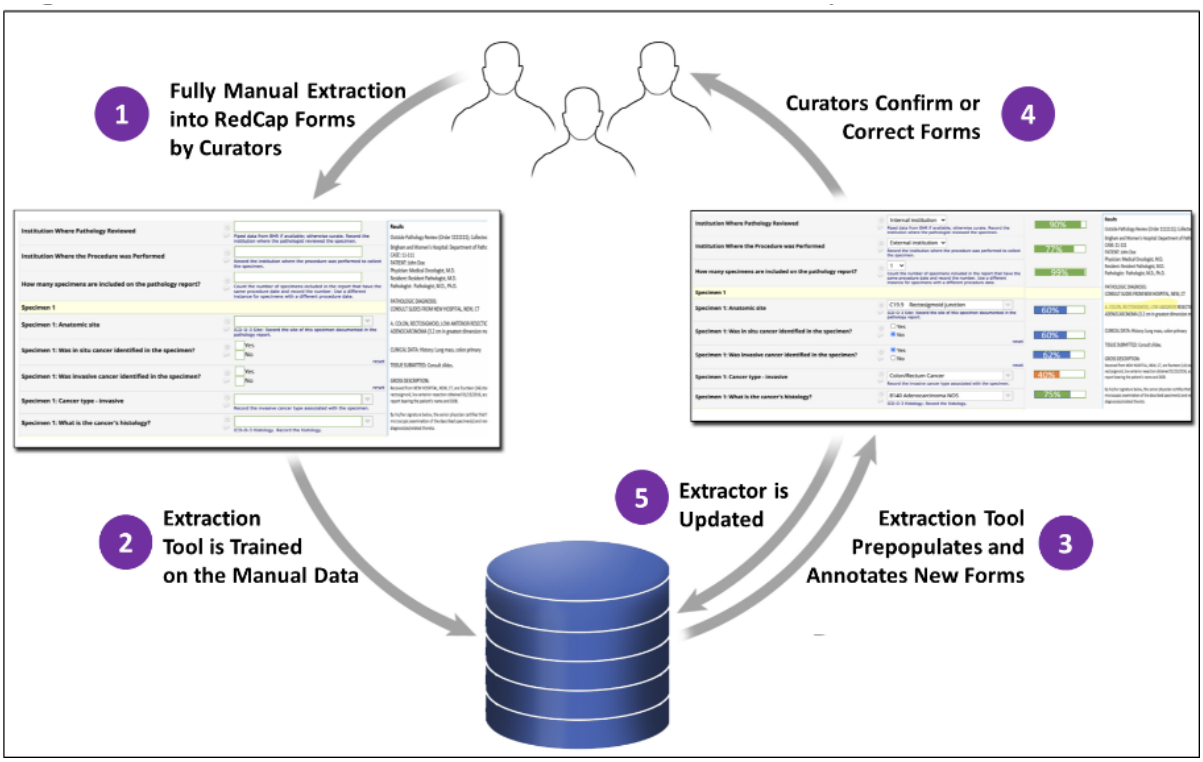

Figure 1. Overview of the Workflow for DECaF

While tools exist for interactive labeling [2] and annotating and extracting medical concepts [3], the underlying models [4] are for niche tasks such as named entity recognition of genes and cannot be generalized to any given set of fields. Moreover, a data engineer is required to set up the tool and customize it to the data. In comparison, the biomedical informatics community requires a tool that leverages existing data extractors but can be used directly by data curators without additional setup. In this work, we present our prototype system, DECaF (Data Extraction and Curation from Free-text) which auto-populates form fields and reduced curation time. DECaF uses multiple state-of-the-art natural language processing tools and is able to achieve 70% accuracy across the categorical fields. Further, our controlled user study shows an average speed-up of 49% across three users when curating with DECaF.

Methods

In this section, we provide details of the DECaF system, which uses an automatic extractor in the backend to extract form field values from a biomedical article and then presents a pre-filled form to the curator for review. The curator reviews the auto-filled values and makes any required updates, which is saved in DECaF's backend database. Researchers can download the curated database and use it for downstream research. We first describe our automatic extraction algorithm, followed by the user-interface, and finally, the user study design.

Extraction

DECaF requires an extractor which automatically populates curation forms. We frame our extraction problem as a separate classification task for each form field. The text from the biomedical article is then the input and the value for the form field is the output or "class". This problem formulation mainly works for binary and categorical fields. Free-text fields pose a challenge. As the baseline for empirical evaluation, we used text matching with regular expressions for extracting form field values. We then compared two state-of-the-art natural language processing tools, Snorkel [5] and Bidirectional Encoder Representations from Transformer (BERT) [6].

Snorkel is a library for weakly supervised data labelling. It takes as input multiple extractors (e.g., keyword search, regular expressions, domain specific rules, etc.) which do not need to have perfect accuracies. Snorkel trains a model to estimate the accuracies of each extractor based on their overlaps and conflicts with each other, and gold labels if provided. It then weights each function according to its accuracy when aggregating extracted values. It needs a minimum of three extractors. Snorkel works well for categorical fields but cannot handle extracting free-text fields with an unbounded label space.

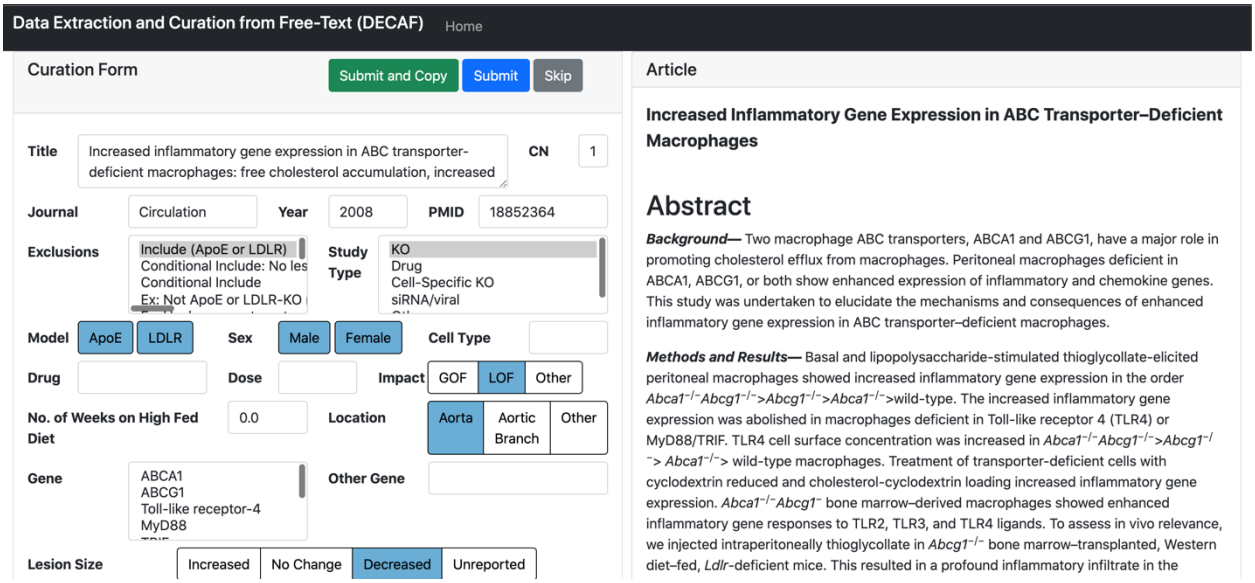

Figure 2. DECaF Interface: Curation Form on the left juxtaposed with the Article to extract from on the right

Next, we compared against variations of the BERT algorithm. BERT masks words in its training corpus and learns a numeric representation, i.e., embedding, for the masked words from its surrounding context. These embeddings can then be used in downstream tasks such as classification and question answering. We first use the raw BERT embeddings trained on the Google News corpus for our extraction task. We then fine-tune for one epoch, i.e., one pass through the training dataset, individually for each form field extraction. One of BERT's limitation is that it can only take sequences of 512 words, which is exceeded by biomedical articles. To overcome this, we divide the article into multiple segments each of length 512. We classify each individual segment, and then use the majority class as the article class for evaluation. Finally, we also compare the performances of BioBERT [7], which is trained on PubMed articles and finetuned on our corpus for one epoch.

Since neither Snorkel or BERT are well-equipped to extract non-categorical fields, we leverage NCBI's Pubtator API[4] to extract genes from abstracts. All genes from the abstract are shown to the curator, who must simply select the correct gene. If the gene is not present in the abstract (e.g., in a drug intervention study), the curator can enter it into the "Other Gene" text field. Empirically, there was little difference between Snorkel and BioBERT, so the latter with Pubtator was used as the extraction backend for DECaF.

Interface Design

Figure 2 shows DECaF's web interface, which is hosted on an AWS instance. The backend uses Python's Django [8] framework, while the frontend was designed using the Bootstrap library. The data entry form is on the left with the article to be curated on the right, so that curator does not lose time and focus in switching between tabs. The data entry widgets were chosen to minimize user effort [9]. For example, segmented trays were used instead of radio buttons and checkboxes since they increase interactive area, making it easier to select. Selection boxes were preferred over dropdowns to increase the number of visible options. Since there are articles that require multiple forms, there is a submit and copy button. Based on end user input, this button copies over the curated data which can then be edited for specific fields. There is also an option to skip if the user does not want to curate a specific article. *DECaF auto-populates the field values extracted from the article.*

User Study Design

We conducted a mini user study to evaluate the current version of the DECaF prototype and to get insights for the next build. We compared the task completion times on DECaF against manual curation of two expert curators who had already curated 576 articles using REDCap. We first performed pilot testing on the tool where each curator practiced with 5 articles on DECaF. We then selected 20 articles that had not been curated before. Each article was curated twice, once on the experimental setup which is curation using DECaF and once on the control which is manual curation. To avoid learning effects[10], we selected a between-user study design, so that each curator-article pair was unique. Articles were counterbalanced between the two curators and the sets were randomly assigned, i.e., for articles 1-10 curator A had the experimental setup while curator B was in the control group, and for articles 11-20 it was vice-versa. For the control setup, curators timed themselves for curating data into an Excel sheet. This imitates their manual workflow, since they first curate multiple articles in Excel and then fill out REDCap forms. Since the latter step can be automated, we did not include it as part of the task time. For the experimental setup, the articles assigned to each curator were preloaded, so they would see their articles one after the other after signing in. After a form is loaded, they would have to click the title to begin the timer. They were asked to finish a data entry form in one sitting but could take breaks in between forms or do them in multiple sessions. After they finished curating all 20 articles, they were asked to fill out the System Usability Scale (SUS)[11], a standard system evaluation survey that measures usability and learnability of the system.

Results

Data

Prior to developing DECaF, 576 papers were manually curated from Arteriosclerosis, Thrombosis, and Vascular Biology Journal [1] for applying PRESCIANT to AVD. These articles look at atherosclerosis mouse models, which modify the ApoE or LDLR gene and look at the impact of this on lesion size, plaque inflammation, and lipids. The mouse model and sex values are not mutually exclusive for each article (an article can study both male and female mice). There are 12 fields of interest (Table 1). Two of these are binary fields (mouse model, sex), six are categorical (study type, impact, location, lesion size, inflammation, lipid content), and four are free text (drug, cell type, gene, number of weeks on diet). These last four are the most challenging. We split our data set so that there are 368 articles in the training set, 116 in the test set, and 92 in the validation set. The training and validation sets are used by BERT

during finetuning, while the test set is used to report results. For our user study, a new set of 25 articles from the Circulation Journal were used.

Extraction Accuracy

In this section we compare results of different extraction models and select the best one for DECaF. The accuracies of the different models on the test set are presented in Table 1. As expected, there is a significant improvement in BERT results after the model is finetuned, reinforcing the need for domain customization. We can see that the BERT models provide reasonable improvement over regular expression for a majority of the fields. The difference between BERT and BioBERT is negligible, with BioBERT having a slight advantage in model identification, study type, impact on function, and plaque inflammation. These fields seem to benefit from the training on Pubmed articles. Snorkel has the best accuracies for model type identification and impact on function, probably because those fields can easily be labelled by regular expressions and keyword search. For impact on function, searching for keywords such as agonist and antagonist can provide accurate labels, however, they come with some noise. Snorkel is able to learn and filter this noise, demonstrated by its superior performance to regular expressions.

The extraction performance is poor on genes throughout. This is partly due to lack of training data since each paper often has a unique gene. This is a difficult field for curators as well, since sometimes the gene is not mentioned in the article and the curator has to look up the drug target to identify the gene. Hence, we leverage Pubtator as an external extractor for this field. Out of the 576 articles, Pubtator accurately extracted 84% of the genes. In terms of performance, when considering all fields, BioBERT has the highest accuracy of 60%. However, if we only look at categorical fields (since none of these methods are suitable for non-categorical fields), Snorkel has the highest performance of 70%. If we use Pubtator as an add-on for extracting genes, then BioBert has an accuracy of 70% while Snorkel achieves 71%. The difference between BioBert and Snorkel is negligible, so for this iteration, we used BioBERT + Pubtator as the extractor for DECaF.

Table 1: Accuracy (percent) of Extraction Methods on Form Fields

Curation Fields	**Regular Expressions**	**BERT (Not Finetuned)**	**BERT**	**BioBERT**	**Snorkel**
Model (ApoE/LDLR)	79	50	77	85	*89*
Sex	*74*	59	68	65	63
Study Type (KO, drug,etc)	41	38	60	*63*	56
Impact (GOF, LOF)	46	25	58	63	*79*
Location of lesion	77	3	*95*	*95*	91
Lesion Size	28	43	45	45	*60*
Plaque Inflammation	43	33	35	50	*55*
Lipid content	71	10	78	78	65
Weeks of Diet	*45*	5	25	25	*45*
Drug	13	3	*53*	*53*	0
Cell Type of KO	60.4	0	*83*	*83*	0
Gene	5	0	*18*	*18*	0
Average (all)	49	25	58	***60***	50
Average (categorical)	57	33	65	68	*70*
Average (categorical+ Pubtator)	60	38	67	70	***71***

As is the case with most machine learning tasks, larger training sets improve model accuracy. Therefore, we evaluated how extraction performance changes with different training sizes. With 57 articles, which is 1% of the 6000 published ApoE studies, DECaF achieved 55% accuracy, and this increases to 60% when using 576 articles or 10% of published studies. So as more articles are curated, we anticipate improvements in model performance.

User Study

We compared the task completion time by article and by curator. Of the 20 articles, 19 were included in the study and one had to be dropped due to missing data. On the control, the average completion time of the 19 papers was 2.73 mins, while on the experimental setup, it was 1.4 mins. This indicates that the average time to curate an article using the DECaF decreased by 49% (p-value .001) compared to manual curation. The average reduction in time per curator is shown in Table 2. The extraction accuracy of DECaF for categorical fields for the 19 papers was 73%. We expect further reduction in completion time as the DECaF's accuracy increases on categorical fields and as the free-text fields are better automated.

Table 2: Reduction in Completion Time Per Curator

	Control (secs)	**Experiment (secs)**	**Difference (secs)**	**Reduction (%)**
Curator A	177.89	81.97	95.92	54
Curator B	151.80	87.99	63.81	42
Average	164.845	84.98	79.865	48

Table 3 shows the answers to the system usability scale by the two curators. A score above 68 is considered good, while one above 80 is considered A grade and in the top 10% of all systems. DECaF was scored very highly by both curators, averaging 92.5, putting it in the A grade category.

Table 3: System Usability Scale Scores

	Question	**Curator A**	**Curator B**
1	I think that I would like to use this system frequently.	Agree	Strongly Agree
2	I found the system unnecessarily complex.	Strongly Disagree	Strongly Disagree
3	I thought the system was easy to use.	Agree	Strongly Agree
4	I think that I would need the support of a technical person to be able to use this system.	Disagree	Disagree
5	I found the various functions in this system were well integrated.	Strongly Agree	Strongly Agree
6	I thought there was too much inconsistency in this system.	Strongly Disagree	Strongly Disagree
7	I would imagine that most people would learn to use this system very quickly.	Strongly Agree	Strongly Agree
8	I found the system very cumbersome to use.	Strongly Disagree	Strongly Disagree
9	I felt very confident using the system.	Agree	Strongly Agree
10	I needed to learn a lot of things before I could get going with this system.	Strongly Disagree	Disagree
	Score	**90**	**95**

Discussion

Our current prototype of DECaF achieves a 49% time-reduction compared to manual curation and has an SUS rating of 92.5. These results come with the caveat that it was evaluated by only two expert curators who self-timed themselves. An obstacle in recruiting more curators is that they do not have the required domain knowledge for curation. However, the goal of this preliminary study was to identify gaps in the system and any additional needs of curators early in the process, in parallel with the development of the extraction model. One of the shortcomings we identified through the user study was that even though DECaF, with the help of Pubtator, was able to extract gene names from the article, curators still had to look up the NCBI gene symbol, which is the required input for pathway analysis. This limitation will be easily automated in the next iteration, further cutting down curation time.

In addition to automating the change to NCBI symbols, we have a couple of other ideas for improving DECaF. First, we can improve the accuracy of our extraction model by increasing our training set as well as by focusing the model to the relevant section of the article. Biomedical articles are large and form field values will be spread across different sections. For example, the sex and diet are most likely to be found in the methods section, while lesion size, inflammation, and lipid content are likely to be in the results section. Specifically, the form field extraction task requires the tool to first identify the relevant section of the article, then the associated sentence and then the exact value. Building a multi-layer approach that understands these semantics will improve extraction accuracy. Training the extraction of form fields together as opposed to individually is another unexplored avenue since form fields sometimes have dependencies that can be learned.

Our second avenue for improvement is providing the curator with additional cues. For example, when the curator clicks on a field, the system could highlight the region of the article from which the field was extracted. We can also add a confidence bar next to each form field to indicate the model's confidence in its extracted value (Figure 3). The confidence bars and location annotation will make the system more transparent. Often, users find it difficult to trust results of an automated system, especially in the healthcare domain [12]. The confidence bars will provide insights into the accuracy of a particular extraction. Various metrics can be used to populate the confidence bar, including accuracy, area under the curve, the F1 score, or support for that value in the dataset. This will draw the curator's attention to fields with less confidence.

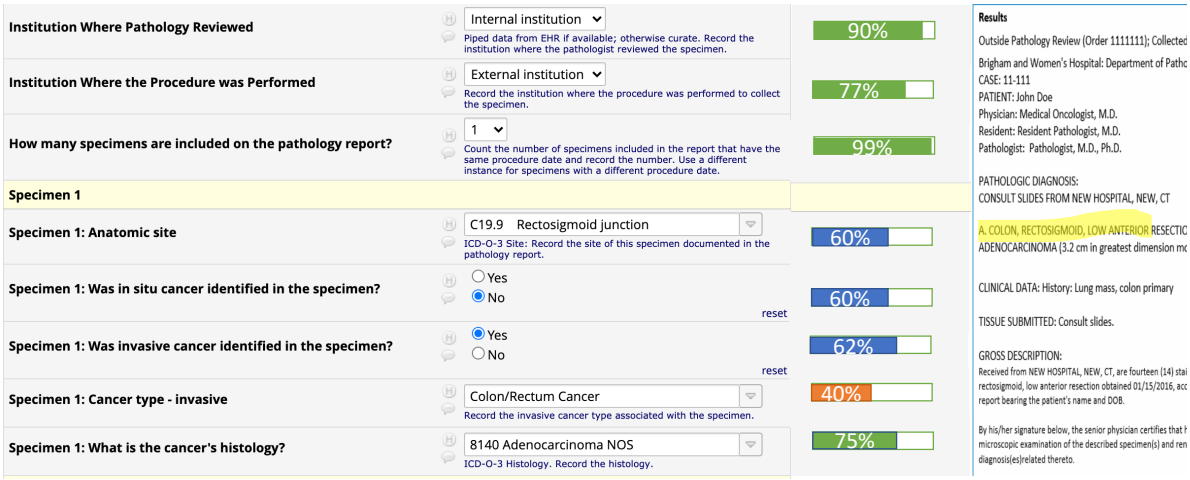

Figure 3. Confidence bars and location guidance can further accelerate curation.

Third, along with correcting the extracted value, the curator could also update the location from which the value was extracted. If the system extracts the correct value but from an incorrect location, it might be incorrectly trained for future tasks. In these cases, the curator can annotate the correct location, giving them finer control over the extraction model as compared to just providing the correct value. These updates will be used by the system to improve its extraction algorithm by updating the parameters of the model as data extraction proceeds. Having more control increases the curator's trust [12] while the finer granularity of data improves the model.

Finally, new extraction tasks will have limited training data for each field. Identifying similar curated fields and applying transfer learning will be key to accelerating curation for new domains. Leveraging existing extractors, as we did with Pubtator, can significantly improve performance. We hope to find and incorporate other open-source extractors into our system. Providing curators with a one-stop shop will allow them to leverage automatic data extraction models without requiring computational expertise.

A possible drawback of semi-automated tools is that there might be an overreliance on the system by humans, decreasing curator performance. One possible solution would be to build in test cases where the answer is known apriori and can be used to assess curator accuracy. These can then be fed back to the curator to improve their performance. We can also double curate some cases to measure inter-rater agreement and retrain curators if necessary. Another drawback of the current study is that it evaluates GUI design and auto extraction together. Future studies should tease out the impact of each individually, by having a control setting where the curator uses the GUI without autocompletion.

Conclusion

In this paper, we present DECaF, a semi-automated data curation system which auto-populates form fields from biomedical literature. Our user study demonstrates a 49% speed-up in curation time with the help of DECaF. The extraction experiments show that Snorkel is able to achieve an accuracy of 71% on categorical fields. However, it cannot extract open-domain labels unless it is constrained and reframed as a classification problem. Extraction of non-categorical fields hence remains an open problem. The next step is to work on automatic identification of the location, which might make free-text extraction more feasible. Our end goal is to build a model that is schema-agnostic and works for any document-domain and form schema. Once the tool has been successfully deployed and used at Vanderbilt, we hope to make it open-source.

Data curation is an integral and time-consuming part of informatics research. Accelerating data curation is vital for widening the bottleneck in the research data pipeline. While multiple tools exist for annotation and labeling, they have a high cost of deployment and usability. These extractors need to be integrated into a single system that is easy to use for non-computational researchers and curators. Allowing curators to seamlessly and symbiotically work with the system, where the system and the curator guide each other, is crucial to the success of informatics research.

References

1. Shuey MM, Xiang RR, Moss ME, Carvajal BV, Wang Y, Camarda N, Fabbri D, Rahman P, Ramsey J, Stepanian A, Sebastiani P. Systems Approach to Integrating Preclinical Apolipoprotein E-Knockout Investigations Reveals Novel Etiologic Pathways and Master Atherosclerosis Network in Humans. Arteriosclerosis, thrombosis, and vascular biology. 2022 Jan;42(1):35-48.
2. Kwon D, Kim S, Wei CH, Leaman R, Lu Z. ezTag: tagging biomedical concepts via interactive learning. Nucleic acids research. 2018 Jul 2;46(W1):W523-9.
3. Wei CH, Allot A, Leaman R, Lu Z. PubTator central: automated concept annotation for biomedical full text articles. Nucleic acids research. 2019 Jul 2;47(W1):W587-93.
4. Leaman R, Lu Z. TaggerOne: joint named entity recognition and normalization with semi-Markov Models. Bioinformatics. 2016 Sep 15;32(18):2839-46.
5. Ratner A, Bach SH, Ehrenberg H, Fries J, Wu S, Ré C. Snorkel: Rapid training data creation with weak supervision. InProceedings of the VLDB Endowment. International Conference on Very Large Data Bases 2017 Nov (Vol. 11, No. 3, p. 269). NIH Public Access.
6. Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805. 2018 Oct 11.
7. Lee J, Yoon W, Kim S, Kim D, Kim S, So CH, Kang J. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics. 2020 Feb 15;36(4):1234-40.
8. Forcier J, Bissex P, Chun WJ. Python web development with Django. Addison-Wesley Professional; 2008 Oct 24.
9. Rahman P, Nandi A. Transformer: a database-driven approach to generating forms for constrained interaction. InProceedings of the 24th International Conference on Intelligent User Interfaces 2019 Mar 17 (pp. 485-496).
10. Rahman P, Jiang L, Nandi A. Evaluating interactive data systems. The VLDB Journal. 2020 Jan;29(1):119-46.

11. Bangor A, Kortum PT, Miller JT. An empirical evaluation of the system usability scale. Intl. Journal of Human–Computer Interaction. 2008 Jul 29;24(6):574-94.
12. Rahman P, Nandi A, Hebert C. Amplifying Domain Expertise in Clinical Data Pipelines. JMIR Medical Informatics. 2020 Nov 5;8(11):e19612.