Predicting the Likelihood of Success for Medicare and Medicaid Appeals

Ben Fogarty, Lilian Huang, Katy Koenig

Problem Statement

- Predict whether a Medicare/Medicaid adjudicatory decision can be successfully appealed
- Medicare/Medicaid appeals
  - Have multiple stages
  - Are time-consuming
  - Are costly
- Enable more informed decisions by petitioners
- If this approach succeeds, can be applied to the numerous other policy domains with a similar structure of administrative law decisions and appeals
  - e.g. the National Labor Relations Board, the Department of Labor, and the Department of Housing and Urban Development
- Providing insight into the possibility of automated policy/legal decision-making

Overview of Policy

- Healthcare providers can file appeals if they disagree with a decision made by Medicare/Medicaid
  - Exclusion from Medicare/Medicaid, i.e. revocation of billing privileges
  - Monetary penalties
- Two stages
  - Initial hearing by Administrative Law Judge (ALJ)
  - Appeal to Departmental Appeals Board (DAB) for review of ALJ's decision

Related Work

- Krass (2019)
  - Using lower court decisions to predict higher court rulings
  - Rulings on veterans’ disability claims
  - Key takeaways:
    - Restrict dataset to cases that are actually appealed
    - General pre-trained embeddings perform poorly on legal texts
- Chalkidis et al. (2019)
  - Using neural networks to predict outcome of court cases
  - States' human rights violations
  - Key takeaways:
    - Limit scope to binary classification problem
    - Text pre-processing steps
    - Consider class and temporal distribution in training/validation/test split

Project Methodology

- Collect dataset through web scraping
  - Store in PostgreSQL cloud database
- Text pre-processing
  - Tokenizing
  - Lemmatizing
- Building and evaluating predictive models

Data Collection: Anatomy of an Appeal

ALJ text (introduction)

Nancy L. Clark, DAB CR5483 (2019)

ALJ Case Number — ALJ Year

Department of Health and Human Services
DEPARTMENTAL APPEALS BOARD
Civil Remedies Division

Nancy L. Clark
(OI File No. H-18-41555-9),
Petitioner,

v.

The Inspector General,

Respondent.

ALJ Text

Docket No. C-19-659
Decision No. CR5483
December 3, 2019

DECISION

The Inspector General of the United States Department of Health and Human Services (the IG) excluded Nancy L. Clark (Petitioner) from participation in Medicare, Medicaid, and all other federal health care programs for five years based on her conviction for a criminal offense related to neglect or abuse of patients, in connection with the delivery of a health care item or service. Petitioner sought review of her exclusion. For the reasons stated below, I affirm the IG's exclusion determination.

DAB text (introduction)

Nancy L. Clark, DAB No. 2989 (2020)

DAB Case Number — DAB Year

Department of Health and Human Services
DEPARTMENTAL APPEALS BOARD
Appellate Division

Nancy L. Clark

Docket No. A-20-25
Decision No. 2989
March 9, 2020

FINAL DECISION ON REVIEW OF ADMINISTRATIVE LAW JUDGE DECISION

ALJ Case Number — ALJ Year

Nancy L. Clark (Petitioner) appeals a decision by an Administrative Law Judge (ALJ) upholding on the written record the Inspector General's (I.G.) exclusion of Petitioner from participation in all federal health care programs for a period of five years. *Nancy L. Clark*, DAB CR5483 (2019) (ALJ Decision). The ALJ concluded that the I.G. properly excluded Petitioner based on her conviction for a criminal offense related to neglect or abuse of patients in connection with the delivery of a health care item or service, pursuant to section 1128(a)(2) of the Social Security Act (Act), [1] which requires a minimum exclusion period of five years (Act § 1128(c)(3)(B)).

Appeal Outcome

For the reasons set out below, we reject Petitioner's arguments and affirm the ALJ's decision.

Data Collection: Overview

- No API to access case information; instead we needed to build a web scraper
- Web scraper tasks
  - Collect all DAB (appeal) decisions from the HHS website
  - Identify each DAB case
  - Scrape the text of each DAB case
  - From the text of each DAB case, identify:
    - Which, if any, ALJ case is associated the the DAB case
    - The outcome of the DAB case (overturned/upheld)
  - Scrape the text of the associated ALJ case

Data Collection: Challenges

- Comprehensive metadata is not available for either DAB cases or ALJ case
  - No well-formatted description of which case was being appealed, the outcome, the date of the case, etc.
  - Instead, this information must be extracted from the text of the DAB case
  - Fortunately, the judges tend to describe their decisions and the originating cases in predictable ways, meaning we can use regex to extract information
- Changes in format over time
  - This affects our ability to extract specific information from the cases
  - Going further back, there is even greater heterogeneity in case formats
  - Developed best and worst case scenarios for extracting information

Data Collection: Further Challenges

- PDFs
  - Some formats presents cases as PDF documents, which generates a number of issues
  - PDF extraction was a major challenge at the time of our mid-quarter presentation
    - Extracting specific information requires regex to locate where the information is likely to be, then additional regex to locate the information itself
    - PDF to text libraries in Python are imperfect -- they add extra whitespace and mistake similar characters for one another
  - Iterated over different PDF to text libraries to find which one works best, eventually settling on textract (h/t: Thanks, Eric Langowski!)
  - The textract library provides extremely high fidelity text representations of PDF documents and is robust to documents produced both by a computer and scanned from a typewriter

Data Collection: Constraints

- We were only able to collect observations from late 1991 onwards
  - Prior to that, DAB decisions did not explicitly state the case number of the ALJ cases they were referring to
  - Instead, described the ALJ case by date and judge's name, which was beyond the capacity of our web scraper
  - A more complex linkage system would only be marginally helpful in expanding the size of the dataset, since the earliest ALJ decisions available online are from May 1985
- Not all cases that arrive at the DAB originated with an ALJ decision
  - Some involve the Centers for Medicare & Medicaid Services, the Administration for Children and Families, or other bureaucratic decision-making bodies
  - We could not consistently collect the original cases unless they came from an ALJ
  - Year-to-year the composition of where cases originates can vary drastically; for example, during the 1990s, there are years where the DAB heard exceedingly few ALJ appeals because their docket was filled with state health departments implementing Medicaid reforms

Data Collection: Final Methodology

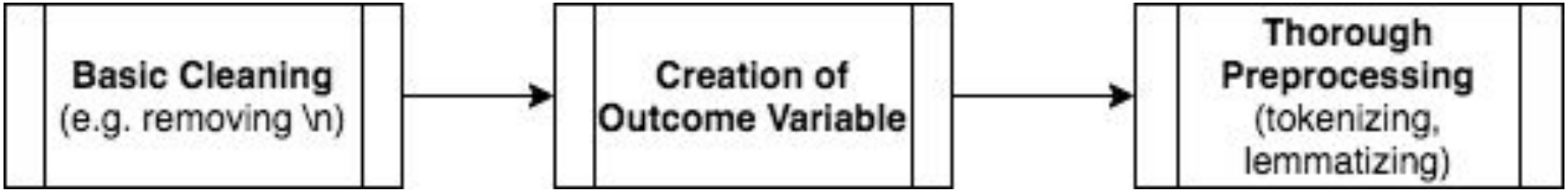
- Begin by cataloguing all the DAB and ALJ cases on the HHS website
- For each DAB cases, create a new Appeal object
    - The Appeals class contains all of our methods for scraping data, linking DAB/ALJ cases, and extracting specific information
    - The constructor for the Appeals class manages the gathering of all the appropriate information in the right order, and contains extensive error handling
- After all DAB cases have been processed, load each successfully processed case to a database table
- Running this process right now, we are able to obtain and process approx. 1,850 cases
- Unfortunately, only 890 of these cases are successfully linked to an ALJ case and have their outcome successfully extracted
    - High success rate at extracting decisions and converting them to a binary variable
    - Primary issues is linking to ALJ cases

Data Collection: Text & Labels

- In our prediction tasks:
  - The text of the ALJ decision is the text
  - The binary decision classification from the DAB (0 if the ALJ decision is affirmed, 1 if the ALJ decision is overturned in part or in full)

Text Preprocessing

- Basic cleaning (e.g. stripping out newline symbols etc)
- Converting outcome into binary variable (overturned or not)
- Tokenizing & preprocessing text using SpaCy
    - Stemming and lemmatizing
    - Removing stop words and punctuation



Exploratory Data Analysis

Of the cases that request an appeal, roughly 31% of cases win their appeals.

Initial Exploratory questions:

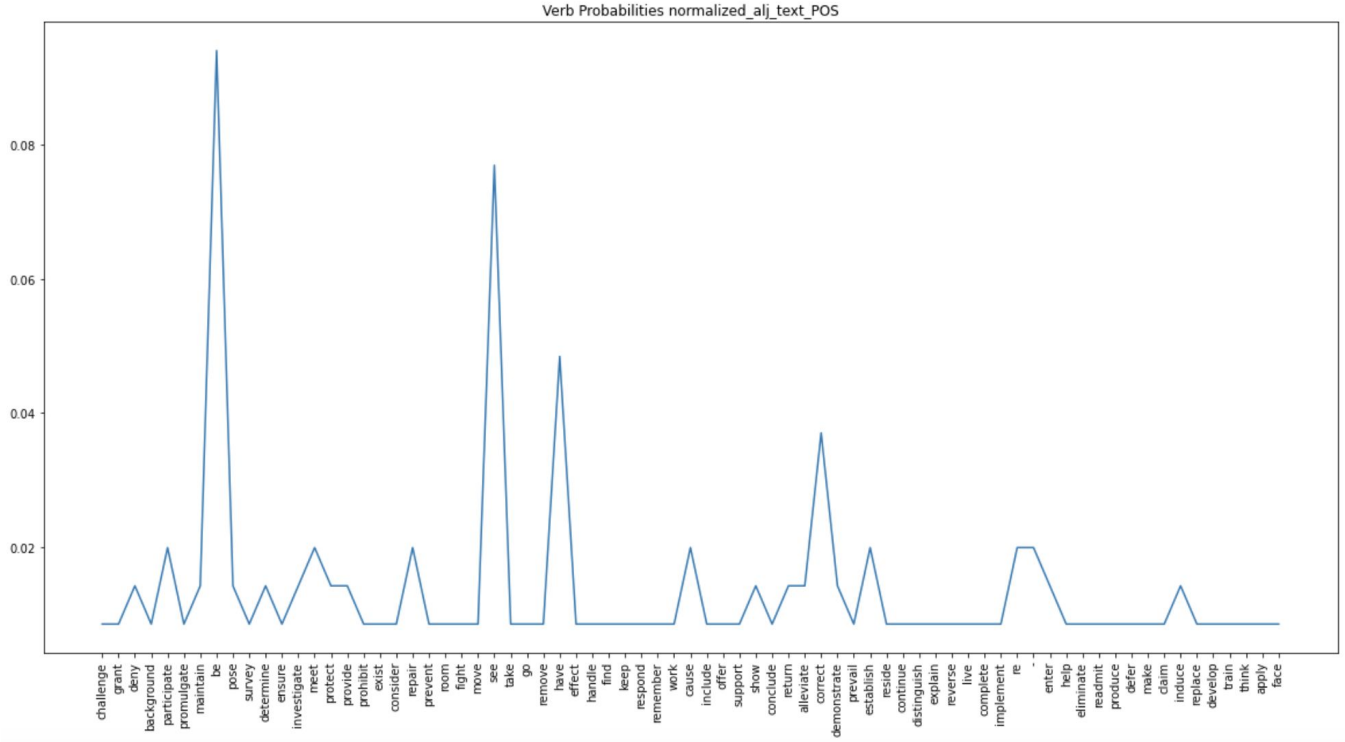
- What are the differences between original decisions and appeals?
- What are the differences between cases that win and lose their appeals?

EDA: Linguistic Divergence

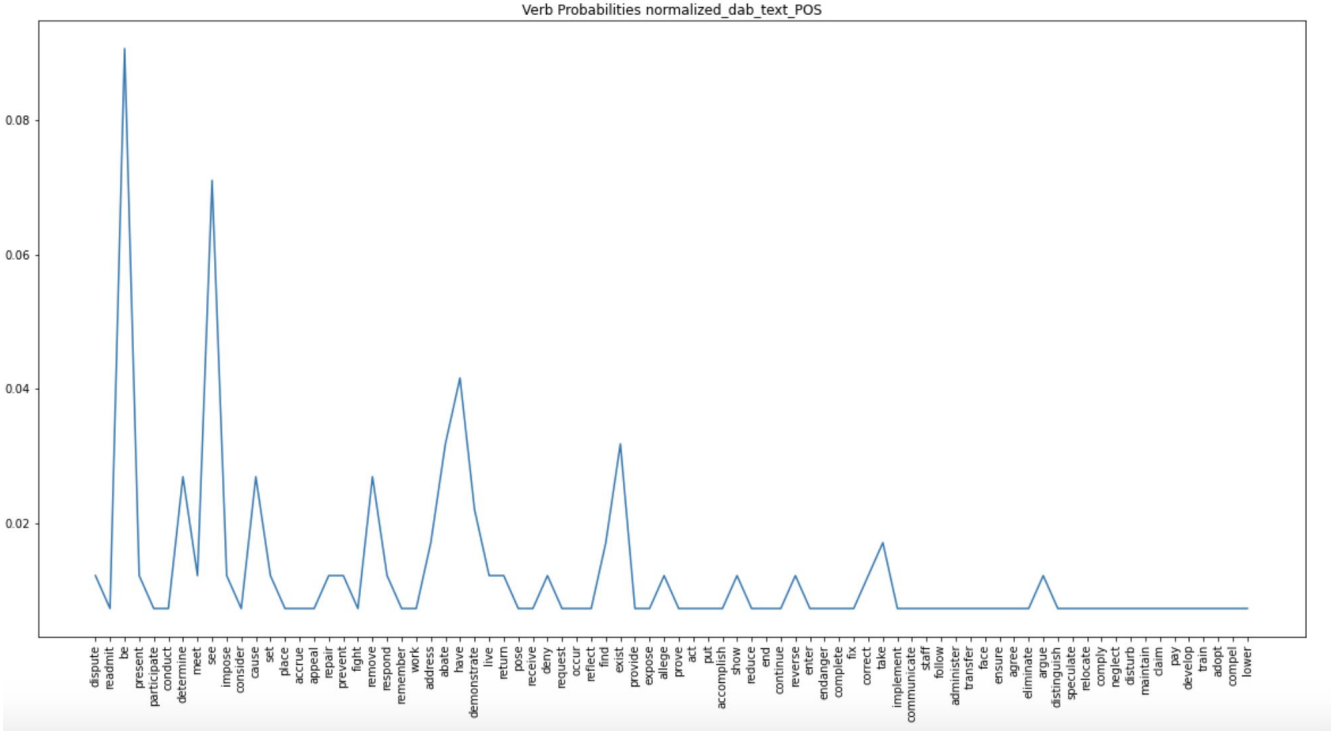
- Wasserstein metric / earth mover's distance
  - The minimum amount of "work" required to transform one text into another

- Applying this metric to successful vs unsuccessful appeals
  - On text of original ALJ decision: 0.00000397
  - On text of final DAB decision: 0.00010519
  - Not much overall difference

Verb Probabilities (ALJ Decision)



Verb Probabilities (DAB Decision)



1-gram and 2-gram Frequencies

From ALJ Text

From DAB Text

Building Models

- Split our data into training, validation, and test sets
  - Temporally ordered, to reflect importance of legal precedent
  - Test set: 2019-2020 cases
  - Validation set: 2017-2018 cases
  - Training set: all previous cases (1991-2016)
- Types of models we used:
  - Simple neural network
  - Recurrent neural network (RNN)
  - Long short-term memory (LSTM)
- Embeddings we used:
  - Self-trained
  - Pre-trained (Law2Vec)
- Weighting to deal with class imbalance
  - Weighted positive examples by ratio of negative:positive examples

Building Models: Tech Stack

- Raw data was stored on an AWS Relational Database Service PostgreSQL instance
- Models were trained using two g4dn.2xlarge EC2 instances with the Amazon Deep Learning Ubuntu 18.04 AMI
- We used ngrok to set up a tunnel to the Jupyter notebooks server running on each instance

Parameters and Hyperparameters

- For simple neural networks:
  - Number of embedding dimensions
  - Number of hidden dimensions
  - Dropout rate
  - Learning rate
- For RNNs and LSTMs:
  - RNN type (simple RNN or LSTM)
  - Number of embedding dimensions
  - Number of hidden dimensions
  - Number of layers
  - Dropout rate
  - Learning rate
  - Bidirectionality

Results

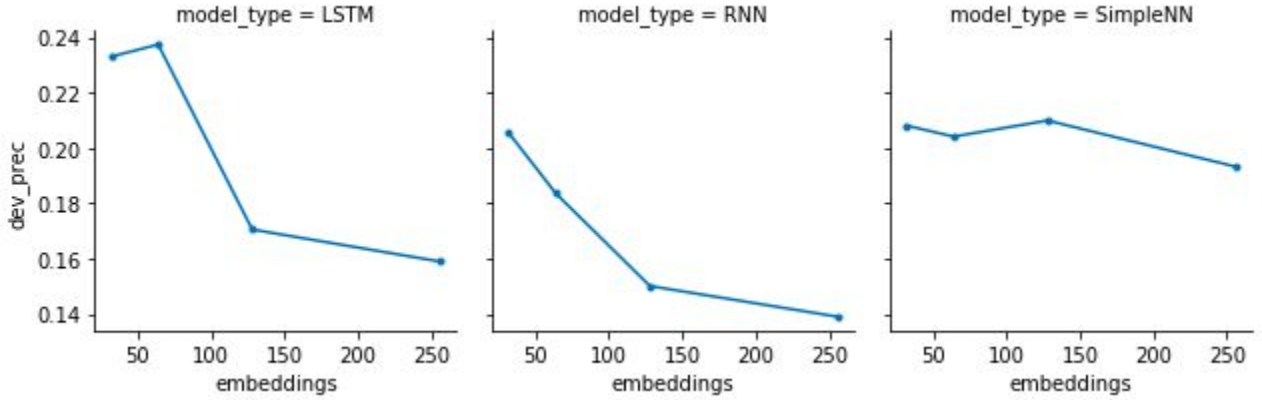
- We have chosen a variety of evaluation metrics to assess our model performance
  - Accuracy, precision, recall
  - But prioritizing precision, since this is a resource allocation problem
- Model performance differs as we vary model architecture and hyperparameters
- Tendency for models to either predict all 0s (not overturned) or all 1s (partially overturned/overturned)
- Difficulty in getting models to converge

Performance of Best Model

- We chose the simple NN model with the highest precision on the validation set
  - Parameters: Embedding dimensions 32, hidden dimensions 25, dropout rate 0.75, 2 layers
  - 100% precision on validation set
- We assessed its performance on our test set
  - Loss: 0.166
  - Accuracy: 79.49%
  - Precision: 18.18%
  - Recall: 11.76%

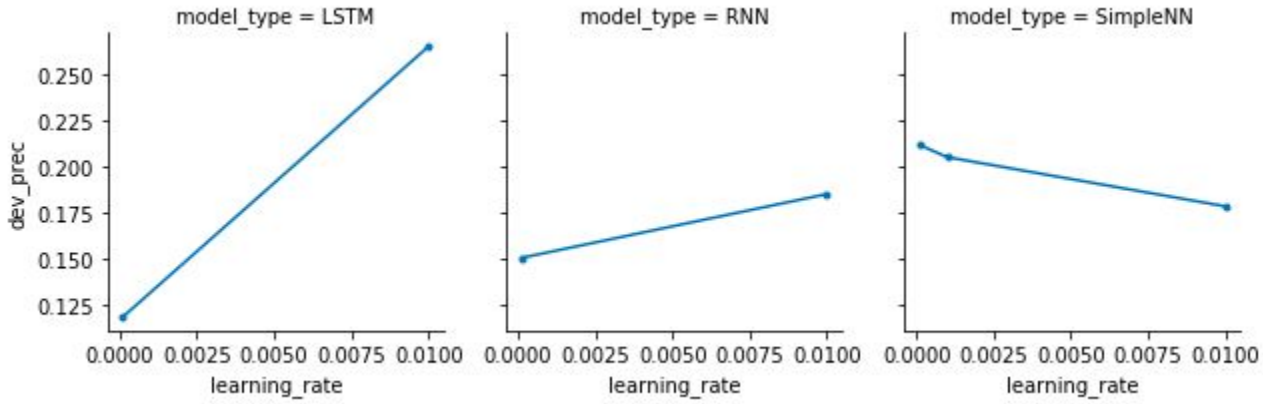
Varying Parameters

- Embedding dimensions:
  - Overall, as the number of embedding dimensions increases, accuracy increases but precision decreases
  - But these effects vary by model type
  - For simple neural networks, after a certain point, an increase in embedding dimensions actually causes accuracy to decrease
  - Overall precision decrease is much more pronounced for RNNs and LSTMs



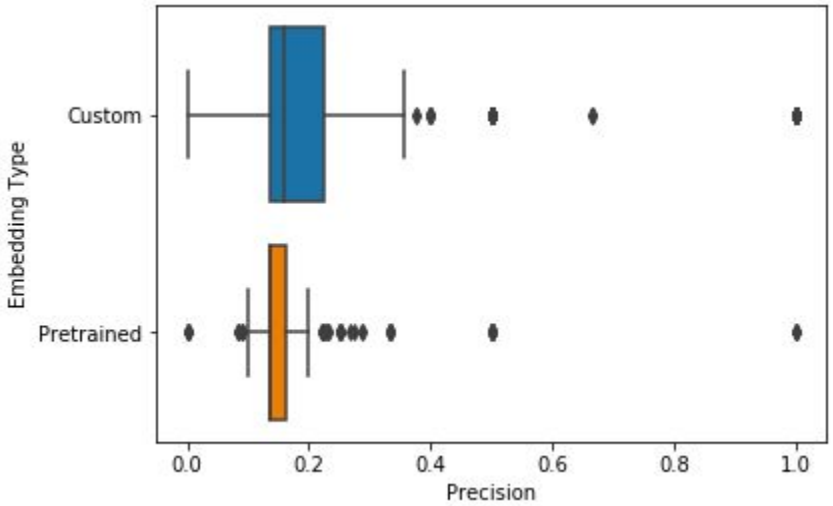
Varying Hyperparameters

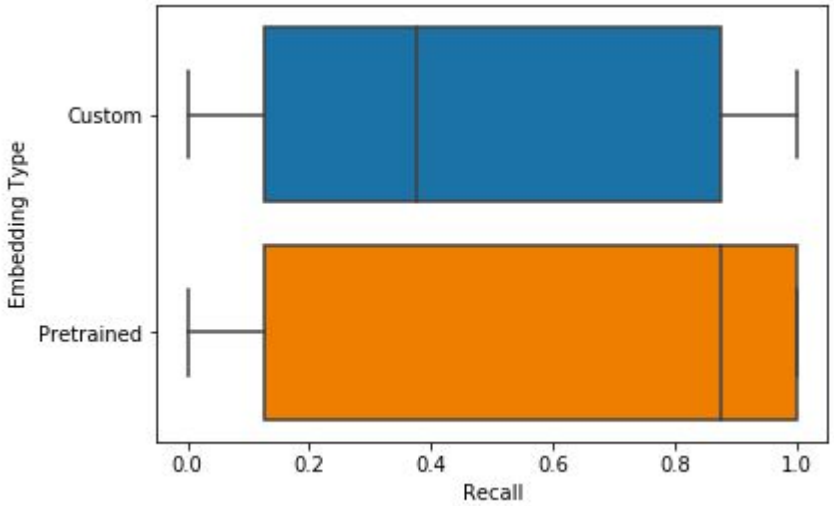
- Learning rate
  - As learning rate increases, precision increases for RNNs
  - Precision increases even more markedly for LSTMs
  - Precision decreases for simple neural networks



Custom vs Pre-trained Embeddings

- Custom embeddings produce higher accuracy and precision, but lower recall





Embedding norms

- For the models in which we trained embeddings, we looked at which words had the embeddings with the highest norm
- However, these words had little qualitative purchase, and included things like "cr02038,", "outreach," and “deferred”
- This lack of qualitative purchase is likely another result of having limited data and being unable to capture nuances in legal decisions
- It may also explain the model’s poor performance on the test set, if the embeddings being trained are very noisy

Caveats

- Small dataset
  - Inevitable given the specific problem we are studying
  - The number of ALJ decisions made in a year is not very large, and the number which are subsequently appealed is even smaller
- Models do not have optimal performance in terms of accuracy
  - However, accuracy is not the primary metric of interest
- 5 models (out of roughly 600 architectures) were not run, due to GPU issues

Future Work

- Ideally, would be able to apply this approach to other court documents
  - Dataset size limits models' ability to pick up nuances in legal documents
  - Applying this method to other legal domains is likely to face one of three issues: (i) inaccessibility, (ii) complex linkage strategies, (iii) small dataset size
  - National Labor Relations Board decisions seems like the most promising starting point
- Creating or leveraging pre-trained word embeddings with suitable legal documents
  - Law2Vec, the set of pre-trained legal word embeddings used in this project, is based on a variety of international court decisions, international laws, US Supreme Court decisions, and US laws
  - LeGloVe, another major set of pre-trained legal word embeddings, is based on Supreme Court documents
  - May not be applicable to other legal contexts
- Adding more traditional features
  - Demographic info regarding judge and defendants
  - Case-specific info, e.g. type, length of time between original case and appeal
- Making use of pre-trained language models, e.g. ELMo and BERT

Reflections on Project

- Existing skills we were able to leverage:
  - Web scraping
  - Databases
  - Text processing with spaCy
  - Machine learning pipelines
- New skills we developed and topics we learned about:
  - Modeling using PyTorch
  - Word embeddings
  - Prior work in applying neural networks to legal decisions

References (Partial)

Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. Neural legal judgment prediction in english. In ACL, 2019.

Mark S. Krass. Learning the rulebook: Challenges facing NLP in legal contexts. 2019.

Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora contain human-like biases. Science, 356(6334):183–186, 2017.

Austin C Kozlowski, Matt Taddy, and James A Evans. The geometry of culture: Analyzing the meanings of class through word embeddings. American Sociological Review, 84(5):905–949, 2019.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From word embeddings to document distances. In International conference on machine learning, pages 957–966, 2015.

Thank you!