Biomedical Corpus Filtering: A Weak Supervision Paradigm With Infused Domain Expertise

Sejal Dua,[1,3] **Ioana Baldini,**[1] **Dmitriy Katz-Rogozhnikov,**[1] **Emily van der Veen,**[2,4] **Allison Britt,**[2,5] **Pradeep Mangalath,**[2] **Laura B. Kleiman,**[2] **Catherine Del Vecchio Fitz**[2]

[1]IBM Research, [2]Reboot Rx, [3]Tufts University, [4]Colby College, [5]Bowdoin College

Abstract

Querying biomedical documents from large databases such as PubMed is traditionally keyword-based and usually results in large volumes of documents that lack specificity. A common bottleneck of further filtering using natural language processing (NLP) techniques stems from the need for a large amount of labeled data to train a machine learning model. To overcome this limitation, we are constructing an NLP pipeline to automatically label relevant published abstracts, without fitting to any hand-labeled training data, with the goal of identifying the most promising non-cancer generic drugs to repurpose for the treatment of cancer. This work aims to programmatically filter a large set of research articles as either relevant or non-relevant, where relevance is defined as those studies that have evaluated the efficacy of non-cancer generic drugs in cancer patient populations. We use Snorkel, a Python-based weak supervision modeling library, which allows domain expertise to be infused into heuristic rules. With a robust set of rules, promising classification accuracy can be cheaply achieved on a large set of documents, making this work easily applicable to other domains.

A Natural Language Processing Pipeline for Drug Repurposing in Cancer

Natural language processing (NLP) is currently being applied at scale to sift through millions of published biomedical studies and synthesize data from a portion that are deemed relevant. In order to successfully extract information from these studies, one must query a database with a combination of keywords related to the scope of the research. As a result, irrelevant studies that happen to match the keyword search but do not actually pertain to the initial intent must be filtered out of the document corpus. This issue motivates the need for a binary filtering model that can determine document relevance based on certain criteria.

The work presented in this paper is part of a collaboration between cancer biology domain experts and data scientists to construct an NLP pipeline for the task of identifying the most promising FDA-approved non-cancer generic drugs to repurpose for the treatment of cancer [9]. While this ambitious endeavor requires several steps in order to extract drug-cancer evidence from scientific documents and ultimately arrive at a small set of drugs for further study, this paper focuses on the corpus filtering task. The premise of the approach presented in this paper is to build a model for understanding document "relevance" by way of de-noising many signals from a set of PubMed titles and abstracts automatically labeled by rules developed by domain experts. While in principle, a state-of-the-art BERT-based model [1] would presumably achieve higher accuracy for a binary classification task like the one under consideration, it also requires a large corpus of manually annotated documents, which is costly and time-consuming. These hand-labeled training sets can take months or years to develop for large benchmark sets, and require annotators with domain expertise since the type of documents under consideration are full of domain-specific jargon. Thus, we aim to circumvent this bottleneck by leveraging the knowledge of domain experts in order to construct a rule-based model that can programmatically label hundreds of thousands of documents with promising accuracy. This type of rule crafting takes considerably less time and it is less tedious than annotating thousands of documents.

Weak Supervision and Snorkel

In practice today, most machine learning systems use some form of weak or distant supervision: noisier, lower quality, but larger-scale training sets constructed via strategies such as using annotators, programmatic scripts, or high-level input from domain experts [6]. The intent is to harness human supervision more cheaply and efficiently. In this work, we encode domain expertise into heuristic rules while taking advantage of existing resources (i.e., knowledge bases, pre-trained models). This method is advantageous for research applications in which a few dozen noisy rules or high-level constraints are able to effectively perform some task with comparable accuracy and at a much lower cost than a large set of labels from domain experts [6].

In order to apply weak supervision to the filtering task within the NLP-based drug repurposing pipeline, we use a software package called Snorkel[1] [8]. Snorkel is an open source framework that is grounded in data programming, a

[1]Snorkel is a data programming paradigm that programmatically builds training data for supervised machine learning.

field in which labels are derived from noisy label sources using generative models. Snorkel effectively de-noises signals from a given corpus without fitting to any labeled data, by implementing the following three key steps:

1. Construct heuristic rules called labeling functions (LFs). These rules are declared by humans, usually domain experts, and represent the only manual step in the Snorkel approach. Apply each of these m rules on all n documents to generate an m x n label matrix.
2. Snorkel pools noisy signals from the label matrix into a generative model using a factor graph approach which learns from the agreements and disagreements of the labeling functions, without access to any ground-truth data [7]. The output of this generative model are predictions for the binary classification of each document.
3. The predictions from the previous step can be used as probabilistic training labels for a noise-aware discriminative model which is intended to generalize beyond the information expressed in the labeling functions.

To make it easier to define labeling rules, Snorkel adds a special label to the set of labels of the classification task: `ABSTAIN`. Whenever a rule can not make a decision for one of the labels for the task, it emits the `ABSTAIN` label. For our task, it is much easier to enumerate inclusion rules (i.e., labeling functions for documents that are considered relevant) than exclusion rules. For this reason, we experiment with marking all `ABSTAIN` labels as non-relevant.

Biomedical Research Corpus

PubMed, provided by the National Center for Biotechnology Information (NCBI), comprises over 40 million biomedical studies from MEDLINE, life science journals, and online books. The large set of unlabeled research studies to be programmatically filtered is sourced from PubMed using a Cochrane highly sensitive search (CHSS) strategy [2] to narrow the scope of our evidence discovery pipeline. Note that this query, even with certain keyword terms listed and publication types specified, does not yield only *relevant* articles, thus motivating the filtering task. In our experience, only about 30% of the articles end up being relevant for our purposes.

The labeled set of documents for testing our procedure was manually generated by our domain experts. In this work, we focus on clinical studies and consider only publication abstracts. In our experience so far, publication abstracts are sufficiently detailed to decide whether an article is of interest or not. The different dataset splits that we used in our work are provided below with a brief explanation on how we used each split.

- **Unlabeled set** [39843 documents]: The largest split, with no ground truth labels.
- **Test set** [1413 documents]: A small, hand-labeled set for final evaluation of our classifier; this dataset is not available for inspection, only for evaluation such that our rules are not biased.
- **Development set** [300 documents]: A small set of labeled documents used for inspection in the creation of rules and error analysis after the model has been applied.

After initially querying PubMed, the datasets were split, duplicates were removed, and metadata was collected. The strongest source of signal was the title, which makes sense since it is the field with the most essential elements of the work described, including the drug and cancer type, and sometimes the type of study. An additional helpful feature was cancer concepts mentioned in the abstract and extracted via the Unified Medical Language System (UMLS) Linker based on ScispaCy [5].

Encoding Biomedical Expertise

We devised a workflow for deeming articles as either relevant for the NLP-based drug repurposing pipeline (`INCLUDE`) or not relevant (`EXCLUDE`). A document is relevant if a non-cancer generic drug was tested for the treatment of cancer and if a phenotype-level outcome was reported. Some of the domain-level expertise encapsulated in this step includes terms that are frequently associated with cancer, deceptive terms that seem to be related to cancer but are actually not related (e.g., tumor necrosis factor), and relevant biomedical processes. The sequential workflow was manually converted into parallel, independent labeling functions in accordance with Snorkel's Label Model package. These rules were treated as a baseline for our Snorkel model.

Labeling Functions

The construction of the rules followed an iterative fashion, starting with simpler rules based on keywords, then leveraging metadata from PubMed, and eventually evolving to more sophisticated rules encapsulating named entity recognition (NER) models and entity linkers.

```
@labeling_function()
def lf_premalignant_and_prevent(x):
    return EXCLUDE if "premalignant" in
        str(x['Abstract']).lower() and "prevent"
        in str(x['Abstract']).lower() else ABSTAIN
```

Figure 1: Keyword-based labeling function

The baseline workflow included several keyword-based rules which were effective in identifying strong sources of signal and eliminating unwanted noise. Some examples are provided below.

- `lf_necrosis_factor`: EXCLUDE if the paper does not have cancer in the title and has a mention of tumor necrosis factor (TNF), a cell signaling protein involved in systemic inflammation.
- `lf_premalignant_and_prevent`: EXCLUDE if the abstract mentions both "premalignant" and "prevent", implying preventative interventions where a patient has not been diagnosed with cancer. (see Figure 1).

Following the baseline set of rules, term frequency-inverse document frequency (TF-IDF) analysis and general exploratory data analysis (EDA) were performed on the corpus, motivating many of the rules. By splitting the documents in the development set by their "ground truth" labels, simple language patterns were identified without relying on a domain expert. As an example, association analysis

Table 1: Corpus Filtering Accuracy

	Accuracy	
Iteration	De-noising	Majority Vote
Baseline	71.1	74.9
Refined	75.4	78.9
Refined + PubmedBERT	75.5	76.6
Optimized	79.1	81.1

was performed using the Apriori principle [4] to generate rules from the most frequent item sets. While this approach required a significant amount of data preprocessing which could not be directly converted into a lightweight labeling function, it inspired the creation of more refined rules which captured these programmatically identified sources of signal.

```
@labeling_function()
def lf_title_triplet(x):
    flag_cancer = get_cancer_flag(x['Title'])
    flag_pt = get_pubtype_flag(x['Title'])
    flag_outcome = get_outcome_flag(x['Title'])
    return INCLUDE if all([flag_cancer, flag_pt,
      flag_outcome]) else ABSTAIN
```

Figure 2: Flag-based labeling function

More robust rules involved a combination of checks to ensure that cancer is the focus of the paper, the publication type is clinical, and a relevant outcome term is mentioned. These rules are characterized by their high precision but low coverage. An example of such a rule is shown in Figure 2.

```
from snorkel.preprocess.nlp import *
spacy=SpacyPreprocessor(language='en_core_sci_lg')
@labeling_function(pre=[spacy])
def lf_neoplastic_process(x):
    for ent in x.doc.ents:
        if ent.label_ == "DRUG":
            return INCLUDE
    return ABSTAIN
```

Figure 3: NER-based labeling function

Finally, a last set of rules involved NER models [5] to detect a wide range of cancer types and salient drug mentions (see Figure 3 for an example). We used ScispaCy [5] with 'en_core_sci_lg' as the language model, in order to identify if the same UMLS cancer concept was being discussed in both the title and abstract. NERs and concept linkers are more robust than keyword-based labeling functions because they can detect that "Acute lymphocytic leukemia", for example, is mentioned several times in the abstract, using the acronym "ALL".

Experimental Results

Corpus Filtering Accuracy

The results in Table 1 show the accuracy of the predictions for the test set as compared to manually annotated ground truth labels. The two rightmost columns denote which Snorkel model was used to pool the signals from the label matrix together and output a single probabilistic label for each document. De-noising refers to the Label Model, which uses a generative model to discern the labels, while Majority Vote applies a simple majority heuristic to choose the final label for a document. While the de-noising model is the most novel part of this approach, the Majority Vote Model achieved better accuracy due to the sparsity of the label matrix caused by an abundance of `ABSTAIN` labels. This implies that the rules were most likely not broad enough to train an optimal generative model. From the baseline rules, including specialized rules that involve NER led to an increase in performance accuracy by several points, as indicated by the Refined row in the table. Following the de-noising step of the Snorkel pipeline, we trained several BERT-variant discriminative models [2] in order to boost the accuracy of the predicted labels and generalize beyond the information expressed in the labeling functions. In Table 1, we show the results for PubmedBERT [3], which yielded the highest accuracy among the BERT-based variants explored in the experiment. The discriminative model, while consistent across several runs, does not seem to help improving the accuracy of the filtering technique.

Table 2: Classification Statistics for Optimized Model

	Precision	Recall	F1-score	Support
`EXCLUDE`	0.88	0.75	0.81	593
`INCLUDE`	0.70	0.85	0.77	419

The final iteration of the model is shown in Table 1 as Optimized. Despite an imbalance in the dataset, performance can be further measured using precision, recall, and F1-score for both classes. Table 2 shows that precision was higher for the `EXCLUDE` class while recall was higher for the `INCLUDE` class. The Majority Vote Model accuracy of 81.1% using the optimized set of rules—including NERs and entity-linking—is promising given the low human effort involved.

Prediction Confidence

From the refined set of rules to the final iteration of the model, many of the noisy rules were modified or removed entirely. Increasing precision at the expense of reduced coverage resulted in lower confidence for a sizable fraction of the corpus, as indicated by the middle portion of the histogram in Figure 4. For some data mining tasks (e.g., identifying cancer types with a considerable amount of clinical studies where non-cancer generic drugs were tested), the ability to extract large amounts of relevant articles in a short time without necessarily knowing all the relevant articles is highly desirable. For this reason, we look at the accuracy of the predictions for the highly confident predictions. When exploring documents for which Snorkel predicts labels with high confidence [3], the Majority Vote Model yields 86% ac-

[2] We experimented with several BERT models that are trained on biomedical and/or scientific data.

[3] High confidence is defined by a probabilistic score less than 0.2 or greater than 0.8 (see Figure 4).

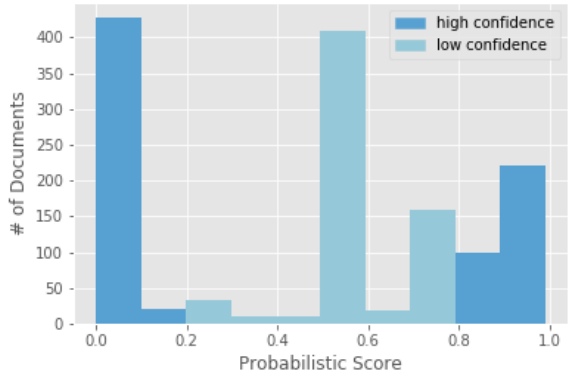

Figure 4: Distribution of probabilistic scores outputted by the Snorkel model

curacy. In the probabilistic score distribution in Figure 4, 0 and 1 indicate the most confident "include" and "exclude" predictions, respectively.

Labeling Function Coverage

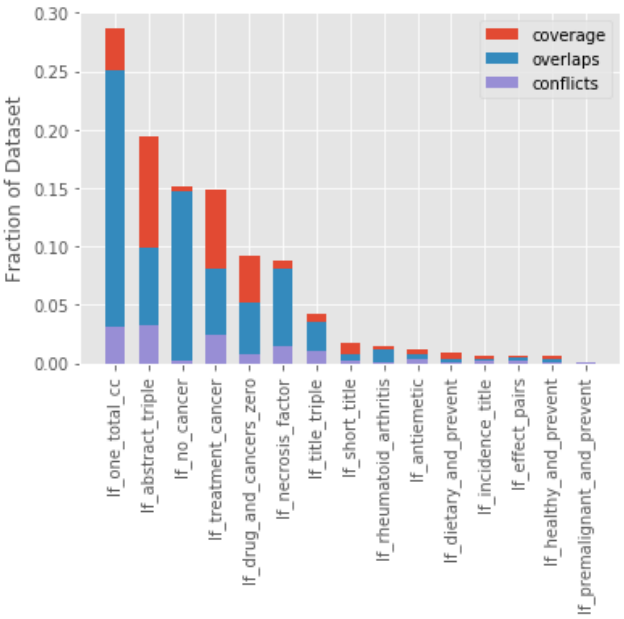

Figure 5: Visualization of coverage, overlaps, and conflicts for each LF, sorted from highest coverage to lowest coverage

In this work, the coverage-precision tradeoff was continually assessed with each iteration of the model. Minimizing the percentage of conflicts introduced with each labeling function was considered the highest priority given the biomedical research application. Fortunately, it was possible to examine a summary of coverage, overlaps, conflicts, and weights for each LF involved in the pipeline [4], shown in Figure 5. We used these statistics to refine the rule-sets used in our experiments. For example, some rules that we initially considered as important and informative ended up having low coverage in practice and were removed in later iterations. By having clear performance metrics associated with each LF, the model can be optimized by both computer science researchers and domain experts alike.

[4]This functionality is encapsulated in the Snorkel Python library in a function called `LFAnalysis`

Conclusions

Snorkel is a promising pipeline for NLP applications in the biomedical research domain due to its ability to achieve significant accuracy on difficult filtering tasks without reliance on a large set of labeled data. In this paper, we demonstrate such a use case that leads to accuracy of 78.9% for filtering PubMed abstracts pertinent to our task of identifying the most promising drug repurposing opportunities for cancer. As a future extension of this work, we would like to understand the interplay between weak supervision and language model-based discriminative models.

References

[1] Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*.

[2] Dickersin, K.; Scherer, R.; and Lefebvre, C. 1994. Identifying Relevant Studies for Systematic Reviews. *BMJ* 309: 1286.

[3] Gu, Y.; Tinn, R.; Cheng, H.; Lucas, M.; Usuyama, N.; Liu, X.; Naumann, T.; Gao, J.; and Poon, H. 2020. Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing.

[4] Kumbhare, T. A.; and Chobe, S. V. 2014. An Overview of Association Rule Mining Algorithms. *International Journal of Computer Science and Information Technologies* 5: 927–930.

[5] Neumann, M.; King, D.; Beltagy, I.; and Ammar, W. 2019. ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing. arXiv:1902.07669.

[6] Ratner, A.; Bach, S. H.; Ehrenberg, H.; Fries, J.; Wu, S.; and Ré, C. 2017. Snorkel: Rapid Training Data Creation with Weak Supervision. *Proc. VLDB Endow.* 11(3): 269–282. ISSN 2150-8097. doi:10.14778/3157794.3157797. URL https://doi.org/10.14778/3157794.3157797.

[7] Ratner, A.; Hancock, B.; Dunnmon, J.; Sala, F.; Pandey, S.; and Ré, C. 2019. Training Complex Models with Multi-Task Weak Supervision. *AAAI Conference on Artificial Intelligence* 33: 4763–4771.

[8] Ratner, A.; and Varma, P. 2019. Snorkel. https://github.com/snorkel-team/snorkel. [Online; accessed 09-09-2020].

[9] Subramanian, S.; Baldini, I.; Ravichandran, S.; Katz-Rogozhnikov, D. A.; Ramamurthy, K. N.; Sattigeri, P.; Varshney, K. R.; Wang, A.; Mangalath, P.; and Kleiman., L. B. 2020. A Natural Language Processing System for Extracting Evidence of Drug Repurposing from Scientific Publications. In *IAAI*.