Labeling data-reuse statements through Active Learning

Gilbert Wong
Loyola University Maryland

April 7, 2020

Table of Contents

List of Figures

Introduction

The National Institute of Mental Health (NIMH) is the lead federal agency for research on mental disorders. The National Institute of Health (NIH) is a large agency that consists of 27 institutes and the NIMH is one of them. In turn, the NIH is an agency of the United States Department of Health and Human Services and the primary agency of the United States government responsible for biomedical and health-related research. The heath field has been an involving area of study and with the advancement of data analytics, more physicians, and medical researchers are leaning towards the area of data science to assist them in their diagnosis, treatment, and research. However, before all the data science magic can happen, researchers, would need a good sample size of training data and this project explores a technique that could be used to lessen the burden of manually labeling data.

The problem with data science at the NIH is figuring out what data is important for their research. Many researchers reuse data for multiple areas but knowing what data is being used is important so that the proper federal funds and grants are put in the right places. This project aims to solve that. If this type of labeling is done manually, someone would take a random sample of sentences in the research papers and then figuring out whether the information is important or not. It's like searching for a needle in a haystack. What the NIMH wants to do is let a classifier tell them which statements they are most likely to gain the most information from and manually label that particular record; think of it like a feedback loop where the model is constantly being retrained as new data comes into the system. This concept is called Active Learning (AL). According to Wikipedia, AL is a special case of machine learning (ML) in which an algorithm is able to interactively query the user (or some other information source) to obtain the desired outputs at new data points. [1] There are an abundance of unlabeled data and manually labeling them is expensive. By using this technique, learning algorithms can actively query the user/teacher for labels. This is what makes the problem interesting since we, as data scientists, know that labeling data is expensive. The client would like to run the classifier on a mobile device and hosted on a cloud-based infrastructure that they can serve up and take input from a mobile-friendly interface.

The two biggest challenges were figuring out how to incorporate the concept of AL into the model but before that can happen, figuring out how to properly vectorize the input, also known as normalization, into something that the ML algorithm could understand and then figuring out how to deploy the model onto a mobile app for the client. Other challenges included exploring different tokenizers that would split the text into desired tokens (as will be explained later in more detail), as well as exploring different methods of natural language processing (NLP) to prepare the text data before it is fed into a ML model. By giving the client a way to label research papers in a more efficient manner, it allows them to tie the reuse data back to its original source and funding which gives them the ability to see which data is more useful so that more funding can be granted to that particular source. The NIH is at the forefront of many health-related discoveries and by having this particular knowledge at their disposure, they can justify the need for more funding/grants and make more medical breakthroughs that would save lives.

Data science management and research platform

The overall plan of analysis was building three programs; one to ingest three common separated value (CSV) files (describe in more detail in a later section) into a SQLite database and then a second program to use a simple JOIN SQL statement to pull only the IDs of interest and passing that ID to an API call so that the proper data is retrieved. The final program is the main analysis which included preprocessing of the data and building a series of ML models. The primary language used is Python and several key packages used were scikit-learn, nltk (Natural Language toolkit), spacy [2] (NLP library), and modAL [3] (AL wrapper around scikit-learn). In terms of what the client asked for about being able to label data on their phones, the application is written in React Native. The three programs described weren't complex (about 1000 lines of code) which involves downloading, filtering, preprocessing and modeling (different ML models were explored). The mobile app, which is still in the development stage, is a little more complex since it involves not only the frontend interface (look and feel of the app) but it also incorporates a backend for the UI to interact with. Some details of the mobile app will be touched upon at the end of this paper (but wouldn't go into much detail since it's not the major point of this project and it's something the client wanted as bonus).

There were no special hardware or software used in the project; only specific packages for NLP and AL. No interactive development environments (IDE) were used and everything was done via the command line and a basic text editor. In the proposal, one topic that wasn't touched on was how to turn text data into an array of normalized values that the ML models could understand. In the beggining, basic vectorizers were explored but as the project progressed and more research was done, a huge part of the preparation stage was missed and that was how these vectorizers worked in preprocessing and tokenizing of the data. The text data consists of email addresses, urls, etc. that weren't properly tokenized using the defaults of the vectorizers. Therefore, more research was done and it turns out that the vectorizers offered by scikit-learn allows one to create custom preprocessors and tokenizers to pass to the initialization of the classes. Therefore, simple custom functions were created to account for this and some parts of the code had to be refactored.

Literature on related work

Researchers, regardless of the domain, have done a lot of work in the areas of labeling. According to an article [4], the market for data labeling passed $500 million in 2018 and it will reach $1.2 billion by 2023. It accounts for 80 percent of the time spent building Artificial Intelligence (A.I.) technology. It is ironic that ML, tool used for the automation of tasks and processes, often starts with the highly manual process of data labeling. The task of creating labels to teach computers new tasks is quickly becoming the blue collar job of the 21st century. There are work being done to create this ability to allow one to automate the process for creating data labels; this is highly desirable from a cost, time and even ethical standpoint (although this is creating thousands of jobs, workers are often underpaid and exploited). Aside from AL, there is another python library called Snorkel. Difference between AL and Snorkel is that AL introduces human expertise into the loop to smartly label a small

set of data where Snorkel removes humans from the labeling process. Snorkel is a really innovation concept since it creates a series of messy label functions and combine these in an intelligent way to build labels for a dataset. [4] The labels then could be used to train a ML model in the same way as a standard ML workflow.

Snorkel has been around since 2016 and continues to improve. It is used by many big names in the industry such as Google, IBM, and Intel. Version 0.9 of the library came out in 2019 which provided a more sophisticated way of building a label model, as well as a suite of well documented tutorials covering all of the key features. Although this library is an innovative concept but the process is very simple. It consides of mainly four steps.

- This first step is optional but is helpful for reviewing performance of the final model. Create a small subset of golden labels for items within the dataset.
- Write a series of label functions which define the different classes across the training data.
- Build a label model and apply this to the dataset to create a set of labels.
- Use labels in the normal ML pipelines.

The process is iterative and most likely, one would evaludate the results and re-think and refine the label functions to improve the output.

This project is focused on the concept of AL and after some research, there have been a number of publications regarding AL published by the NIH. Two specific articles found talks about interactive ML (iML) for health informatics [5], and using AL for electronic medical record de-identification [6]. As a research institute, the NIH publishes many research papers (available at *https://www.ncbi.nlm.nih.gov/pmc/*; it is also where all the text data for this project comes from) that touches on a variety of health related topics. Interestingly, the [5] was cited 25 times by other publications and [6] was cite once.

Many ML researchers concentrate on automatic ML (aML) which works really well for speech recognition, recommender systems, or autonomous vehicles but these automatic approaches only works from big data with many training sets. However, according to [5], in the health domain, there are many instances where they have to deal with a small number of data sets or rare events, where aML isn't efficient due to insufficient training samples. Therefore, the concept of AL can help since it optimizes the learning behavior through interactions with agents (where agents could be a human). This human-in-the-loop can be extremely important in solving computationally hard problems such as subspace clustering, protein folding or k-anonymization of health data, where human expertise can help to reduce an exponential search space. This need of AL in the health domain is crucial because biomedical data sets are full of uncertainty, incompleteness, etc. and they can contain missing data, noisy data, dirty data, unwanted data, and more importantly, some problems in this domain are hard, which makes fully automated approaches difficult or even impossible. In the below figure, there is four ML workflows. A illustrates an unsupervised pipeline, B supervised, C semi-supervised, and D shows the iML approach where one input data, pre-process the data, human agent(s) interacting with the computational agent(s)), and final check done by the human expert. Scenario A illustrates the pipeline where learning is fully automatic and

does not require a human to manually to label the data. Scenario B is where humans are providing labels for the training data then selecting features to feed the algorithm to learn (the more samples the better) and then the human expert can check results at the end of the pipeline.)Scenario C is kind of a mixture of A and B where one mixes labeled and unlabeled data, so that the algorithm can find labels according to a similarity measure to one of the given groups. Scenario D, as briefly mentioned above, is where the human expert is seen as an agent directly involved in the actual learning phase, step-by-step influencing measures; however, many questions remain open and needs further research, in terms of evaluation, robustness, etc.

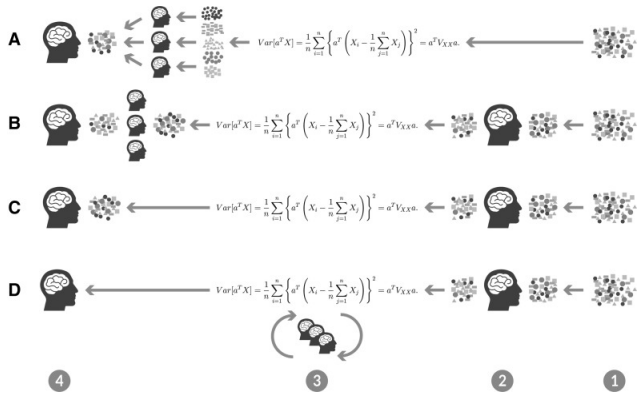

Figure 1: ML pipelines [5]

There are still many evidence that humans sometimes still outperform ML algorithms (ex. diagnostic radiologic imaging). The goal is to integrate the physicians high-level expert knowledge into the retrieval process by acquiring his/her relevance judgments regarding a set of initial retrieval results. The reason for the popularity of aML approaches is that it is much better to evaludate and therefore, more publishable, as opposed to iML, where correct experiments and evaluations are not just more difficult and time-consuming, but very difficult to replicate, due to the fact that human agents are subjective compared to data, algorithms, and computational agents.

Privacy is a huge issue in today's world where data is more widely assessible. Therefore, in the medical field, ensuring privacy for the patients remains one of the primary challenges to disseminating such data. [6] To protect someone's privacy, heath care organizations rely upon the de-identification standard of the Privacy Rule of the Health Insurance Portability and Accountability of 1996. It is straightforward to protect health information (PHI) (personal name, dates of birth, geocodes of residence) but it is more challenging to do so for clinical information where data is more free or semi-structured form. As mentioned earlier, ML can assist in such task, however, this ML approach requires the presense of sufficiently high-quality training data and it must be accomplished under limited budgets to informatics team running such systems. Therefore, the paper focuses on using AL in the process to reduce the overall cost for annotation and support the establishment of a more scable de-identification pipeline. Figure 2 below illustrates the overall pipeline of such tasks which is very similar to the pipeline for this project where, instead of tagging PHI, we're tagging re-use statements from research papers. Similarly, we use a small batch of data that is selected randomly from the

dataset as the starting point of the AL then humans manually tag the data in the initial batch of data to create a gold standard for model traning. The concept of AL has proven to be an effective tool in named entity recognition tasks in clinical text and studies show that AL is more efficient than more passive learning. It also suggests that uncertainty sampling (which will be touched upon later) was the best strategy for reducing the annotation cost; therefore, it should be taken account when evaluating the performance of AL. There are still lots of research being done to see the full impact of what AL can do for the health related field, but it has already proven useful in some research areas and this concept will continue to evolve.

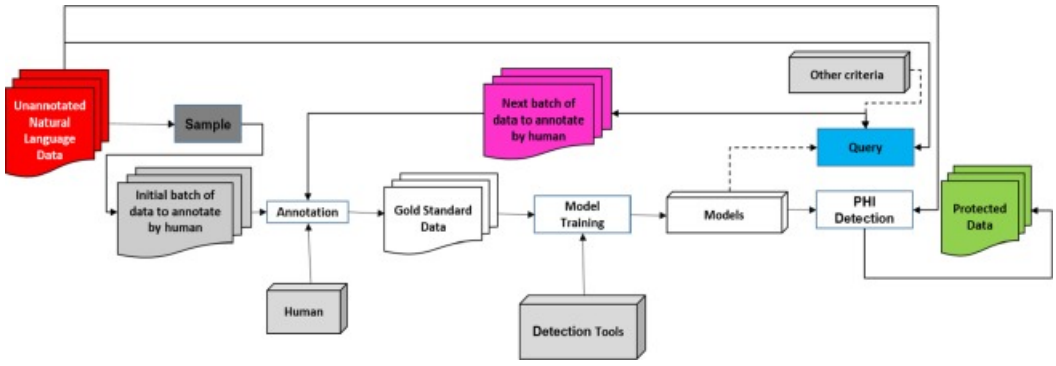

Figure 2: Pipeline for AL framework for natural language EMR de-identification [6]

Hypothesis/Dataset

References

[1] Active learning (Machine Learning) 23 *March* 2020. In Wikipedia. Retrieved from *https://en.wikipedia.org/wiki/Active_learning_(machine_learning)*

[2] Industrial-Strength Natural Language Processing. Available at *https://spacy.io/*

[3] modAL: A modular active learning framework for Python3. Available at *https://modal-python.readthedocs.io/en/latest/index.html*

[4] Tayor, Josh (2020, March 5). No labels? No problem! towards data science. *https://towardsdatascience.com/no-labels-no-problem-30024984681d*

[5] Holzinger, Andreas *Interactive machine learning for health informatics: when do we need the human-in-the-loop?*, Brain Informations. 2016

[6] Li M, Scaiano M, El Emam K, Malin B. *Efficient active learning for electronic medical record de-identification*. AMIA Jt Summits Transl Sci Proc. 2019

[7] Comeau DC, Wei CH, Islamaj Doğan R, and Lu Z. *PMC text mining subset in BioC: about 3 million full text articles and growing*, Bioinformatics, btz070, 2019