Title

Machine Learning Applications in Healthcare: Parsing Pathology Reports Across Multiple Cancers at UCSF and County-level Death Projections of COVID-19

Author

Altieri, Nicholas Lewis

Publication Date

2020

Peer reviewed|Thesis/dissertation

Machine Learning Applications in Healthcare: Parsing Pathology Reports Across Multiple Cancers at UCSF and County-level Death Projections of COVID-19

by

Nicholas L Altieri

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Statistics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Bin Yu, Co-chair
Associate Teaching Professor John DeNero, Co-chair
Assistant Professor Sam Pimentel

Fall 2020

Machine Learning Applications in Healthcare: Parsing Pathology Reports Across Multiple Cancers at UCSF and County-level Death Projections of COVID-19

Abstract

Machine Learning Applications in Healthcare: Parsing Pathology Reports Across Multiple Cancers at UCSF and County-level Death Projections of COVID-19

by

Nicholas L Altieri

Doctor of Philosophy in Statistics

University of California, Berkeley

Professor Bin Yu, Co-chair

Associate Teaching Professor John DeNero, Co-chair

This thesis will focus on two major applications of machine learning in healthcare and will be divided into two sections.

In the first, we discuss our work extracting information from pathology reports across cancers at UCSF. Personalized healthcare is at the frontier of machine learning and medicine and has the potential to revolutionize patient care. A major resource for personalized healthcare is the large amounts of medical textual data and our ability to leverage such data depends on how accurately we can extract information. As a result, there has been much interest in automated text analytics and information extraction methods to tackle healthcare text data which have been used in health informatics, precision medicine, and clinical research. Implementing such extraction systems in practice is a challenge, since many automated extraction systems rely on large amounts of annotated textual data to perform adequately well. However, annotating healthcare text is a largely manual effort, a time-consuming and expensive process that requires training and medical knowledge. It is thus difficult to obtain sufficient amounts of annotated data across a variety of clinical domains. Consequently, while deep learning has been shown to be extremely powerful in natural language processing, it can occasionally underperform in biomedical applications due to the lower number of labeled examples. Therefore, it is of considerable practical importance to develop methods in biomedical natural language processing that perform well in the absence of large amounts of labeled data. In this work, we build natural language processing systems for extracting information from pathology reports across cancers at UCSF, investigating practical problems in the deployment of such systems as well as developing methods that require fewer data points, which leads to systems that perform as well as the state of the art while only requiring 40% of the labeled data. Beyond natural language processing in healthcare, we also develop

machine learning methods for COVID-19 county level death predictions.

For the second section, we discuss models for forecasting short-term death predictions from COVID-19. As the COVID-19 outbreak evolves, accurate forecasting continues to play an extremely important role in informing policy decisions. In this paper, we present our continuous curation of a large data repository containing COVID-19 information from a range of sources. We use this data to develop predictions and corresponding prediction intervals for the short-term trajectory of COVID-19 cumulative death counts at the county-level in the United States up to two weeks ahead. Using data from January 22 to June 20, 2020, we develop and combine multiple forecasts using ensembling techniques, resulting in an ensemble we refer to as Combined Linear and Exponential Predictors (CLEP). Our individual predictors include county-specific exponential and linear predictors, a shared exponential predictor that pools data together across counties, an expanded shared exponential predictor that uses data from neighboring counties, and a demographics-based shared exponential predictor. We use prediction errors from the past five days to assess the uncertainty of our death predictions, resulting in generally-applicable prediction intervals, Maximum (absolute) Error Prediction Intervals (MEPI). MEPI achieves a coverage rate of more than 94% when averaged across counties for predicting cumulative recorded death counts two weeks in the future. Our forecasts are currently being used by the non-profit organization, Response4Life, to determine the medical supply need for individual hospitals and have directly contributed to the distribution of medical supplies across the country. We hope that our forecasts and data repository at `https://covidseverity.com` can help guide necessary county-specific decision-making and help counties prepare for their continued fight against COVID-19.

To my family

Contents

Acknowledgments

My PhD would not have been possible without the help and guidance of many colleagues and mentors and I am extremely grateful to have been able to interact with so many kind and brilliant people.

Principally, I'd like to thank my advisors John DeNero and Bin Yu. I met John as a first year, back when I was transitioning from doing work in machine learning theory to doing natural language processing. I will be forever grateful for him for taking me under his wing and believing in me, teaching me a great deal about natural language processing and research, being open to my diverse and often wandering research interests, his support through the often rocky journey through the PhD, and his unwavering kindness and empathy. Bin Yu as well has had a tremendous influence on my life and my research philosophy. Her perspective on research, as well as life, has profoundly influenced and changed my life direction. I consider myself very lucky to have been able to learn from her wisdom, leadership, and insight. Furthermore, I am incredibly thankful for her constant support and exposing me to so many amazing opportunities and collaborations.

I am grateful too for the mentorship of Anobel Odisho through our long collaboration at UCSF. Not only was he the one who suggested the project that would result in most of my thesis, he has been instrumental in providing medical knowledge and expertise. Beyond that he has been a constant source of inspiration about the exciting opportunities in precision medicine and cancer research broadly. I consider him, along with my advisor Bin Yu, to be the major reasons I continue to work in the intersection of machine learning and medicine with a focus on applications in cancer.

I have had many great collaborators over my years at Berkeley, but I'd like to thank Briton Park especially, who has been involved equally in the UCSF collaboration which comprises most of this thesis. Throughout this long journey of this project, he has been truly wonderful to work with and consistently impressive. I am also thankful to members of the Yu group, former and current, many of whom I had the privilege to work with on the COVID-19 chapter of this thesis, particularly, Xiao Li, Chadan Singh, Raaz Dwivedi, Hue Wang, Tiffany Tang, James Duncan, Robert Netzorg, Rebecca Barter, Wooseok Ha, Merle Behr, Yan Shuo Tan, Jamie Murdoch, and Karl Kumbier. Additionally, I'd like to thank the Berkeley NLP group, whose weekly lunch discussions and random chats in Sutardja Dai Hall are among my favorite memories in the PhD, in particular Dan Klein, Daniel Freid, Sam Ibraheem, Mitchell Stern, Jacob Andreas, Nikita Kitaev, Katie Stasaski, and David Gaddy. Finally, I'd like to thank Ashia Wilson, Nick Boyd, Robert Nishihara, and Philipp Moritz, whose friendship and guidance during the beginning of my PhD I feel very fortunate to have had.

I would also like to thank members from the Information Commons at UCSF, in particular Eugenia Rutenberg, Lakshmi Radhakrishnan, and Dima Lituiev, not only for a grant that funded a large portion of my PhD, but also as excellent collaborators who allowed to take our academic research and work with us to deploy it at UCSF.

During my PhD I am grateful to have had some wonderful experiences in industry. At Lilt, along with my mentorship from John DeNero, I appreciated stimulating machine translation discussions with Spence Green, Joern Wuebker, and Patrick Simianer. At Amazon, I had a great time being mentored by Silja Hildebrand and Alon Lavie, where I had the chance to work on interesting quality estimation problems in machine translation. At Robin Healthcare, I'm particularly fortunate to have worked with Noah Auerhahn and Emilio Galan who taught me so much about start ups and how to be effective in industry and Ben Golan for the great collaborations. Finally, I'm extra thankful for my time as an intern at Genentech, where I had the chance to be mentored by Brandon Arnieri who expanded my horizons and allowed me to collaborate with many others and work on a wonderfully diverse and interesting range of projects. Furthermore, I'd like to thank my additional mentors and collaborators Ryan Copping, Gunther Jansen, Nayan Chaudary, Diego Saldana, Paul Pakzuki, and Svetlana Lyalina.

I have been fortunate to have been financially supported by an Adobe Research grant and a UCSF research grant from the Bakar Institute of Computational Health Science.

I'd especially like to thank La Shana in the statistics department, who has helped me immensely and miraculously never lost her patience among my seemingly never ending administrative mistakes and requests.

Most importantly, I'd like to thank my family, and especially my parents, for supporting me throughout my PhD as well as the entirety of my academic career. No matter how challenging it was, they were always there and encouraged me. I feel very lucky to have their unconditional love and support. Finally, I'd like to thank my girlfriend Jess Li, for being a constant source of joy and happiness in my life, being extremely understanding and supportive when my PhD work was at times all-consuming, and for more fond memories than I can hope to count.

Chapter 1

Overview

1.1 Extracting Information Across Pathology Reports Across UCSF

By enabling patients to receive tailored risk assessment and treatment decisions, precision medicine has the potential to improve healthcare quality [1] [12] However, effective delivery of precision medicine depends on accurate and detailed patient data. Unfortunately, much of the relevant clinical data, such as cancer stage and histology, are stored as free text in lengthy unstructured or semi-structured reports. [12] Leveraging the data contained in these reports for precision medicine applications relies on manual efforts by annotators with domain expertise for many downstream automated methods. Due to the time-consuming and expensive nature of manual information extraction, researchers and clinicians have worked to develop algorithms to automatically extract pertinent data from pathology reports with mixed success, with machine learning-based methods underlying some of the more effective solutions. [12] However, generating sufficient training data for different cancer types is challenging, due to the large number of data elements and their specificity, as well as the need for highly trained annotators. This is a substantial obstacle for automatically structuring biomedical text across clinical conditions and healthcare facilities. Thus, it is critical to develop methods that can provide high accuracy using small training sets. In particular, biomedical information extraction has the potential to make a major impact on cancer treatment and within cancer treatment, pathology reports contain vital clinical information.

An estimated 1.8 million Americans will be diagnosed with cancer in 2020. [85] In nearly all cases, diagnosis is made via tissue analysis, described in detail in a pathology report, which is stored in most electronic medical record systems as unstructured free text. Without manual data abstraction, these important details are unavailable for scalable and algorithmic approaches for case identification, risk stratification, prognostication, treatment selection, clinical trial screening, and surveillance. [79, 99] Moreover, access to these data in structured formats can drive algorithmic personalized treatment strategies based on pathologic information. For nearly 50 years investigators have worked to develop natural language pro-

cessing (NLP) algorithms to extract these details from pathology reports. [55, 12] However, only a limited number of categorical data elements are typically extracted, model outputs often lack reliable uncertainty estimates and furthermore these methods typically require a large labeled dataset, limiting the clinical applicability of these systems, only 10% of which have been reported to be in real-world use. [12]

In this work, we address these major challenges while building a pathology information extraction system which we deploy across multiple cancers and over 2 million notes at the University of California San Francisco.

In the first section of this work, we investigate sample efficiency of state of the art pathology information extraction methods and the accuracy of their uncertainty estimates. In the second section, we create a novel annotation method by annotating where relevant information is located and create a corresponding method to take advantage of these annotations, which leads to a reduction in annotation time of 40% relative to state of the art methods. Finally, we improve sample efficiency even further creating novel transfer learning methods across cancers and using string similarity based methods to allow for accurate information extraction when there are a very large number of labels relative to the number of labeled data points.

Beyond natural language processing in healthcare, we also develop machine learning methods for COVID-19 county level death predictions.

1.2 Curating a COVID-19 data repository and forecasting county-level death counts in the United States

In recent months, the COVID-19 pandemic has dramatically changed the shape of our global society and economy to an extent modern civilization has never experienced. Unfortunately, the vast majority of countries, the United States included, were thoroughly unprepared for the situation we now find ourselves in. There are currently many new efforts aimed at understanding and managing this evolving global pandemic. This chapter, together with the data we have collated (and are collating continuously), represents one such effort. In this chapter, we provide access to a large data repository combining data from a range of different sources and to forecast short-term (up to two weeks) COVID-19 mortality at the county level in the United States. We also provide uncertainty assessments of our forecasts in the form of prediction intervals based on conformal inference. [93]

Chapter 2

Pathology Parsing in Limited Data Environments with Uncertainty Estimation

Extraction information from pathology reports has traditionally been approached using rule-based methods [64, 66, 36, 67]. However, designing rules is labor intensive and requires deep involvement of clinical experts. The complexity and conflicts between rules grow rapidly as the number of rules increases, and as the underlying documents shift, rules quickly become ineffective. [29] NLP has been applied to pathology report information extraction with promising results, using both classic NLP (boosting over a bag-of-ngrams representation of the document) and deep learning approaches (convolutional, recurrent, and hierarchical attention networks).[98, 34] While most work focuses on classification tasks involving fields with a small number of labels (such as histology or margin status), Li and Martinez (2010) investigate categorical fields as well as numeric fields such as the tumor size and the number of lymph nodes examined. [58] Furthermore, many other information extraction tasks and methods have been applied to pathology reports, such as Coden et al [21] which creates a knowledge representation model to represent cancer disease characteristics; Si et al [84] which uses a frame-based representation to extract information from clinical narratives focusing on cancer diagnosis, cancer therapeutic procedure, and tumor description; Xu et al [97] which considers attribute detection as a sequence labeling problem; and Oliwa et al [69] uses NLP to classify gastrointestinal pathology reports into internal and external reports and uses Named Entity Recognition to label accession number, location, date, and sub-labels.

Despite these developments, there has been comparatively little effort in understanding two additional important criteria that are the basis for reproducibility and real-world use. The first is evaluating performance as a function of training data size, which informs practitioners about how much data they may need to deploy similar systems. Creating an annotated

corpus is costly and time consuming, and accurate assessment of necessary sample size can aid deployment. [25, 77, 68, 86, 33] Second, accurate uncertainty estimates for the predicted results are critical for clinical deployment, as different uses have varying acceptability thresholds. Having accurate uncertainty estimates means that for all cases where the model score outputs a probability p, it is correct p percent of the time. An example of a model with inaccurate uncertainty estimates would be one that gives a predicted probability of correctness of 90% on all examples, but is actually only correct 10% of the time. Accurate uncertainty estimates are important for deployment, as lower certainty may be acceptable if the results are used for initial screening with manual verification to follow, but higher certainty is required for a clinical decision support system. Resources can be directed to verification for cases of high uncertainty, supplanting the need for full manual abstraction. The source code for this project will be made available under an open source license to facilitate adoption of NLP tools in cancer pathology.

OBJECTIVE

Our objective was to investigate two practical issues that arise when deploying machine learning-based information extraction systems to pathology reports, using prostate cancer as a test case. First, we evaluate the performance of models as a function of dataset size for tasks that involve categorical values, such as histologic grade or presence of lymphovascular invasion, as well as numeric values, such tumor size Second, we describe an approach to model calibration and calculation of uncertainty estimates for each prediction and assessing the quality of the model's uncertainty estimates. We address these gaps in the literature to guide practitioners as they implement these systems in real-world settings.

MATERIALS AND METHODS

Data Sources

We used a corpus of 3,232 free text pathology reports for patients that had undergone radical prostatectomy for prostate cancer at the University of California, San Francisco (UCSF) from 2001 - 2018, which were extracted from UCSFâs electronic health record (Epic Systems, Verona, WI). For each document, annotations for 17 pathologic features, such as Gleason scores, margin status, extracapsular extension, and seminal vesicle invasion were extracted (Table 2.1) in the Urologic Outcomes Database, which is a prospective database that contains clinical and demographic information about patients treated for urologic cancer. Since 2001, data have been manually abstracted by trained abstractors under an institutional review board (IRB) approved protocol. This study was separately approved by the IRB.

The full corpus was divided into four parts, 64% training, 20% validation, 10% test, and 10% true test. We looked at the true test only while compiling results. In order to handle our diverse set of fields, we used two separate information extraction methods. For categorical fields, we used a document-based classification method which has been previously applied to information extraction from pathology reports. [98, 34] For fields with a large number of possible values (such as numeric quantities), we used a sequence labeling task to extract individual tokens from the document. [50] We applied our methods to the full training dataset as well as randomly selected subsets of 16, 32, 64, 128, and 256 reports. All models are implemented in using scikit-learn and pytorch. Detailed explanation of the pre-processing pipeline and dataset statistics are presented in the supplementary material.

Document Classifier Methods

For categorical data fields, such as the presence of lymphovascular invasion, we treat it as a document classification problem. These fields have between two to six possible classes (Table 2.1). We apply classical methods, such as logistic regression, random forests, support-vector machines (SVM), and adaptive boosting (AdaBoost) on bag-of-n-gram features, as well as deep learning methods, such as convolutional neural networks (CNNs), and long short-term memory networks (LSTMs).

Token Extractor Methods

Many critical clinical data elements, such as tumor volume, are not suited for classification because they are not categorical in nature. In order to broaden the variety of data fields extracted from the reports, we employ an additional approach which we refer to as *token extractor methods*. These methods are well-suited to extract numerical quantities from a document (such as the estimated tumor volume or the patientâs medical record number, Table 2.1). For these fields, we take each tokenâs surrounding context of k words represented as a bag of n-grams as the primary features. We additionally append the token type encoded as a vector to the bag of n-grams context vector. The token type vector specifies whether a particular token is an ordinary word, a numeric value, or a hybrid of the two. These features are used to predict whether or not the token is the token we aim to extract using logistic regression, AdaBoost, or random forest methods. Unlike the document classifier methods, we excluded SVMs and deep learning methods for the token classifier due to the impractical computational requirements for our compute resources. Because our labeled data did not contain the location information of the token of interest within the document, we labeled all tokens that matched our label as a positive example at the time of training. At test time for each token we compute the score under the model that this token should be extracted and

then choose the token with the largest score as our final prediction. This token extraction method is applied to the following fields: pathologic T, N, and M stage, prostate weight, and tumor volume. For additional details regarding the pathologic stage field, we refer the reader to the supplementary material. We would like to give a comparison with a related but slightly different information extraction task of Named Entity Recognition (NER), which classifies named entities in text into categories. Like token extraction, this too is a sequence labeling task. In NER, this involves labeling each token into a predefined category and in our case, for a given field, we label each token with a 0 or 1 as to whether or not it is the desired token for this field and document. As a clarifying example for the distinction between the tasks, an NER system with procedure as a predefined category would label all mentions of procedures in a pathology report as the procedures class. However, this is not what we want, as pathologists will often discuss multiple procedures in a report, but we are interested in only the specific procedure that resected the tumor.

Dataset Size and Performance

We investigate the performance over varying data-regimes, since for informaticists who wish to build a machine learning parser on their data, a critical question is the quantity of data points needed for adequate performance and which methods are most likely to perform well. We fixed the training set size to 16, 32, 64, 128, and 256 reports, which were randomly drawn from the full training set and averaged the results over 5 random draws.

Evaluation Metrics

For each field we report the weighted F1 score of the classifier, which is the weighted sum of the F1 scores for each class in the field, where the term for each class is weighted by the portion of true instances of the class. In the Supplementary Tables 1-4, we report the micro F1 and macro F1 to better compare to existing literature. For token extractor models, we compute the accuracy of whether the token extracted from the report was correct.

Hyperparameter Tuning

To tune hyperparameters for the classification models on the full data, we used random search with a validation set to tune each method. For each model, we randomly select 20 model-specific hyperparameter configurations, train the model on the training set, and obtain weighted F1 scores on the validation set. The model with the hyperparameter configuration

with the highest score is used to obtain results on the test set. To tune hyperparameters for the classification models in the low data regimes (training on ≤ 256 reports), we used random search across 20 configurations of hyperparameters but with 4-fold cross validation to calculate weighted F1 scores. For extractor models, we used random search with 20 hyperparameter configurations and 4-fold cross-validation for both the full data and low data regimes.

We chose 4-fold cross validation as it provided a good balance between performance and computational cost in preliminary experiments. For more details regarding hyperparameter ranges for different models, we refer the reader to the supplementary materials.

Calibration of Systems

To support multiple use cases for the outputs of our model, it is desirable to estimate the modelâs uncertainty reflecting the true probability of correctness for each predicted value. For example, values that have a low probability of being correct can be flagged for manual verification, or results can be limited to only those with a high probability of being correct. More rigorously, for a model f and data distribution X ideally we would like a function P^* such that

$$P_x\left(f\left(x\right) = y | P^*\left(x\right) = p\right) = p \text{ for all } p \epsilon \left[0, 1\right]$$

One common definition of the discrepancy between the modelâs predicted probability of correctness and its true probability of correctness is given by the expected calibration error which is the expected difference between the models confidence and its true probability of being correct. [100]

$$E_x\left[\left|P\left(f\left(x\right) = Y | \widehat{P}\left(x\right) = p\right) - p\right|\right]$$

where $f\left(x\right)$ is the modelâs prediction for a datapoint point x, Y is the true value, and $\hat{P}$ (x) is the modelâs predicted probability of being correct for point x. However, this is typically not able to be measured in practice, for example if $\widehat{P}\left(x\right)$ takes on a continuous set of values, so instead $\widehat{P}\left(x\right)$ is discretized into bins and the Expected Calibration Error (ECE) [24] is defined as follows:

$$ECE = \sum_{m=1}^{M} |B_m|/n|acc\,(B_m) - conf\,(B_m)\,|$$

Where B_m is the mth bin, $acc\,(B_m)$ is the average accuracy of the model in bin m, and $conf\,(B_m)$ is the average value of $\hat{P}$ (x) of the model in bin m.

To improve the calibration of our system we apply isotonic regression. [100] In the binary case it takes the confidence of the models output of the positive class and fits a monotonic function where the x-axis represents the modelâs confidence score and the y-axis represents whether or not the model was correct. In the multivariate case, the calibration method attempts to calibrate the probability estimate of each class. It does this by first calibrating the probability of each class in a one-vs-all setting, then after fitting, estimating the probabilities by normalizing the one-vs-all probability for each class.

Error Analysis

To understand the potential failure modes of our models, for each field we manually analyzed 10 errors randomly chosen in our test set split of the best models in Table 2.2 by comparing the model output and annotated label with the text of the report to check the source of the error. If there were fewer than *10* errors for a field, we analyzed all the modelâs errors

If the error was a result of an incorrect label in our original data set, it was named as an annotation error. Model errors occurred when the model extracted the incorrect value for a certain field. Next, an error was classified as a report anomaly if there was something wrong with the raw text of the report, such as if the sentences of a report were repeated many times in the text or there was internal inconsistency in the report. Lastly, the evaluation error means that the extracted value was correct but the evaluation method incorrectly penalized the model such as if the correct extracted token was 2 for volume of tumor and the model extracted 2cm.

RESULTS

Document Classifier Performance

We calculated the weighted F1-score for each data field using the true test set (Table 2.2). When working with the full training corpus (n= 2,066), convolutional networks perform the best (mean weighted F1 0.972 across all 12 clinical data elements). However, we see that the best non-deep-learning method is not far behind with AdaBoost having a weighted F1 score of 0.965.

Token Extractor Performance

For token extraction we measure the accuracy of extracting the correct token from each document (Table 2.2). In greater detail, we choose the most probable token over all tokens in the document and compare this to the ground truth. We observe that random forests perform the best out of all the methods with a mean accuracy of 0.883 across 5 fields.

Performance as a Function of Dataset Size

For the classification fields, the classical machine learning methods (logistic regression, SVM, AdaBoost, and random forests) clearly outperform the deep learning methods on average, likely due to the small amount of training data available. The results also show that 128 reports are needed for the best methods to achieve a 0.90 weighted F1 on average across all classification fields. For the token extractor fields, the results seem to plateau at 64 reports. Our experiments show that a training set size in the thousands is not always needed to adequately extract structured data from these pathology reports, an important observation for practitioners weighing the cost of creating an annotated dataset.

Effect of Calibration

We apply calibration to two of our models. For the classification model, we apply isotonic calibration to boosting and for the extractor model we apply isotonic regression to the random forest model. [26] For the extractor case, we treat the probability of the token with the highest probability as the confidence score of the model. We fit our isotonic regression calibration methods on the development test set and evaluate the expected calibration error on the test set, binning our uncertainty estimates $\hat{P}$ (x) into bins of width .1. (Table 4.2). Additional experiments investigating the expected calibration error (ECE) as a function of

the bin size, which we include in supplementary Figures 1 and 2, show that while the average expected calibration error increased, the difference in the average expected calibration error between the smallest bin size (4) and the largest (64) was less than .02 for both classification and extraction tasks. [24]

We find that for most classifications fields, the model had expected calibration scores less than .1 and that isotonic regression generally improves upon this. Since for each class the one-vs-all probabilities are calibrated, the calibrated modelâs predictions may differ from the original model if it is not a binary classification problem, so in addition to the expected calibration error of the model, we list the weighted F1 score of the calibrated model. Conversely, extractor models are not well calibrated out of the box in general, but surprisingly, by only using the probability of the token with greatest probability, performing isotonic regression on this single value is enough to achieve sub .05 expected calibration errors.

We also examined when the model was most overconfident, where we look for examples with high estimated probabilities of being correct, but which were nevertheless wrong. We found the most overconfident example in each field and observed that in 10 of the 15 examples the algorithm was correct and the label was actually incorrect.

Error Analysis

The most common type of evaluation error for the token extractor occurred when the model extracted the right token, but the evaluation method did not correctly score the model (Table 6). For example, if the label for the estimated volume of tumor was 2 (in centimeters) and the model extracted 2cm, the model would be penalized. The most common type of report anomaly occurred when the text in the report was repeated. For example, in one case, each sentence in the report was repeated 3 times. This was an issue in the raw text of the report and was not an aberration in preprocessing. Overall, error analysis shows that the scores given for the models are likely underestimates and the models actually perform better than the raw results show.

For a comprehensive breakdown of errors, we refer the reader to Table 5) in the supplementary material. Because the pathologic stage errors are highly correlated (due to the fact that the different types of stages are encoded in the same token in the text), only the results for the pathologic T-stage are shown.

DISCUSSION

We have investigated several practical issues in the clinical deployment of a machine learning based pathology parsing system and developed a system that can accurately parse prostate reports across a variety of fields and provide reliable per-label uncertainty estimates. Furthermore, we evaluated the number of samples required for adequate performance to guide outside practitioners who are considering using a learning based parsers for their datasets.

The dual classification/extraction approach to our pipeline was developed to accommodate a larger variety of data fields. Yala et al (2017) applied boosting across twenty binary fields on 17,000 labeled breast cancer reports and observed strong performance with F1 scores above .9 for many fields.[6] Gao et al (2018) applied hierarchical attention networks to predict tumor site and grade from pathology reports within the NCI-SEER dataset and noted improvement in micro-F1 (up to 0.2 greater) compared to baselines across two fields (primary site and histologic grade) for a dataset of lung and breast cancer pathology reports. [12] Much of the previous work does not attempt to extract all relevant data fields since they rely primarily on document classification methods which cannot handle continuous values, such as tumor size or prostate weight or perform the related but slightly different task of NER. Although Li and Martinez (2010) attempt to extract data fields based on numeric values using a hierarchical prediction method, the final prediction step relies on a rule based method that has no clear way to be calibrated. [64] Furthermore, while our two methods are not run on the same fields, our algorithm appears to have higher performance in general. Our solution is developing a sequence tagging algorithm that extracts tokens corresponding to the desired values directly, as well as employing classifier methods to extract categorical data fields. Each method is also capable of outputting a score that can be directly calibrated using isotonic regression. One limitation of our extraction methods is that we only consider simple bag-of-n grams based representations and it would be interesting to see how sample efficiency or calibration errors change under a deep learning approach.

Second, we investigated the necessary number of reports needed for accurate classification for our pathology reports by varying the size of the training set of reports from 16 to 256 across both classification and extraction. While others have performed sample efficiency analysis of NLP algorithms across many tasks [4, 44, 83] to our knowledge, this has not been investigated for the important application of clinical information extraction from pathology reports, with the exception of Yala et al. who plot dataset size vs performance over only one method (boosting) and over fields that only take two values. [13] Overall, we found that only 128 labeled reports were needed for the best methods for classification and only 64 for the token extractor, a small number compared to the dataset sizes used in prior work. It is important for practitioners who have a smaller dataset to understand approximately

how much performance to expect from a machine learning based approach as it can be expensive and time-consuming to create a large corpus of annotated documents. We hope this encourages more groups to explore these approaches, as large datasets may not always be required. Our study is limited by focusing on a single cancer from one institution, and further work can assess generalizability to other cancers and sites. Of note, there was heterogeneity in report structure and style over 20 years.

Another important observation is that the classical statistical learning methods outperformed deep learning methods by a large margin when fewer than 256 data points were available, while deep learning only slightly outperformed logistic regression when using all 2,066 reports in the training set. This suggests deep learning only adds marginal value and the complexity of the problem, at least for the reports we worked with, is more suited to classical methods.

Finally, we investigated the reliability of uncertainty estimates of the model, which to the authorsâ knowledge, has not been investigated in other pathology information extraction work. Knowing which reports are likely to be incorrect can decrease the time needed to manually verify extracted data and filter uncertain predictions for tasks like clinical research with small populations, where each predicted value may have a large impact on conclusionsThrough our calibration work, we observed that the classification model was typically well calibrated without any modification, whereas our token extraction algorithm was not. However, by just using the probability of the selected token, isotonic regression was a very effective calibration solution. We furthermore investigated when the model is most likely to be overconfident and found that two-thirds of these errors were due to incorrect annotation labels, not incorrect algorithm outputs.

CONCLUSION

Creating annotated datasets and reliable systems are serious practical concerns when developing and deploying biomedical information extraction systems due to the high cost of creating annotations and the impact of errors on patients outcomes. We show when applying machine learning to pathology parsing, accurate results can be obtained using relatively small annotated datasets and calibration methods can improve the reliability of per-label uncertainty estimates.

TABLES AND FIGURES

Table 2.1. Data elements extracted from pathology reports

Document Classifier Algorithm Fields	
Gleason Grade Primary, secondary, tertiary	Histologic grading of tumor aggressiveness based on the Gleason grading system. Each specimen is assigned a primary, secondary, and occasionally a tertiary score, each of which are whole numbers from 1-5
Tumor histologic type	Primary histologic type, such as acinar adenocarcinoma, ductal adenocarcinoma, small cell neuroendocrine carcinoma
Cribriform pattern	Whether the cells exhibit a cribriform growth pattern (Gleason 4 only)
Treatment effect	Indicator whether there is evidence of a prior treatment, such as hormone treatment or radiation therapy
Margin status for tumor	To evaluate surgical margins, the entire prostate surface is inked after removal. The surgical margins are designated as "negative" if the tumor is not present at the inked margin and "positive" if tumor is present at the inked margin.
Margin status for benign glands	The benign margins are designated as "positive" if there are benign prostate glands present at the inked margin and "negative" otherwise
Perineural Invasion	Whether cancer cells were seen surrounding or tracking along a nerve fiber within the prostate
Seminal vesicle invasion	Invasion of tumor into the seminal vesicle
Extraprostatic extension	Presence of tumor beyond the prostatic capsule
Lymph node status	Whether tumor was identified in lymph nodes
Token Extractor Algorithm Fields	
Pathologic Stage Classification T (primary tumor) N (regional lymph nodes) M (distant metastasis)	Based on American Joint Committee on Cancer TNM staging system for prostate cancer. Based on the edition used in each report (5^{th} - 8^{th} edition)

continued on next page

continued from previous page

Tumor volume	Amount of tumor identified in prostate specimen (cubic centimeters)
Prostate weight	Overall weight of the prostate (grams)

Table 2.2. Weighted F1 scores for classification fields and mean accuracy for token extractor fields on full training data sample (n = 2,066)

Data Element	**Logistic regression**	**Adaboost classifier**	**Random Forest**	**SVM**	**CNN**	**LSTM**	**Majority Class Accuracy**
Gleason Grade - Primary	0.978	0.971	0.941	0.932	0.981	0.628	0.709
Gleason Grade - Secondary	0.958	0.943	0.913	0.912	0.968	0.576	0.467
Gleason Grade - Tertiary	0.923	0.930	0.844	0.886	0.930	0.741	0.901
Tumor histology	0.989	0.995	0.995	0.993	0.995	0.994	0.991
Cribriform pattern	0.963	0.981	0.963	0.968	0.987	0.966	0.997
Treatment effect	0.981	0.979	0.981	0.981	0.981	0.973	0.985
Tumor margin status	0.941	0.953	0.888	0.918	0.950	0.630	0.799
Benign margin status	0.977	0.975	0.972	0.981	0.978	0.967	0.997
Perineural invasion	0.944	0.978	0.938	0.929	0.972	0.613	0.771
Seminal vesicle invasion	0.943	0.974	0.940	0.965	0.976	0.784	0.904
Extraprostatic extension	0.954	0.953	0.882	0.939	0.961	0.778	0.712
Lymph node status	0.983	0.952	0.983	0.973	0.986	0.824	0.570
Mean weighted F1	0.961	0.965	0.937	0.948	0.972	0.790	0.817
T Stage	0.951	0.954	0.948	-	-	-	-
N Stage	0.954	0.954	0.948	-	-	-	-
M Stage	0.972	0.969	0.969	-	-	-	-
Estimate Tumor Volume	0.605	0.765	0.873	-	-	-	-
Prostate Weight	0.846	0.855	0.914	-	-	-	-
Mean Accuracy for token extractor models	0.866	0.899	0.930	-	-	-	-

LSTM: Long Short-Term Memory Neural Network
CNN: Convolutional Neural Network
SVM: Support Vector Machine

Table 2.3. Mean weighted F1 score ± standard deviation for classification models for classification models and mean accuracy ± standard deviation for token extractor models on

increasing numbers of reports (n) after 5 trials

Model	**n = 16**	**n = 32**	**n = 64**	**n = 128**	**n = 256**
Classification Models (mean weighted F1 score across all classification fields ± SD)					
Logistic	0.781 ± 0.175	0.846 ± 0.117	0.875 ± 0.090	0.911 ± 0.059	0.934 ± 0.041
AdaBoost	0.829 ± 0.140	0.878 ± 0.100	0.907 ± 0.066	0.928 ± 0.049	0.945 ± 0.034
Random Forest	0.795 ± 0.169	0.835 ± 0.128	0.867 ± 0.101	0.882 ± 0.088	0.901 ± 0.070
SVM	0.738 ± 0.214	0.763 ± 0.209	0.786 ± 0.194	0.842 ± 0.112	0.860 ± 0.140
CNN	0.720 ± 0.225	0.790 ± 0.163	0.851 ± 0.122	0.893 ± 0.086	0.935 ± 0.055
LSTM	0.688 ± 0.205	0.729 ± 0.187	0.743 ± 0.203	0.739 ± 0.214	0.739 ± 0.212
Token Extractor Models (mean accuracy across all token extractor fields ± SD)					
Logistic	0.844 ± 0.085	0.897 ± 0.079	0.892 ± 0.096	0.902 ± 0.087	0.896 ± 0.092
Adaptive Boost	0.877 ± 0.097	0.892 ± 0.080	0.890 ± 0.084	0.896 ± 0.082	0.890 ± 0.092
Random forest	0.897 ± 0.180	0.898 ± 0.064	0.915 ± 0.054	0.920 ± 0.041	0.924 ± 0.038

LSTM: Long Short-Term Memory Neural Network
CNN: Convolutional Neural Network
SVM: Support Vector Machine

Table 2.4: Upper: Classifier accuracy and expected calibration error for boosting before and after isotonic calibration. Lower: Expected Calibration Error for Random Forest Model before and after isotonic calibration.

Classification Calibration				
Data Element	**Weighted-F1**	**ECE**	**Isotonic Weighted-F1**	**Isotonic ECE**
Gleason Grade - Primary	0.95	0.03	0.93	0.03
Gleason Grade - Secondary	0.94	0.08	0.92	0.14
Gleason Grade - Tertiary	0.91	0.05	0.91	0.03
Tumor histology	0.99	0.009	0.99	0.007
Cribriform pattern	0.995	0.007	0.995	0.017
Treatment effect	0.99	0.007	0.99	0.003
Tumor margin status	0.96	0.15	0.94	0.013
Benign margin status	0.994	0.007	0.995	0.019
Perineural invasion	0.95	0.26	0.96	0.02
Seminal vesicle invasion	0.987	0.16	0.97	0.02
Extraprostatic extension	0.96	0.12	0.96	0.01
Lymph node status	0.96	0.04	0.98	0.01

Extractor Calibration		
Data Element	**ECE**	**Isotonic ECE**
T Stage	0.155	0.016
N Stage	0.144	0.013
M Stage	0.007	0.005
Estimated Volume of Tumor	0.221	0.021
Prostate Weight	0.278	0.033

Chapter 3

Enriched Annotations for Tumor Attribute Classification from Pathology Reports

In this previous chapter, we found that performing document classification with a small number of labeled reports can be a major limitation for pathology information extraction systems in practice. In this work, we develop a novel hierarchical annotation and corresponding classification method to address the need for high accuracy methods in the presence of a small amount of annotated data. We apply this method to classifying tumor attributes from 250 colon cancer pathology reports and 250 kidney cancer reports at the University of California, San Francisco. Compared to state of the art approaches, we find that our methods typically require half the labeled data to achieve the same level of performance.

BACKGROUND

The abundance of textual data in the clinical domain has led to increased interest in developing biomedical information extraction systems. These systems aim to automatically extract pre-specified data elements from medical documents, such as physician notes, radiology reports, and pathology reports, and store them in databases. Converting the originally free-text data into a structured form makes them easily available to clinical practitioners or researchers.

For categorical attributes, the information extraction task can be viewed as an instance of document classification that classifies the tumor attribute based on document contents. For a given attribute, the value is one of a fixed set of options selected based on information in the document. As an illustration, the set of values for the attribute "presence of lymphovascular invasion" could consist of the values "present", "absent", and "not reported". Both classical and deep learning classification methods have been applied to this task in the prior work discussed below.

There has been success in applying classical machine learning techniques to classifying attributes of tumors from pathology reports. Yala et al. classified over 20 binary attributes from breast cancer pathology reports using boosting over n-gram features. [98] Jouhet et al investigated applications of Support Vector Machines (SVMs) and Naive Bayes classifiers to the task of predicting International Classification of Diseases for Oncology (ICD-O-3) from cancer pathology reports. [49]. More recently, there has been success in applying deep learning techniques to pathology report classification. Qiu et al. applied convolutional neural networks (CNNs) to predicting ICD-O-3 from breast and lung cancer pathology reports. [76] Gao et al. applied hierarchical attention networks to predict tumor site and grade from pathology reports within the NCI-SEER dataset and noted improvement in micro-f1 of up to 0.2 compared to baselines across primary site and histologic grade for lung cancer and breast cancer reports. [34]

There has also been work addressing pathology report classification in the absence of a large amount of labeled data. Odisho et al. analyzed performance of machine learning methods for extracting clinical information from prostate pathology reports across various data regimes and found that, while deep learning performed best when trained on the full dataset of 2,066 labeled documents and achieved a mean weighted-F1 score of 0.97 across classification attributes, simpler methods such as logistic regression and adaBoost performed best in smaller data regimes ($<$256 reports). [67] Additionally, Zhang et al, investigated the problem of unsupervised adaptation across attributes in breast cancer pathology reports. [7] Given a set of attributes with labels and a new attribute without labels but with relevant keywords, they used adversarial adaptation with semi-supervised attention to extract data. We use all of the above methods as baselines for our system to compare against, with the exception of Zhang et al. due to the difference in tasks.

MATERIALS AND METHODS

Data Sources

Our data consists of 250 colon cancer pathology reports and 250 kidney cancer reports from 2002-2019 at the University of California, San Francisco. The data was split into two sets,

a set of 186, which we used for training and validation, and a test set of size 64. We list the tumor attributes and their corresponding possible values in Table 3.1. Institutional Review Board approval was obtained for this study.

Our data consists of 250 colon cancer pathology reports and 250 kidney cancer reports from 2002-2019 at the University of California, San Francisco. The data was split into two sets, a set of 186, which we used for training and validation, and a test set of size 64. We list the tumor attributes and their corresponding possible values in Table 3.1. Institutional Review Board approval was obtained for this study.

Data Annotation Methods

Pathology reports consist of free text describing a patient's clinical history and attributes describing the excised specimen, such as surgical procedure, cancer stage, tumor histology, grade, cell differentiation, and presence of invasion to surrounding tissues. More recent pathology reports also contain a synoptic comment section, which is a condensed semi-structured summary of relevant cancer attributes. While many of the most clinically important attributes are reported in this synoptic comment, this is not always the case. The presence and completeness of the synoptic comment also varies significantly over time. All attributes in the College of American Pathology reporting guidelines are annotated for each cancer, but for this paper we restrict our investigation to the most frequently occurring attributes in our pathology reports. These include tumor site, histologic type, procedure, laterality, tumor grade, and lymphovascular invasion for both cancers. Additionally, we have the cancer specific attributes of laterality and perineural invasion for kidney and colon cancer specifically.

Enriched Annotations

In previous work, annotations consisted of only the label for each attribute in a document. [98, 67, 34] However, in this work, for each attribute of interest the annotator highlighted all occurrences relevant to the label throughout the document, in addition to the label itself. This provides us with the specific location within the text that directly indicates the attribute's label. Each highlight is classified into the corresponding College of American Pathologists (CAP)-derived category. We investigate two types of annotation: the first we refer to as the "reduced annotation set", a minimal set of annotations containing the line of a given attribute value's first occurrence in the synoptic comment, or, if not in the synoptic comment, the line of where that information is referenced elsewhere in the document. The incremental time required to annotate this location is marginal because the annotator does not need to read any more of the document than that required to annotate the first occurrence

of the attribute value. In fact, "We investigated the amount of additional time required to create these enriched annotations and found that it took 20 percent longer on average, primarily due to the time it took the annotator to navigate the attribute drop-down menu. This could perhaps be improved through user interface (UI) considerations. In addition to a reduced annotation set, we also investigate performance with all the occurrences relevant to the final classification highlighted, a more laborious annotation scheme. For our results, unless stated otherwise, we are using the reduced annotation set due to its comparable annotation time to labeling the attribute values alone.

Data Preprocessing

For all methods, we replace all words that occur fewer than two times in the training data with a special <UNK> token, and remove commas, backslashes, semi-colons, tildes, periods, and the word "null" from each report in the corpus. For colons, forward slashes, parentheses, plus, and equal signs, we added a space before and after the character. The spaces were artificially added to preserve semantic value important to the task. For instance, colons often appear in the synoptic comment, and so if an n-gram contains a colon, it can indicate that the n-gram contains important information. If multiple labels for an attribute occurred within a report, we concatenate them to form a single composed label. For example, if the report contains both grade 1 and grade 2 as labels for histologic grade, we label the histologic grade of the report as "grade 1 and grade 2" .

Baselines

For all classical baselines, we represent each document as a union of a set of n-grams where n varies from 1 to N, where N is a hyperparameter. For all methods we use random search [9] with 40 trials to tune our hyperparameters according to the 4-fold cross validation error which we found in preliminary experiments to be a good compromise between performance and computational efficiency.

Logistic regression

We use sklearn's [71] logistic regression model with L1 regularization and the liblinear solver. We use balanced class weights to up-weight the penalty on rare classes. We generate 500 points from -6 to 6 logspace for the regularization penalty, and sample 40 points at random.

Support Vector Classifier

We use sklearn's SVC model with balanced class weights. We define our parameter space as 500 points evenly generated from -6 to 6 in log space for the error penalty C of the model; the kernel as linear or rbf; and the parameter of the kernel as either 0.001, 0.01, 0.1, or 1. We then sample 40 points at random from this space.

Random Forest

We use sklearn's random forest classification model with balanced class weights. The parameter space consists of the number of estimators from 25, 50, 100, 200, 400, 600, 800, and 1000; the minimum number of samples for a leaf from 1 to 128 in powers of 2; max depth of a tree from 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100; and whether to bootstrap samples to build trees or not. We sample 40 points randomly from this parameter set.

Boosting

We use sklearn's adaboost classifier with decision trees of depth 3 and with the SAMME.R boosting algorithm. Our parameter space is 500 points generated evenly from -4 to 1 in logspace for the learning rate and either 25, 50, 100, 25, or 500 for the number of estimators. We then sample 40 points at random from this space.

Hierarchical Attention Network

We implement the hierarchical attention method without unsupervised pre-training from Gao et al. This model represents the document as a series of word-vectors. For each sentence in the document it runs a gated recurrent unit (GRU) [20] over the word vectors. It then uses an attention module to create a sentence representation as a sum of the attention-weighted outputs of the GRU. To generate the document representation, a GRU is run over the sentence representations, followed by another attention module applied to the GRU outputs. The document representation is the attention-weighted sum of the GRU outputs.

For our hyperparameters we use random search across the learning rate, which is either 1e-2, 1e-3, or 1e-4; the width of the hidden layer of the attention module, which is either 50, 100, 150, 200, 250, or 500; the hidden size of the GRU, which is either 50, 100, 150, 200, 250, or 500; and the dropout rate applied to the document representation, which is either 0, 0.2, 0.4, 0.6, or 0.8. We use a batch size of 64 and ADAM [54] as our optimizer.

Our method: Supervised Line Attention

In order to take advantage of annotations enriched with location data, we propose a two-stage prediction procedure in which we first predict which lines in the document contain relevant information. We then concatenate the predicted relevant lines and use this string to make the final class prediction using logistic regression.

Finding Relevant Lines

The first stage predicts which lines are relevant to the attribute. We do this by training an xgboost binary classification model that takes a line represented as a bag of n-grams as its input and outputs whether or not the line is relevant to the attribute. The relevance of each line is predicted independently by this initial classifier.

We then take the top-k lines with the highest scores under the model (where k is a hyperparameter). Groups of adjacent lines are conjoined into one line so that sentences which span multiple lines are presented to the model as a single line.

Finally, we represent each line as a set of n-grams vectors and compose a document representation as the weighted sum of each vector representation, which is weighted by the score of that line under the xgboost model. If a line is conjoined, its weight is the maximum of all the xgboost scores for each line in the conjoined line. Mathematically, this is represented as

$$d_r\left(l_1, ..., l_n\right) = \sum_{l_i \epsilon S_k} v\left(l_i\right) m\left(l_i\right)$$

where d_r represents the vector representation of a document d , S_k are the top-k lines with the highest scores under the xgboost model, v is the mapping from a line l_i to its set of n-grams representation, and $m(l_i)$ is the xgboost score for line l_i .

With this final weighted representation, we train an L1 regularized logistic regression model with balanced class weights to predict the final class.

We refer to this method as "supervised line attention" due to its relationship to supervised attention in the deep learning literature which predicts relevant locations and creates a weighted representation of the relevant regions' features. Supervised attention in the deep learning literature has been used to match a neural machine translations attention distribution to match an unsupervised aligner [59] and to match a sequence-to-sequence neural constituency parser's attention mechanism with traditional parsing features [52], for example. Our approach can be viewed as a form of supervised attention for document classification. The principle difference from existing work is that in supervised attention in the deep learning literature the method is trained in an end-to-end fashion with neural networks, whereas we train each module independently with classical methods and our feature representation for sentences are sets of n-grams instead of dense real-valued vectors.

Rule-based line classifier

As a baseline, we also include a line classifier that selects relevant lines by searching for expert-generated keywords and phrases. After the lines are selected, the final representation is generated the same way, with the exception that all lines are given a weight of 1; thus,for all $l_i \, \varepsilon S_k$, $m(l_i) = 1$.

Oracle Model

In addition to the line attention model, we also evaluate a model that uses the correct relevant lines from the annotator directly as input to the final classifier, which we refer to as the "oracle model" . Using the oracle lines, the final representation is generated the same way as the rule-based line classifier, where all lines are given a weight of 1.

Hyperparameter tuning

Similar to our baselines, we perform random search for 40 iterations and choose the hyperparameters that minimize 4-fold cross-validation error. The hyperparameters for our shallow attention method are an n-gram size for finding relevant lines between 1 and 4; an n-gram size for the second stage of making the final classification between 1 to 4.

For *xgboost*, the hyperparameters were 500 points from -2 to -0.5 in logspace for the learning rate; a max depth between 3 to 7; a minimum split loss reduction to split a node that is 0, 0.01, 0.05, 0.1, 0.5, or 1; a subsample ratio that is 0.5, 0.75, or 1; and an L2 regularization on the weights that is 0.1, 0.5, 1, 1.5, or 2.

For the final classifier, the L1 penalty is chosen from 500 evenly spaced points from -6 to 6 in logspace. Additionally, since the final representation is a weighted representation of the features of the top-k lines under the line classifier model, we have a hyperparameter k which determines how many lines to use, where k is between 1 and 5.

Ablation Experiments

For our ablation experiments, we investigate the relative contribution of each component in our model.

No weighting

Here we investigate if weighting the features in each line by the classifier scores increases performance compared to weighting the features in each line by one.

No joining

Here we investigate how joining affects the results when information spans multiple lines. Instead of conjoining lines that occur adjacent to each other, we leave them as separate lines for our final classifier.

No weighting and no joining

Here we neither weight the features vectors representing each line nor do we join adjacent predicted lines.

Error Analysis

To better understand model performance, we inspect all errors that the supervised line attention model makes for each attribute and cancer domain. In our investigation we find 6 primary types of errors, which we define below:

Attribute Qualification Error occurs when the model correctly extracts the relevant lines, but fails to classify the final label correctly because the label text is negated or qualified by an additional phrase indicating information is not available, such as in the following example: "If we were to classify the tumor, it would be grade 2 but due to the treatment effect it is unclassified."

Rare Phrasing Error occurs when the model correctly predicts the relevant lines, but the relevant lines contain rare or unusual phrasing and the model assigns an incorrect final classification.

Irrelevant Lines Error occurs when the model includes irrelevant lines in its final predictions, which can influence the final classification.

Multi-Label Error occurs when a report contains a conjoined label (such as “grade 1 and grade 2”), but the model only correctly predicts one of the labels.

Annotator Error occurs when the modelâs prediction is correct, but on re-review we noted that the annotatorâs label was incorrect.

U*nknown error* occurs when the underlying cause of the error is not known. This often occurs when the model correctly extracts out the relevant line but assigns an incorrect final label.

RESULTS

We trained our methods using various training set sizes of 32, 64, 128, and 186 with 4-fold cross validation. We take the average of 10 runs where we reshuffle the data and generate new splits each time and compute 95% confidence intervals for all methods using bootstrap resampling, with the exception of the HAN method due to computational limitations. For our results, unless stated otherwise, we are using the reduced annotation set due to its comparable annotation time to labeling the attribute values alone. As shown in Figure 3.1 and Table 3.4, our shallow attention model frequently improves substantially over existing methods in terms of micro and macro-f1, particularly in the lowest data regimes. For example, for colon cancer we see an absolute improvement of 0.10 micro-f1 and 0.17 macro-f1 over previously existing methods with 32 labeled data points. Furthermore, SLA frequently tends to perform as well or better than state of the art methods with only half the labeled documents. Two exceptions are in kidney cancer micro-f1 scores, where boosting performs .01 better in micro-f1. We see that the rule-based line classifier method tends to do better than existing methods with 64 labeled data points or fewer, but its performance plateaus and XGBoost outperforms it with 128 and 186 labeled data points. Furthermore, we see that the rule-based line classifier consistently performs worse than supervised line attention.

Ablation Results

We plot the results of our ablation experiments in Figure 3.2, using the same setup as our main result where we have training set sizes of 32, 64, 128, and 186 with 4-fold cross validation. Again, we take the average of 10 runs where we reshuffle the data and generate new splits each time and compute 95% confidence intervals for all methods using bootstrap resampling. We see mixed results for joining adjacent predicted lines; it appears to be inconsequential for colon cancer and detrimental for kidney cancer. However, weighting the features by line predictor seems beneficial for the macro-f1 scores. This seems to suggest that weighting helps primarily for rare classes since the macro-f1 score weights the f1 scores of each class equally.

Full Annotations

Here we compare how well reduced annotation compares to the more laborious full annotation setting where we highlight all areas in the document relevant to final classification. We use the same setup as for our main results as wells as our ablation experiments and present our results in Figure 3.3. We can see that the full annotation set leads to a consistent increase in performance. However, it is unclear whether the extra time required to create this full annotation scheme is beneficial overall as it would lead to fewer documents annotated in the same amount of time.

Error Analysis

We provide a compilation of the number of errors across attributes in Table 3.2 and Table 3.3 for colon and kidney cancer, respectively. We see that the most common error is the multi-label error. This is primarily problematic for colon cancer histologic grade, where pathologists will describe a range of grades such as "grade 1-2" and tumor site for colon and kidney cancer as tumors can inhabit multiple sites. This suggests that treating this as a multi-label classification problem instead of naively conjoining multiple labels may reduce many of the errors.

DISCUSSION

We have investigated the efficacy of location-enriched annotations and a corresponding simple and interpretable method, which we call Supervised Line Attention, for extracting data elements from pathology reports across colon and kidney cancers at UCSF. By leveraging location annotations, our two-stage modeling approach can lead to increases of micro-f1 scores up to 0.1 and macro-f1 scores up to 0.17 over state-of-the-art methods and typically reduces the required annotation by 40% to achieve the performance of existing methods.

Our SLA approach with enriched annotations was primarily developed to tackle the problem of achieving accurate performance with minimal labeled data. Previous approaches that attempt to leverage additional data use multi-task learning and transfer learning using information from other cancer domains with complex modeling architectures. For example, [76] investigated using transfer learning with convolutional neural networks to extract data from 942 breast and lung cancer reports, achieving 0.685 and 0.782 micro-f1 scores, respectively. [2] implemented multitask learning with convolutional neural networks to classify tumor attributes in 942 pathology reports for breast and lung cancers, and achieved 0.77, 0.79, and 0.96 micro-f1 scores for tumor site, histologic grade, and laterality, respectively.

An important observation is that our approach is more interpretable than previous machine learning methods, since addition to outputting the probability and predicted value for a certain report, our system outputs the exact lines of the text used to make the classification as well. This enables practitioners to easily check predictions by examining the lines output by the extraction system, and verify the system is working as expected before making clinical decisions. The hierarchical attention approach used by [35] also can output the most pertinent sentences for a classification by using the attention mechanism to hierarchically filter out pieces of text. However, our experiments show that HAN requires a large training size to achieve adequate performance due to the more complex architecture used, and requires significantly more development and computational time to search the hyperparameter space. Additionally, there have been recent concerns regarding the interpretability of attention distributions from neural networks [48].

Our study has a few limitations. Although we observe high performance of our methodology in both colon and kidney cancer reports at UCSF, our investigation was done at a single institution; this may limit the generalizability of our findings to other institutions that use different pathology reporting or data collection systems. Second, within the field of natural language processing, there has been strong empirical evidence showing the benefit of pre-trained contextualized representations for a variety of tasks, both in and out of clinical applications [73, 27, 57, 46]. In preliminary experiments, we investigated the efficacy of using biomedical word vectors [74] as feature representation input to our SLA model, but did not see an improvement in results. However, it would be interesting to investigate the effect that more sophisticated contextualized representations may have on downstream performance, and this may increase performance of SLA even further.

CONCLUSION

We have shown that including location information in annotation and applying our supervised line attention mechanism can vastly reduce the number of labeled documents needed for accurate tumor attribute classification compared to state of the art approaches. Furthermore, our supervised line attention method allows for greater interpretability due to its hierarchical nature, which can allow for easy verification of its outputs for clinicians. We hope these methods will advance the application of information extraction in medicine, where labeled data is scarce and expensive to acquire.

Tables

Table 3.1. Extracted attributes and their possible values

Tumor Site	
Colon	Cannot be determined, cecum, colon not otherwise specified, hepatic flexure, ileocecal valve, left descending colon, other, rectosigmoid junction, rectum, right ascending colon, sigmoid colon, splenic flexure, transverse colon, or not reported
Kidney	Upper pole, middle pole, lower pole, other, not specified, or not reported
Histologic Type	
Colon	Adenocarcinoma, adenosquamous carcinoma, carcinoma, type cannot be determined, large cell neuroendocrine carcinoma, medullary carcinoma, micropapillary carcinoma, mucinous adenocarcinoma, neuroendocrine carcinoma poorly differentiated, other histologic type not listed, serrated adenocarcinoma, signet-ring cell carcinoma, small cell neuroendocrine carcinoma, squamous cell carcinoma, undifferentiated carcinoma, or not reported
Kidney	Acquired cystic disease associated renal cell carcinoma, chromophobe renal cell carcinoma, clear cell papillary renal cell carcinoma, clear cell renal cell carcinoma, collecting duct carcinoma, hereditary leiomyomatosis and renal cell carcinoma-associated renal cell carcinoma, mit family translocation renal cell carcinoma, mucinous tubular and spindle renal cell carcinoma, multilocular cystic clear cell renal cell neoplasm of low malignant potential, oncocytoma, other histologic type, papillary renal cell carcinoma, papillary renal cell carcinoma type 1, papillary renal cell carcinoma type 2, renal cell carcinoma unclassified, renal medullary carcinoma, succinate dehydrogenase sdh deficient renal cell carcinoma, t611 renal cell carcinoma, tubulocystic renal cell carcinoma, xp11 translocation renal cell carcinoma, or not reported
Procedure	
Colon	Abdominoperineal resection, left hemicolectomy, low anterior resection, not specified, other, polypectomy, right hemicolectomy, sigmoidectomy, total abdominal colectomy, transanal disk excision, transverse colectomy, or not reported

continued on next page

continued from previous page

Kidney	Total nephrectomy, partial nephrectomy, radical nephrectomy, other, or not reported
Laterality	
Colon	Not applicable to colon cancer
Kidney	Left, right, or not reported
Grade	
Kidney, Colon	Grade 1, 2, 3, 4, not applicable, or not reported
Lymphovascular Invasion	
Kidney, Colon	Present, absent, or not reported
Perineural Invasion	
Colon	Present, absent, or not reported
Kidney	Not applicable for kidney cancer

Table 3.2: Error analysis: Colon cancer

Attribute	Histologic Grade	Histologic Type	Perineural invasion	Lymphovascular invasion	Procedure	Tumor Site	Total
Attribute Qualification Error	1	0	0	0	0	0	1
Rare phrasing	0	0	0	**1**	3	0	4
Irrelevant Lines Error	1	0	0	0	5	0	6
Annotator Error Error	3	**1**	**1**	0	5	0	10
Multi-label Error	**6**	0	0	0	0	**6**	**12**
Unknown error	1	0	0	0	**6**	0	7
Total by attribute	12	1	1	1	**19**	6	40

Table 3.3: Error analysis: Kidney cancer

Attribute	Histologic Grade	Histologic Type	Specimen Laterality	Lymphovascular invasion	Procedure	Tumor Site	Total
Attribute Qualification Error	0	0	0	0	0	0	0
Rare phrasing	0	0	0	**1**	**5**	0	6
Irrelevant Lines Error	**1**	0	0	**1**	1	1	4
Annotator Error	**1**	2	0	**1**	1	0	5
Multi-label Error	0	**4**	0	0	2	**6**	**12**
Unknown error	**1**	**4**	0	**1**	1	5	**12**
Total by attribute	3	10	0	4	10	**12**	39

Figure 3.1: Average micro-f1 and macro-f1 performance across attributes of different methods as a function of 32, 64, 128, 186 labeled examples on colon cancer and kidney cancer pathology reports. SLA: supervised line attention; oracle: oracle model that gets access to the true lines as input; rules: line prediction done with a rule-based method and logistic regression to predict the final class; boost: XGBoost; SVM: Support Vector Machine; logistic; logistic regression; RF: Random forest; HAN: Hierarchical attention network. We present the mean result across 10 random shufflings of the data as well as 95% bootstrap confidence intervals. We see that our method SLA outperforms existing methods in almost all cases. Furthermore, we see that predicting relevant lines outperforms our rule-based method to extract relevant lines.

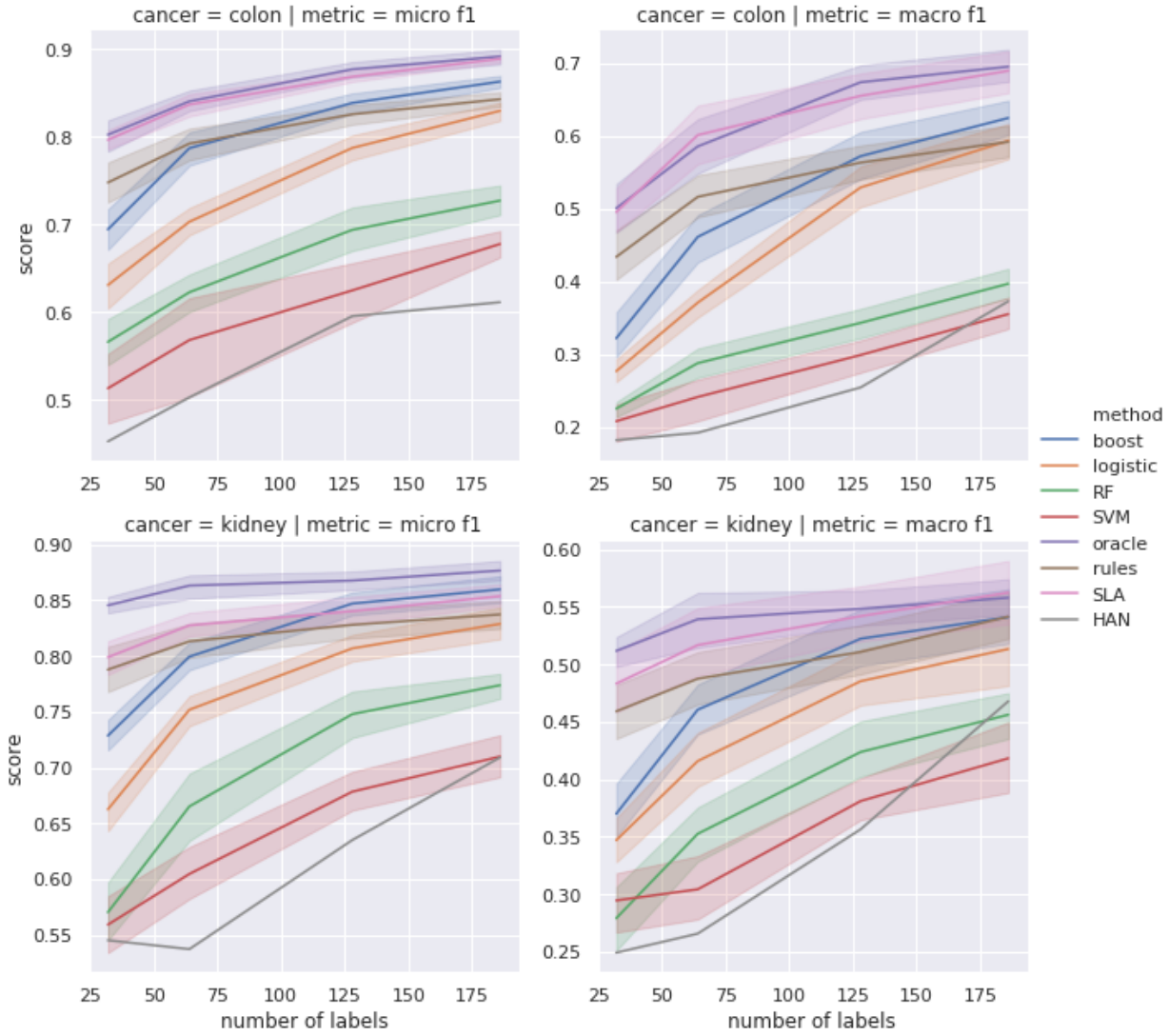


Table 3.4: Average micro-f1 and macro-f1 performance across attributes of different methods as a function of 32, 64, 128, 186 labeled examples on colon and kidney cancer. Highest performing non-oracle method is bolded.

Colon								
	HAN	RF	SVM	Boost	Logistic	Rules	SLA	Oracle
Micro-F1								
32	.45	.57	.51	.69	.63	.75	**.80**	.80
64	.50	.62	.57	.79	.70	.79	**.84**	.84
128	.60	.69	.62	.84	.79	.83	**.87**	.88
186	.61	.73	.68	.86	.83	.84	**.89**	.89
Macro-F1								
32	.18	.22	.21	.32	.28	.43	**.50**	.50
64	.19	.29	.24	.46	.37	.52	**.60**	.59
128	.25	.34	.30	.57	.53	.56	**.66**	.67
186	.37	.40	.35	.62	.59	.59	**.69**	.70
Kidney								
Micro-F1								
32	.54	.57	.56	.73	.66	.79	**.80**	.85
64	.54	.67	.60	.80	.75	.81	**.83**	.86
128	.63	.75	.68	**.85**	.81	.83	.84	.87
186	.71	.77	.71	**.86**	.83	.84	.85	.88
Macro-F1								
32	.25	.28	.29	.37	.35	.46	**.48**	.51
64	.27	.35	.30	.46	.42	.49	**.52**	.54
128	.36	.42	.38	.52	.49	.51	**.54**	.55
186	.47	.46	.42	.54	.51	.54	**.56**	.56

Figure 3.2: Ablation studies for SLA measuring the average micro-f1 and macro-f1 performance across attributes of different methods as a function of 32,64,128,186 labeled examples on colon cancer and kidney cancer pathology reports. We investigate the impact of joining adjacent selected lines prior to featurization as well as the impact of weighting the features by the line classifier scores. We present the mean result across 10 random shufflings of the data with 95% bootstrap confidence intervals. While it appears that joining adjacent predicted lines leads to mixed or potentially even negative performance over not joining adjacent predicted lines, weighted methods seem to outperform their unweighted alternatives, especially for macro-f1 scores, suggesting that weighting helps in particular for rare classes

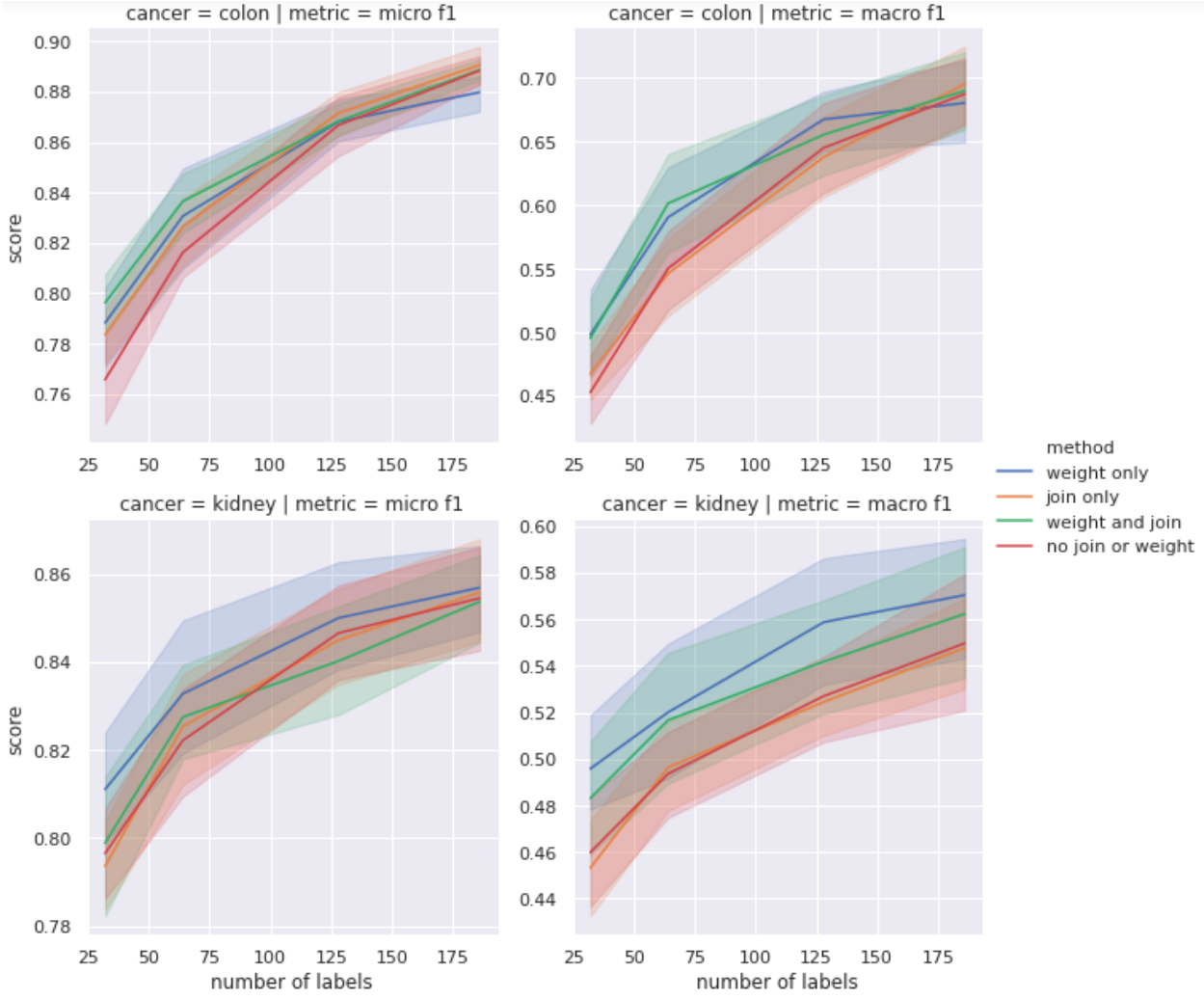
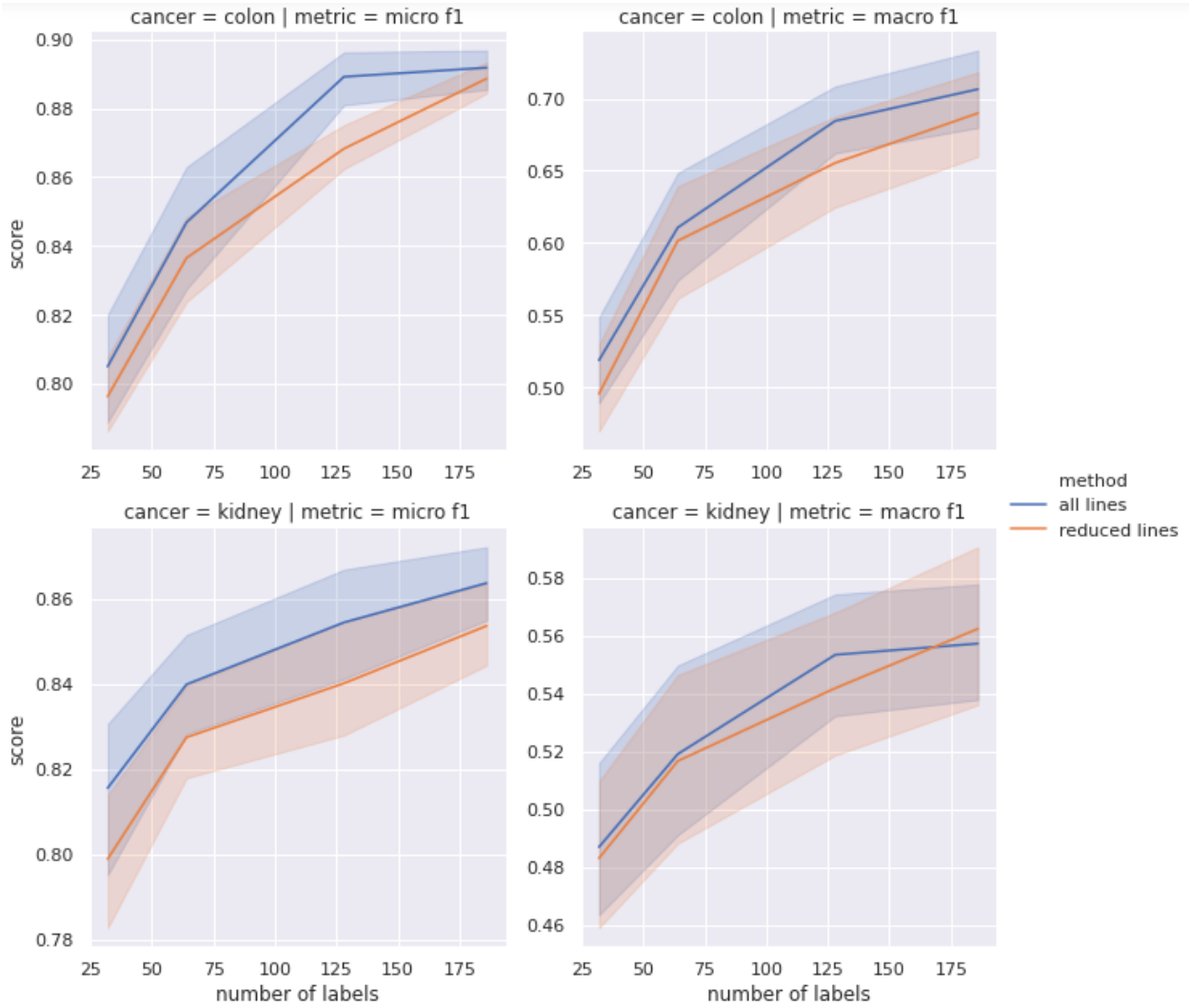

cancer = colon | metric = micro f1
cancer = colon | metric = macro f1
cancer = kidney | metric = micro f1
cancer = kidney | metric = macro f1
score
number of labels
method
weight only
join only
weight and join
no join or weight

Figure 3.3: Comparing the more laborious scheme of annotating the location information for all relevant lines for a given attribute as compared to the more lightweight annotation method of only annotating the first line in the synoptic comment, if the synoptic comment contains the information, or the first relevant line in the document otherwise. We see that having the additional information yields a consistent, though sometimes small, benefit.



Chapter 4

Transfer Learning and String Similarity for Information Extraction from Pathology Reports

Our contributions in this chapter are creating methods for extracting information from clinical documents under the presence of much fewer examples, by leveraging transfer learning techniques and string similarity to facilitate zero-shot learning, for predicting tumor attributes from pathology reports at UCSF across multiple cancers. In particular, we develop a novel transfer learning procedure that allows us to leverage pathology reports from outside cancers to increase performance on a cancer of interest. Furthermore, to address the case of a large number of classes and a limited amount of labeled data points, we develop zero-shot classification techniques by leveraging string similarity techniques. By doing this, we can perform as well as the most data efficient methods in the existing literature while requiring typically only half of the data.

In this work, we develop novel natural language process methods that extract information from unstructured pathology reports from very few labeled examples. In particular, we build upon the work in in the previous chapter which introduces a novel two-stage approach to extracting data from electronic health records using location-based annotations and dramatically improves data efficiency and often only using required half of the labeled data. Specifically, we improve the two-stage method with transfer learning and string similarity and evaluate their performance on extracting categorical data fields from lung, kidney, and colon pathology reports We compare these improvements with the original two-stage method and other state-of-the-art approaches.

BACKGROUND

There has been considerable interest in developing algorithmic approaches to automatically extract data elements from clinical text and convert them into structured form such as a table [95, 12, 64, 61]. Many of the learning-based approaches rely on training document-level classifiers which take in the full text of a report and predict a value for a given data field, such as the histologic type of a cancer. Training these types of methods requires adequate amounts of annotations which are expensive and time consuming to obtain. Due to these limitations, it is important to investigate how to build reliable extraction methods which can be built using small amounts of annotated reports.

We develop a data extraction system in the data sparse setting where we have access to small amounts of annotated pathology reports across lung, kidney, and colon cancers. We focus specifically on categorical data fields where the extraction task is formulated as a document classification learning problem. An example of a categorical data field is the lymphovascular invasion where the fixed categories are "present", "absent", or "not reported".

Previous works have explored reliably extracting in limited data settings. Yala et al. carried out a performance analysis of boosting tree extraction models and found that around 400 training examples were required to obtain an accuracy of 0.9 for 20 breast cancer classification fields [98]. Odisho et al. showed that non-deep learning methods largely outperformed deep learning methods using annotated data sizes below 256 for prostate cancer reports [67]. The previous chapter developed a novel two-stage approach using more fine-grained, location-based annotations and showed that the fully supervised location-based approach outperformed the state-of-the-art using training data sizes below 186 for colon, kidney, and lung cancer reports. However, none of these works contain a performance analysis using transfer learning from other cancer types or consider the case where many labels donât show up in the training data.

Transfer learning has been shown as a promising approach to improve medical data extraction performance. Qiu et al. found that intra-class transfer learning on convolutional neural networks provided improvements of up to 0.04 in micro-F1 and macro-F1 scores for lung and breast tumor sites [76]. Alawad et al. show that multi-task convolutional neural networks trained across cancer types achieve up to 0.17 improvement in the macro-F1 score over single cancer registry models [3].

In this work, we explore extending the existing two-stage approach using transfer learning and string similarity methods. For data fields with categories that are shared across colon, kidney, and lung cancers, we study whether applying transfer learning to the two-stage method improves performance. For data fields with categories that are unique to a specific

cancer type, we study whether interpolating string similarity methods with the two-stage method can lead to improvements in performance.

4.1 MATERIALS AND METHODS

Data Sources

The pathology reports we use span across colon, lung, and kidney cancers from the University of California, San Francisco from 2002 to 2019. For each cancer type, we have access to 250 annotated reports which we use to create 10 random train-validation-test splits. Each of the 10 splits consists of the same 250 annotated reports overall, but the individual training, validation, and test sets are different due to randomness. The train, validation and test sets consist of 40, 20, and 190 annotated reports, respectively. Each experiment is run 10 times separately on each of the 10 random splits. We obtain confidence intervals for the evaluation metrics using the test set of each split.

Data fields

Pathology reports contain information about a patient's clinical history and data fields, such as the cancer stage, tumor histology and grade, and presence of invasion to surrounding tissues. The chosen data fields are categorical attributes that appear in the College of American Pathology reporting guidelines, but we filter out data fields where the tumor attribute appears in less than 90% of our reports.

We then specifically look at fields that are shared between cancers; examples of such fields are the histologic type and grade of the cancer. These fields are relevant across many cancer types unlike the presence of rhabdoid features, which is relevant to only kidney cancer from the 3 cancers we study. Out of these shared fields, we specifically study the histologic grade and presence of lymphovascular invasion in the context of transfer learning (Table 4.1). For example, a relevant question is if a trained model performs better for extracting the histologic grade for kidney cancer if it has been previously trained to extract the histologic grade from colon cancer. The histologic grade and presence of lymphovascular invasion fields are specifically chosen for this task since they appear across all three cancer domains and have labels that are independent of the cancer domain. For example, the labels for the presence

of lymphovascular invasion field arenot identified, present, or not available regardless of the cancer domain.

We additionally focus on the following data fields: the procedure carried out on a patient, the site of a tumor, and the histologic type of a tumor (Table 4.2). These fields appear across all three cancer domains but have labels specific to a cancer domain. For example, the categories for kidney tumor site include the upper pole, middle pole, and lower pole. For these fields, our goal is to study whether a string similarity type approach enhances a learned information extraction model trained on small amounts of data.

Enriched Annotations

Our pathology reports contain annotations for ground-truth labels as well as highlighted text throughout the report relevant to the label as in the previous chapter. These fine-grained annotations provide the lines in the report that determine the value of a tumor attribute. Similar to the previous chapter we use the "creduced annotation set", which consists of the minimal set of annotations containing the line of a given data field's value in the synoptic comment or the first line that contains the relevant information if the report does not contain a synoptic comment. These synoptic comments are typically common in more recent pathology reports and are a brief standardized portion of the text where relevant cancer attributes are reported. As mentioned in the previous chapter, reduced location-specific annotations take 20 percent longer on average than typical annotations.

Data Preprocessing

Before any kind of vectorization of the text, we replace words that occur once in the training data and words that never appear in the training data with a special <UNK> token. We then remove commas, backslashes, semi-colons, tildes, periods, and the word "null" from each report and add spaces around colons, forward slashes, parentheses, plus and equal signs. Additionally, we also concatenate labels if multiple labels are assigned to a report for a specific field. For example, if the report contains both "upper" and "lower" labels for the field tumor site, then we aggregate the labels so the final label is "upper and lower."

SUPERVISED LINE ATTENTION

Our models are based on the two-stage framework introduced in the previous chapter. At a high level, the goal is to predict the lines in the report which contain information on a specific data field and then use the predicted lines to make the final class prediction for a given report. There are two separate classifiers for the line prediction task and the class prediction task which are both trained directly using our location-based annotations. As in previous work, the xgboost model is used for the line prediction task, and logistic regression is used for the class prediction task.

To help us tackle this task, we break up our tumor attributes of interest into two distinct categories. On the one hand we have tumor attributes with shared labels across cancers. One such example is the histologic grade; every cancer is graded on the same scale. On the other hand we have tumor attributes whose labels are not shared across cancers. One such example is the procedure; for each cancer type there are a different set of procedures used for resecting tumors. We break these up into two distinct categories since, the first case is a natural candidate for a transfer learning approach, whereas for the second, since the labels are not shared across cancer types, transfer learning is less applicable. We propose two methods to perform extraction depending on whether the labels are shared across cancers or not.

1) Shared labels case

In the case where the labels are the same across cancer domains, knowledge can be transferred from one cancer type to another. For example, for the presence of lymphovascular invasion, identifying the relevant lines in a report is domain-independent because lymphovascular invasion is a relevant attribute for many cancers. Furthermore, identifying the correct label of the lymphovascular invasion field is again domain-independent because the categories (present, not identified, and not reported) are the same across cancer types. In the low-data setting, this shared knowledge is especially important because reports from other domains can be used to improve extraction performance through data augmentation.

To take advantage of this property, we create a transfer learning technique to learn data extraction using the shared information across cancer types for the relevant data fields. We build off the supervised line attention method by training both the line classifier and the final classifier on reports from all domains, which we refer to as hierarchical cancer to cancer learning (HCTC). We include in our results ablations to HCTC where we only share information for the line classifier (HCTC-line) or the final classifier (HCTC-final).

We note that, to the authors' best knowledge, this is the first work to investigate transfer learning techniques across multiple cancers.

2) Unique labels case

Unlike the shared labels case, we opt against a transfer learning approach and only focus

on an individual cancer at a time, as it is unclear the level of applicability of transfer learning since the label space is different for each cancer. Furthermore, many of our fields have a large number of labels, such as kidney histologic type which has 32 different labels (Table 2). Therefore, it is possible that we will see labels at test time that we have never seen at training time.

A challenge is that typical machine learning models need a sufficient number of examples for each possible label to learn classification tasks. As seen by the set of possible labels for our fields (Table 2), this is not possible for small training data sizes. Consequently, a technique that can handle a large set of labels is essential here and in particular, a method capable of zero-shot learning is necessary.

We develop a novel string similarity method that allows us to have a more sample efficient method that is capable of zero-shot learning. We refer to this method as zero-shot string similarity (ZSS). At a high level, ZSS first predicts the relevant lines for the label as in the previous chapter, then from the set of labels, finds the label with the highest string similarity from the relevant lines. The labels we use are defined in the College of American Pathology reporting guidelines. For example, the set of labels for the site of the tumor for kidney cancer include the labels upper pole, mid-pole, and lower pole.

ZSS involves calculating pairwise character-based similarity scores between a predicted line of a report and each possible label. We use the line in the report with the highest probability computed using the line classifier as the predicted line. The similarity between words in a predicted line and a candidate label is computed with the Ratcliff-Obershelp algorithm which is based on the longest contiguous matching subsequence. We tried several character-level string similarity algorithms, such as the Jaro-Winkler similarity, Levenshtein similarity, and Hamming distance but found that the Ratcliff-Obershelp approach performed best on the training set of the first split. The label with the highest Ratcliff-Obershelp score is used as the final prediction for a report. The full algorithm is described below.

Zero-shot similarity (ZSS)

Input: Predicted line in a report, the probability output from the line classifier, a set of labels for a given field (the possible values that a data field can take), and a learned cutoff parameter used to predict 'NA' or not reported Output: Predicted label

Output: Predicted label

1. *For eachcandidate label, calculate its fuzzy jaccard score with the predicted line*
2. *Take the label with the highest fuzzy jaccard score. If there is a tie between multiple labels, take the label with the most characters as the prediction*
3. *If the fuzzy jaccard score is less than 0.5, then predict "other"*
4. *If the line probability is less than the cutoff, then predict "NA"*

Fuzzy jaccard algorithm

Input: Predicted line in a report and a candidate label
Output: Similarity score

1. *For each unique word in the candidate label, calculate its string similarity score with each unique word in the predicted line using the Ratcliff-Obershelp contiguous matching subsequence algorithm and find the max similarity score.*
2. *Sum up the max similarity scores across unique words for the candidate label*
3. *Scale the resulting sum by the number of unique words in the label*

Ensembling String Similarity with the General Two-Stage Approach: While we found ZSS to be effective on its own, we found that it suffered from a few weaknesses. In particular, we found that the "other" class particularly challenging, as it consists of all possible values the field can take outside of the defined label set in the CAP protocols. For example, if the field is "procedure", then the "other" class corresponds to all other possible procedures not listed in the CAP protocols, which will all have low string similarity to the label "other". Furthermore, there are cases where the label in general is very dissimilar to how it occurs in the text. With these examples, no matter how many data points our algorithm is trained on, it will never get these right. Therefore, we aim to get the best of both ZSS and the two stage approach by developing a hybrid approach to the problem. If the final string similarity score is above a learned threshold, then we output the ZSS prediction. Otherwise, we output the two-stage prediction. We call this method ZSS-thresholding. We also include an oracle method for the sake of comparison that chooses the two-stage prediction if it is equal to the ground truth and the ZSS prediction otherwise. This oracle method serves as an upper bound on the performance of our string similarity enhancement of the two-stage method. Our final method is ZSS-doc which is using ZSS on the entire text of the report instead of the lines output from the line classifier. This allows us to gauge how necessary the location targeting approach is for the ZSS methods.

BASELINE METHODS

1) Shared labels

Our first set of baselines is document-level classification methods, such as logistic regression, XGBoost, random forest, and support vector machines. These methods take as input all the tokens in a given report and predict the class value of a particular data field. The bag of words approach is used to vectorize the text in each document and train the outlined machine learning methods. This approach only uses the final document-level labels and is trained on the cancer of interest as well as the out-domain cancer reports.

Our next baseline is the hierarchical attention network (HAN) for document classification [35]. In particular, we study the hierarchical attention network (HAN) in terms of transfer learning. We pretrain the model on out-domain reports for a shared field and then fine-tune the model on in-domain reports.

We also use the two-stage approach outlined in the previous chapter as an additional baseline. XGBoost is used as the line prediction model, while the logistic regression is used as the final label classifier. Location-based annotations are used to train the line prediction model, while the document-level label annotations are used to train the final classifier.

2) Unique labels

The baselines in the unique labels scenario include the ordinary document-level classifiers and the two-stage approach similar to the previous case. All the methods in the unique labels case are trained on a single cancer domain.

HYPERPARAMETER TUNING

For all our experiments, we train each method with 40 randomly chosen hyperparameter values and choose the set that maximizes the average 4-fold cross-validation score.

Model-based parameters

For the *xgboost* method we sample from 1000 points from -2 to -0.5 in logspace for the learning rate; values 3 to 7 for max depth; a minimum split loss reduction to split a node from 0, 0.01, 0.05, 0.1, 0.5 and 1; a subsample ratio values from 0.5, 0.75, or 1; and L2 regularization values from 0.1, 0.5, 1, 1.5, or 2. For *logistic regression*, we sample 1000 evenly spaced points from -6 to 6 in logspace for the L1 regularization parameter.

Two stage parameters

The two-stage method has additional model-independent hyperparameters which are the n-gram size for the line classifier which is between 1 and 4, the n-gram size for the final classifier which is again between 1 and 4, the number of selected lines to use as input to the final classifier which is between 1 and 5.

String similarity parameters

The string similarity approach has an additional model-independent cutoff parameter sampled from 0.05, 0.1, 0.15, 0.2, 0.25, and 0.3 which is used to handle NA values for a given data field. We specifically use the maximum line classifier probability from the positively predicted lines. If this probability is less than the cutoff value, then the predicted value from the string similarity method is replaced by "NA."

Two stage + string similarity parameters

The two-stage data augmented variation contains a threshold parameter that is sampled from 0.8, 0.85, 0.9, 0.95, and 1.0. This parameter is used to select instances outside the train set to augment the training data. All instances which have a similarity score with a label greater than the threshold value is used along with the string similarity prediction for data augmentation.

Similarly, the two-stage thresholding variation contains a threshold parameter sampled from 0.8, 0.85, 0.9, 0.95, and 1.0. If a given instance has a string similarity score with a label greater than the threshold value, then the two-stage prediction is replaced with the string similarity prediction.

RESULTS

We run two sets of experiments across lung, colon, and kidney cancers: one for the shared field and shared labels case and another for the shared field and unique labels case. Each method is trained on training data sizes of 10, 20, and 40 in-domain reports along with 372 out-domain reports with validation set sizes of 5, 10, and 20 in domain reports, respectively. The test set consists of 186 held out reports from the domain in question.

Each experiment is run 10 times where the training, validation, and test splits are randomly formed. We compare across methods using the mean micro-F1 and mean macro-F1 scores and obtain confidence bounds around the means. Due to computational reasons we only run HAN one time for all experiments.

Shared field and shared labels

HCTC consistently improves performance across the data sizes compared to the two-stage method (Figure 4.1). Comparing between the methods on the same data sizes, performance increases by 0.09 average micro-F1 score and 0.05 average macro-F1 score when using HCTC over the two stage method on 10 training points. or training data size 40, the increase in performance narrows to about 0.02 for the average micro-F1 score and 0.03 for the average macro-F1 score. HCTC also requires only 10 training points to achieve 97% of the micro-F1 performance and 92% of macro-F1 performance of the two stage method using 20 annotated training reports. Compared to the two stage method using 40 annotated training reports, HCTC only requires 20 reports to achieve 99% of the micro-F1 performance and 96% of the macro-F1 performance.

Shared field and unique labels

ZSS-based methods outperform the two-stage method across dataset sizes for the shared field and unique labels case (Figure 1). ZSS-thresholding consistently outperforms the other

methods for both micro-F1 and macro-F1. For training data size 10, ZSS-thresholding achieves an improvement of 0.11 micro-F1 and 0.23 macro-F1 over the two-stage method. For training data size 40 the improvement is 0.00015 for micro-F1 and 0.14 macro-F1 over the vanilla two-stage method. Comparing across data sizes, ZSS-thresholding with 10 data points achieves an increase of 0.02 in micro-F1 and 0.17 in macro-F1 over the two-stage method with 20 data points. Using 20 data points, ZSS-thresholding achieves 98% of the performance in micro-F1 and an increase of 0.13 in macro-F1 compared to the two stage method trained on 40 data points.

Our ZSS methods greatly outperform the baseline methods for the shared field and unique labels case (Figure 4.4). Compared to the best baseline method, xgboost, trained on 40 reports, ZSS-thresholding trained on 10 reports achieves an increase of 0.04 in micro-F1 and 0.045 in macro-F1. Similarly, ZSS trained on 10 reports achieves an increase of 0.035 in macro-F1 and 0.031 in micro-F1 over xgboost trained on 40 reports.

DISCUSSION

We have developed two ways to improve the performance of learning-based extraction systems when the amount of annotated reports are limited. For fields where the data field and labels are shared across domains, it is natural to aggregate annotations across domains to augment the data used to train the models. Our experiments with enhancing the two-stage method show that much of the gain in performance occurs within the lower end of the data sizes; there is a 0.09 increase in micro-F1 for data size 10 and 0.02 increase in micro-F1 for data size 40 for our reports.

In the case of fields where the labels are not shared across domains, we opt for a string similarity enhancement instead of a transfer approach. Because the categories for these fields are unique for each domain, there is less room for improvement via transfer learning due to cancer-unique labels. String similarity is a more viable approach because typically in this case the text will contain strings close to the label names. Our experiments in this work show that interpolating learning-based solutions with string similarity prediction can lead to a significant increase in performance - up to 0.1 micro-F1 and 0.23 macro-F1 for data size 10 and similar performance for micro-F1 and 0.14 macro-F1 for data size 40.

Our findings motivate future directions for information extraction with small data regimes. One promising direction is taking advantage of models pre-trained using large corpuses of text on language modeling tasks. Recent work in NLP has shown fine-tuning such models on specific tasks with small amounts of data lead to improvements in performance for a given task. Combining such models, such as BERT, [22] with the two-stage framework could be a promising direction for future research. Furthermore, we did not study how much transfer learning benefits learning across different fields for a particular cancer. Even though the

majority of categories are different between data fields even for one particular cancer, there can be similarities within the text. For example, one such case is when a pathologist denotes that a particular attribute is not reported in the text which is applicable to many data fields. Hence a fully unified extraction model may perform better than a model trained on a specific data field. In practice, it is also easier to maintain one integrated model over maintaining many individual models.

Another promising direction is improving the ensembling method between machine learning methods and rules-based or string similarity methods for the unique labels case. Our results on the oracle ensembling model shows that there is still much room for improvement when combining string similarity predictions with machine learning predictions. For example, the oracle model has up to 0.075 improvement in both macro and micro-F1 over our method of thresholding based on the learned similarity score cutoff. Potential approaches include basing the decision-making process on the uncertainties of each algorithm or combining model probabilities and string similarity scores for each possible label and outputting the label with the highest score

CONCLUSION

Data is the fuel for machine learning development in personalized healthcare, and much of this data is found in text data. However, in an unstructured form, utilizing this data is quite challenging; furthermore, due to privacy concerns and the technical nature of annotation, acquiring a large labeled dataset is typically infeasible, making it difficult to apply state of the art general domain natural language processing methods. Therefore, it is important for personalized healthcare to develop efficient natural language processing methods. In this work we have created two methods to tackle this problem, one investigating novel cancer to cancer transfer learning techniques and the other utilizing string similarity techniques to handle the problem of zero-shot learning or when the number of classes is relatively large compared to the number of labels. We find that our transfer learning method achieves an increase of up to 0.09 in micro-F1 and 0.05 in macro-F1. Furthermore, we find that our string similarity method achieves an increase of up to 0.11 in micro-F1 and 0.23 in macro-F1.

TABLES

Table 4.1. Extracted attributes and their possible values for the shared fields and shared labels case

Field	Possibles values
Histologic grade	Grade 1, grade 2, grade 3, grade 4, or not reported
Lymphovascular invasion	Present, absent, or not reported

Table 4.2. Extracted attributes and their possible values for the shared fields and unique labels case

Tumor Site	Possible values
Colon	Cannot be determined, cecum, colon not otherwise specified, hepatic flexure, ileocecal valve, left descending colon, other, rectosigmoid junction, rectum, right ascending colon, sigmoid colon, splenic flexure, transverse colon, or not reported
Kidney	Upper pole, middle pole, lower pole, other, not specified, or not reported
Lung	Upper lobe, middle lobe, lower lobe, bronchus, or not reported
Histologic Type	Possible values
Colon	Adenocarcinoma, adenosquamous carcinoma, carcinoma, type cannot be determined, large cell neuroendocrine carcinoma, medullary carcinoma, micropapillary carcinoma, mucinous adenocarcinoma, neuroendocrine carcinoma poorly differentiated, other histologic type not listed, serrated adenocarcinoma, signet-ring cell carcinoma, small cell neuroendocrine carcinoma, squamous cell carcinoma, undifferentiated carcinoma, or not reported

continued on next page

continued from previous page

Kidney	Acquired cystic disease associated renal cell carcinoma, chromophobe renal cell carcinoma, clear cell papillary renal cell carcinoma, clear cell renal cell carcinoma, collecting duct carcinoma, hereditary leiomyomatosis and renal cell carcinoma-associated renal cell carcinoma, mit family translocation renal cell carcinoma, mucinous tubular and spindle renal cell carcinoma, multilocular cystic clear cell renal cell neoplasm of low malignant potential, oncocytoma, other histologic type, papillary renal cell carcinoma, papillary renal cell carcinoma type 1, papillary renal cell carcinoma type 2, renal cell carcinoma unclassified, renal medullary carcinoma, succinate dehydrogenase sdh deficient renal cell carcinoma, t611 renal cell carcinoma, tubulocystic renal cell carcinoma, xp11 translocation renal cell carcinoma, or not reported
Lung	Adenocarcinoma in situ mucinous, adenocarcinoma in situ nonmucinous, adenocarcinoma acinar predominant, invasive adenocarcinoma acinar predominant, pulmonary adenocarcinoma acinar predominant, adenosquamous carcinoma, atypical carcinoid tumor, carcinoma, combined small cell carcinoma, fetal adenocarcinoma, invasive adenocarcinoma micropapillary predominant, invasive adenocarcinoma papillary predominant, invasive adenocarcinoma lepidic predominant, invasive mucinous adenocarcinoma, invasive squamous cell carcinoma keratinizing, invasive squamous cell carcinoma non-keratinizing, invasive squamous cell carcinoma basaloid, large cell carcinoma, large cell neuroendocrine carcinoma, lymphoepithelioma, minimally invasive adenocarcinoma, minimally invasive adenocarcinoma mucinous, mucinous adenocarcinoma, mucoepidermoid carcinoma, non-small cell carcinoma, small cell carcinoma, squamous cell carcinoma in situ, solid adenocarcinoma with mucin, typical carcinoid tumor, squamous cell carcinoma, other, or not reported
Procedure	Possible values
Colon	Abdominoperineal resection, left hemicolectomy, low anterior resection, not specified, other, polypectomy, right hemicolectomy, sigmoidectomy, total abdominal colectomy, transanal disk excision, transverse colectomy, or not reported

continued on next page

continued from previous page

Kidney	Total nephrectomy, partial nephrectomy, radical nephrectomy, other, or not reported
Lung	Bilobectomy, completion lobectomy, wedge resection, lobectomy, segmentectomy, pneumonectomy, other, or not reported

FIGURES

Figure 4.1: Average micro-f1 and macro-f1 performance across transfer learning variations of the two-stage method as a function of 10, 20, and 40 labeled examples on colon, kidney, and lung cancer pathology reports. The results presented include the mean performance across 10 random splits of the data and 95% confidence intervals for the shared labels case. The two-stage method with transfer learning for both the line and final classifiers consistently outperforms the other variations of two-stage including no transfer learning across the different training set sizes.

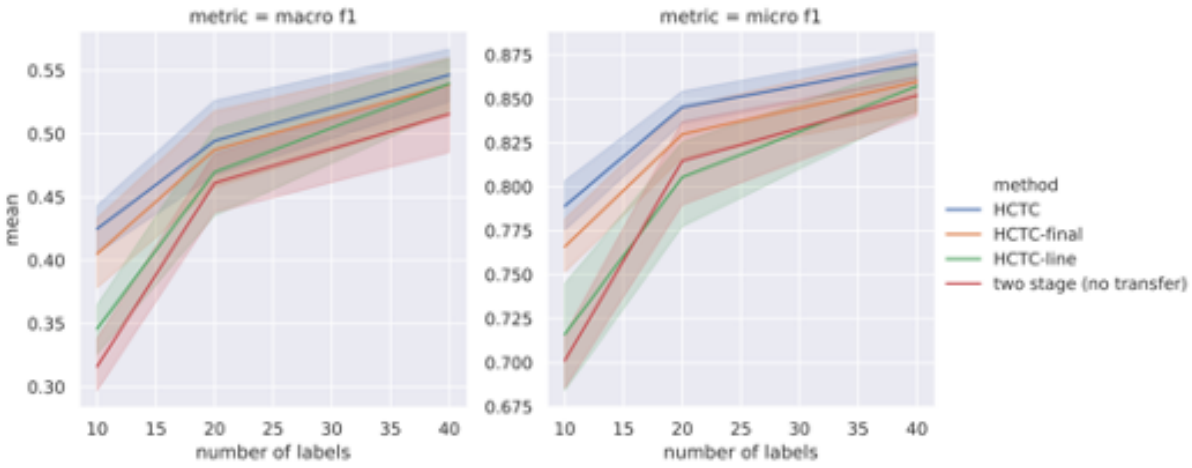


Figure 4.2: Average micro-f1 and macro-f1 performance across baseline methods as a function of 10, 20, and 40 labeled examples on colon, kidney, and lung cancer pathology

reports. The results presented include the mean performance across 10 random splits of the data and 95% confidence intervals for the shared labels case.

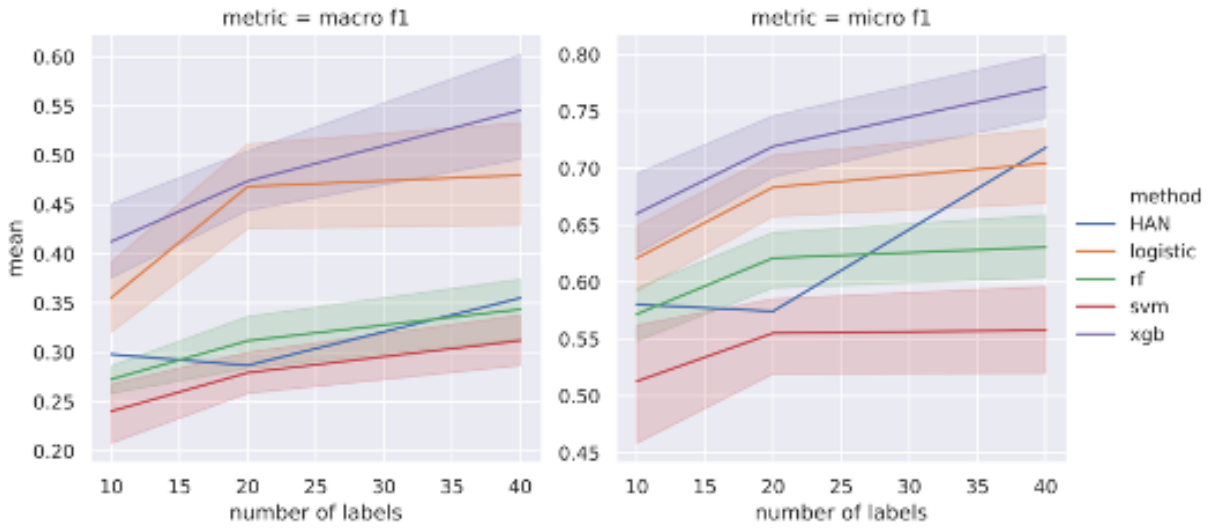


Figure 4.3: Average micro-f1 and macro-f1 performance across string similarity variations of the two-stage method as a function of 10, 20, and 40 labeled examples on colon, kidney, and lung cancer pathology reports. The results presented include the mean performance across 10 random splits of the data and 95% confidence intervals for the shared field and unique labels case. The thresholding variation consistently outperforms the other methods across the different training set sizes for both micro-F1 and macro-F1 metrics.

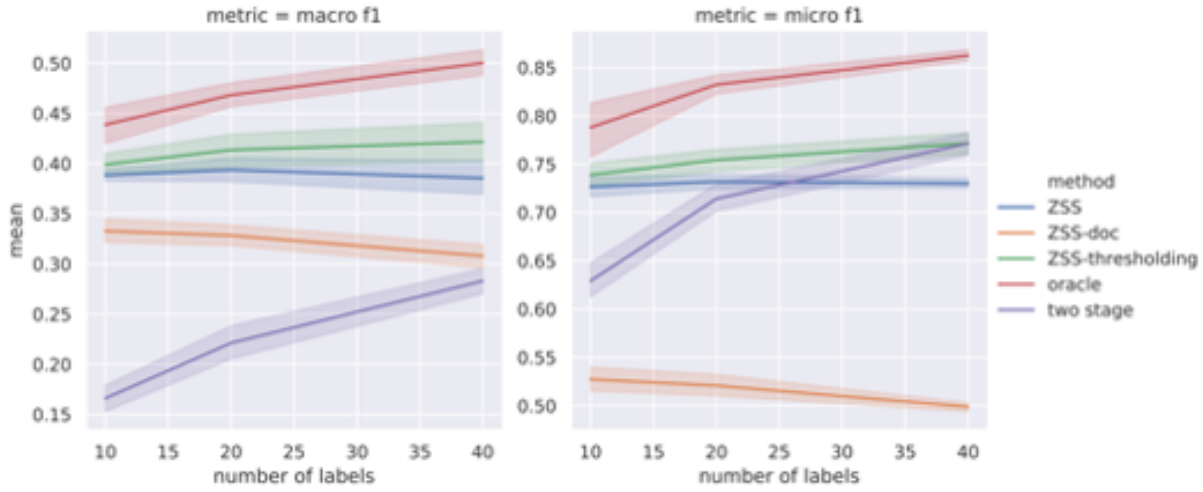
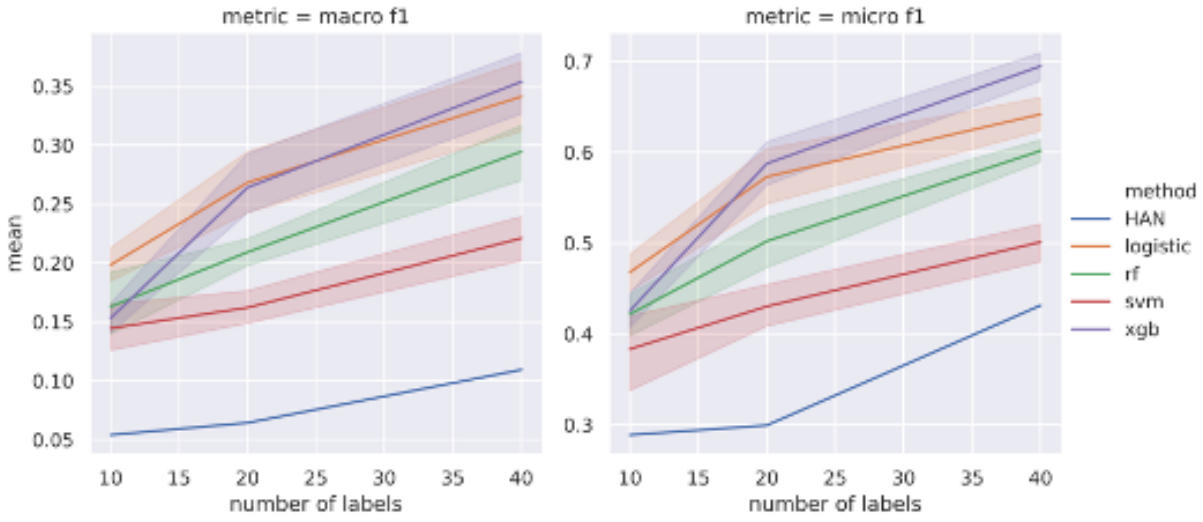


Figure 4.4: Figure 4: Average micro-f1 and macro-f1 performance across baselines as a function of 10, 20, and 40 labeled examples on colon, kidney, and lung cancer pathology reports. The results presented include the mean performance across 10 random splits of the data and 95% confidence intervals for the unique labels case.



Chapter 5

Curating a COVID-19 data repository and forecasting county-level death counts in the United States

5.1 Introduction

In recent months, the COVID-19 pandemic has dramatically changed the shape of our global society and economy to an extent modern civilization has never experienced. Unfortunately, the vast majority of countries, the United States included, were thoroughly unprepared for the situation we now find ourselves in. There are currently many new efforts aimed at understanding and managing this evolving global pandemic. This paper, together with the data we have collated (and are collating continuously), represents one such effort.

Our goals are to provide access to a large data repository combining data from a range of different sources and to forecast short-term (up to two weeks) COVID-19 mortality at the county level in the United States. We also provide uncertainty assessments of our forecasts in the form of prediction intervals based on conformal inference [93].

Predicting the short-term impact, e.g., over the next week, of the virus in terms of the number of deaths is critical for many reasons. Not only can it help elucidate the overall impacts of the virus, but it can also help guide difficult policy decisions, such as whether or not to impose or ease lock-downs or whether to re-open. While many other studies focus on predicting the long-term (several months or over the year) trajectory of COVID-19, these approaches are currently difficult to verify due to a lack of long-term COVID-19 data.[1] On the other hand, predictions for immediate short-term trajectories are much easier to verify and are likely to be much more accurate than long-term forecasts due to comparatively fewer uncertainties involved, e.g., due to policy change or behavioral changes in society. Short-term predictions are also necessary for PPE distribution planning and policy decisions such

[1]Given the longer trajectory of the COVID-19 since the first version of our paper, the longer-term predictions can probably be given a better scrutiny now.

as safe re-opening of the counties and states. So far, a vast majority of predictive efforts have focused on modeling COVID-19 case-counts or death-counts at the national or state-level [32], rather than the more fine-grained county-level that we consider in this paper. To the best of our knowledge, ours was the first work on county-level forecasts.[2]

The predictions we produce in this paper focus on recorded cumulative death counts, rather than recorded cases since recorded cases fail to accurately capture the true prevalence of the virus due to previously limited testing availability. Moreover, comparing different counties based on *the number of* recorded cases is difficult since some counties have performed many more tests than others: the number of positive tests does not equal the number of actual cases. While the *proportion of positive tests* is more comparable across different counties, our modeling approach focuses on recorded death counts rather than proportions. The original motivation to predict death counts was to provide a proxy for the severe case counts, where individuals would need intense care in a hospital; see Section 5.7 for further discussion. We note that the recorded death count is also likely to be an under-count of the number of true COVID-19 deaths because it seems as though in many cases only deaths occurring in hospitals are being counted.[3] Nonetheless, the recorded death count is generally believed to be more reliable than the recorded case count.[4] We also note that, more recently, several efforts are being made to obtain better recorded (not using any algorithms or models) COVID-19 death counts, e.g., by including probable deaths and deaths occurring at home.[5]

In Section 5.2, we introduce our data repository and summarize the data sources contained within. This data repository is being updated continuously (as of September 2020) and includes a wide variety of COVID-19 related information in addition to the county-level case-counts and death-counts; see Tables 5.1–5.4 for an overview. Given the rapidly evolving and dynamic nature of COVID-19, several biases arise in the COVID-19 infection data. We provide a detailed discussion on these biases in the context of our forecasts in Section 5.2.

In Section 5.3, we introduce our predictive approach, wherein we fit a range of different exponential and linear predictor models using our curated data. Each predictor captures a different aspect of the behaviors exhibited by COVID-19, both spatially and temporally, i.e., across regions and time. The predictions generated by the different methods are combined using an ensembling technique by [80], which we refer to as Combined Linear and Exponential Predictors (CLEP). Additional predictive approaches, including those using social distancing information, are presented in Appendix .1, even though they do not outperform the CLEP predictors in the main text based on the COVID-19 case and death counts in the past and neighboring counties.

[2]By the time of our first submission to arXiv on May 16, 2020, we were not aware of any concurrent work on county-level forecasts. See Section 5.6 for discussion on related work where we also discuss the county-level forecasts by [19] (time stamp of June 8) which we became aware of in mid-June while revising the manuscript.

[3]This comment is applicable for the period up to June 21, 2020 considered in this paper, e.g., see `https://www.nytimes.com/interactive/2020/04/28/us/coronavirus-death-toll-total.html`

[4]`https://coronavirus.jhu.edu/data/cumulative-cases`

[5]`https://covidtracking.com/blog/confirmed-and-probable-covid-19-deaths-counted-two-ways`

In Section 5.4, we develop uncertainty estimates for our predictors in the form of prediction intervals, which we call Maximum (absolute) Error Prediction Intervals (MEPI). The ideas behind these intervals come from conformal inference [93] where the prediction interval coverage is well defined as the empirical proportion of times when the observed cumulative death counts fall into the prediction intervals over a period of time. Since their guarantees rely on an exchangeability property of the prediction errors in the past several days, we also examine this property in the context our prediction tasks.

Section 5.5 details the evaluation of the predictors and the prediction intervals for the 3, 5, 7, and 14-days-ahead forecasts. We use the data from January 22, 2020, the day of the first COVID-19 death in the US[6], and report the prediction performance over the period March 22, 2020 to June 20, 2020. Overall, we find that CLEP predictions are adaptive to the exponential and sub-exponential nature of COVID-19 outbreak, with about 15% error for 7-day-ahead and 30% error for 14-day-ahead predictions; e.g., see Table 5.9. We also provide detailed results for our prediction intervals MEPI from April 11, 2020 to June 20, 2020. And we observe that MEPIs are reasonably narrow and cover the recorded number of deaths for more than 90% of days for most of the counties in the US, e.g., see Figures 5.13 and 5.14.

Finally, we describe related work by other authors in Section 5.6, discuss the impact of our work in distributing medical supplies across the country in Section 5.7, and conclude in Section 5.8.

Making both the data and the predictive algorithms used in this paper accessible to others is key to ensuring their usefulness. Thus the data, code, and predictors we discuss in this paper are open-source on GitHub (`https://github.com/Yu-Group/covid19-severity-prediction`) and are also updated daily with several visualizations at `https://covidseverity.com`. The results in this paper contain case and death information at county level in the U.S. from January 22, 2020 to June 20, 2020 but the data, forecasts, and visualizations in the GitHub repository and on our website are updated daily. See Figure 5.1 for a high-level summary of the contributions made in this work.

5.2 COVID-19 data repository

One of our primary contributions is the curation of a COVID-19 data repository that we have made publicly available on GitHub. It is updated daily with new information. Specifically, we have compiled and cleaned a large corpus of hospital-level and county-level data from 20+ public sources to aid data science efforts to combat COVID-19.

Overview of the datasets on June 20, 2020

At the *hospital-level*, our dataset covers over 7000 US hospitals and over 30 features including the hospital's CMS certification number (a unique ID of each hospital used by Centers for

[6] `https://www.cdc.gov/mmwr/volumes/69/wr/mm6924e2.htm`

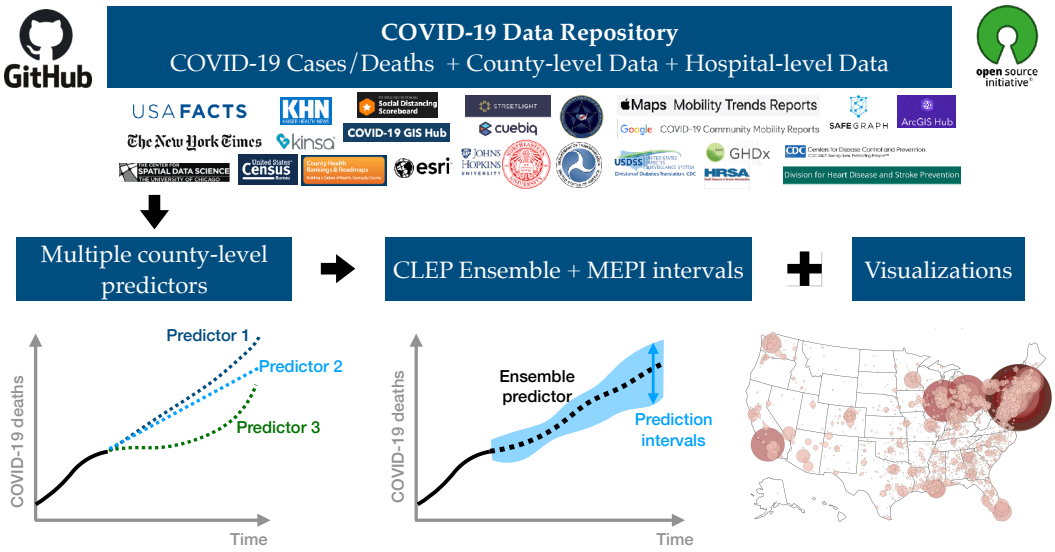


Figure 5.1: An overview of the paper. We curate an extensive data repository combining data from multiple data sources. We then build several predictors for county-level predictions of cumulative COVID-19 death counts, and develop an ensembling procedure (CLEP) and a prediction interval scheme (MEPI) for these predictions. Both CLEP and MEPI are generic machine learning methods and can be of independent interest (see Sections 5.3 and 5.4 respectively). All the data, and predictions are publicly available at GitHub repo (link in footnote). Visualizations are available at `https://covidseverity.com/` and `https://geodacenter.github.io/covid/map.html`, in collaboration with the Center for Spatial Data Science at the University of Chicago.

Medicare and Medicaid Services), the hospital's location, the number of ICU beds, the hospital type (e.g., short-term acute care and critical access), and numerous other hospital statistics.

There are more than 3,100 counties in the US. At the *county-level*, our repository includes data on

(i) daily recorded COVID-19-related case count and (recorded) death count by [88] and [92];

(ii) demographic features such as population distribution by age and population density;

(iii) socioeconomic factors including poverty levels, unemployment, education, and social vulnerability measures;

(iv) health resource availability such as the number of hospitals, ICU beds, and medical staff;

(v) health risk indicators including heart disease, chronic respiratory disease, smoking, obesity, and diabetes prevalence;

(vi) mobility measures such as the percent change in mobility from a pre-COVID-19 baseline; and

(vii) other relevant information such as county-level presidential election results from 2000 to 2016, county-level commute data that includes the number of workers in the commuting flow, and airline ticket survey data that includes origin, destination, and other itinerary details.

In total, there are over 8000 features in the county-level dataset. We provide a feature-level snapshot of the different types of data available in our repository, highlighting features in the county-level datasets in Table 5.1 and the hospital-level datasets in Table 5.2. Alternatively, in Tables 5.3 and 5.4, we provide an overview of the county-level and hospital-level data sources in our repository, respectively, organized by the dataset.

The full corpus of data, along with further details and extensive documentation, are available on GitHub. In particular, we have created a comprehensive data dictionary with the available data features, their descriptions, and source dataset for ease of navigation on our github. We have also provided a quick-start guide for accessing the unabridged county-level and hospital-level datasets with a single Python code line.

Datasets used by our predictors: In this paper, we focus on predicting the number of recorded COVID-19-related cumulative death counts in each county. For our analysis, we primarily use the county-level case and death reports provided by USAFacts from January 22, 2020 to June 20, 2020 (pulled on June 21, 2020) along with some county-level demographics and health data. We have marked these datasets with an asterisk (*) in Table 5.3. We discuss our prediction algorithms in detail in Sections 5.3, 5.4 and 5.5.

Other potential use-cases for our repository: The original intent of our data repository was indeed to facilitate our work with Response4Life and aid medical supply allocation efforts. However, with time the data repository has grown to encompass more resources and now supports investigations into a wider range of COVID-19 related problems. For instance, using the breadth of travel information in our repository, including (aggregated) air travel and work commute data, researchers can investigate the impact of both local and between-city travel patterns on the spread of COVID-19. Our repository also includes data on the prevalence of various COVID-19 health risk factors, including diabetes, heart disease, and chronic respiratory disease, which can be used to stratify counties. Furthermore, one can also potentially leverage socioeconomic and demographic information, as well as health resource data (e.g., number of ICU beds, medical staff) to gain a better understanding of the severity of the pandemic in a county. Stratification using these covariates is particularly crucial for assessing the COVID-19 status of rural communities, which are not directly comparable, both in terms of people and resources, to the larger cities and counties that have

received more attention.

Comparison with the repository collated by [53] at Johns Hopkins University: Note that similar but complementary county-level data was recently aggregated and released in another study [53]. Both our county-level repository and the repository in [53] include data on COVID-19 cases and deaths, demographics, socioeconomic information, education, and mobility, albeit some are from different sources. For example, the repository by [53] uses COVID-19 cases and deaths data from the John Hopkins University CSSE COVID-19 dashboard by [28] whereas our data is pulled from [92] and [88]. The main difference, however, between the two repositories is that our data repository also includes data on COVID-19 health risk factors. Furthermore, while the repository in [53] provides additional datasets at the state-level, we provide additional datasets at the hospital-level (given our initial goal of helping the allocation of medical supplies to hospitals, in partnership with the non-profit Response4Life). While their data repository contains both overlapping and complementary information to our repository, a thorough dataset-by-dataset comparison is beyond the scope of this work for two reasons: (i) We learned about this repository towards the completion of our work, and (ii) we were unable to find detailed documentation of how the datasets in their repository were cleaned.

Data quality and bias

Before introducing our prediction algorithms, it is vital to discuss the quality and limitations of the available COVID-19 data. Many downstream analyses, including ours, rely on accurate COVID-19 infection data, including accurate case and death counts. In this subsection, we focus our discussion and evaluation on the data quality of the county-level COVID-19 case and death count data. We also conduct some preliminary exploratory data analysis (EDA) to shed light on the scale of bias and the possible directions of the biases in the data.

Though discussions on data quality issues and their possible consequences are relatively sparse in the existing literature, [5] discuss a variety of possible data biases in the context of estimating the case fatality ratio. They proposed a method that can theoretically account for two biases: time lag and imperfect reporting of deaths and recoveries. Unfortunately, it is hard to evaluate their method's performance since the actual death counts due to COVID-19 remain unknown. Moreover, some data biases (e.g., under-ascertainment of mild cases) for estimating the case fatality ratio do not affect estimation of future death counts. Nonetheless, many of the ideas we present with our EDA here in uncovering possible biases in the data are inspired by [5].

Imperfect reporting and attribution of deaths due to COVID-19: Numerous news articles have suggested that the official US COVID-19 death count is an underestimate [96].

Description of County-level Features	Data Source(s)
COVID-19 Cases/Deaths	
Daily # of COVID-19-related recorded cases by US county	[92]; [88]
Daily # of COVID-19-related deaths by US county	[92]; [88]
Demographics	
Population estimate by county (2018)	[41] (Area Health Resources Files)
Census population by county (2010)	[41] (Area Health Resources Files)
Age 65+ population estimate by county (2017)	[41] (Area Health Resources Files)
Median age by county (2010)	[41] (Area Health Resources Files)
Population density per square mile by county (2010)	[41] (Area Health Resources Files)
Socioeconomic Factors	
% uninsured by county (2017)	[22]
High school graduation rate by county (2016-17)	[22]
Unemployment rate by county (2018)	[22]
% with severe housing problems in each county (2012-16)	[22]
Poverty rate by county (2018)	[90]
Median household income by county (2018)	[90]
Social vulnerability index for each county	[14] (Social Vulnerability Index)
Health Resources Availability	
# of hospitals in each county	[51]
# of ICU beds in each county	[51]
# of full-time hospital employees in each county (2017)	[41] (Area Health Resources Files)
# of MDs in each county (2017)	[41] (Area Health Resources Files)
Health Risk Factors	
Heart disease mortality rate by county (2014-16)	[13] (Interactive Atlas of Heart Disease and Stroke)
Stroke mortality rate by county (2014-16)	[13] (Interactive Atlas of Heart Disease and Stroke)
Diabetes prevalence by county (2016)	[15] (Diagnosed Diabetes Atlas)
Chronic respiratory disease mortality rate by county (2014)	[47]
% of smokers by county (2017)	[22]
% of adults with obesity by county (2016)	[22]
Crude mortality rate by county (2012-16)	[91]
Mobility	
Start date of stay at home order by county	[53]
% change in mobility at parks, workplaces, transits, groceries/pharmacies, residential, and retail/recreational areas	[38]

Table 5.1: A list of select relevant features from across all county-level datasets contained in our COVID-19 repository grouped by feature topic. See Table 5.3 for an overview of each of the individual county-level datasets.

Description of Hospital-level Features	Data Source(s)
CMS certification number	[18] (Case Mix Index File)
Case Mix Index	[18] (Case Mix Index File); [17] (Teaching Hospitals)
Hospital location (latitude and longitude)	[43]; [23]
# of ICU/staffed/licensed beds and beds utilization rate	[23]
Hospital type	[43]; [23]
Trauma Center Level	[43]
Hospital website and telephone number	[43]

Table 5.2: A list of select relevant features from across all hospital-level datasets contained in our COVID-19 repository. See Table 5.4 for an overview of each hospital-level dataset.

According to The New York Times[7], on April 5, the Council of State and Territorial Epidemiologists advised states to include both the confirmed cases based on laboratory testing, and probable cases—using specific criteria for symptoms and exposure. The Centers for Disease Control adopted these definitions, and national CDC data began including confirmed and probable cases on April 14. The infection data included in our data repository (USAFacts and NY Times) contains both the probable death and the confirmed deaths beginning April 14. Although the probable death counts address imperfect reporting and attribution, there is still the possibility of under-reporting in some counties (for the period up to June 21, 2020). However, the magnitude of the possible under-reporting and the counties where under-reporting occurred is unclear. Going forward, we use the term recorded death counts and recorded case counts to reflect that the recorded counts are based on both confirmed and probable deaths and cases.

Inconsistency across different data sources: There exist multiple sources of COVID-19 death counts in the US. In our data repository, we include data from [92] and data from [88]. According to USAFacts and the NY Times websites, they both collect data from state and local agencies or health departments and manually curate the data. However, these websites do not scrape data from those sources at the same time. While USAFacts states that "they mostly collect data in the evening (Pacific Time)", NY Times mentions they update data throughout the day. Furthermore, while there are some discussions on how they collect and process the data on their websites, the specific data curation rules are not shared publicly. Possibly due to different scrapping times and curation rules, there are a few discrepancies in their case and death counts. In Figure 5.2(a), we plot the absolute

[7] https://www.nytimes.com/interactive/2020/06/19/us/us-coronavirus-covid-death-toll.html

County-level Dataset	Description
COVID-19 Cases/Deaths Data	
[92]*†	Daily cumulative number of reported COVID-19-related death and case counts by US county, dating back to Jan. 22, 2020
[88]†	Similar to the USAFacts dataset, but includes aggregated death counts in New York City without county breakdowns
Demographics and Socioeconomic Factors	
[41] (Area Health Resources Files)*	Includes data on health facilities, professions, resource scarcity, economic activity, and socioeconomic factors (2018-2019)
[22]*	Estimates of various health behaviors and socioeconomic factors (e.g., unemployment, education)
[14] (Social Vulnerability Index)	Reports the CDC's measure of social vulnerability from 2018
[90]	Poverty estimates and median household income for each county
Health Resources Availability	
[41] (Area Health Resources Files)*	Includes data on health facilities, professions, resource scarcity, economic activity, and socioeconomic factors (2018-2019)
[42] (Health Professional Shortage Areas)	Provides data on areas having shortages of primary care, as designated by the Health Resources & Services Administration
[51]*	# of hospitals, hospital employees, and ICU beds in each county
Health Risk Factors	
[22]*	Estimates of various socioeconomic factors and health behaviors (e.g., % of adult smokers, % of adults with obesity)
[13] (Interactive Atlas of Heart Disease and Stroke)*	Estimated heart disease and stroke death rate per 100,000 (all ages, all races/ethnicities, both genders, 2014-2016)
[15] (Diagnosed Diabetes Atlas)*	Estimated percentage of people who have been diagnosed with diabetes per county (2016)
[47]*	Estimated mortality rates of chronic respiratory diseases (1980-2014)
[16] (Chronic Conditions)	Prevalence of 21 chronic conditions based upon CMS administrative enrollment and claims data for Medicare beneficiaries
[91]	Overall mortality rates (2012-2016) for each county from the National Center for Health Statistics
Mobility	
[53] (JHU Date of Interventions)	Dates that counties (or states governing them) took measures to mitigate the spread by restricting gatherings
[38] (Google Community Mobility Reports)†	Reports relative movement trends over time by geography and across different categories of places (e.g., retail/recreation, groceries/pharmacies)

Hospital-level Dataset	Description
[43]	Includes number of ICU beds, and location for US hospitals
[23]	Provides data on number of licensed beds, staffed beds, ICU beds, and the bed utilization rate for hospitals in the US
[18] (Case Mix Index File)	Reports the Case Mix Index (CMI) for each hospital
[17] (Teaching Hospitals)	Lists teaching hospitals along with address (2020)

Table 5.4: A list of hospital-level datasets contained within in our COVID-19 repository. Currently, all hospital-level sources are static. See Table 5.2 for an overview of select features from these hospital-level datasets.

difference in death counts from the two datasets for each county. In Figure 5.2(b), we plot the number of counties whose recorded COVID-19 deaths on a given day differ by more than 5. The proportion of counties with observably different death-counts (difference > 5) is in general small ($< 1\%$), although sometimes the differences are quite significant (>100). This fact is further highlighted in Figure 5.2(c), where we plot the histogram of these differences across all counties and all days. We observe that a majority of these differences (4737/7281) are equal to 1. Since two datasets are curated under different rules (which are unknown to us), it is not obvious how to combine them or assess their validity. For our analysis, we choose to use the USAFacts COVID-19 deaths data as they provide county-level death counts for New York City while the NY Times data aggregates the death counts over the five boroughs in that region.

Weekday patterns: The recorded case counts and death counts have a significant weekly pattern in both the USAFacts and NY Times data; such a pattern can possibly be attributed to the reporting delays as discussed in [5]. We show the total number of deaths recorded on each day of the week in the USAFacts data in Figure 5.3(a). The total number of deaths on Monday and Sunday is significantly lower than that for any other day. We try to account for these weekly patterns in our prediction methods later in Appendix .1.

Historical data revision: We observed that some of the historical infection data was revised after initially being recorded. According to USAFacts, these revisions are typically due to earlier mistakes from local agencies which revised their previously recorded death counts. Note that these data revisions are not related to the probable deaths as we discussed earlier and therefore we regard this phenomenon as a distinct source of bias. This kind of revision is not common: until June 21, we observe that only 2.1% of counties across the U.S. had one or more historical revisions. Figure 5.3(b) shows a histogram of the amount of time

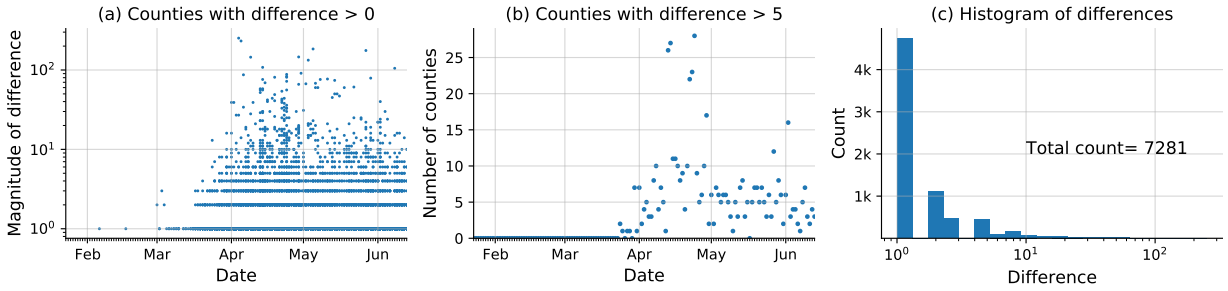


Figure 5.2: Plots illustrating the differences in county-wise recorded daily death counts between the USAFacts and NY Times datasets. In panel **(a)**, we plot the magnitude of difference in death counts for each county as a function of time, where one dot represents a particular county on a particular day. We notice that while there are counties where the discrepancy can be larger than 100 deaths recorded in a day, the majority of the discrepancies are not large. The large discrepancies are possibly due to different data curation protocols used by USAFacts and NY Times. In panel **(b)**, we plot the number of counties that have a discrepancy of more than 5 on any given day between the two datasets. We notice that on any given day, no more than 30 counties, i.e., $< 1\%$ of the more than 3,100 counties, have a difference larger than 5 between the two datasets. In panel **(c)**, we plot the histogram of number of counties (counted separately for each day) that have a certain difference in the daily death counts between the two datasets. We notice that for an majority fraction (4736/7281), the death counts between the two datasets differ by 1.

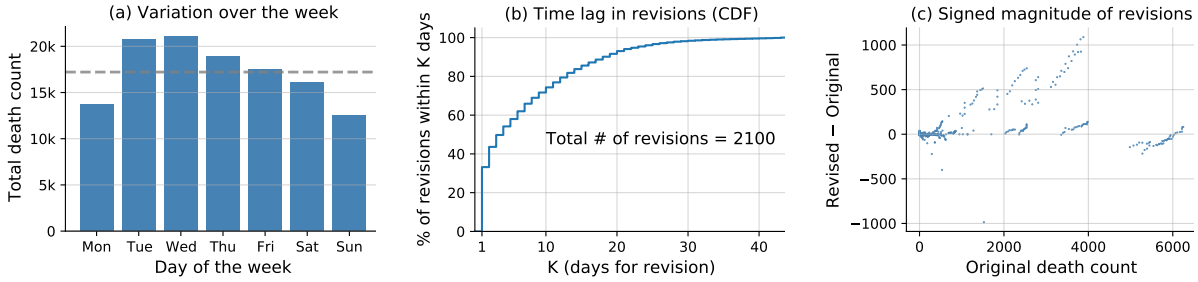


Figure 5.3: Exploratory data analysis (EDA) plots for identifying potential biases in USAFacts data. **(a)** Variation of recorded death counts over different days of the week. We observe that the total death counts (for March-June) shows a trend across different weekdays, and the total count is significantly smaller for Sunday and Monday. In panel **(b)**, we plot the cumulative distribution for the fraction of revisions made versus the number of days for the revisions. In panel **(c)**, we present the signed magnitude of the change, i.e., the "revised count minus the original count" against the original count for each revision.

from the initial record to the revision. It can be seen that almost half of the changes happen within 2 days from the day data was initially recorded. Figure 5.3(c) shows the signed magnitude of the change in death count that results from the revisions versus the initial recorded death counts. Note that there are a few stripes (consecutive upward revisions) in the plot. Each stripe corresponds to the revision of a particular county on different dates. Since the reported data is cumulative death counts, when the deaths from a few days ago get revised, all the data after that day until the day when the revision is made also get revised accordingly, thereby explaining the short stripy trends.

However, only 582 out of 2100 revisions (around 27%) have absolute magnitude > 2 deaths, and 354 of these 582 revisions (around 67%) are in the positive direction (i.e., more deaths than initially recorded). Furthermore, amongst the 61 revisions with an absolute magnitude larger than 200, almost all of them (57/61) lead to an increase in the number of recorded deaths. The four most significant downward revisions, i.e., the points with large negative "revised-original count" in Figure 5.3(c), correspond to counties in the Washington State. This finding can be corroborated by the media news that Washington State admitted errors in reporting the death counts, and subsequently lowered these counts in the revisions. It is natural for our predictions to vary if the training data (for a fixed period) varies with time, i.e., when the COVID-19 counts are adjusted for a backdate. Most of these revisions are minor, in which case the general performance of our predictors does not change significantly. However, when the revisions are a significant uptick, the predictions can become unstable for a few days (depending on the uptick, and the prediction-horizon). See Section 5.5 and 5.5 for further discussion on these biases. In this paper, we use the initial infection data available on June 21, 2020 to evaluate our algorithm performance, i.e., we do not use data that was revised after June 21. Nonetheless, we caution the reader to keep the following fact in mind while interpreting results from our work as well as other related COVID-19 studies: The recorded death counts themselves are an under-estimate and the consequent bias is hard to adjust for due to the lack of ground truth.

5.3 Predictors for forecasting short-term death counts

Figure 5.4 provides a visualization of the COVID-19 outbreak across the United States. We plot (a) the cumulative recorded death counts due to COVID-19 up to June 20, and (b) the new death counts from June 1 to June 20, 2020. Each bubble denotes a county-level count, a darker and larger bubble denotes a higher death count, and the absence of a bubble denotes that the count is zero. Panel (a) captures the extent of the outbreak in a region, while (b) captures the recent trends in the outbreak. The color scale differs between the two plots to better illustrate the respective counts in each plot, but the size scales are held constant between the two plots to help provide a comparison between the extent and recent trends of COVID-19. Overall, Figure 5.4 clearly shows that the COVID-19 outbreak in the United States is incredibly dynamic both in time and across different regions. The worst-affected regions include the states of New York, New Jersey, Massachusetts, Michigan,

Illinois, Florida, Louisiana, Georgia, Washington, and California. Moreover, most of these areas continue to face a substantial COVID-19 burden in the first two-thirds of June.

We develop several different statistical and machine learning prediction algorithms to capture the dynamic behavior of COVID-19 death counts. Since each prediction algorithm captures slightly different trends in the data, we also develop various weighted combinations of these prediction algorithms. The five prediction algorithms or predictors for cumulative recorded death counts that we devise in this paper are as follows:

1. **A separate-county exponential predictor (the "separate" predictors)**: a series of predictors built for predicting cumulative death counts for each county using only past death counts from that county.

2. **A separate-county linear predictor (the "linear" predictor)**: a predictor similar to the separate county exponential predictors, but uses a simple linear format, rather than the exponential format.

3. **A shared-county exponential predictor (the "shared" predictor)**: a single predictor built using death counts from all counties, used to predict death counts for individual counties.

4. **An expanded shared-county exponential predictor (the "expanded shared" predictor)**: a predictor similar to the shared-county exponential predictor, which also includes COVID-19 case numbers and neighboring county cases and deaths as predictive features.

5. **A demographics shared-county exponential predictor (the "demographics shared" predictor)**: a predictor also similar to the shared-county exponential predictor, but which also includes various county demographic and health-related predictive features.

An overview of these predictors is presented in Table 5.5. We use the python package statsmodels [81] to train all the five predictors: ordinary least squares for predictor (2), and Poisson regression for predictors (1), (3), (4) and (5) (where the set of features for each predictor is different).[8] To combine the different trends captured by each of these

[8]We use the default parameters in the Python statsmodels package (version 0.11.1) while training our predictors. For predictors (1) and (2), *glm.fit* was used which always converged. For predictors (3)-(5), we first tried *glm.fit_ regularized* since we experimented with the use of explicit ℓ_1 and ℓ_2 regularization in the beginning. While we ended up using reguralization for only predictor (5) and not for (3) and (4), it turns out that default settings (algorithm and stopping criterion) in the functions *glm.fit* and *glm.fit_ regularized* are different leading to different implicit regularizations even in predictors (3) and (4), and consequently different performance. We still chose to use *glm.fit_ regularized* for fitting predictors (3) and (4) since it led to better performance for predictor (4) (which forms the basis of our best performing predictor CLEP). We also note that the function *glm.fit_ regularized*, in fact, calls another function *fit_ elasticnet* which uses block coordinate descent (BCD) by default to solve a generalized linear model. The default value for the maximum number of iterations of BCD (max_iter) is set to be 50, which, in some cases, resulted in early stopping (a form of implicit regularization) before the iterative algorithm converges.

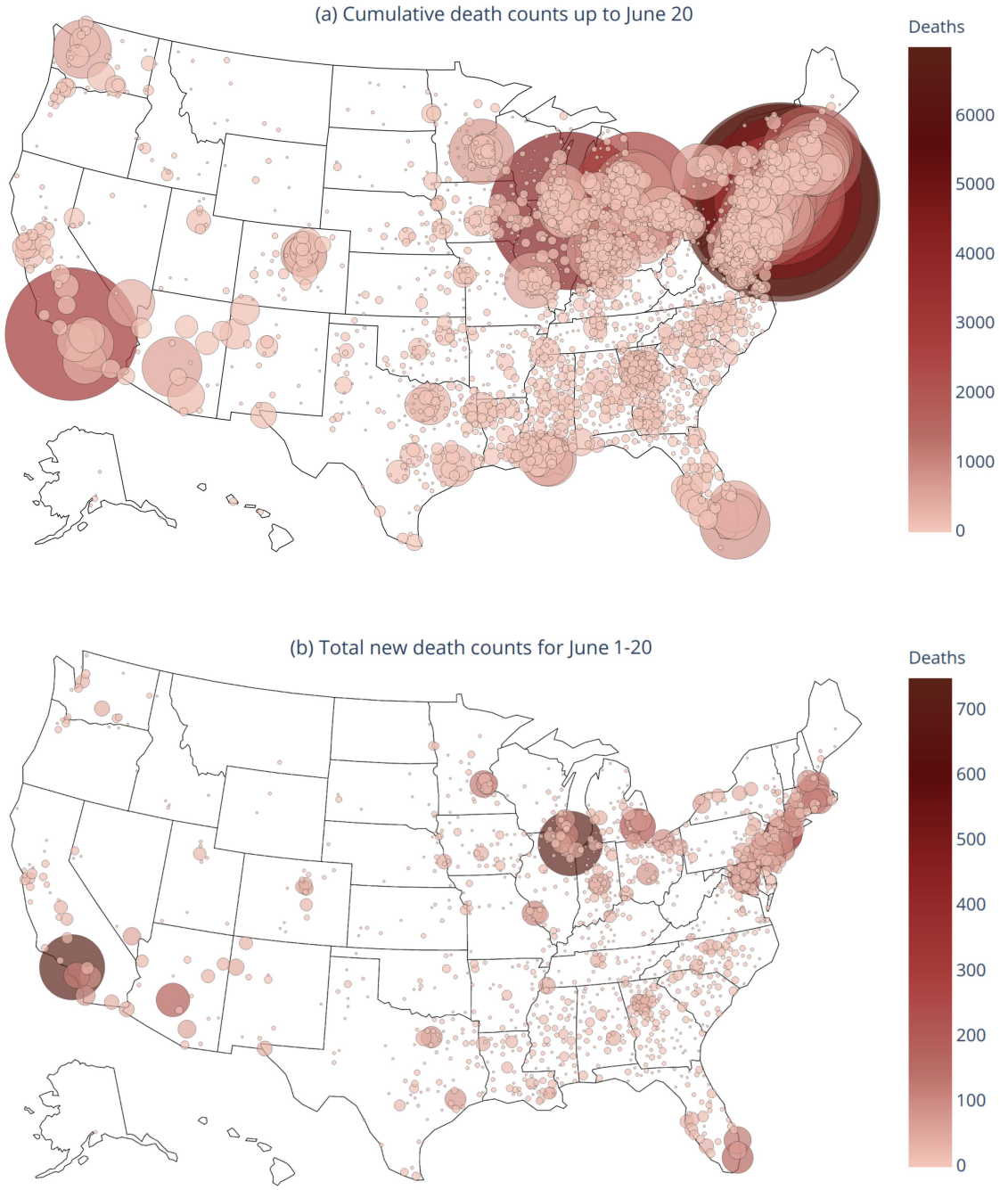


Figure 5.4: Visualization of the COVID-19 outbreak in the US. We depict the cumulative recorded death counts up to June 20 in panel **(a)** and newly recorded death counts for the period June 1-20 in panel **(b)**. Each bubble denotes the death count for a county (the absence of a bubble denotes a zero count). The bubble size (area) is proportional to the death counts in the region. The two panels' bubble sizes are on the same scale, but the color scale is different as shown respectively on each plot.

predictors, we also fit various combinations of them, which we refer to as Combined Linear and Exponential Predictors (CLEP). CLEP produces a weighted average of the predictions from the individual predictors, where we borrow the weighting scheme from prior work [80]. In this weighting scheme, a higher weight is given to those predictors with more accurate predictions, especially on recent time points. We find that the CLEP that combines only the *linear predictor* and the *expanded shared predictor* consistently has the best predictive performance when compared to the individual predictors and the CLEP that combines all five predictors. (We did not try all possible combinations to avoid over-fitting; also see Table 5.9).

For the rest of this section, we expand upon the individual predictor models and the weighting procedure for the CLEP ensembles. In addition, Appendix .1 contains results on variants of the two best single predictors (linear and expanded shared), which include features for social-distancing and features that account for the under-reporting of deaths on Sunday and Monday (as observed in Figure 5.3(a)). These additional features did not lead to better performance.

We note that although in this paper, we discuss our algorithms for predicting cumulative recorded death counts, the methods can be more generally applied to predict other quantities of interest, e.g., case counts or new death counts for each day. Moreover, the combination scheme used for combining different predictors can be of independent interest in developing ensembling schemes with generic machine learning methods.

Predictor name	Type	Fit separately to each county?	Fit jointly to all counties?	Use neighboring counties?	Use demographics?
Separate	Exponential	✓			
Linear	Linear	✓			
Shared	Exponential		✓		
Expanded shared	Exponential		✓	✓	
Demographics shared	Exponential		✓		✓

Table 5.5: Overview of the 5 predictors used here. The best model is a combination of the linear predictor and the expanded shared predictor (see Section 5.3).

The separate-county exponential predictors (the "separate" predictors)

The separate-county exponential predictor aims to capture the observed exponential growth of COVID-19 deaths [65]. We approximate an exponential curve for death count separately for each county using the most recent 5 days of data from that county. These predictors

have the following form:

$$\widehat{\mathrm{E}}[\mathrm{deaths}^c_{t+1}|t] = \exp\big(\beta^c_0 + \beta^c_1(t+1)\big), \tag{5.1}$$

where $\widehat{\mathrm{E}}[\mathrm{deaths}^c_{t+1}|t]$ denotes the (fitted) cumulative death count by the end of day $t+1$ for county c, and it is trained on the data until day t, and computed on the morning of day $t+1$. Note that we use $t+1$ on the RHS of equation (5.1) just for notational exposition, and in practice we just use $\beta^c_0 + \beta^c_1 t$ in the exponent in our code.

Here we fit a separate predictor for each county, and the coefficients β^c_0 and β^c_1 for each county c are fit using maximum likelihood estimation under a Poisson generalized linear model (GLM) with t as the independent variable and deaths_t as the observed variable. In simple words, on the morning of day $t+1$, the coefficients are estimated using the cumulative recorded death counts for day $\{t, t-1, t-2, t-3, t-4\}$. And to predict k-days-ahead cumulative death count on the morning of day $t+1$—denoted by $\widehat{\mathrm{E}}[\mathrm{deaths}^c_{t+k}|t]$—we simply replace $t+1$ with $t+k$ on the RHS of equation (5.1). Note that although the prediction $\widehat{\mathrm{E}}(\mathrm{deaths}^c_{t+1}|t)$ is being made on day $t+1$, we call it 1-day-ahead prediction since it is made in the morning of day $t+1$ using the data for up to day t. Moreover, the recorded count deaths^c_{t+1} is reported only late in the night of day $t+1$ or early morning of the next day $t+2$.

If the first death in a county occurred less than 5 days prior to fitting the predictor, only the days from the first death were used for the fit. If there is less than three days' worth of data or the cumulative deaths remain constant in the past days, we simply use the most recent deaths as the predicted future value. We also fit exponential predictors to the full time-series (as opposed to just the most recent 5 days) of available data for each county. However, due to the rapidly shifting trends, these performed worse than our 5-day predictors. We also found that predictors fit using 6 days of data yielded similar results to predictors fit using 5 days of data, and using 4 days of data performed slightly worse.

To handle possible overdispersion of data (when the variance is larger than the mean), we also explored estimating $\{\beta^c_0, \beta^c_1\}$ by fitting a negative binomial regression model (in place of Poisson GLM) with inverse-scale parameter taking values in $\{0.05, 0.15, 1\}$.[9] However, we found that this approach yields a larger mean absolute error than the Poisson GLM for counties with more than 10 deaths.

The separate-county linear predictor (the "separate linear" predictor)

The separate linear predictor aims to capture linear growth, based on the most recent 4 days of data in each county. In the early stages of tuning, we tried using 5 and 7 days of data, and obtained worse performance (also see the discussion in Appendix .1). The motivation for the linear model is that some counties are exhibiting sub-exponential growth. For these

[9]These values were tried keeping in mind the typical permissible range of [0.02, 2] as per the documentation at `https://www.statsmodels.org/stable/generated/statsmodels.genmod.families.family.NegativeBinomial.html`

counties, the exponential predictors introduced in the previous section may not be a good fit to the data. The separate linear predictors are given by

$$\widehat{\mathrm{E}}[\mathrm{deaths}^c_{t+1}|t] = \beta^c_0 + \beta^c_1(t+1), \tag{5.2}$$

where we fit the coefficients β^c_0 and β^c_1 via ordinary least squares using the cumulative death count for county c for most recent 4 days. Like equation (5.1), we use $t+1$ on the RHS simply for notational exposition. Put simply, on the morning of day $t+1$, the coefficients $\{\beta^c_0, \beta^c_1\}$ are estimated using the death counts for day $t, t-1, t-2, t-3$. To predict k-days-ahead, i.e., predict cumulative death counts by the end of day $t+k$ on the morning of day $t+1$ (in our notation, $\widehat{\mathrm{E}}[\mathrm{deaths}^c_{t+k}|t]$), we simply replace $t+1$ by $t+k$ on the RHS of equation (5.2).

The shared-county exponential predictor (the "shared" predictor)

To incorporate additional data into our predictions, we fit a predictor that combines data across different counties. Rather than producing a separate predictor model for each county (as in the separate predictor approach above), we instead produce a single shared predictor that pools information from counties across the nation. The shared predictor is then used to predict future deaths in the individual counties. These changes allows us to leverage the early-stage trends from counties that are now much further along in the pandemic trajectory to inform the predictions for other current earlier-stage counties.

The data underlying the shared predictor is slightly different from the separate county predictors. For each county, instead of only including the most recent 5 days, we include all days after the third death in the county. (In the earlier stages of tuning, we also tried including the counties after first and fifth death, and then selected the choice of third death due to better performance.) Thus the data from many of the counties extend substantially further back than 5 days, and for each county, $t = 0$ is the day on which the third death occurred. Instead of basing the exponential predictor prediction on time $t+1$ (as was the case for the separate predictors above), we base the prediction on the logarithm of the previous day's death count. This choice makes the counties comparable since the outbreaks began at different time points in each county. The shared predictor is given as follows:

$$\widehat{\mathrm{E}}[\mathrm{deaths}^c_{t+1}|t] = \exp\Big(\beta_0 + \beta_1 \log(\mathrm{deaths}^c_t + 1)\Big), \tag{5.3}$$

where $\widehat{\mathrm{E}}[\mathrm{deaths}^c_{t+1}|t]$ denotes the (fitted) cumulative death count by the end of day $t+1$ for a county c, and deaths^c_t denotes the recorded cumulative death count for that county by the end of day t. The coefficients β_0 and β_1 are shared across all counties and fitted by maximizing the log-likelihood corresponding to Poisson GLM (like that in the separate county predictor given by equation (5.1)). We normalize the feature matrix to have zero mean and unit variance before fitting the coefficients. To predict k-days-ahead cumulative death count $\widehat{\mathrm{E}}[\mathrm{deaths}^c_{t+k}|t]$, we first obtain the estimate $\widehat{\mathrm{E}}[\mathrm{deaths}^c_{t+1}|t]$ using equation (5.3). Next, we plug-in $\log(\widehat{\mathrm{E}}[\mathrm{deaths}^c_{t+j}|t] + 1)$ (after normalizing across all counties) on the RHS

of equation (5.3) to compute $\widehat{\mathrm{E}}[\mathrm{deaths}^c_{t+j+1}|t]$ in a sequential manner for $j = 1, \ldots, k-1$, and finally obtain $\widehat{\mathrm{E}}[\mathrm{deaths}^c_{t+k}|t]$ (k-day-ahead prediction computed on the morning of day $t+1$).

The expanded shared exponential predictor (the "expanded shared" predictor)

Next, we expand the shared county exponential predictor to include other COVID-19 dynamic (time-series) features. In particular, we include the number of recorded *cases* in the county, as this may give an additional indication to the severity of an outbreak. We also include the total sum of cumulative death (and case) counts in the *neighboring* counties. Let cases^c_t, $\mathrm{neigh\_deaths}^c_t$, $\mathrm{neigh\_cases}^c_t$ respectively denote the (recorded) cumulative case count in the county c at the end of day t, the total sum of cumulative death counts across all its neighboring counties at the end of day t, and the total sum of cumulative recorded case counts across all its neighboring counties at the end of day t. Then our (expanded) predictor to predict the number of recorded cumulative deaths k days into the future is given by

$$\widehat{\mathrm{E}}[\mathrm{deaths}^c_{t+1}|t] = \exp\Bigg(\beta_0 + \beta_1 \log(\mathrm{deaths}^c_t + 1) + \beta_2 \log(\mathrm{cases}^c_{t-k+1} + 1) + \beta_3 \log(\mathrm{neigh\_deaths}^c_{t-k+1} + 1) + \beta_4 \log(\mathrm{neigh\_cases}^c_{t-k+1} + 1)\Bigg), \tag{5.4}$$

where the coefficients $\{\beta_i\}_{i=0}^4$ are shared across all counties and are fitted using the Poisson GLM after normalization of each feature (in the exponent) to have zero mean and unit variance. When *fitting* the predictor on the morning of day $t+1$, we use the death counts for the county up to the end of day t. However, we only use the new features (cases in the current county, cases in neighboring counties, and deaths in neighboring counties) up to the end of day $t-k+1$, since when predicting $\widehat{\mathrm{E}}[\mathrm{deaths}^c_{t+k}|t]$ these covariates would only be available up to day t, i.e., k days before. Moreover, we normalize the feature matrix to have zero mean and unit variance before fitting the predictor. While *predicting* the death count for a given county k days into the future (i.e, the cumulative death count by the end of day $t+k$), we iteratively use the daily sequential predictions for the death counts for that county, and use the information for the other features only up to time t (the time up to which we have data available). More precisely, first we estimate $\widehat{\mathrm{E}}[\mathrm{deaths}^c_{t+1}|t]$ by plugging in the normalized features $\log(\mathrm{deaths}^c_t)$, $\log(\mathrm{cases}^c_{t-k+1})$, $\log(\mathrm{neigh\_deaths}^c_{t-k+1})$, and $\log(\mathrm{neigh\_cases}^c_{t-k+1})$ in equation (5.4), where the normalization is done across counties so that each feature has zero mean and unit variance. Then, for $j = 1, 2, \ldots, k-1$, we recursively plug-in $\log(\widehat{\mathrm{E}}[\mathrm{deaths}^c_{t+j}|t])$, $\log(\mathrm{cases}^c_{t-k+j+1})$, $\log(\mathrm{neigh\_deaths}^c_{t-k+j+1})$, $\log(\mathrm{neigh\_cases}^c_{t-k+j+1}))$ in equation (5.4) (again after normalizing each of these features) to compute $\widehat{\mathrm{E}}[\mathrm{deaths}^c_{t+j+1}|t]$, and finally compute $\widehat{\mathrm{E}}[\mathrm{deaths}^c_{t+k}|t]$ for k-day-ahead prediction made with data until day t. It

may be possible to jointly predict the new features along with the number of deaths, but we leave building such a predictor for future work. As before, the predictor is fitted by including all days after the third death in each county.

The demographics shared exponential predictor (the "demographics shared" predictor)

The demographics shared county exponential predictor is again very similar to the shared exponential predictor. However, it includes several static county demographic and healthcare-related features to address the fact that some counties will be affected more severely than others, for instance, due to (a) their population makeup, e.g., older populations are likely to experience a higher death rate than younger populations, (b) their hospital preparedness, e.g., if a county has very few ICU beds relative to their population, they might experience a higher death rate since the number of ICU beds is correlated strongly (0.96) with the number of ventilators [78], and (c) their population health, e.g., age, smoking history, diabetes, and cardiovascular disease are all considered to be likely risk factors for acute COVID-19 infection [40, 75, 39, 37, 101].

For a county c, given a set of demographic and healthcare-related features $d_1^c, \ldots, d_m^c$ (such as median age, population density, or number of ICU beds), the demographics shared predictor is given by

$$\widehat{\mathrm{E}}[\mathrm{deaths}_{t+1}^c | t] = \exp\left(\beta_0 + \beta_1 \log(\mathrm{deaths}_t^c + 1) + \beta_{d_1} d_1^c + \cdots + \beta_{d_m} d_m^c\right). \tag{5.5}$$

Here the coefficients $\{\beta_0, \beta_1, \beta_{d_1}, \ldots, \beta_{d_m}\}$ are shared across all counties, and are fitted by maximizing the log-likelihood of the corresponding Poisson generalized linear model, where we include all the observations since the third death in each county. Moreover, we also normalize the feature matrix to have zero mean and unit variance before fitting the coefficients. The features we choose fall into three categories:

1. County density and size: population density per square mile (2010), population estimate (2018)

2. County healthcare resources: number of hospitals (2018-2019), number of ICU beds (2018-2019)

3. County health demographics: median age (2010), percentage of the population who are smokers (2017), percentage of the population with diabetes (2016), deaths due to heart diseases per 100,000 (2014-2016).

The k-day-ahead predictions for this predictor are obtained in a very similar manner to the shared predictor (5.3): We first obtain the estimate $\widehat{\mathrm{E}}[\mathrm{deaths}_{t+1}^c | t]$ using equation (5.5) and then, sequentially plug-in $\log(\widehat{\mathrm{E}}[\mathrm{deaths}_{t+j}^c | t] + 1)$ on the RHS of the equation (5.5)

(after normalization to obtain zero mean and unit variance) to compute $\widehat{\mathrm{E}}[\mathrm{deaths}^c_{t+j+1}|t]$ in a sequential manner for $j = 1, \ldots, k-1$. We found that regularization was quite helpful in addressing overfitting in this predictor and found that ℓ_1-penalized Poisson regression with a penalty of 0.5 performed the best.

The combined predictors: CLEP

Finally, we consider various combinations of the five predictors we have introduced above using an ensemble approach similar to that described in [80]. Specifically, we use the recent predictive performance (e.g., over the last week) of different predictors to guide an adaptive tuning of the corresponding weights in the ensemble. To simplify notation, let us denote the predictions for cumulative death count by the end of day $t+k$—where the prediction is made on the morning of day $t+1$—by $\{\widehat{y}^m_{t+k}\}$ with $m = 1, \ldots, M$ denoting the index of various linear and exponential predictors.[10] Then, their Combined Linear and Exponential Predictor (CLEP) is given by

$$\widehat{y}^{\mathrm{CLEP}}_{t+k} = \sum_{m=1}^{M} w^m_{t+1} \widehat{y}^{\mathrm{m}}_{t+k}. \tag{5.6}$$

Here the weight, w^m_{t+1}—used for combining the predictions made on the morning of day $t+1$—for predictor m, is computed according to the recent performance as follows:

$$w^m_{t+1} \propto \exp\left(-0.5 \sum_{i=t-6}^{t} (0.5)^{t-i} \left|\sqrt{\widehat{y}^m_i} - \sqrt{y_i}\right|\right), \tag{5.7}$$

where $\widehat{y}^m_i$ is the 3-day-ahead prediction from the predictor m trained on data up to time $i-3$ (and computed on the morning of day $i-2$). In addition, the weights are normalized so that $\sum_{m=1}^{M} w^m_{t+1} = 1$ for each $t+1$. The weights $\{w^m_{t+1}, m = 1, \ldots, M\}$ are computed separately for each county. We now turn to our discussion on the general combination scheme that leads to the equation (5.7) with a certain choice of hyperparameters (and how those hyperparameters were chosen).

The weights in equation (5.7) are based on the general ensemble weighting format introduced in [80]. This general format is given by

$$w^m_{t+1} \propto \exp\left(-c(1-\mu) \sum_{i=t_0}^{t} \mu^{t-i} \ell(\widehat{y}^m_i, y_i)\right), \tag{5.8}$$

[10] Our predictions are released around 11:30 AM Pacific Time each day, both on GitHub and our website (`https://covidseverity.com`). The released predictions on day $t+1$ include the county-wise predictions for cumulative death counts by the end of day $t+1$ itself. To summarize, 1-day-ahead prediction for day $t+1$ is denoted by $\widehat{\mathrm{E}}[\mathrm{deaths}^c_{t+1}|t]$ earlier is now written simply as $\widehat{y}_{t+1}$. Similarly, the k-day-ahead prediction $\widehat{\mathrm{E}}[\mathrm{deaths}^c_{t+k}|t]$ is denoted by $\widehat{y}_{t+k}$ in the simplified (and slightly abused) notation.

where $\mu \in (0,1)$ and $c > 0$ are tuning parameters, t_0 represents some past time point, and the weights are computed on the morning of day $t+1$. Since $\mu < 1$, the μ^{t-i} term represents the greater influence given to more recent predictive performance. For a given day i and predictor m, we measure the predictive performance of the predictor via the term $\ell(\widehat{y}_i^m, y_i)$, which denotes the loss incurred due to the discrepancy between its predicted number of deaths $\widehat{y}_i^m$ and the recorded death counts y_i. The hyperparameter c controls the relative importance of predictors depending on their recent predictive performance. Given the same recent predictive performance and μ, a larger c gives a higher weight to the better predictors. The hyperparameter t_0 denotes the number of recent days used for evaluating the predictor performance to influence the weight (5.8).

Choice of hyper-parameters: equation (5.7) corresponds to equation (5.8) with appropriate hyper-parameters, c, μ, t_0, and a specific loss format, ℓ. In [80], the authors used the loss function $\ell(\widehat{y}_i^m, y_i) = |\widehat{y}_i^m - y_i|$, since their errors roughly had a Laplacian distribution. In our case, we found that this loss function led to vanishing weights due to our error distribution's heavy-tailed nature. To help address this, we apply a square root to the predictions and the true values, and define $\ell(\widehat{y}_i^m, y_i) = |\sqrt{\widehat{y}_i^m} - \sqrt{y_i}|$. We found that this transformation improved performance in practice. We also considered a logarithmic transform instead of a square root (i.e., $\ell(\widehat{y}_i^m, y_i) = |\log(1+\widehat{y}_i^m) - \log(1+y_i)|$), but we found that using the logarithm yielded worse performance than using the square root transformation.[11]

To generate our predictions, we use the default value of c in [80] which is 1. However, we change the value of μ from the default of 0.9 to 0.5 for two reasons: (i) we found $\mu = 0.5$ yielded better empirical performance, and (ii) it ensured that performance more than a week ago had little influence over the predictor. We chose $t_0 = t - 6$ (i.e., we aggregate the predictions of the past week into the weight term), since we found that performance did not improve by extending further back than 7 days. Moreover, the information from more than a week effectively has a vanishing effect due to our choice of μ.

Finally, we found that for computing the weights in (5.7), using 3-day-ahead predictions in the loss terms $\ell(\widehat{y}_i^m, y_i)$ led to best predictive performance; i.e., these weights are computed based on the 3-day-ahead predictions generated over the course of a week starting with the predictor built 11 days ago (for predicting counts 8 days ago) up to the predictor built 4 days ago (for predicting yesterday's counts). In principle, the five hyper-parameters—c, μ, t_0, ℓ, and the choice of the prediction horizon to use for evaluating the loss ℓ—can be tuned jointly via a grid or randomized search. Nevertheless, to keep the computations tractable and our choices interpretable, we selected them sequentially. Moreover, a dynamic tuning of these hyper-parameters (over time) is left for future work (see last paragraph of Section 5.6).

[11] In our first submission on May 16, 2020 to arXiv, we had presented results for March 22 to May 10, 2020. During the preparation of manuscript, we had updated the transform to be the square-root transform in our code, but we did not update the CLEP equation in the paper, and erroneously reported that our CLEP weights used a logarithmic transform.

Ensuring monotonicity of predictions

In this work, we predict county-wise cumulative death count, which is a non-decreasing sequence. However, the predictors discussed in the previous sections need not provide monotonic estimates for different prediction horizon, i.e., $\widehat{\mathrm{E}}[\mathrm{deaths}^c_{t+k}|t]$ may decrease as k increases for a fixed t. Moreover, the predictors may estimate a future count that is smaller than the last observed cumulative death count, i.e., $\widehat{\mathrm{E}}[\mathrm{deaths}^c_{t+k}|t] < \mathrm{deaths}^c_t$. In our setting, expanded shared predictor exhibited both these issues. To avoid these pitfalls, we use post-hoc maxima adjustments for all the predictors as follows. First, we replace the estimate $\widehat{\mathrm{E}}[\mathrm{deaths}^c_{t+1}|t]$ by $\max\{\widehat{\mathrm{E}}[\mathrm{deaths}^c_{t+1}|t], \mathrm{deaths}^c_t\}$ to make sure that the predicted counts in the future are at least as large as the latest observed cumulative death counts. Next, we iteratively replace the estimate $\widehat{\mathrm{E}}[\mathrm{deaths}^c_{t+j}|t]$ by $\max\{\widehat{\mathrm{E}}[\mathrm{deaths}^c_{t+j}|t], \widehat{\mathrm{E}}[\mathrm{deaths}^c_{t+j-1}|t]\}$ for $j = 2, 3, \ldots 21$. Imposing these constraints for the individual predictors also ensures the monotonicity of predictions by the CLEP. Note that we use these monotonic predictions (after the maxima calculations) to determine the weights in equation (5.7).[12]

Note that even after imposing the previous monotonicity corrections, it is still possible that $\widehat{\mathrm{E}}[\mathrm{deaths}^c_{t+k}|t] > \widehat{\mathrm{E}}[\mathrm{deaths}^c_{t+k+1}|t+1]$ since the predictors are re-fitted over time. Hence, when plotted over time, k-day-ahead predictions, need not be monotonic with respect to t. For example, see the plots of 7-day-ahead predictions in Figure 5.11 and 14-day-ahead predictions in Figure 5.12.

5.4 Prediction intervals via conformal inference

Accurate assessment of the uncertainty of forecasts is necessary to help determine how much emphasis to put on them, for instance, when making policy decisions. As such, the next goal of the paper is to quantify the uncertainty of our predictions by creating prediction intervals. A common method to do so involves constructing (probabilistic) model-based confidence intervals, which rely heavily on the probabilistic assumptions made about the data. However, due to the highly dynamic nature of COVID-19, assumptions on the distribution of death and case rate are challenging to check. Moreover, such prediction intervals based on probability models are likely to be invalid when the underlying probability model does not hold to the desired extent. For instance, a recent study [60] reported that the 95% credible intervals for state-level daily mortality predicted by the initial IHME model [87], had a coverage of a mere 27% to 51% of recorded death counts over March 29 to April 2. The authors of the IHME model noted this behavior and have since updated their uncertainty intervals so that they now provide more than 95% coverage (where coverage is defined below in equation (5.12a)). However, while the previous releases of the intervals were based on asymptotic confidence intervals, the IHME authors have not precisely described the methodology for their more

[12] We report partial results up to 21-day-ahead predictions, and detailed results up to 14-day-ahead predictions in Section 5.5. In the first arXiv submission of this work on May 16, 2020, we had not implemented monotonicity of predictions. The monotonicity implementation improved the overall results both for predictions and prediction intervals.

recent intervals. In this section, we construct prediction intervals that attempt to avoid these pitfalls by taking into account the recent observed performance of our predictors; and later in Section 5.5, we show that these intervals obtain high empirical coverage while maintaining reasonable width.

Maximum-absolute-Error Prediction Interval (MEPI)

We now introduce a generic method to construct prediction intervals for sequential or time-series data. In particular, we build on the ideas from conformal inference [93] and make use of the past errors made by a predictor to estimate the uncertainty for its future predictions.

To construct prediction intervals for county-level cumulative death counts caused by COVID-19, we calculate the largest (normalized absolute) error for the death count predictions generated over the past 5 days for the county of interest and use this value (the "maximum absolute error") to create an interval surrounding the future (e.g., tomorrow's) prediction. We call this interval the Maximum absolute Error Prediction Interval (MEPI).

Let y_t be the actual recorded cumulative deaths by the end of day t, and $\widehat{y}_t$ denote the estimate for y_t made k days earlier (in our case on the morning of day $t-k+1$) by a prediction algorithm. We call $\widehat{y}_t$ the k-day-ahead prediction for day t. (Note that we suppress the dependence on county and prediction horizon k for brevity and ease of exposition; and the notation here is a slightly abused version of that used in Section 5.3. In particular, we have $\widehat{y}_t = \widehat{\mathrm{E}}[\mathrm{deaths}^c_t | t-k]$ for some fixed county c.) We define the normalized absolute error, Δ_t, of the prediction, $\widehat{y}_t$, to be

$$\Delta_t := \Big| \frac{y_t}{\max\{\widehat{y}_t, 1\}} - 1 \Big|. \tag{5.9}$$

We use the normalization so that y_t (when non-zero) is equal to either $\widehat{y}_t(1-\Delta_t)$ or $\widehat{y}_t(1+\Delta_t)$. This normalization addresses the fact that the counts are increasing over time, and thus the un-normalized errors, $|y_t - \widehat{y}_t|$, also tend to be increasing over time. The normalization ensures that the errors across time are comparable in magnitude, which is essential for the exchangeability of the errors (see Section 5.4).

To compute the k-day-ahead prediction interval for day $t+k$ on the morning of day $t+1$, we first compute the k-day-ahead prediction $\widehat{y}_{t+k}$ ($= \widehat{\mathrm{E}}[\mathrm{deaths}^c_{t+k} | t]$) using a CLEP. Next, we compute the normalized errors for the k-day-ahead predictions for the most recent 5 days $\Delta_t, \Delta_{t-1}, ..., \Delta_{t-4}$ (a window of 5 days was chosen to balance the trade-off between coverage and length, see Appendix .2 for more details). The largest of these normalized errors is then used to define the *maximum absolute error prediction intervals* (MEPI) for the k-day-ahead prediction as follows:

$$\widehat{\mathrm{PI}}_{t+k} := \left[\max\left\{ \widehat{y}_{t+k}(1-\Delta_{\max}), y_t \right\},\ \widehat{y}_{t+k}(1+\Delta_{\max}) \right], \tag{5.10a}$$

$$\text{where} \quad \Delta_{\max} := \max_{0 \le j \le 4} \Delta_{t-j}; \tag{5.10b}$$

where the lower bound for the interval involves a maximum operator to account for the fact that y_t is a cumulative count, and thereby non-decreasing. This maxima calculation ensures that the lower bound for the interval is not smaller than the last observed value.

For a general setting beyond increasing time-series, this maxima calculation can be dropped, and the MEPIs can be defined simply as

$$[\widehat{y}_{t+k}(1-\Delta_{\max}),\ \widehat{y}_{t+k}(1+\Delta_{\max})]. \tag{5.11}$$

In our case, we construct the MEPIs (5.10a) separately for each county for the cumulative death counts. We remind the reader that when constructing k-day-ahead MEPIs, the Δ_t defined in equation (5.9) is computed using k-day-ahead predictions (our notation does not highlight this fact), so that the maximum error $\Delta_{\max}$ would be typically different, say, for 7-day-ahead and 14-day-ahead predictions.

Evaluation metrics

For any time-series setting, stationary or otherwise, the quality of a prediction interval can be assessed in terms of the percentage of time—over a sufficiently long period—that the prediction interval covers the observed value of the target of interest (e.g., recorded cumulative death counts as in this paper). A good prediction interval should both contain the true value most of the time, i.e., have a good coverage, and have a reasonable width or length.[13] Indeed, one can trivially create very wide prediction intervals that would always contain the target of interest. We thus consider two metrics to measure the performance of prediction intervals: *coverage* and *normalized length*.

Let y_t denote a positive real-valued time-series of interest, which in this case is the target variable: COVID-19 deaths (t denotes the time index). Let $\{\widehat{\mathrm{PI}}_t = [a_t, b_t]\}$ denote the sequence of prediction intervals produced by an algorithm. The coverage of this prediction interval, Coverage($\mathcal{T}$), over a specified period, $\mathcal{T}$, corresponds to the fraction of days in this period for which the prediction interval contained the observed values of y_t (cumulative COVID-19 death counts in our case). This notion of *coverage* for streaming data has been used extensively in prior works on conformal inference [93] and can be calculated for a given evaluation period $\mathcal{T}$ (which we set to be from April 11 to June 20) as follows:

$$\mathrm{Coverage}(\mathcal{T}) = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} \mathbb{I}(y_t \in \widehat{\mathrm{PI}}_t), \tag{5.12a}$$

where $\mathbb{I}(y_t \in \widehat{\mathrm{PI}}_t)$ takes value 1 if y_t belongs to the interval $\widehat{\mathrm{PI}}_t$ and 0 otherwise. The average *normalized length* of the prediction intervals, NL($\mathcal{T}$), is calculated as follows:

$$\mathrm{NL}(\mathcal{T}) = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} \frac{b_t - a_t}{y_t}. \tag{5.12b}$$

[13] We use the terms width and length for an interval interchangeably in this paper.

For our case, we replace the denominator on the RHS of equation (5.12b) with $\max\{1, y_t\}$ to avoid possible division by 0. We use normalized length to address the fact that the death counts across different counties can differ by orders of magnitude.

Importantly, the definitions of coverage (equation (5.12a)) and the average length (equation (5.12b)) are entirely data-driven and do not rely on any probabilistic or generative modeling assumptions.

Exchangeability of the normalized prediction errors

While the ideas from MEPI are a special case of conformal prediction intervals [93, 82], there are some key differences. While conformal inference uses the raw errors in predictions, MEPI uses the normalized errors, and while conformal inference uses a percentile (e.g., the 95th percentile) of the errors, MEPI uses the maximum. Furthermore, we only make use of the previous five days instead of the full sequence of errors. The reason behind these alternate choices is because the validity of prediction intervals constructed in this manner relies crucially on the assumption that the sequence of errors is exchangeable. Our choices are designed to make this assumption more reasonable. Due to the dynamic nature of COVID-19, considering a longer period (e.g., substantially longer than five days) would mean that it is less likely that the errors across the different days are exchangeable. Meanwhile, the normalization of the errors eliminates a potential source of non-exchangeability by removing the sequential growth of the errors resulting from the increasing nature of the counts themselves. Since we only use five time points, to construct the interval, we opt for the more conservative choice of simply taking the maximum—or the 100th percentile—of the five errors. Moreover, note that 95th percentile is not well defined for five data points.

Before turning to certain theoretical guarantees, we first discuss some evidence for exchangeability of the errors in our setting. In Figures 5.5 and .6, we show that the assumption of exchangeability of the past 5 *normalized* errors for CLEP is indeed reasonable for both 7-day-ahead and 14-day-ahead predictions. For k-day-ahead prediction, we rank the errors $\{\Delta_{t+k}, \Delta_t, \Delta_{t-1}, \ldots, \Delta_{t-4}\}$ in increasing order so that the largest error has a rank of 6. (Interested readers may first refer to Section 5.4 for how exchangeability of these 6 errors is useful for establishing theoretical guarantees for MEPI.) For a given $j \in \{0, ..., 4\}$, Δ_{t-j} denotes the error in k-day-ahead prediction for day $t-j$, where the prediction was made on the morning of day $t-j-k+1$, but the error can be computed only by the end of day $t-j$ (or the morning of day $t-j+1$). If the errors were exchangeable, then for each of them, the rank has a uniform distribution on $\{1, 2, 3, 4, 5, 6\}$, and in particular has a mean of 3.5. To approximate the rank distributions and average ranks numerically for 7-day-ahead predictions, we measure the rank of the errors Δ_{t+7}, and $\Delta_{t-j}, j = 0, \ldots, 4$, for each day t between March 26 to June 13, and take an average. Figure 5.5 plots the empirical rank distributions and average ranks for each of the 6 worst hit counties as well as 6 randomly-selected counties for these errors (see Section 5.5 for further discussion on these counties.). Corresponding results for 14-day-ahead predictions, where we rank the errors Δ_{t+14} and $\Delta_{t-j}, j = 0, \ldots, 4$ is presented in Figure .6 in the Appendix. In both figures, we see that across most counties,

the average rank for almost all the errors is around 3.5 as would be expected if the errors were exchangeable. Thus, the observations from Figures 5.5 and .6 provide a heuristic justification for the construction of MEPI, albeit not a formal proof since average rank being close to 3.5 is not sufficient to claim exchangeability of the six errors. Moreover, we refer the interested reader to Appendix .2 for further discussion on MEPIs, where we provide more evidence on why we chose only past 5 errors and normalization to define MEPI (see Figure .5).

Theoretical guarantees for MEPI coverage

In order to obtain a rough baseline coverage for the MEPIs, we now reproduce some of the theoretical computations from the conformal literature. For a given county and a fixed time t, and a parameter k, if the six errors in the set $\{\Delta_{t+k}, \Delta_t, \Delta_{t-1}, \Delta_{t-2}, \Delta_{t-3}, \Delta_{t-4}\}$ are exchangeable, then we have

$$\mathbb{P}\left(y_{t+k} \in \widehat{\mathrm{PI}}_{t+k}\right) = \mathbb{P}\left(\Delta_{t+k} < \Delta_{\max}\right) = 1 - \mathbb{P}\left(\Delta_{t+k} = \Delta_{\max}\right) = \frac{5}{6} \approx 0.83. \tag{5.13}$$

Recall the definition (equation (5.12a)) for Coverage($\mathcal{T}$) for a given period of days $\mathcal{T}$. Given equation (5.13), we may believe that the Coverage($\mathcal{T}$) $\approx 83\%$ holds for large $|\mathcal{T}|$, where the coverage was defined in equation (5.12a). However, we now elaborate that a few challenges remain to take claim (5.13) as a proof for the stronger claim that MEPI achieves 83% coverage as defined by equation (5.12a).

On the one hand, the probability in equation (5.13) is taken over the randomness in the errors, and the time-index $t+k$ remains fixed. This observation, in conjunction with the law of large numbers, implies the following: Over multiple independent runs of the time-series, for a given county and a given time $t+k$, the fraction of runs for which the MEPI $\widehat{\mathrm{PI}}_{t+k}$ contains the observed value y_{t+k} converges to 5/6 as the number of runs goes to infinity. However, analyzing such a fraction over several different independent runs of the COVID-19 outbreak is not relevant for our work.

On the other hand, the evaluation metric we consider is the average coverage of the MEPI over a single run of the time-series, c.f., the definition (equation (5.12a)) for Coverage($\mathcal{T}$). Thus, we require an online version of the law of large numbers in order to guarantee that Coverage($\mathcal{T}$) $\to 83\%$ as $|\mathcal{T}| \to \infty$. Such a law of large numbers, established in prior works [82], has been crucial for establishing theoretical guarantees in conformal inference. In our case, this law—stated as Proposition 1 in Section 3.4 in their paper [82]—guarantees that, when the entire sequence of errors $\{\Delta_t, t \in \mathcal{T}\}$ for a given county is exchangeable, the corresponding Coverage($\mathcal{T}$) $\approx 83\%$, when the period $\mathcal{T}$ is large. Unfortunately, such an assumption (exchangeability over the entire period) is both hard to check and unlikely to hold for the prediction errors obtained from CLEP for the COVID-19 cumulative death counts.

Despite the challenges listed above, later, we show in Section 5.5 that MEPIs with CLEP achieved good coverage with narrow widths for COVID-19 cumulative death count predictions.

5.5 Prediction results for March 22 to June 20

In this paper, we focus on predictive accuracy for up to 14 days. In this section, we first present and compare the results of our various predictors, and then give further examinations of the best performing predictor: the CLEP ensemble predictor that combines the expanded shared exponential predictor and the linear predictor (the best two among the individual predictors). Finally, we report the performance of the coverage and length of the MEPIs for this CLEP. (We note that CLEP that combined all five predictors performed worse since even bad performing predictors get non-zero weight (5.7) in the ensemble and can adversely affect the prediction performance.) A Python script to reproduce all the results in this section is made available on Github at `https://github.com/Yu-Group/covid19-severity-prediction/tree/master/modeling`.

Empirical performance of the single predictors and CLEP

Table 5.9 summarizes the Mean Absolute Errors (MAEs) of our predictions for cumulative recorded deaths on raw, square-root and logarithm scale. We now explain how these errors are computed.

First, on the morning of day $t+1$, we compute $\mathcal{C}_t$—the collection of counties in the US that have at least 10 cumulative recorded deaths by the end of day t. Let $\widehat{y}_t^c$ and y_t^c respectively denote the predicted and recorded cumulative death count of county $c \in \mathcal{C}_t$ by the end of day t. We note that while the set of counties $\mathcal{C}_t$ varies with time, it is computable on the day the error is computed (i.e., $\mathcal{C}_t$ does not depend on future information). We define the set of counties in this manner, to ensure that only the counties with non-trivial cumulative death counts are included in our evaluation on a given day. Moreover this definition satisfies the condition $\mathcal{C}_t \subseteq \mathcal{C}_{t+1}$, that is, only new counties can be added in the set $\mathcal{C}_t$ as t progresses (and a county once included is never removed).

Given the set $\mathcal{C}_t$, the mean absolute percentage error (MAPE), the raw-scale MAE, and the square-root-scale MAE for day t are given by

$$\text{MAPE}_t(\%\text{ error}) = 100 \times \frac{1}{|\mathcal{C}_t|} \sum_{c \in \mathcal{C}_t} \frac{|\widehat{y}_t^c - y_t^c|}{y_t^c}, \quad \text{and} \tag{5.14a}$$

$$\text{Raw-scale MAE}_t = \frac{1}{|\mathcal{C}_t|} \sum_{c \in \mathcal{C}_t} |\widehat{y}_t^c - y_t^c|\,, \tag{5.14b}$$

$$\text{Sqrt-scale MAE}_t = \frac{1}{|\mathcal{C}_t|} \sum_{c \in \mathcal{C}_t} \left|\sqrt{\widehat{y}_t^c} - \sqrt{y_t^c}\right|. \tag{5.14c}$$

The percentage error (MAPE) captures the relative errors of the predictors without paying attention to the scale of the counts while the raw-scale MAE would be heavily affected by the counties with large death counts. Both of these MAEs are commonly used to report the prediction performance for regression tasks with machine learning methods. We report the MAE at square-root-scale to be consistent with the square-root transform used in the CLEP weighting scheme (5.7). Each row in Table 5.9 corresponds to a single predictor, and we report different statistics of errors made for k-day-ahead predictions over the period $t \in \{\text{March } 22, \ldots, \text{June } 20\}$ for $k \in \{3, 5, 7, 14\}$.[14] For any given predictor, we compute these errors for each day and report the 10th percentile (p10), 50th percentile (median), and 90th percentile (p90) values over the evaluation periods mentioned above. From Table 5.9, we find that the CLEP ensemble that combines the expanded shared exponential predictor and the separate county linear predictors has the best overall performance, with median MAPE of 8.18%, 12.21%, 15.14% and 26.45% for 3-, 5-, 7-, and 14-day-ahead predictions.[15] We note that the (p90) MAEs for the separate (exponential) and demographics shared predictors are too large, especially for larger horizons. For separate (exponential) predictors, we can attribute these large errors directly to the fact that exponential fit (5.1) for large horizons is very likely to over-predict. On the other hand, the demographics shared predictor has large errors potentially due to over-fitting as well as the recursive plug-in procedure to obtain longer horizon estimates in the exponential fit (5.5). In the early stages of this project (and the COVID-19 outbreak in the US), these predictors had provided a reasonable fit for short-term (3- and 5-day-ahead) predictions in late-March to mid-April.

In Figure 5.6, we plot all three errors from the display 5.14 as a function of time over the past 3 months for the expanded shared exponential predictor, the separate county linear predictor, and the CLEP that combines the two. We found that the MAE of the CLEP is often similar to, and usually slightly smaller than the smaller MAE of the two single predictors.

Visualization of CLEP weights: Figure 5.7 plots the weight of the linear predictor in the CLEP (for 7-day-ahead predictions) over time for certain counties. We found that for counties with large number of cumulative deaths, the prediction of the CLEP has become much more similar to the prediction of the linear predictor in late May and June. For example, for the six worst affected counties on June 20 (panel (a)), the average weight of the linear predictor in the CLEP is larger than 0.91 from May 17 to June 20. In contrast,

[14] As the expanded shared predictor is trained on counties with at least 3 deaths, there was not enough data to train 14-day-ahead CLEP that predicts recorded deaths before March 29. Hence for $t \in \{\text{March } 22, \ldots, \text{March } 28\}$, we use the 14-day-ahead predictions of the linear predictor to impute the 14-day-ahead predictions of the CLEP.

[15] In the first version of this paper submitted on May 16, 2020 on arXiv, we reported results for the period March 22 to May 10, 2020 for 3-, 5-, 7-day ahead predictions. We made an error while computing aggregate statistics for 5-day and 7-day ahead predictions (reported in Table 3 of that version) and reported errors that were smaller than the actual errors made by CLEP. Correcting our aggregation code revealed that the actual performance of median of raw-scale MAE of CLEP for 5-day and 7-day predictions was worse by 16% and 29% respectively when compared to the reported errors.

[t]

	3-day-ahead			5-day-ahead			7-day-ahead			14-day-ahead		
	p10	median	p90	p10	median	p90	p10	median	p90	p10	median	p90
separate	3.80	13.16	59.63	6.26	22.56	114.07	9.95	39.56	300.53	30.37	226.26	>1000
shared	7.05	12.55	25.99	11.68	19.77	37.73	16.59	28.65	55.01	36.55	62.45	224.75
demographics	14.80	24.08	106.97	23.08	37.60	>1000	31.52	51.52	>1000	89.71	190.97	>1000
expanded shared	6.86	9.59	35.55	11.17	14.54	44.28	15.09	18.52	52.13	23.13	31.18	>1000
linear	3.39	9.37	29.67	5.27	14.25	40.26	7.18	18.60	56.10	15.58	33.16	87.21
CLEP	4.34	8.18	22.60	6.59	12.21	31.99	8.79	15.14	42.47	14.61	26.45	93.03

Table 5.6: Summary statistics of mean absolute percentage error (MAPE)

[t]

	3-day-ahead			5-day-ahead			7-day-ahead			14-day-ahead		
	p10	median	p90	p10	median	p90	p10	median	p90	p10	median	p90
separate	2.35	8.10	25.13	3.67	13.94	57.03	5.33	24.30	124.61	14.58	105.64	>1000
shared	7.54	12.04	19.43	13.12	19.93	36.74	18.81	28.09	72.74	33.69	69.35	325.50
demographics	21.81	39.61	77.78	44.57	79.36	596.04	93.83	147.66	>1000	656.55	>1000	>1000
expanded shared	8.07	10.69	14.32	13.10	16.68	23.02	18.24	22.95	42.84	29.90	36.56	329.21
linear	2.15	5.93	13.81	3.67	9.49	20.02	4.91	12.05	26.89	10.24	25.47	56.73
CLEP	2.76	5.98	11.93	4.09	8.64	18.67	5.42	10.64	27.29	9.18	22.50	81.77

Table 5.7: Summary statistics of raw-scale MAE

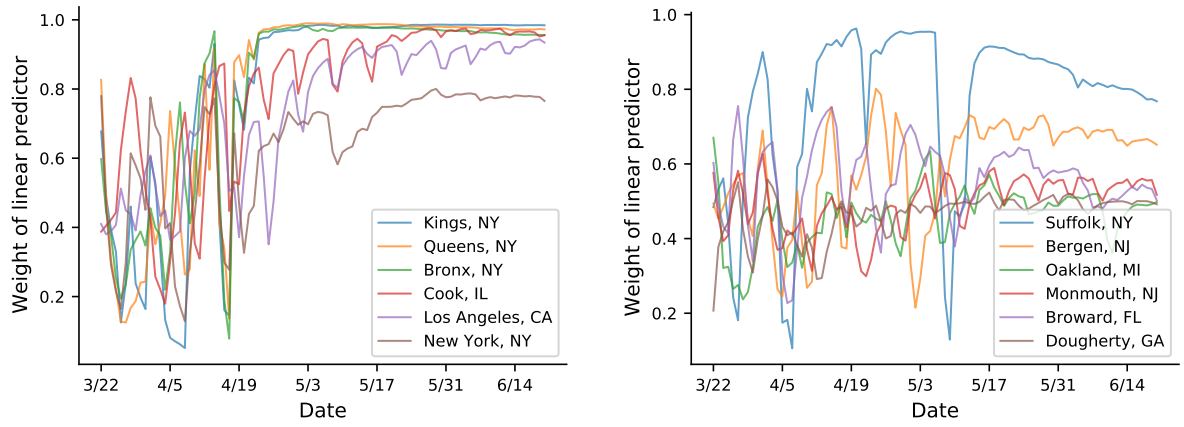
[t]

	3-day-ahead			5-day-ahead			7-day-ahead			14-day-ahead		
	p10	median	p90	p10	median	p90	p10	median	p90	p10	median	p90
separate	0.11	0.39	1.66	0.19	0.63	2.91	0.25	1.03	3.83	0.67	3.40	18.26
shared	0.22	0.40	0.76	0.36	0.63	1.22	0.50	0.90	1.82	1.09	2.06	5.70
demographics	0.73	1.03	2.78	1.24	1.79	6.63	1.84	2.65	35.56	5.49	8.33	>1000
expanded shared	0.22	0.34	0.75	0.35	0.52	1.15	0.47	0.67	1.59	0.73	1.12	10.62
linear	0.11	0.29	0.99	0.17	0.46	1.45	0.23	0.58	2.15	0.50	1.10	4.62
CLEP	0.13	0.26	0.66	0.19	0.37	0.93	0.26	0.47	1.51	0.43	0.92	4.13

Table 5.8: Summary statistics of sqrt-scale MAE

Table 5.9: Summary statistics of Mean Absolute Errors (equation (5.14)) based on **(A)** the mean absolute percentage error (MAPE), **(B)** the raw-scale MAE, and **(C)** the square-root-scale MAE. The results are presented for the 3, 5, 7, and 14-days-ahead forecasts for each of the predictors considered in this paper, and the CLEP that combines the expanded shared and separate linear predictors. The evaluation period is March 22, 2020 to June 20, 2020 (91 days). "p10", "median", and "p90" denote the 10th-percentile, median, and 90th-percentile of the 91 mean absolute errors computed daily in the evaluation period. The smallest error in each column is displayed in bold.

the average weight of linear predictor of these six counties is less than 0.5 from March 23 to March 31.



(a) Worst affected counties (b) Random counties

Figure 5.7: Weight of the linear predictor in the CLEP (for *7-day-ahead* predictions) over time, for six worst affected counties in panel **(a)**, and six random counties in panel **(b)**.

Furthermore, in Figure 5.8, we also provide a US-wide visualization for the weight of the linear predictor for all counties, displaying the values on April 1 in panel (a), and those on June 10 in panel (b). As expected, the weight of the linear predictor is high when the COVID-19 outbreak (in terms of death counts) is in a linear growth regime, which can be seen generally for counties with low number of death counts, or counties with large number of deaths where the COVID-19 outbreak is in a post-exponential growth stage.

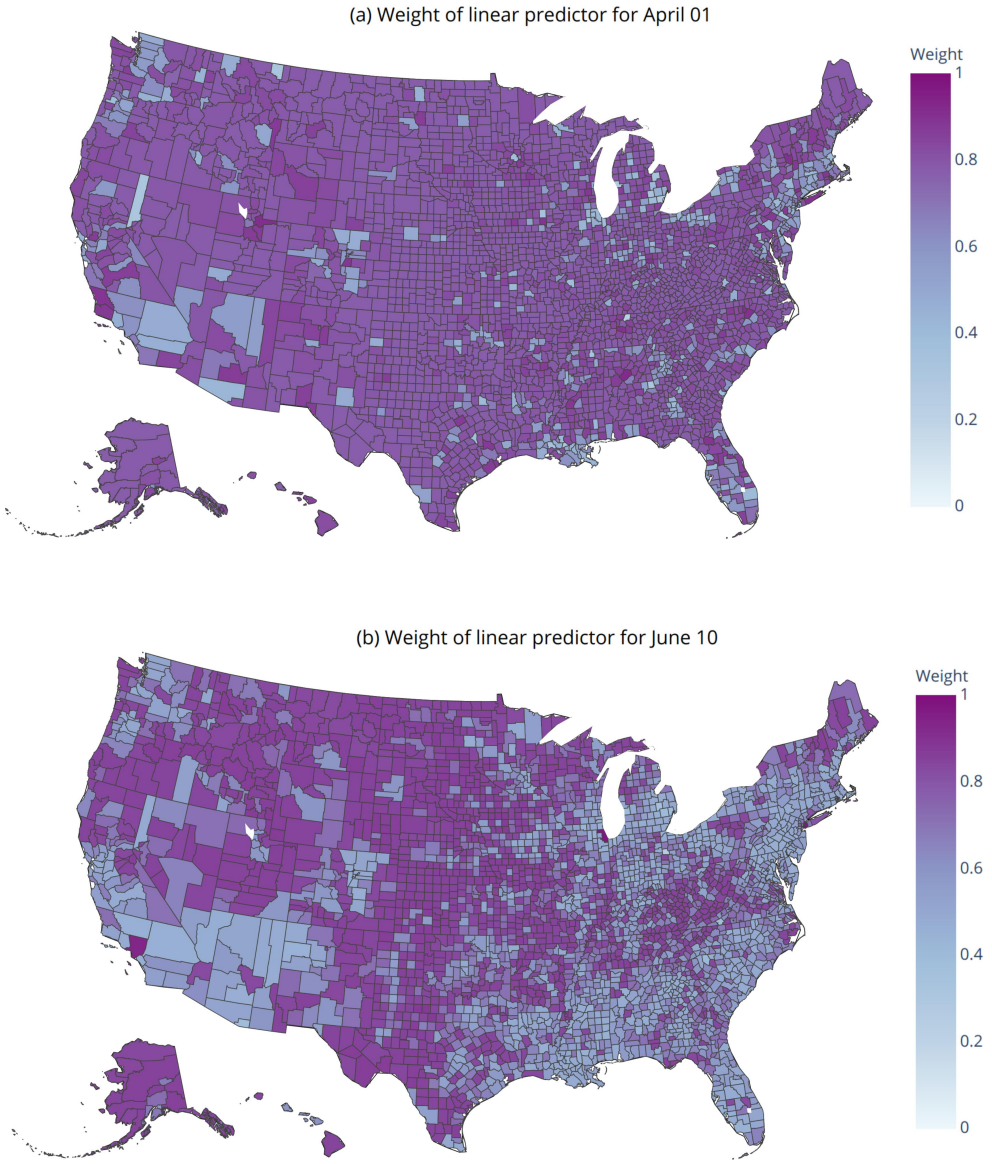


Figure 5.8: Weight of the linear predictor in the CLEP (for *7-day-ahead* predictions) for every county in the US at two timepoints: April 1 in panel **(a)**, and June 10, 2020 in panel **(b)**. Generally speaking, for both the panels, the weight for the linear predictor in CLEP gets close to 1 for counties with linear growth of COVID-19 (in terms of death counts) which can be observed relatively easily for two scenarios: (i) when the counties have low number of COVID-19 death counts (for such counties, the expanded shared predictor generally overestimates the future death counts), and (ii) when the counties that are in the later stage of COVID-19 and have had a large number of deaths.

CLEP performance for longer horizons: Next, in Figure 5.9, we plot the performance of this CLEP for longer horizons. In particular, we plot the three MAEs, raw-scale, percentage-scale, and square-root-scale, for k-day-ahead predictions for $k = 7$, 10, and 14. Notice that the 7-day-ahead CLEP predictor has the lowest MAE and that the MAE increases as the prediction horizon increases. (Recall that precise statistics for 7-day-ahead and 14-day-ahead MAEs are listed in Table 5.9.) The increases in MAE in mid-late April was due to the state of New York adding thousands of deaths (3,778) that were previously reported as "probable" to their counts on a single day, April 14. This change led the CLEP to greatly over-predict deaths in New York in mid-late April—(i) 7 days later on April 21 for the 7-day-ahead CLEP, (ii) 10 days later on April 24 for the 10-day-ahead CLEP, and (iii) 14 days later on April 28 for the 14-day-ahead CLEP. As further evidence that this is indeed the case, when we manually removed this uptick in the death counts in New York, the raw-scale MAE for the 14-day-ahead CLEP on April 28 was 29.5, which is much smaller than the original raw-scale MAE on April 28, which was 91.9 in Figure 5.9(b).

We further evaluate the performance of CLEP for longer prediction horizons in Figure 5.10, where all predictions were made over the period April 11-June 20. In panels (a-c) of Figure 5.10, we show the box plots of the different MAEs for up to 14-day prediction horizon. From these plots, and the panels (d-f), we observe that the MAEs increase roughly linearly with the horizon for up to 21 days.

Putting together the results from Table 5.9, Figures 5.6, 5.9 and 5.10, we find that the adaptive combination used for building our ensemble predictor CLEP is able to leverage the advantages of linear and exponential predictors, and, by improving upon the MAE of single predictors, is able to provide very good predictive performance for up to 14 days in future.

Performance of CLEP and MEPI at the county-level

Having examined the overall performance of our predictors, we now take a closer look at how our predictors are performing at the county level. In this section, we focus on the performance of our CLEP predictor (based on the best-performing CLEP of the expanded shared and separate linear predictor models) for the period April 11 to June 20 for 7-day and 14-day-ahead predictions (Figures 5.11 and 5.12 respectively). Since there are over 3,000 counties in the United States, we present results for two sets of counties:

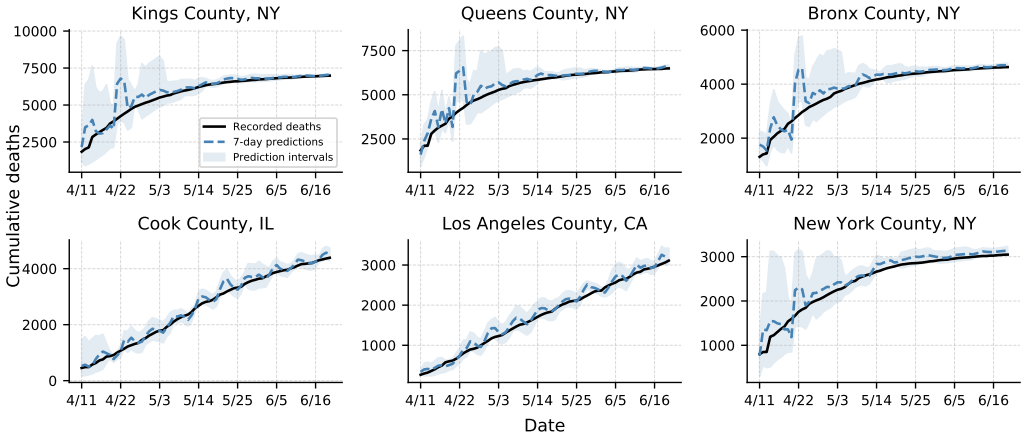
- The six worst-affected counties on June 20: Kings County, NY; Queens County, NY; Bronx County, NY; Cook County, IL; Los Angeles County, CA; and New York County, NY.
- Six randomly selected counties: Suffolk County, NY; Bergen County, NY; Oakland County, MI; Monmouth County, NJ; Broward County, FL; Dougherty County, GA.

7-day-ahead predictions: In Figure 5.11, we present the 7-day-ahead prediction results for the worst-affected counties in the top panel (a), and for the randomly-selected counties

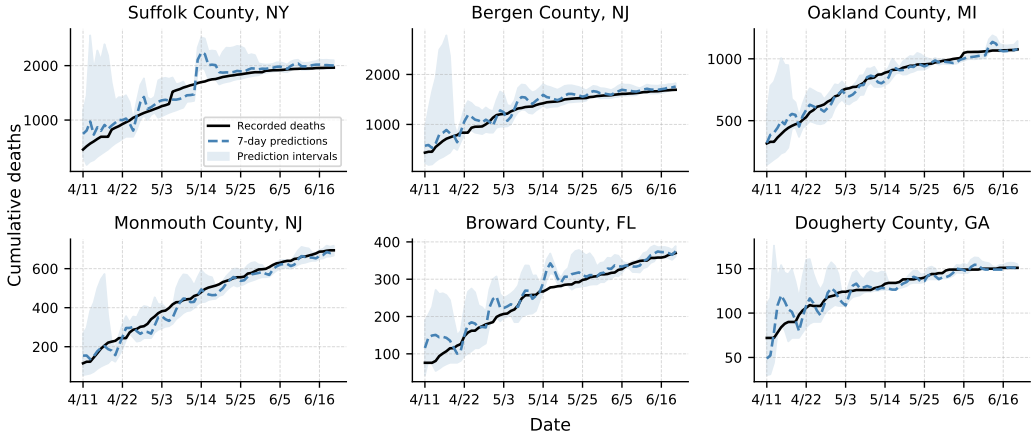
in the bottom panel (b). The solid black line denotes the recorded death counts, dashed blue line denotes the CLEP 7-day-ahead predictions, and shaded blue region denotes the corresponding MEPI (prediction interval). The predictions and prediction intervals for a given day t (t = April 11, ..., June 20) are based on data up to day $t - 7$. Corresponding results for 14-day-ahead predictions are plotted in Figure 5.12. Although the recorded cumulative death counts are monotonically increasing, our predictions for it (blue dashed lines in Figures 5.11 and 5.12) need not be since the predictions are updated daily. (Recall from Section 5.3 that we did impose monotonicity constraints for predictions with respect to k for k-day-ahead predictions made on a given day t.)

From Figure 5.11(a), we observe that, among the worst-affected counties, the CLEP appears to fit the data very well for Cook County, IL, and Los Angeles County, CA. After initially over-predicting the number of deaths in the four NY counties in mid-late April, our predictor also performs very well on these NY counties. Moreover, the MEPIs—plotted as blue shaded region—have reasonable width and appear to cover the recorded values for the majority of the days (detailed results on MEPIs are presented in Section 5.5).

The rapid increase in our predictions and MEPI interval widths observed in mid-late April in Figure 5.11(a) is caused by a major upward revision of cumulative death counts by the New York state for Kings, Queens, Bronx, and New York counties on a single day, April 14. The effect of this sudden change in recorded counts on CLEP performance was also discussed in the previous section in the context of Figure 5.9. Our predictors quickly corrected themselves after the recorded counts stabilized. From Figure 5.11(b), we find that our predictors and MEPI perform well for each of our six randomly-selected counties (Broward County, FL, Dougherty County, GA, Monmouth County, NJ, Bergen County, NJ, and Oakland County, MI). However, for Suffolk County, NY, there is a sudden uptick in cumulative deaths on May 5, leading to a fluctuation in the predictions shortly thereafter, in a similar fashion to the other NY counties in mid-April. In both panels, our predictions have higher uncertainty at the beginning of the examination period when recorded death numbers are low, which is reflected in the slightly wider MEPIs in the bottom left of each plot.



(a) Worst-affected counties on June 20



(b) Randomly-selected counties

Figure 5.11: A grid of line charts displaying the performance of *7-day-ahead* CLEP and MEPI for the cumulative death counts due to COVID-19 between April 11, 2020 and June 20, 2020. The observed data is shown in black, CLEP predictions are shown in the dashed blue, and the corresponding 7-day-ahead MEPIs are shown as shaded blue regions. In panel **(a)**, the MEPI coverage for the 6 counties are Kings (92%), Queens (80%), Bronx (90%), Cook (90%), Los Angeles (89%) and New York (86%). In panel **(b)**, Suffolk (85%), Bergen (96%), Oakland (87%), Monmouth (86%), Broward (89%) and Dougherty county (86%). (It is worth noting that the theoretical coverage guarantees under certain assumptions is 83%; see equation (5.13).)

14-day-ahead predictions: Next, in Figure 5.12, we plot the visualizations for 14-day-ahead CLEP and MEPI predictions for the two sets of counties discussed above. On the one hand, overall, the CLEP predictions appear to track the cumulative death counts and MEPIs

cover the observed death counts most of the times. On the other hand, we find that the MEPIs are generally wide for all the counties in the beginning of the prediction period. This phenomenon is due to the fact that till mid-April, predictions made 14 days ago were trained on data with very few death counts, which led the expanded shared predictor to significantly underestimate the death counts in the beginning period (leading to larger normalized errors and thereby wider MEPIs). Furthermore, we also observe that for the counties in the NY State, CLEP greatly overestimates (sharp peak around April 22) the cumulative recorded death counts towards the end of April. Like in Figure 5.11, this overestimation is caused by the major upward revision of the death counts in the NY State on April 14. It is worth noting that these sharp peaks for 14-day-ahead CLEP predictions in Figure 5.12 occurs after 7 days of the sharp peaks observed for 7-day-ahead CLEP predictions in Figure 5.11.[16]

Empirical performance of MEPI

We now present the performance of our MEPI at the county level for cumulative death counts, with respect to both coverage (equation (5.12a)) and average normalized length (equation (5.12b)) (see Section 5.4). Since the average performance may change with time, we report the results for two time periods: {April 11, ..., May 10}, and {May 11, ..., June 20}. We choose these time periods for a simple reason: The first draft of this paper presented results until May 10, and thus the period May 11 to June 20 serves as an additional validation. (Our methods were proposed and tuned prior to May 10, except for the change of transform from logarithm to square-root in equation (5.7).)

We evaluate the 7-day-ahead and 14-day-ahead MEPIs, i.e., $k = 7$ and 14 in equation (5.10a), designed with the CLEP that combines the expanded shared and separate linear predictors, and summarize the results in Figure 5.13 and Figure 5.14 respectively. We now discuss these results, first for 7-day-ahead and then for 14-day-ahead MEPIs.

Coverage: We compute the observed coverage(5.12a) of 7-day-ahead MEPIs across all counties in the US for April 11–May 10, and May 11–June 20, and plot the histogram of these values in Figure 5.13(a) and 5.13(c), respectively. Panel (e) of Figure 5.13 shows the observed coverage of 7-day-ahead MEPIs for the 700 counties that had at least 10 deaths on June 11 (each such county has had significant counts for at least 10 days by the end of our evaluation period June 20). For each county, we include only the days between April 11, 2020 and June 20, 2020 for which the county had at least 10 cumulative deaths. On June 20, the median number of days since 10 deaths is 58. From these plots, we observe that we achieve excellent coverage for the majority of the counties. Finally, Figure 5.14 shows the corresponding results for 14-day-ahead MEPIs, i.e., coverage for April 11–May 10 in panel

[16]These observations also suggest that one may possibly use large errors from our predictors as a warning flag for anomaly or reporting error/sudden revision in the data. We leave a detailed investigation on this aspect as an interesting future direction.

(a), May 11–June 20 in panel (c), and over the county-specific period for 700 counties that had at least 10 deaths on June 11 in panel (e).

The coverage observed for the two periods in panels (a) and (c) of Figure 5.13 are very similar. For the earlier period—April 11 to May 10—in panel (a), the 7-day-ahead MEPIs have a median coverage of 100% and mean coverage of 95.6%. On the other hand, for the later period—May 11 to June 20—in panel (c), the 7-day-ahead MEPIs have a median coverage of 100% and mean coverage of 96.2%. However, the coverage is slightly decreased when restricting to counties with at least 10 deaths in panel (e), for which we observe a median coverage of 88.7%, and mean coverage of 87.9%. This observation is consistent with the fact that at the beginning of the pandemic, several counties had zero or very few deaths resulting in very good coverage with the prediction interval. On the other hand, note that smaller death counts would also imply a relatively larger normalized length for the MEPI intervals.

Figure 5.14(a), (c) and (e) show that, in general, our 14-day-ahead MEPIs achieve similar coverage as our 7-day-ahead MEPIs. For example, over the April 11–May 10 and May 11-June 20 periods, our prediction intervals have mean coverage of 95.0% and 97.0% (median is 100% for both periods). For panel (e)—counties with at least 10 deaths on June 11—the coverage has a median of 89.7% and a mean of 87.9%.

Overall, the statistics discussed above show that both 7-day-ahead and 14-day-ahead MEPIs achieve excellent coverage in practice. In fact, for the counties with poor coverage, we show in Appendix .2 that there is usually a sharp uptick in the number of recorded deaths at some point in the evaluation period, possibly due to recording errors, or backlogs of counts. Modeling these upticks and obtaining coverage for such events is beyond the scope of this paper.

Normalized length: Next, we discuss the other evaluation metric of the MEPIs, their normalized length (5.12b). In panels (b) and (d) of Figure 5.13, we plot the histogram of the observed average normalized length of 7-day-ahead MEPIs for the periods April 11–May 10, and May 11–June 20 respectively. Panel (f) covers the same counties as did panel (e): those with at least 10 deaths for at least 10 days in the period April 11 to June 20, 2020.

Recall that the normalized length is defined as the length of the MEPI over the recorded number of deaths (equation (5.12b)). And, more than 70% of counties in the US recorded 2 or less COVID-19 deaths by May 1. For these counties, having a normalized length of 2 means the actual length of the prediction interval is 4 (or less). And thus, it is not surprising to see that the average normalized length of MEPI for a non-trivial fraction of counties is larger than 2 in panels (b) and (d). When considering counties with at least 10 deaths in panel (f), the average normalized length over these (county-specific) periods is much smaller; and the median is 0.470.

Turning to 14-day-ahead MEPIs in Figure 5.14, panels (b) and (d) show that that the normalized length for 14-day-ahead MEPIs can be quite wide for counties with a small

number of deaths. Nevertheless, panel (f) shows that the 14-day-ahead MEPIs are reasonably narrow for counties with more than 10 deaths, with a median average normalized length of 1.027—which is roughly two times the median size of 0.470 for the 7-day-ahead MEPIs in Figure 5.13(f).

Overall, Figures 5.13 and 5.14 show that our MEPIs provide a reasonable balance between coverage and length for up to 14 days in future, especially when the cumulative counts are not too small.

5.6 Related work

Several recent works have tried to predict the number of cases and deaths related to COVID-19. Even more recently, the Center for Disease Control and Prevention (CDC) has started aggregating results from several models.[17] But to the best of our knowledge, ours was the first work focusing on predictions at the county-level. During the time period of our work, a direct comparison with other models' to our own were difficult for several other reasons: (1) the models relied on strong assumptions and did not usually provide predictive checks on their models, (2) we did not have access to a direct implementation of their models (or results), and (3) their models focus on substantially longer time horizons. During the revisions, we became aware of a recent work by [19] which we return to after a brief summary of the other related works on predictive modeling for COVID-19 release.

Two recent works [63, 32] have modeled the death counts at the state level in the US. The earlier versions of the model by Murray et al. (also referred to as the IHME model) was based on Farr's Law with feedback from Wuhan data. On the other hand, the Imperial College model [32] uses an individual-based simulation models with parameters chosen based on prior knowledge. On the topic of Farr's Law, we note that [10] used Farr's Law to predict that the total cases from the AIDS pandemic would diminish by the mid-1990s and the total number of cases would be around 200,000 in the entire lifetime of the AIDS pandemic. It is now estimated that 32 million people have died from the disease so far. While the AIDS pandemic is very different from the COVID-19 pandemic, it is still useful to keep this historical performance in mind.

Another approach uses exponential smoothing from time-series predictors to estimate day-level COVID-19 cases [30]. In addition, several works use compartmental epidemiological predictors such as SIR, SEIR, and SIRD [31, 72, 8] to provide simulations at the national level. Other works [70, 45] simulate the effect of social distancing policies either in the future for the US or in a retrospective manner for China. Finally, several papers estimate epidemiological parameters retrospectively based on data from China [94, 56].

During the revision of our paper, another work was published by [19] that appeared in medRxiv on June 8, 2020[18] after the submission of our paper to arXiv in mid-May. Chiang

[17] Forecasts available at `https://www.cdc.gov/coronavirus/2019-ncov/covid-data/forecasting-us.html`

[18] Accessed at `https://www.medrxiv.org/content/10.1101/2020.06.06.20124149v1.full.pdf`

et al. use models based on Hawkes' process to provide county-level predictions for new daily cases as well as new death counts. Of note is that the authors also explore a CLEP with adaptive tuning of c and μ (whereas we used fixed values for these parameters). Such a tuning approach might present a promising improvement of CLEP performance in general and we plan to investigate adaptive tuning of various hyper-parameters in the CLEP in our own future work. Unfortunately, we were unable to properly reproduce the CLEP results provided in Chiang et al.'s work using their provided documentation. During a private email exchange the authors kindly provided further information regarding some of our questions about their methodology[19], but several of their choices make it difficult to compare their work to ours. For instance, their work focuses on daily counts, rather than cumulative counts as ours does. More importantly, their prediction numbers are not available on their GitHub repository (both for the period till May 20 analyzed in their paper and the days since then). The authors do not report the performance of their confidence intervals in the paper, and report the MAE performance metric only for the counties that fall in the top quantiles of cumulative counts at the end of the evaluation period. Such a quantile-based group of counties is not interpretable (since it is time-varying and not spatially meaningful) and does not allow for real-time use, since one must wait until the end of the evaluation period to calculate the performance. In addition, the authors compute their predictions in blocks of days, e.g., once a week for the 7-day-ahead predictions (rather than daily as in this paper). Thus, from our point of view, these decisions unfortunately make their work ill-suited to real-time usage for making fast-paced policy decisions related to COVID-19.

5.7 Impact: a hospital-level severity index for distributing medical supplies

We are using our models to support the non-profit Response4Life[20] in determining which hospitals are most urgently in need of medical supplies and have subsequently been directly involved in the distribution of medical supplies across the country. To do this, we translate our forecasts into the COVID pandemic severity index, which is a simple measure of the COVID-19 outbreak severity for each hospital.

To generate this hospital-level severity index, we divided the total county-level deaths among all of the hospitals in the county proportional to their number of employees. Next, for each hospital, we computed its percentile among all US hospitals with respect to total deaths so far and also with respect to predicted new deaths in the next seven days. These two percentiles are then averaged to obtain a single score for each hospital. Finally, this score is quantized evenly into three categories: low, medium, and high severity. Evaluation

[19]In particular, via this email exchange we learned that: (i) they had implemented the adaptive tuning of CLEP; and (ii) they had computed the % error (in Table S1 of their paper) for total new counts over the entire k-day-block (for k-day-ahead predictions) summed over all the counties in a given quantile thereby explaining the (surprising at first) decrease in % error as the prediction horizon k increases.

[20]`https://response4life.org/`

and refinement of this index are ongoing, as more hospital-level data becomes available. The interested reader can find a daily-updated map of the COVID pandemic severity index and additional hospital-level data at our website `https://covidseverity.com`.

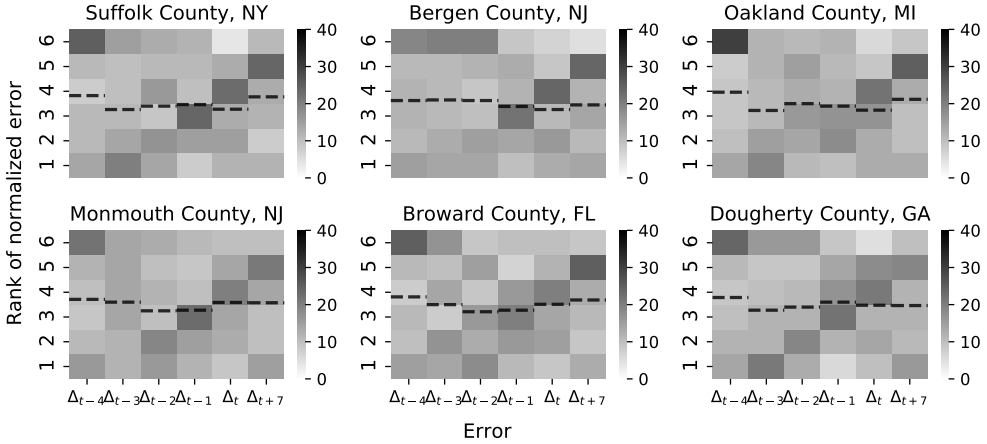
5.8 Conclusion

In this paper, we made three key contributions. We (1) introduced a data repository containing COVID-19-related information from a variety of public sources, (2) used this data to develop CLEP predictors for short-term forecasting at the county level (up to 14 days), and (3) introduced a novel yet simple method MEPI for producing prediction intervals for these predictors. By focusing on county-level predictions, our forecasts are at a finer geographic resolution than those from a majority of other relevant studies. By comparing our predictions to real observed data, we found that our predictions are accurate and that our prediction intervals are reasonably narrow and yet provide good coverage. We hope that these results will be useful for individuals, businesses, and policymakers in planning and coping with the COVID-19 pandemic. Indeed, our results are already being used to determine the hospital-level need for medical supplies and have been directly influential in determining the distribution of these supplies.

Furthermore, our data repository as well as forecasting and interval methodology will be useful for academic purposes. Our data repository has already been used for data science education, and by other teams interested in analyzing the data underlying the COVID-19 pandemic. Our CLEP ensembling techniques and MEPI methodology can be applied to other models for COVID-19 forecasting, as well as to online methods and time-series analysis more broadly. Our data, codes and models can be found at `https://covidseverity.com`.

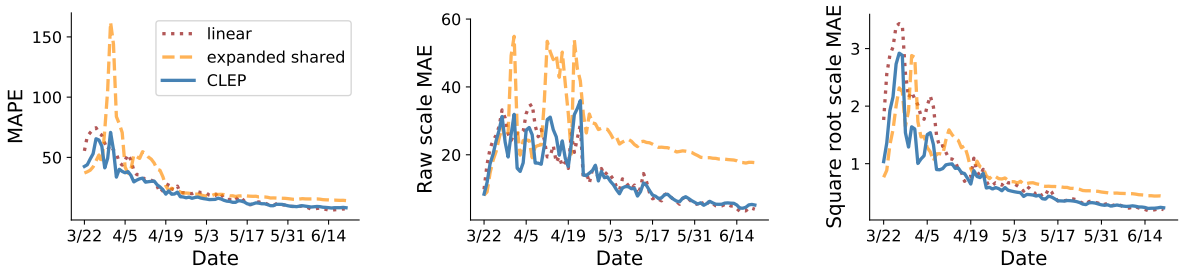
Lastly, inspired by the recent work of [19], we are beginning our investigation into adaptive tuning (over time) of μ, c and other hyperparameters for CLEP, in the hope of improving its performance.

(a) Six worst-affected counties



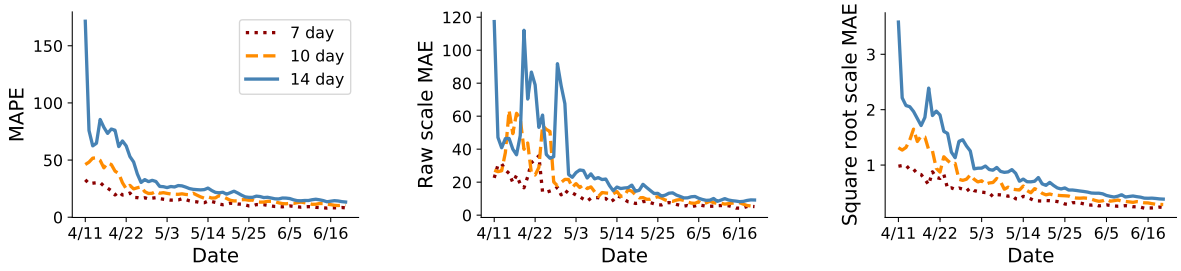
(b) Six randomly selected counties

Figure 5.5: EDA plot for investigating exchangeability of normalized errors of *7-day-ahead* CLEP predictions with its last 5 errors made at time t, over the period $\mathcal{T} = \{\text{March 26}, \ldots, \text{Jun 13}\}$ (80 days). These heatmaps are obtained as follows. First, for each day t, we rank the errors $\{\Delta_{t+7}, \Delta_t, \Delta_{t-1}, \ldots, \Delta_{t-4}\}$ of our CLEP (with the expanded shared and linear predictors) in increasing order so that the largest error has a rank of 6. Then, for each of the six errors $\{\Delta_{t+7}, \Delta_t, \Delta_{t-1}, \ldots, \Delta_{t-4}\}$, we count the number of days when it is ranked 1, 2, 3, 4, 5 and 6. Finally, we plot these numbers of days in the above heatmaps for **(a)** the six worst affected counties, and **(b)** six random counties. In addition, we plot the average rank of each error in dashed black lines. If $\{\Delta_{t+7}, \Delta_t, \Delta_{t-1}, \ldots, \Delta_{t-4}\}$ are exchangeable for any day t, then the expected average rank for each of the six errors would be 3.5.



(a) MAPE (b) Raw-scale MAE (c) Square-root-scale MAE

Figure 5.6: Plots of mean absolute error (MAE) of different predictors for *7-day-ahead* predictions from March 22 to June 20. We plot the **(a)** mean absolute percentage error (MAPE), **(b)** raw-scale MAE, and **(c)** square-root-scale MAE versus time. Results are shown for expanded shared exponential predictor (orange dashed line), the separate county linear predictor (red dotted line), and the CLEP that combines the two predictors (solid blue line).



(a) MAPE (b) Raw-scale MAE (c) Square-root-scale MAE

Figure 5.9: Plots of mean absolute error (MAE) of CLEP with different prediction horizons from April 11, 2020 to June 20, 2020. We plot the **(a)** mean absolute percentage error (MAPE), **(b)** raw-scale MAE, and **(c)** square-root-scale MAE versus time, for k-day-ahead predictions with $k \in \{7, 10, 14\}$.

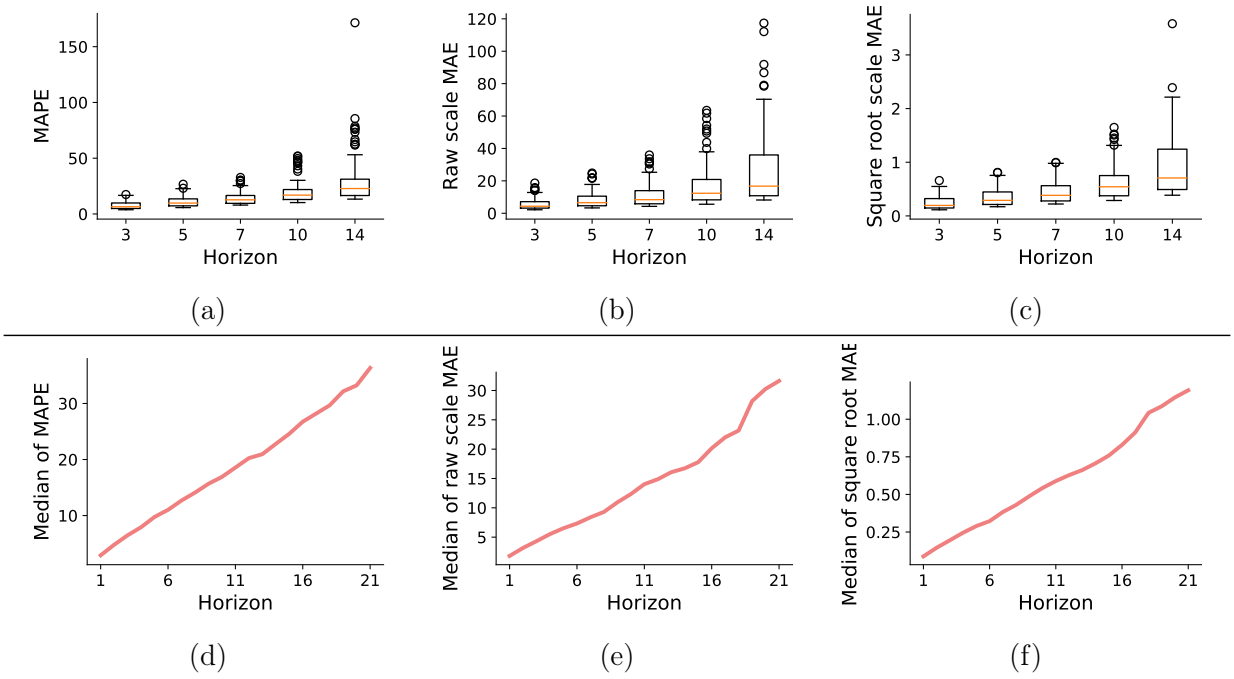
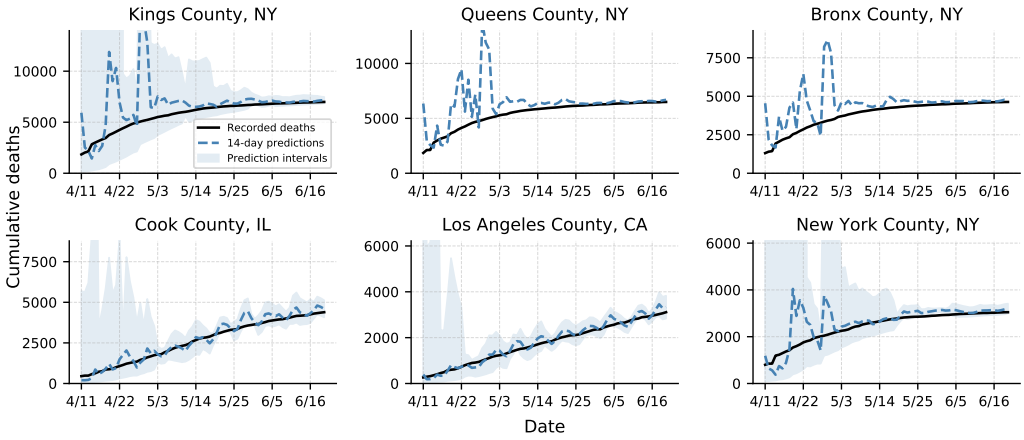
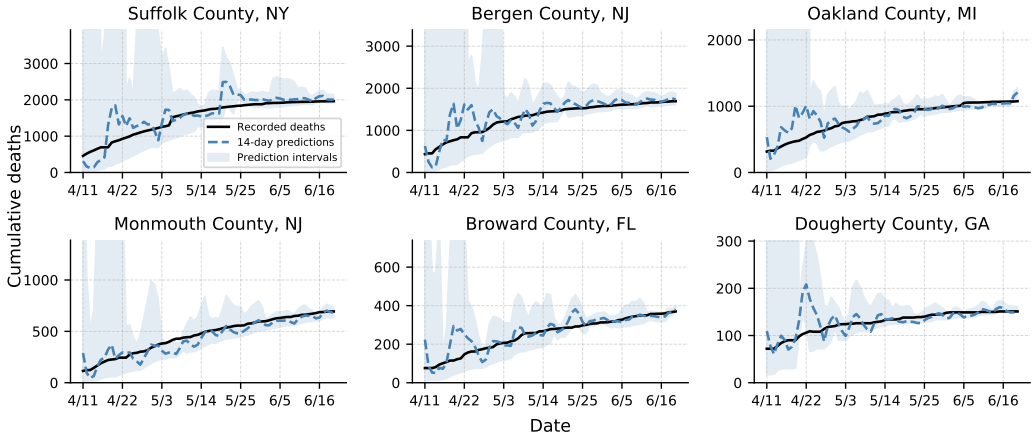


Figure 5.10: Plots of MAE distribution of CLEP for different prediction horizon over the period April 11, 2020 to June 20, 2020. In panels **(a)**, **(b)** and **(c)**, we show the box-plots (over time) of the mean absolute percentage error (MAPE), raw-scale MAE, and square-root-scale MAE for k-day ahead predictions $k = \{3, 5, 7, 10, 14\}$. In panels **(d)**, **(e)** and **(f)**, we plot the median value of the different MAEs for even longer horizons, up to 21 days.



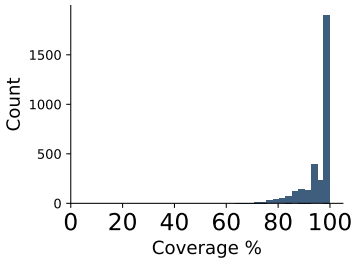
(a) Worst-affected counties on June 20



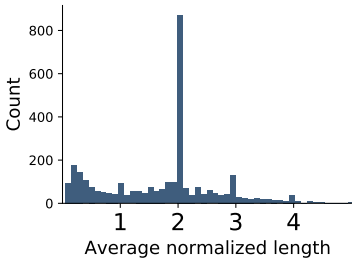
(b) Randomly-selected counties

Figure 5.12: A grid of line charts displaying the performance of *14-day-ahead* CLEP and MEPI for the cumulative death counts due to COVID-19 between April 11, 2020 and June 20, 2020. The observed data is shown in black, CLEP predictions are shown in the dashed blue, and the corresponding 14-day-ahead MEPIs are shown as shaded blue regions. In panel **(a)**, the MEPI coverage for the 6 counties are Kings (99%), Queens (99%), Bronx (99%), Cook (93%), Los Angeles (96%) and New York (89%). In panel **(b)**, Suffolk (92%), Bergen (99%), Oakland (89%), Monmouth (94%), Broward (97%) and Dougherty county (94%). Note that for these counties, the coverage of 14-day-ahead MEPIs is higher than that of 7-day-ahead MEPIs (shown in Figure 5.11) due to the wider intervals in the beginning of the period. (It is worth noting that the theoretical coverage guarantees under certain assumptions is 83%; see equation (5.13).)

Evaluation period: April 11–May 10

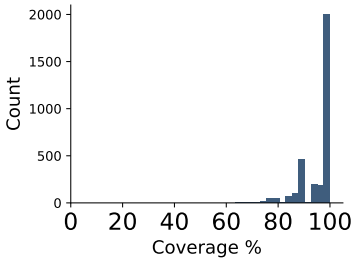


(a) Coverage for all counties

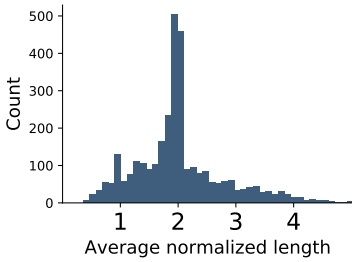


(b) Average length for all counties

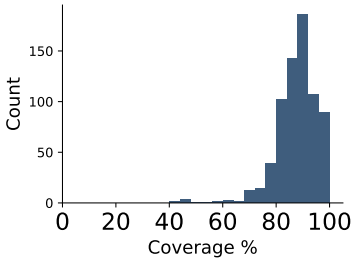
Evaluation period: May 11–June 20



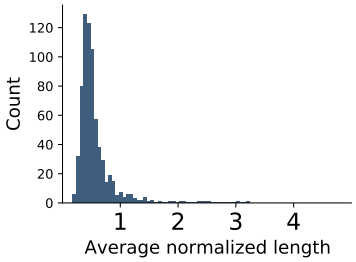
(c) Coverage for all counties



(d) Average length for all counties



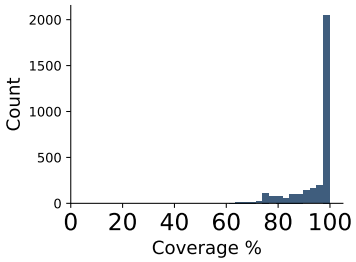
(e) Coverage for selected counties



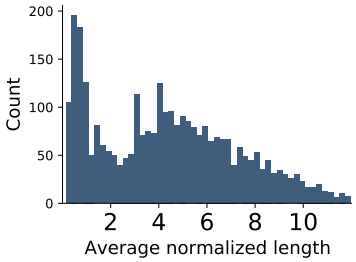
(f) Average length for selected counties

Figure 5.13: Histograms showing the performance of *7-day-ahead MEPI intervals* for county-level cumulative death counts. For each county, we compute the observed coverage and average normalized length for the period April 11–May 10, 2020, and plot the histogram of these values across counties respectively in the top panels **(a)** and **(b)**. Respective results for the period May 11–June 20, 2020 are plotted in the middle panels **(c)** and **(d)**. For the bottom two panels **(e)** and **(f)**, we plot the histogram across only those counties that had at least 10 cumulative deaths by June 11. In these two panels, for each county the observed coverage and average normalized length are computed only over those days in April 11 to June 20 for which that county's cumulative death count is at least 10. See Figure 5.14 for similar plots for 14-day-ahead MEPIs.

Evaluation period: April 11–May 10

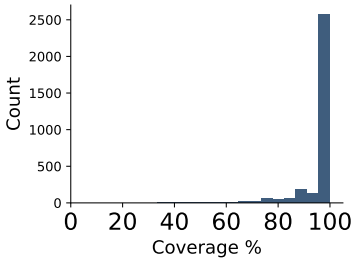


(a) Coverage for all counties

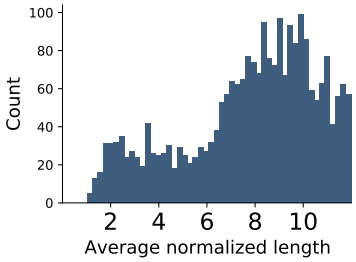


(b) Average length for all counties

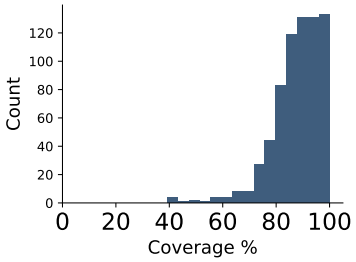
Evaluation period: May 11–June 20



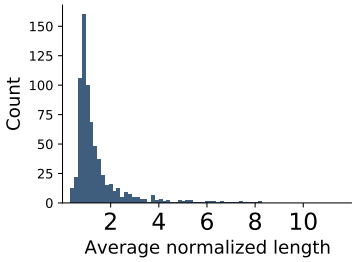
(c) Coverage for all counties



(d) Average length for all counties



(e) Coverage for selected counties



(f) Average length for selected counties

Figure 5.14: Histograms showing the performance of the *14-day-ahead MEPI intervals* for county-level cumulative death counts. For each county, we compute the observed coverage and average normalized length for the period April 11–May 10, 2020, and plot the histogram of these values across counties respectively in the top panels **(a)** and **(b)**. Respective results for the period May 11–June 20, 2020 are plotted in the middle panels **(c)** and **(d)**. For the bottom two panels **(e)** and **(f)**, we plot the histogram across only those counties that had at least 10 cumulative deaths by June 11. In these two panels, for each county the observed coverage and average normalized length are computed only over those days in April 11 to June 20 for which that county's cumulative death count is at least 10. See Figure 5.13 for similar plots for 7-day-ahead MEPIs.

Bibliography

[1] National Research Council (US) Committee on A Framework for Developing a New Taxonomy of Disease. *Toward Precision Medicine: Building a Knowledge Network for Biomedical Research and a New Taxonomy of Disease*. eng. The National Academies Collection: Reports funded by National Institutes of Health. Washington (DC): National Academies Press (US), 2011. ISBN: 978-0-309-22222-8. URL: http://www.ncbi.nlm.nih.gov/books/NBK91503/ (visited on 05/06/2020).

[2] Mohammed Alawad et al. "Automatic extraction of cancer registry reportable information from free-text pathology reports using multitask convolutional neural networks". In: *Journal of the American Medical Informatics Association* 27.1 (Nov. 2019), pp. 89–98. ISSN: 1527-974X. DOI: 10.1093/jamia/ocz153. eprint: https://academic.oup.com/jamia/article-pdf/27/1/89/33744947/ocz153.pdf. URL: https://doi.org/10.1093/jamia/ocz153.

[3] Mohammed Alawad et al. "Automatic extraction of cancer registry reportable information from free-text pathology reports using multitask convolutional neural networks". In: *Journal of the American Medical Informatics Association* 27.1 (2020), pp. 89–98.

[4] Christoph Alt, Marc Hübner, and Leonhard Hennig. *Improving Relation Extraction by Pre-trained Language Representations*. June 2019.

[5] Anastasios Nikolas Angelopoulos et al. "On Identifying and Mitigating Bias in the Estimation of the COVID-19 Case Fatality Rate". In: *Harvard Data Science Review* (June 9, 2020). URL: https://hdsr.mitpress.mit.edu/pub/y9vc2u36.

[6] Apple Inc. "Apple Mobility Trends Reports". In: (2020). Accessed on 05-15-2020 at https://www.apple.com/covid19/mobility.

[7] *Aspect-augmented Adversarial Networks for Domain Adaptation | Transactions of the Association for Computational Linguistics | MIT Press Journals*. URL: https://www.mitpressjournals.org/doi/abs/10.1162/tacl_a_00077 (visited on 02/24/2020).

[8] Michael Becker and Corey Chivers. "Announcing CHIME, A tool for COVID-19 capacity planning". In: (2020). Accessed on 04-02-2020 at http://predictivehealthcare.pennmedicine.org/2020/03/14/accouncing-chime.html.

[9] James Bergstra and Yoshua Bengio. "Random Search for Hyper-Parameter Optimization". In: *Journal of Machine Learning Research* 13.Feb (2012), pp. 281–305. ISSN: ISSN 1533-7928. URL: http://www.jmlr.org/papers/v13/bergstra12a.html?source=post_page---------- (visited on 02/24/2020).

[10] Dennis J. Bregman and Alexander D. Langmuir. "Farr's Law Applied to AIDS Projections". In: *JAMA* 263.11 (Mar. 1990), pp. 1522–1525. ISSN: 0098-7484. DOI: 10.1001/jama.1990.03440110088033. URL: https://doi.org/10.1001/jama.1990.03440110088033.

[11] Bureau of Transportation Statistics. "Airline Origin and Destination Survey (DB1B)". In: (2020). Accessed on 04-20-2020 at https://transtats.bts.gov/Databases.asp?Mode_ID=1&Mode_Desc=Aviation&Subject_ID2=0.

[12] Gerard Burger et al. "Natural language processing in pathology: a scoping review". eng. In: *Journal of Clinical Pathology* (July 2016). ISSN: 1472-4146. DOI: 10.1136/jclinpath-2016-203872.

[13] Centers for Disease Control and Prevention. "Interactive Atlas of Heart Disease and Stroke". In: (2018). Accessed on 04-02-2020 at http://nccd.cdc.gov/DHDSPAtlas.

[14] Centers for Disease Control and Prevention, Agency for Toxic Substances and Disease Registry, and Geospatial Research, Analysis, and Services Program. "Social Vulnerability Index Database". In: (2018). Accessed on 04-03-2020 at https://svi.cdc.gov/data-and-tools-download.html.

[15] Centers for Disease Control and Prevention, Division of Diabetes Translation, and US Diabetes Surveillance System. "Diagnosed Diabetes Atlas". In: (2016). Accessed on 04-02-2020 at https://www.cdc.gov/diabetes/data.

[16] Centers for Medicare & Medicaid Services. "Chronic Conditions Prevalence State/County Level: All Beneficiaries by Age, 2007-2017". In: (2017). Accessed on 04-02-2020 at https://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/Chronic-Conditions/CC_Main.

[17] Centers for Medicares & Medicaid Services. "2020 Reporting Cycle: Teaching Hospital List". In: (2020). Accessed on 04-01-2020 at https://www.cms.gov/OpenPayments/Downloads/2020-Reporting-Cycle-Teaching-Hospital-List-PDF-.pdf.

[18] Centers for Medicares & Medicaid Services. "Case Mix Index File". In: (2018). Accessed on 04-01-2020 at https://www.cms.gov/Medicare/Medicare-Fee-for-Service-Payment/AcuteInpatientPPS/FY2020-IPPS-Final-Rule-Home-Page-Items/FY2020-IPPS-Final-Rule-Data-Files.

[19] Wen-Hao Chiang, Xueying Liu, and George Mohler. "Hawkes process modeling of COVID-19 with mobility leading indicators and spatial covariates". In: *medRxiv preprint 2020.06.06.20124149* (2020).

[20] Kyunghyun Cho et al. "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation". en-us. In: Oct. 2014, pp. 1724–1734. DOI: 10.3115/v1/D14-1179. URL: https://www.aclweb.org/anthology/D14-1179 (visited on 02/24/2020).

[21] Anni Coden et al. "Automatically extracting cancer disease characteristics from pathology reports into a Disease Knowledge Representation Model". en. In: *Journal of Biomedical Informatics* 42.5 (Oct. 2009), pp. 937–949. ISSN: 15320464. DOI: 10.1016/j.jbi.2008.12.005. URL: http://linkinghub.elsevier.com/retrieve/pii/S1532046408001585 (visited on 01/27/2018).

[22] County Health Rankings & Roadmaps. "County Health Rankings & Roadmaps 2020 Measures". In: (2020). Accessed on 04-02-2020 at https://www.countyhealthrankings.org/explore-health-rankings/measures-data-sources/2020-measures.

[23] Definitive Healthcare. "Definitive Healthcare: USA Hospital Beds". In: (2020). Accessed on 04-01-2020 at https://coronavirus-resources.esri.com/datasets/definitivehc::definitive-healthcare-usa-hospital-beds.

[24] Morris H. Degroot and Stephen E. Fienberg. "The Comparison and Evaluation of Forecasters". en. In: *Journal of the Royal Statistical Society: Series D (The Statistician)* 32.1-2 (1983), pp. 12–22. ISSN: 1467-9884. DOI: 10.2307/2987588. URL: https://rss.onlinelibrary.wiley.com/doi/abs/10.2307/2987588 (visited on 01/18/2020).

[25] Louise Deleger et al. "Building Gold Standard Corpora for Medical Natural Language Processing Tasks". In: *AMIA Annual Symposium Proceedings* 2012 (Nov. 2012), pp. 144–153. ISSN: 1942-597X. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3540456/ (visited on 01/25/2018).

[26] Leon Derczynski. "Complementarity, F-score, and NLP Evaluation". en. In: (), p. 6.

[27] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". en. In: (Oct. 2018). URL: https://arxiv.org/abs/1810.04805v2 (visited on 06/21/2020).

[28] Ensheng Dong, Hongru Du, and Lauren Gardner. "An interactive web-based dashboard to track COVID-19 in real time". In: *The Lancet infectious diseases* 20.5 (2020), pp. 533–534.

[29] Glenn A. Edwards. "Expert systems for clinical pathology reporting". eng. In: *The Clinical Biochemist. Reviews* 29 Suppl 1 (Aug. 2008), S105–109. ISSN: 0159-8090.

[30] Haytham H Elmousalami and Aboul Ella Hassanien. "Day Level Forecasting for Coronavirus Disease (COVID-19) Spread: Analysis, Modeling and Recommendations". In: *arXiv preprint arXiv:2003.07778* (2020).

[31] Duccio Fanelli and Francesco Piazza. "Analysis and forecast of COVID-19 spreading in China, Italy and France". In: *Chaos, Solitons & Fractals* 134 (2020), p. 109761.

[32] NM Ferguson et al. *Impact of non-pharmaceutical interventions (NPIs) to reduce COVID19 mortality and healthcare demand*. Accessed on 04-02-2020 at `https://www.imperial.ac.uk/media/imperial-college/medicine/sph/ide/gida-fellowships/Imperial-College-COVID19-NPI-modelling-16-03-2020.pdf`. 2020.

[33] Allan Fong et al. "Assessment of Automating Safety Surveillance From Electronic Health Records: Analysis for the Quality and Safety Review System". In: *Journal of patient safety* (2016).

[34] Shang Gao et al. "Hierarchical attention networks for information extraction from cancer pathology reports". en. In: *Journal of the American Medical Informatics Association* 25.3 (Mar. 2018), pp. 321–330. ISSN: 1067-5027, 1527-974X. DOI: `10.1093/jamia/ocx131`. URL: `https://academic.oup.com/jamia/article/25/3/321/4636780` (visited on 10/09/2019).

[35] Shang Gao et al. "Hierarchical attention networks for information extraction from cancer pathology reports". en. In: *Journal of the American Medical Informatics Association* 25.3 (Mar. 2018), pp. 321–330. ISSN: 1067-5027. DOI: `10.1093/jamia/ocx131`. URL: `https://academic.oup.com/jamia/article/25/3/321/4636780` (visited on 02/24/2020).

[36] Alexander P. Glaser et al. "Automated Extraction of Grade, Stage, and Quality Information From Transurethral Resection of Bladder Tumor Pathology Reports Using Natural Language Processing". In: *JCO Clinical Cancer Informatics* 2 (2018), pp. 1–8. ISSN: 2473-4276. DOI: `10.1200/CCI.17.00128`. URL: `http://ascopubs.org/doi/10.1200/CCI.17.00128`.

[37] Ken J Goh, Shirin Kalimuddin, and Kian Sing Chan. "Rapid Progression to Acute Respiratory Distress Syndrome: Review of Current Understanding of Critical Illness from COVID-19 Infection." In: *Annals of the Academy of Medicine, Singapore* 49.1 (2020), p. 1.

[38] Google LLC. "Google COVID-19 Community Mobility Reports". In: (2020). Accessed on 05-15-2020 at `https://www.google.com/covid19/mobility/`.

[39] Weijie Guan et al. "Clinical characteristics of Coronavirus disease 2019 in China". In: *New England Journal of Medicine* (2020).

[40] Weijie Guan et al. "Comorbidity and its impact on 1590 patients with COVID-19 in China: A Nationwide Analysis". In: *European Respiratory Journal* (2020).

[41] Health Resources and Services Administration. "Area Health Resources Files". In: (2019). Accessed on 04-02-2020 at `https://data.hrsa.gov/data/download`.

[42] Health Resources and Services Administration. "Health Professional Shortage Areas - Primary Care". In: (2020). Accessed on 04-04-2020 at `https://data.hrsa.gov/data/download`.

[43] Homeland Infrastructure Foundation-Level Data. "Hospitals". In: (2020). Accessed on 06-23-2020 at https://hifld-geoplatform.opendata.arcgis.com/datasets/6ac5e325468c4cb9b905f1728d6fbf0f_0.

[44] Jeremy Howard and Sebastian Ruder. "Universal Language Model Fine-tuning for Text Classification". en. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, 2018, pp. 328–339. DOI: 10.18653/v1/P18-1031. URL: http://aclweb.org/anthology/P18-1031 (visited on 05/22/2020).

[45] Solomon Hsiang et al. "The Effect of Large-Scale Anti-Contagion Policies on the Coronavirus (COVID-19) Pandemic". In: *medRxiv* (2020). DOI: 10.1101/2020.03.22.20040642. URL: https://www.medrxiv.org/content/early/2020/03/31/2020.03.22.20040642.

[46] Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. "ClinicalBERT: Modeling Clinical Notes and Predicting Hospital Readmission". en. In: (Apr. 2019). URL: https://arxiv.org/abs/1904.05342v2 (visited on 06/21/2020).

[47] Institute for Health Metrics and Evaluation. "United States Chronic Respiratory Disease Mortality Rates by County 1980-2014". In: (2017). Accessed on 04-02-2020 at http://ghdx.healthdata.org/record/ihme-data/united-states-chronic-respiratory-disease-mortality-rates-county-1980-2014.

[48] Sarthak Jain and Byron C. Wallace. "Attention is not Explanation". In: *arXiv:1902.10186 [cs]* (May 2019). arXiv: 1902.10186. URL: http://arxiv.org/abs/1902.10186 (visited on 04/23/2020).

[49] V. Jouhet et al. "Automated classification of free-text pathology reports for registration of incident cases of cancer". eng. In: *Methods of Information in Medicine* 51.3 (2012), pp. 242–251. ISSN: 2511-705X. DOI: 10.3414/ME11-01-0005.

[50] Dan Jurafsky and James H. Martin. *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition*. 2nd ed. Prentice Hall series in artificial intelligence. Pearson Prentice Hall, 2009. ISBN: 978-0-13-187321-6.

[51] Kaiser Health News. "ICU Beds by County". In: (2020). Accessed on 04-02-2020 at https://khn.org/news/as-coronavirus-spreads-widely-millions-of-older-americans-live-in-counties-with-no-icu-beds/.

[52] Hidetaka Kamigaito et al. "Supervised Attention for Sequence-to-Sequence Constituency Parsing". en-us. In: Nov. 2017, pp. 7–12. URL: https://www.aclweb.org/anthology/I17-2002 (visited on 02/24/2020).

[53] Benjamin D Killeen et al. "A County-level Dataset for Informing the United States' Response to COVID-19". In: *arXiv preprint arXiv:2004.00756* (2020).

[54] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". en. In: (Dec. 2014). URL: https://arxiv.org/abs/1412.6980v9 (visited on 02/24/2020).

[55] Kory Kreimeyer et al. "Natural language processing systems for capturing and standardizing unstructured clinical information: A systematic review". In: *Journal of Biomedical Informatics* 73 (Sept. 2017), pp. 14–29. DOI: 10.1016/j.jbi.2017.07.012. URL: http://dx.doi.org/10.1016/j.jbi.2017.07.012.

[56] Adam J Kucharski et al. "Early dynamics of transmission and control of COVID-19: A mathematical modelling study". In: *medRxiv* (2020). DOI: 10.1101/2020.01.31.20019901. URL: https://www.medrxiv.org/content/early/2020/02/18/2020.01.31.20019901.

[57] Jinhyuk Lee et al. "BioBERT: a pre-trained biomedical language representation model for biomedical text mining". en. In: (Jan. 2019). DOI: 10.1093/bioinformatics/btz682. URL: https://arxiv.org/abs/1901.08746v4 (visited on 06/21/2020).

[58] Yue Li and David Martinez. "Information Extraction of Multiple Categories from Pathology Reports". In: *Proceedings of the Australasian Language Technology Association Workshop 2010*. Melbourne, Australia, Dec. 2010, pp. 41–48. URL: https://www.aclweb.org/anthology/U10-1008 (visited on 12/16/2019).

[59] Lemao Liu et al. "Neural Machine Translation with Supervised Attention". en-us. In: Dec. 2016, pp. 3093–3102. URL: https://www.aclweb.org/anthology/C16-1291 (visited on 02/24/2020).

[60] Roman Marchant et al. "Learning as We Go: An Examination of the Statistical Accuracy of COVID19 Daily Death Count Predictions". In: *arXiv preprint arXiv:2004.04734* (2020).

[61] David Martinez and Yue Li. "Information extraction from pathology reports in a hospital setting". In: *Proceedings of the 20th ACM international conference on Information and knowledge management*. 2011, pp. 1877–1882.

[62] MIT Election Data and Science Lab. "County Presidential Election Returns 2000-2016". Version V6. In: (2018). DOI: 10.7910/DVN/VOQCHQ. URL: https://doi.org/10.7910/DVN/VOQCHQ.

[63] Christopher JL Murray and Institute Health Metrics Evaluation COVID-19 health service utilization forecasting team. "Forecasting COVID-19 impact on hospital bed-days, ICU-days, ventilator-days and deaths by US state in the next 4 months". In: *medRxiv* (2020). DOI: 10.1101/2020.03.27.20043752. URL: https://www.medrxiv.org/content/early/2020/03/30/2020.03.27.20043752.

[64] Giulio Napolitano et al. "Pattern-based information extraction from pathology reports for cancer registration". en. In: *Cancer Causes & Control* 21.11 (Nov. 2010), pp. 1887–1894. ISSN: 0957-5243, 1573-7225. DOI: 10.1007/s10552-010-9616-4. URL: http://link.springer.com/10.1007/s10552-010-9616-4 (visited on 07/22/2019).

[65] S. Nebehay and K. Kelland. "COVID-19 cases and deaths rising, debt relief needed for poorest nations: WHO". In: *Reuters* (Apr. 1, 2020). Accessed on 04-01-2020 at https://www.reuters.com/article/us-health-coronavirus-who/covid-19-infections-growing-exponentially-deaths-nearing-50000-who-idUSKBN21J6IL?il=0.

[66] Anthony N Nguyen et al. "Symbolic rule-based classification of lung cancer stages from free-text pathology reports". en. In: *Journal of the American Medical Informatics Association* 17.4 (July 2010), pp. 440–445. ISSN: 1527-974X, 1067-5027. DOI: 10.1136/jamia.2010.003707. URL: https://academic.oup.com/jamia/article-lookup/doi/10.1136/jamia.2010.003707 (visited on 07/22/2019).

[67] Anobel Odisho* et al. "PD58-09 EXTRACTING STRUCTURED INFORMATION FROM PATHOLOGY REPORTS USING NATURAL LANGUAGE PROCESSING AND MACHINE LEARNING". en. In: *The Journal of Urology* (Apr. 2019). URL: https://www.auajournals.org/doi/abs/10.1097/01.JU.0000557177.97226.63 (visited on 02/24/2020).

[68] Philip V. Ogren et al. "Building and Evaluating Annotated Corpora for Medical NLP Systems". In: *AMIA Annual Symposium Proceedings* 2006 (2006), p. 1050. ISSN: 1942-597X. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1839264/ (visited on 01/27/2018).

[69] Tomasz Oliwa et al. "Obtaining Knowledge in Pathology Reports Through a Natural Language Processing Approach With Classification, Named-Entity Recognition, and Relation-Extraction Heuristics". In: *JCO Clinical Cancer Informatics* 3 (July 2019). Publisher: American Society of Clinical Oncology, pp. 1–8. DOI: 10.1200/CCI.19.00008. URL: https://ascopubs.org/doi/full/10.1200/CCI.19.00008 (visited on 04/18/2020).

[70] Corey M Peak et al. "Modeling the Comparative Impact of Individual Quarantine vs. Active Monitoring of Contacts for the Mitigation of COVID-19". In: *medRxiv* (2020). DOI: 10.1101/2020.03.05.20031088. URL: https://www.medrxiv.org/content/early/2020/03/08/2020.03.05.20031088.

[71] Fabian Pedregosa et al. "Scikit-learn: Machine learning in Python". In: *Journal of machine learning research* 12.Oct (2011), pp. 2825–2830.

[72] Sen Pei and Jeffrey Shaman. "Initial Simulation of SARS-CoV2 Spread and Intervention Effects in the Continental US". In: *medRxiv* (2020). DOI: 10.1101/2020.03.21.20040303. URL: https://www.medrxiv.org/content/early/2020/03/27/2020.03.21.20040303.

[73] Matthew E. Peters et al. "Deep contextualized word representations". en. In: (Feb. 2018). URL: https://arxiv.org/abs/1802.05365v2 (visited on 06/21/2020).

[74] Sampo Pyysalo et al. *Distributional Semantics Resources for Biomedical Text Processing*. en. Library Catalog: www.semanticscholar.org. 2013. URL: /paper/Distributional-Semantics-Resources-for-Biomedical-Pyysalo-Ginter/e2f28568031e1902d4f8ee818261f0f2 (visited on 06/21/2020).

[75] Di Qi et al. "Epidemiological and clinical features of 2019-nCoV acute respiratory disease cases in Chongqing municipality, China: A retrospective, descriptive, multiple-center study". In: *medRxiv* (2020). DOI: 10.1101/2020.03.01.20029397. URL: https://www.medrxiv.org/content/early/2020/03/03/2020.03.01.20029397.

[76] John X. Qiu et al. "Deep Learning for Automated Extraction of Primary Sites From Cancer Pathology Reports". eng. In: *IEEE journal of biomedical and health informatics* 22.1 (2018), pp. 244–251. ISSN: 2168-2208. DOI: 10.1109/JBHI.2017.2700722.

[77] Angus Roberts et al. "The CLEF Corpus: Semantic Annotation of Clinical Text". In: *AMIA Annual Symposium Proceedings* 2007 (2007), pp. 625–629. ISSN: 1942-597X. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2655900/ (visited on 01/27/2018).

[78] Lewis Rubinson et al. "Mechanical ventilators in US acute care hospitals". In: *Disaster medicine and public health preparedness* 4.3 (2010), pp. 199–206.

[79] Florian R Schroeck et al. "Development of a Natural Language Processing Engine to Generate Bladder Cancer Pathology Data for Health Services Research". In: *URL* 110 (Dec. 2017), pp. 84–91. DOI: 10.1016/j.urology.2017.07.056. URL: https://doi.org/10.1016/j.urology.2017.07.056.

[80] Gerald DT Schuller et al. "Perceptual audio coding using adaptive pre-and post-filters and lossless compression". In: *IEEE Transactions on Speech and Audio Processing* 10.6 (2002), pp. 379–390.

[81] Skipper Seabold and Josef Perktold. "statsmodels: Econometric and statistical modeling with python". In: *9th Python in Science Conference*. 2010.

[82] Glenn Shafer and Vladimir Vovk. "A tutorial on conformal prediction". In: *Journal of Machine Learning Research* 9.Mar (2008), pp. 371–421.

[83] Yanyao Shen et al. "Deep Active Learning for Named Entity Recognition". en. In: (July 2017). URL: https://arxiv.org/abs/1707.05928v3 (visited on 05/22/2020).

[84] Yuqi Si and Kirk Roberts. "A Frame-Based NLP System for Cancer-Related Information Extraction". In: *AMIA Annual Symposium Proceedings* 2018 (Dec. 2018), pp. 1524–1533. ISSN: 1942-597X. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6371330/ (visited on 04/18/2020).

[85] Rebecca L. Siegel, Kimberly D. Miller, and Ahmedin Jemal. "Cancer statistics, 2020". en. In: *CA: A Cancer Journal for Clinicians* 70.1 (Jan. 2020), pp. 7–30. ISSN: 0007-9235, 1542-4863. DOI: 10.3322/caac.21590. URL: https://onlinelibrary.wiley.com/doi/abs/10.3322/caac.21590 (visited on 04/15/2020).

[86] Brett R South et al. "Developing a manually annotated clinical document corpus to identify phenotypic information for inflammatory bowel disease". en. In: *BMC Bioinformatics* 10.Suppl 9 (2009), S12. ISSN: 1471-2105. DOI: 10.1186/1471-2105-10-S9-S12. URL: http://www.biomedcentral.com/1471-2105/10/S9/S12 (visited on 01/27/2018).

[87] The Institute for Health Metrics and Evaluation. *COVID-19: What's New for April 5, 2020*. http://www.healthdata.org/sites/default/files/files/Projects/COVID/Estimation_update_040520_3.pdf, Last accessed on 2020-04-13. 2020.

[88] The New York Times. *COVID-19 Data in the United States*. https://github.com/nytimes/covid-19-data. Accessed on 04-01-2020 at https://github.com/nytimes/covid-19-data. 2020.

[89] United States Census Bureau. "County Adjacency File". In: (2018). Accessed on 05-15-2020 at https://www.census.gov/geographies/reference-files/2010/geo/county-adjacency.html.

[90] United States Department of Agriculture, Economic Research Service. "Poverty estimates for the U.S., states, and counties". In: (2018). Accessed on 04-24-2020 at https://www.ers.usda.gov/data-products/county-level-data-sets/download-data/.

[91] United States Department of Health and Human Services, Centers for Disease Control and Prevention, and National Center for Health Statistics. "Compressed Mortality File (CMF) on CDC WONDER Online Database, 2012-2016". In: (2017). Accessed on 04-02-2020 at https://wonder.cdc.gov/cmf-icd10.html.

[92] USAFacts. "COVID-19 Deaths Data". In: (2020). Accessed on 03-31-2020 at https://www.reuters.com/article/us-health-coronavirus-who/covid-19-spread-map.

[93] Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer Science & Business Media, 2005.

[94] Chaolong Wang et al. "Evolving Epidemiology and Impact of Non-pharmaceutical Interventions on the Outbreak of Coronavirus Disease 2019 in Wuhan, China". In: *medRxiv* (2020). DOI: 10.1101/2020.03.03.20030593. URL: https://www.medrxiv.org/content/early/2020/03/06/2020.03.03.20030593.

[95] Yanshan Wang et al. "Clinical information extraction applications: a literature review". In: *Journal of biomedical informatics* 77 (2018), pp. 34–49.

[96] Jin Wu et al. "109,000 Missing Deaths: Tracking the True Toll of the Coronavirus Outbreak". In: *The New York Times* (2020).

[97] Jun Xu et al. "Applying a deep learning-based sequence labeling approach to detect attributes of medical concepts in clinical text". en. In: *BMC Medical Informatics and Decision Making* 19.5 (Dec. 2019), p. 236. ISSN: 1472-6947. DOI: 10.1186/s12911-019-0937-2. URL: https://doi.org/10.1186/s12911-019-0937-2 (visited on 04/18/2020).

[98] Adam Yala et al. "Using machine learning to parse breast pathology reports". eng. In: *Breast Cancer Research and Treatment* 161.2 (2017), pp. 203–211. ISSN: 1573-7217. DOI: 10.1007/s10549-016-4035-1.

[99] Wen-wai Yim et al. "Natural Language Processing in Oncology: A Review". en. In: *JAMA Oncology* 2.6 (June 2016), p. 797. ISSN: 2374-2437. DOI: 10.1001/jamaoncol.2016.0213. URL: http://oncology.jamanetwork.com/article.aspx?doi=10.1001/jamaoncol.2016.0213 (visited on 07/22/2019).

[100] Bianca Zadrozny and Charles Elkan. *Transforming Classifier Scores into Accurate Multiclass Probability Estimates*. 2002.

[101] Fei Zhou et al. "Clinical course and risk factors for mortality of adult inpatients with COVID-19 in Wuhan, China: A retrospective cohort study". In: *The Lancet* (2020).

Appendix A

Appendix

SUPPLEMENTAL NOTES

Corpus Statistics

The prostate cancer reports have a mean length of 912 words and 22382 unique words in the vocabulary. After replacing rare words with the special <UNK> token, there are 6803 unique words in the vocabulary. We include the percentage of reports containg each data element in supplemental table 6.

Text Preprocessing Methods

We removed commas, backslashes, semi-colons, tildes, periods, and the word "null" from each report in the corpus. For colons, forward slashes, parentheses, plus, and equal signs we added a space before and after the character, because these frequently had semantic value important for the classification task, for example colons typically occur in the synoptic comment where most of the relevant is contained, so if an n-gram has a colon in it, then this is indicative that it may contain important information. Each report was preprocessed so that every token in the report had one space preceding and one space following it. We then created a vocabulary of words using only reports in the traing corpus. If a token occurred fewer than 10 times in the traing corpus vocabulary, it was replaced with a "<UNK>" token for every report.

Pathologic Stage Information

We applied the token extractor methods rather than the classifiers to the pathologic stages, even though the number of classes for each stage type is small. The reason is the way the pathologic stages are encoded in the reports. For example, each pathologic stage is encoded in a single token, such as *pt2an0m0.* The letters *t, n,* and *m* denote the pathologic stage type, while *2a* and *0s* that follow immediately denote the class for each stage type. Classifier methods are not well suited for this task, because the number of possible token encodings is large (equal to the number of classes for the t-stage multiplied by the number of classes for the n and m-stages). It may even be the case that a certain encoding may show up in the test set and does not show up in the train set if a certain combination of the stages is new. For this reason, we first extract the stage token (eg *pt2an0m0*) and then determine the values for the *t, n,* and *m* stages using a regular expression rule-based method.

Model hyperparameters

We sampled regularization values from -6 to 6 in logspace for logistic regression models. For the random forest, we varied parameters such as bootstrapping or not, the max depth of each decision tree from 10 to 50 in increments of 10, the number of trees from 200 to 1000 in increments of 200, and the minimum number of samples per leaf from 2 to 64 by factors of two. For the support vector machines method, we sampled the regularization parameter from -6 to 6 in logspace, the kernel from either linear or radial basis kernel, and the gamma parameter from -3 to 1 in logspace. For the boosting classifier we sampled the learning rate from -4 to 1 in logspace using the SAMME.R algorithm in scikit-learn.

For the convolutional neural network, we sampled the learning rate from -6 to -1 in logspace, the number of convolutional filters from 50 to 400 in increments of 50, drop out parameter from 0, 0.125, 0.25, and 0.5, batch sizes from 16 to 32, and filter sizes from 3 to 6. For the LSTM, we sampled the learning rate from -6 to -1 in logspace, the drop out from 0, 0.125, 0.25, and 0.5, the hidden dimension size from 50 to 300 in increments of 50.

Supplemental Tables

Supplemental Table 1. Macro F1 scores for classification fields across on full traing data sample (n = 2,066)

Data Element	**LSTM**	**Logistic regression**	**Adaboost classifier**	**SVM**	**Random Forest**
Gleason Grade - Primary	0.329	0.864	0.865	0.751	0.624
Gleason Grade - Secondary	0.300	0.680	0.561	0.649	0.536
Gleason Grade - Tertiary	0.268	0.692	0.771	0.492	0.385
Tumor histology	0.498	0.496	0.499	0.498	0.499
Cribriform pattern	0.583	0.493	0.807	0.593	0.493
Treatment effect	0.492	0.620	0.496	0.496	0.496
Tumor margin status	0.553	0.932	0.953	0.902	0.864
Benign margin status	0.489	0.593	0.701	0.496	0.972
Perineural invasion	0.600	0.942	0.978	0.927	0.936
Seminal vesicle invasion	0.550	0.888	0.946	0.884	0.876
Extraprostatic extension	0.555	0.898	0.953	0.856	0.687
Lymph node status	0.550	0.657	0.636	0.650	0.657
Mean Macro F1 across classification data elements	**0.481**	**0.730**	**0.764**	**0.683**	**0.669**

LSTM: Long Short-Term Memory Neural Network
CNN: Convolutional Neural Network
SVM: Support Vector Machine

Supplemental Table 2. Micro F1 scores for classification fields across on full traing data sample (n = 2,066)

Data Element	**LSTM**	**CNN**	**Logistic regression**	**Adaboost classifier**	**SVM**	**Random Forest**
Gleason Grade - Primary	0.604	0.981	0.978	0.972	0.935	0.947
Gleason Grade - Secondary	0.577	0.969	0.959	0.947	0.913	0.919
Gleason Grade - Tertiary	0.750	0.935	0.925	0.935	0.907	0.876
Tumor histology	0.993	0.997	0.984	0.997	0.993	0.996
Cribriform pattern	0.972	0.975	0.975	0.981	0.975	0.975
Treatment effect	0.972	0.981	0.981	0.984	0.987	0.987
Tumor margin status	0.645	0.951	0.941	0.953	0.919	0.891
Benign margin status	0.959	0.981	0.975	0.966	0.987	0.969
Perineural invasion	0.614	0.972	0.944	0.978	0.929	0.938
Seminal vesicle invasion	0.787	0.975	0.941	0.975	0.941	0.941
Extraprostatic extension	0.753	0.960	0.953	0.953	0.941	0.901
Lymph node status	0.824	0.988	0.984	0.953	0.975	0.984
Mean Micro F1 across classification data elements	0.788	0.972	0.962	0.966	0.950	0.944

LSTM: Long Short-Term Memory Neural Network
CNN: Convolutional Neural Network
SVM: Support Vector Machine

Supplemental Table 3. Mean macro F1 scores and standard deviations across columns for classification models on varying number of reports (N) across 5 runs

Model	**N = 16**	**N = 32**	**N = 64**	**N = 128**	**N = 256**
Logistic	0.434 ± 0.121	0.495 ± 0.116	0.532 ± 0.129	0.586 ± 0.154	0.630 ± 0.170
AdaBoost	0.475 ± 0.133	0.545 ± 0.155	0.590 ± 0.169	0.620 ± 0.182	0.658 ± 0.183
Random forest	0.445 ± 0.148	0.476 ± 0.135	0.508 ± 0.143	0.529 ± 0.134	0.562 ± 0.140
SVM	0.420 ± 0.141	0.444 ± 0.147	0.460 ± 0.148	0.503 ± 0.131	0.531 ± 0.161
CNN	0.386 ± 0.159	0.416 ± 0.162	0.460 ± 0.173	0.504 ± 0.171	0.577 ± 0.174
LSTM	0.387 ± 0.135	0.408 ± 0.128	0.405 ± 0.131	0.417 ± 0.126	0.430 ± 0.127

Supplemental Table 4. Mean micro F1 scores and standard deviations across columns for classification models on varying number of reports (N) across 5 runs

Model	**N = 16**	**N = 32**	**N = 64**	**N = 128**	**N = 256**
Logistic	0.808 ± 0.164	0.851 ± 0.123	0.880 ± 0.089	0.913 ± 0.060	0.937 ± 0.041
AdaBoost	0.845 ± 0.131	0.883 ± 0.098	0.907 ± 0.067	0.931 ± 0.050	0.949 ± 0.034
Random forest	0.828 ± 0.148	0.860 ± 0.117	0.888 ± 0.088	0.901 ± 0.077	0.913 ± 0.065
SVM	0.764 ± 0.202	0.774 ± 0.205	0.798 ± 0.179	0.853 ± 0.106	0.868 ± 0.138
CNN	0.722 ± 0.216	0.763 ± 0.180	0.786 ± 0.183	0.852 ± 0.122	0.901 ± 0.077
LSTM	0.691 ± 0.208	0.740 ± 0.186	0.740 ± 0.207	0.768 ± 0.182	0.778 ± 0.166

Supplemental Table 5. Counts for each outcome of error analysis for each field from 10 randomly sampled errors (or all errors if fewer than 10 errors were made)

Field	Annotation Error	Model error	Report anomaly	Evaluation Error	Not reported in text	Total	Best model
Gleason Grade - Primary	3	6	1	0	0	10	Cnn
Gleason Grade - Secondary	4	5	1	0	0	10	Cnn
Gleason Grade - Tertiary	4	6	0	0	0	10	Cnn
Tumor histology	0	3	0	0	0	3	Cnn
Cribriform pattern	3	1	0	0	0	4	Cnn
Treatment effect	3	3	1	0	0	7	Cnn
Tumor margin status	1	1	0	0	0	2	Boost
Benign margin status	0	1	0	0	0	1	Svm
Perineural invasion	3	6	1	0	0	10	Boost
Seminal vesicle invasion	1	8	1	0	0	10	Cnn
Extraprostatic extension	4	5	1	0	0	10	Cnn
Lymph node status	2	8	0	0	0	10	Cnn
Prostate weight	3	3	0	2	2	10	Boost
Estimated volume of tumor	2	1	0	3	4	10	Random forest
Pathologic T-stage	3	0	1	0	6	10	Logistic

Supplemental Table 6. Number of annotated data elements by number of reports used in the traing set

Field	Percent of reports containg field

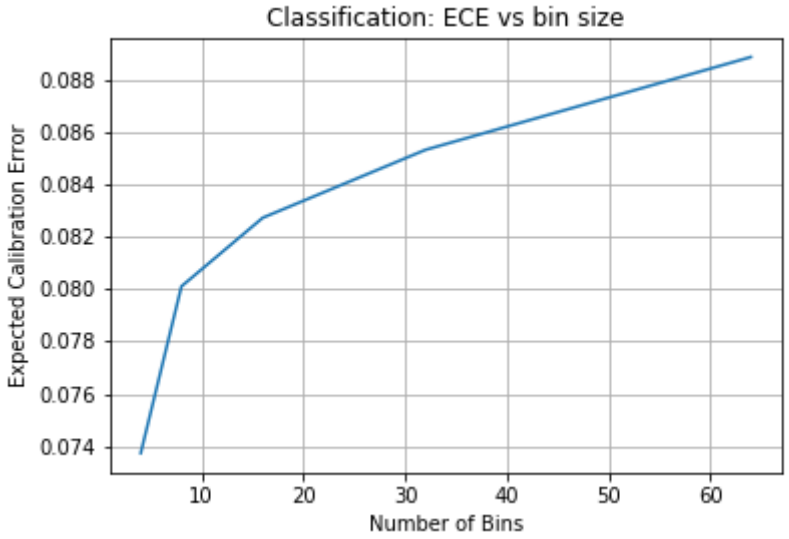
continued on next page

continued from previous page

Gleason Grade - Primary	0.99
Gleason Grade - Secondary	0.99
Gleason Grade - Tertiary	0.15
Tumor histology	1.00
Cribriform pattern	1.00
Treatment effect	1.00
Tumor margin status	1.00
Benign margin status	1.00
Perineural invasion	1.00
Seminal vesicle invasion	1.00
Extraprostatic extension	1.00
Lymph node status	0.99
Prostate weight	0.98
Estimated volume of tumor	0.98
Pathologic T-stage	0.95
Pathologic M-stage	0.94
Pathologic N-stage	0.38

SUPPLEMENTAL FIGURES

Supplemental Figure 1. Expected Calibration Error as a function of the bin size for boosting classification model averaged across fields



Supplemental Figure 2. Expected Calibration Error as a function of bin size for random forest extractor models averaged across fields

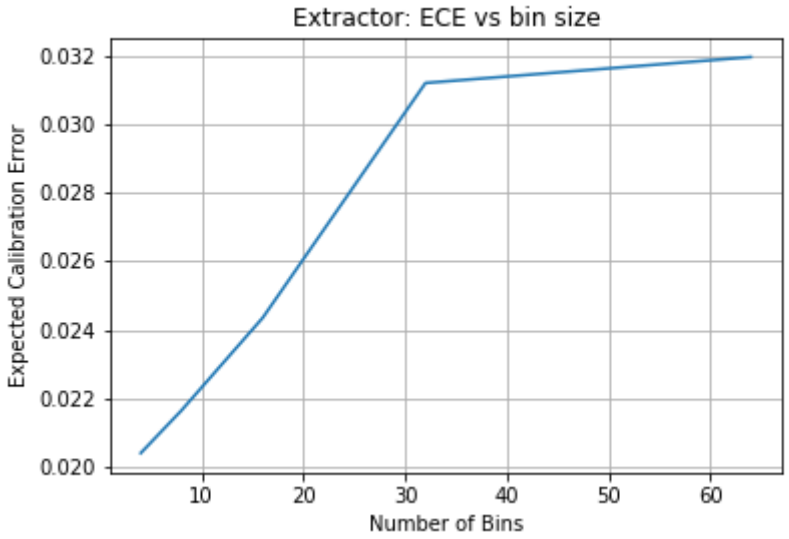


Figure 1A: Average pre-calibration and post-calibration ECE estimates across transfer learning variations of the two-stage method as a function of 10, 20, and 40 labeled examples on colon, kidney, and lung cancer pathology reports. The results presented include the mean performance across 10 random splits of the data and 95% confidence intervals. The two-stage method with transfer learning for both the line and final classifiers consistently outperforms other variations of the two-stage method including no transfer. Calibrating the models via isotonic regression is shown to reduce ECE scores across all methods.

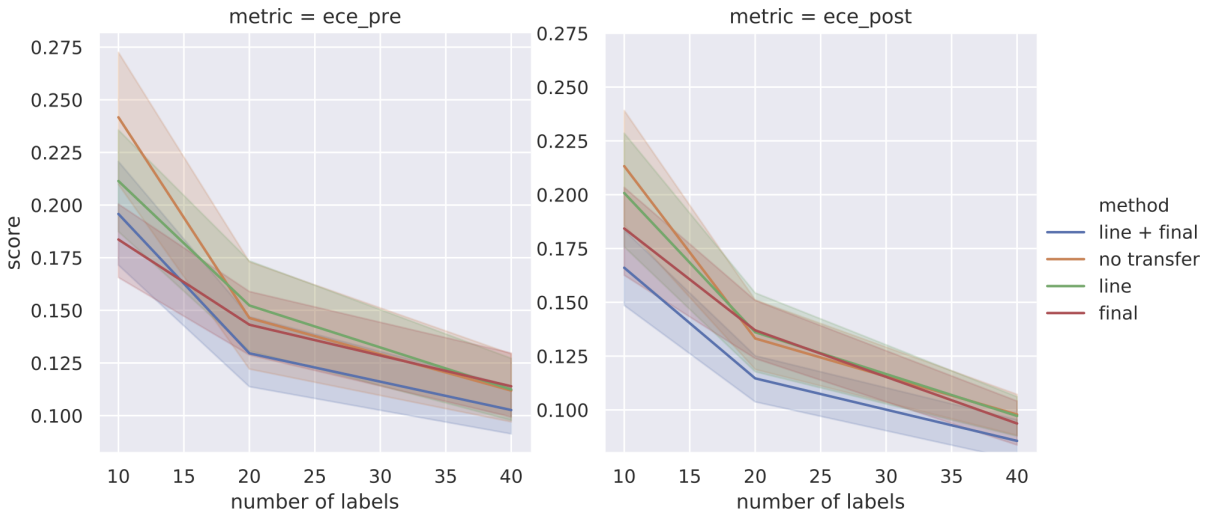


Figure 2A: Average pre-calibration and post-calibration ECE estimates across baseline methods as a function of 10, 20, and 40 labeled examples on colon, kidney, and lung cancer pathology reports. The results presented include the mean performance across 10 random splits of the data and 95% confidence intervals for the shared field and shared labels case. Compared to the transfer learning variations of the two-stage method, the baselines are less calibrated.

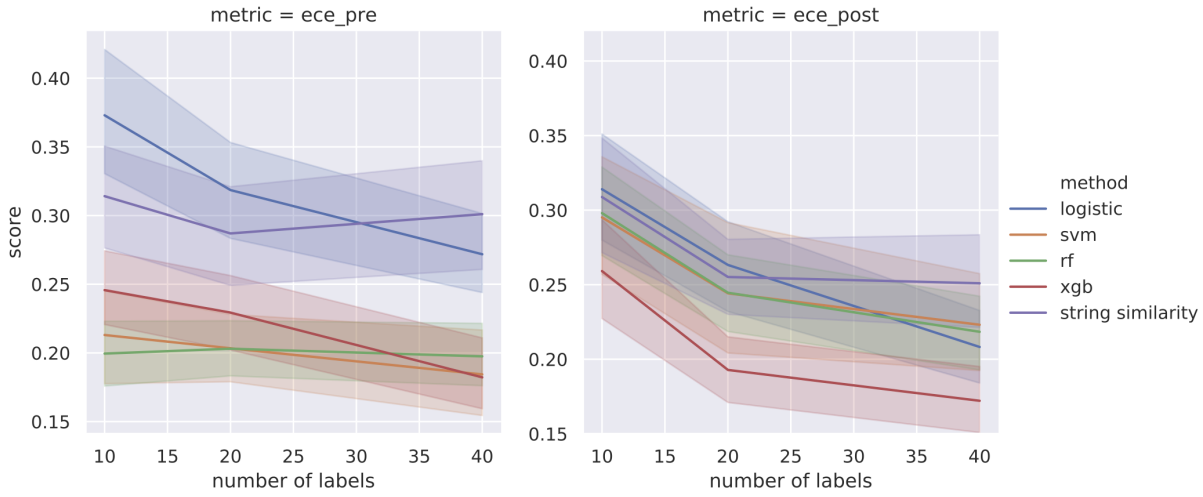
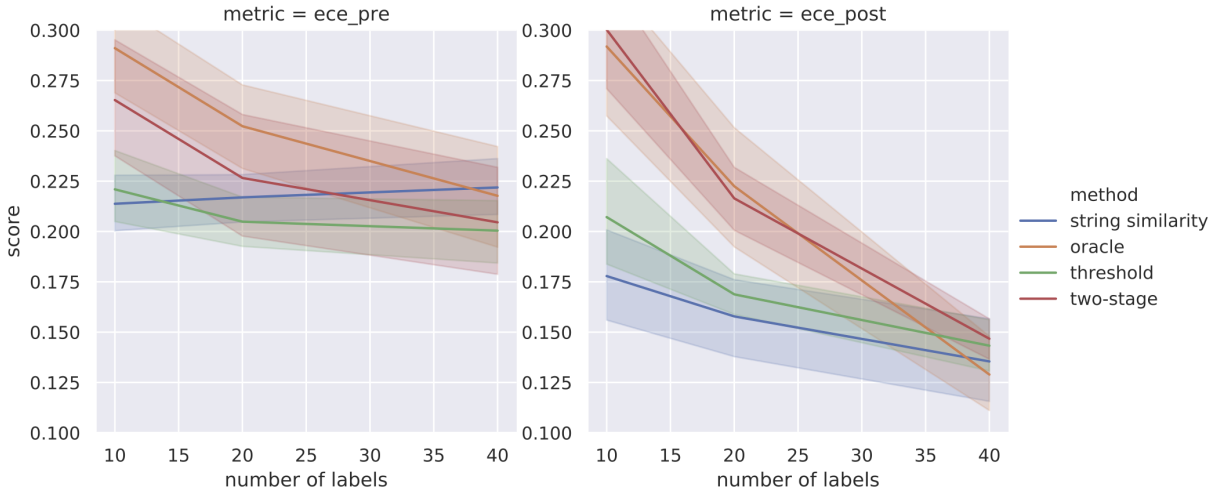


Figure 3A: Average pre-calibration and post-calibration ECE estimates across string similarity variations of the two-stage method as a function of 10, 20, and 40 labeled examples on colon, kidney, and lung cancer pathology reports. The results presented include the mean performance across 10 random splits of the data and 95% confidence intervals for the shared field and unique labels case. The string similarity variations consistently outperform the two-stage method using the ECE metric. Surprisingly the string similarity method in isolation has lower ECE after calibration than the other string similarity variations of the two-stage approach.



.1 Predictors with additional features

We now describe a few additional features that were considered to potentially improve our predictors (but did not lead to any significant improvements). We included these features after our first submission (on May 16, 2020), and hence tried the new features only in the context of the CLEP that combines the expanded shared and linear predictors.[1]

[1]We note that the expanded shared predictor in this appendix is implemented without the monotonicity adjustment (discussed in Section 5.3). The linear predictor does not need such adjustments in our setting. Since our attempts with new features considered in this appendix did not lead to any improvement, we did

Social-distancing feature

Here we consider adding a social distancing feature to the expanded shared model (discussed in Section 5.3). We included an indicator feature in equation (5.4) for every county that takes value 1 on a day if at least two weeks have passed since social distancing was first instituted in a county, and 0 otherwise. We chose two weeks as the time lag to account for the two week progression time for the illness to the recovery of the COVID-19. We found it necessary to regularize this predictor since, without regularization, our 7-day-ahead predictions became infinite in some cases. We regularized this model with the elastic net and an equal penalty of .01 for both ℓ_1 and ℓ_2 regularization.

We now compare CLEP with the social-distancing feature included in the expanded shared predictor, with the original CLEP from the main paper, for 7-day-ahead prediction of the recorded cumulative death counts. We found that the new variant (with the social distancing feature) performed slightly worse than our original CLEP. Over the period March 22 to June 20, the original CLEP has a mean (over time) raw-scale MAE (equation (5.14)) of 13.95, while the social-distancing variant has an MAE of 14.2. In Figure .1, we plot the behavior of raw-scale MAEs with time for the evaluation period from March 22 to June 20. We observe that the performance of the new CLEP variant is similar to that of the original CLEP, with the exception of a couple of the peaks, where the new CLEP variant performs slightly worse.

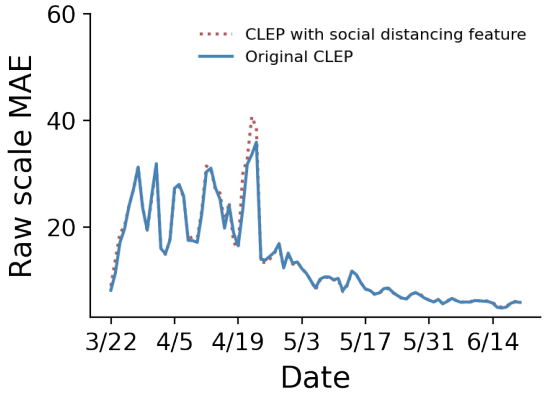


Figure .1: Plots of raw-scale MAE for *7-day-ahead* predictions of two variants of CLEP combing expanded shared and linear predictors: CLEP with a social distancing indicator feature for whether social distancing was in place in a county for more than two weeks or not, and the original CLEP considered in the main paper. The social distancing feature is included in the expanded shared model.

not re-do the investigation with the monotonicity adjustment. We leave any further investigation with these new features (or their variants) for future work.

Weekday feature

As illustrated in Section 5.2 and Figure 5.3(a), the COVID-19 death counts are under-reported on Sunday and Monday, which could potentially lead to increased errors for our prediction algorithm. As a result, we considered an additional feature for the two best predictors that we used in our CLEP earlier: the expanded shared and separate linear predictors. We now discuss the details of our investigations with these two predictors one by one.

Weekday feature for expanded shared predictor

To address this, we first investigated our expanded shared predictor's (5.4) performance for 3-day-ahead prediction on a per weekday basis and plotted the results in Figure .2(a). We observe that the average raw-scale MAE is slightly higher for the days when the 3-day-ahead period included both Sunday and Monday. For example, 3-day-ahead predictions made on Saturday would require making predictions for Saturday, Sunday, and Monday, and that made on Sunday would require making predictions for Sunday, Monday, and Tuesday.

To help account for this bias, we introduced an additional indicator feature in equation (5.4) that takes a value of 1 when the day—for which the prediction is made—is either Sunday or Monday, and 0 otherwise. For instance, when we make 3-day-ahead predictions on Saturday, this feature would take value 0 while computing the prediction for Saturday, and 1 when we compute the prediction for Sunday and Monday. We plot the error distribution over days of this new variant in Figure .2(b). For the new variant, we find that the raw-scale MAEs for the days, when the 3-day-ahead period does not include both Sunday and Monday, typically have higher MAE. Overall when averaging across all days for March 22 to June 20, we find that the new variant of expanded shared predictor performed slightly worse than the original version. The raw-scale MAE (5.14b) for the new variant is 11.7, while the original variant had a raw-scale MAE of 11.5.

We also experimented with a feature that accounted for the predicted day being either a Tuesday or Wednesday, days which are typically overcounted to compensate for undercounting on Sunday and Monday, but initial experiments were not promising.

Weekday feature for separate linear predictors

Next, we experiment with adding a weekday feature to the separate linear predictors (discussed in Section 5.3) for 3-day-ahead predictions, by adding a binary feature that takes value 1 if the day—for which the prediction is made—is either Sunday or Monday, and 0 otherwise. Thus, the new variant of the separate linear predictors is given by

$$\widehat{\mathrm{E}}[\mathrm{deaths}^c_{t+1}|t] = \beta^c_0 + \beta^c_1(t+1) + \beta^c_2 v_{t+1}, \tag{1}$$

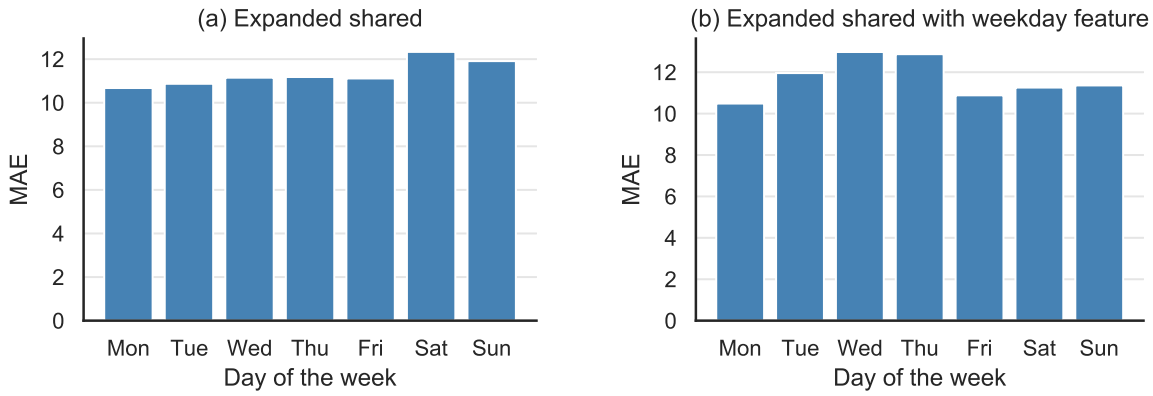


Figure .2: Mean raw-scale MAEs by weekdays for the expanded shared predictor for *3-day-ahead* predictions with and without the weekday feature. The MAE for a given day is the MAE for the 3-day-ahead prediction *computed on that day*. So the MAE for Wednesday is the MAE for predicting the cumulative deaths on Friday.

where v_{t+1} indicates whether day $t+1$ is Sunday/Monday or not. For 3-day-ahead predictions ($\widehat{\mathrm{E}}[\mathrm{deaths}^c_{t+3}|t]$), we simply replace $t+1$ by $t+3$, and v_{t+1} by v_{t+3} on the RHS of equation (1).

Recall that the original separate linear predictors in Section 5.3 were fit only with the four most recent days data. For some choices of days, the new feature v_t takes only a single value 0 in the traing data. For instance, when day $t+1$ is Saturday, we have $v_t = v_{t-1} = v_{t-1} = v_{t-2} = v_{t-3} = 0$, i.e., the new feature is identically zero in the traing data. For such cases, the parameter β^c_2 is not identifiable. To address this issue of non-identifiability, for these experiments, we use the 7 most recent days to fit the linear predictors. For comparison, using the 7 most recent days instead of the 4 most recent days for the original linear predictor increases the raw-scale MAE (5.14b) from 7.0 to 7.2. Adding the new indicator feature v_t increases this error further to 7.4. As with the expanded shared model, we plot the mean raw-scale MAE per day of the week for 3-day-ahead predictions in Figure .3. With the weekday feature (panel (b)), we see that errors for most days are lower than that without the weekday feature (panel (a)), but this gain is offset by a high error for predictions made on Tuesday. In addition to this Sunday/Monday feature, we incorporated a feature to account for the overcounting of deaths on Tuesday and Wednesday. However, we did not see any improvement in results from initial experiments, and thereby omit further details here.

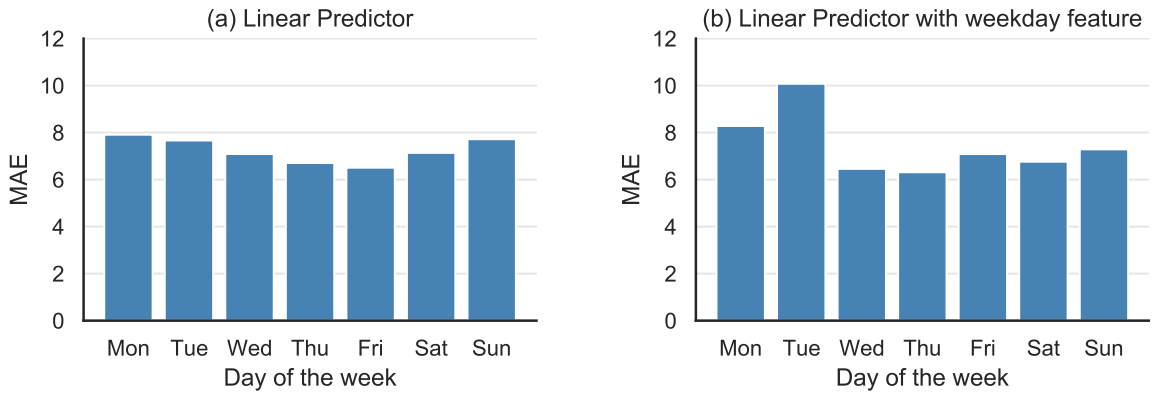


Figure .3: Mean raw-scale MAEs by weekdays for the separate linear predictors for *3-day-ahead* predictions with and without the weekday feature. The MAE for a given day is the MAE for the 3-day-ahead prediction *computed on that day*. So the MAE for Wednesday is the MAE for predicting the cumulative deaths on Friday.

.2 Further discussion on MEPI

We now first shed light on why MEPI had slightly worse coverage for some of the counties. And then, we provide a further discussion on various choices made for designing MEPI.

Counties with poor coverage

While Figure 5.13(a) shows that MEPI intervals achieve higher than 83% coverage for the vast majority of counties over the April 11–May 10 period, there are also counties with coverage below the targeted level. We provide a brief investigation of counties where the coverage of MEPIs for cumulative death counts is below 0.8. Among 198 such counties, Figure .4 shows the cumulative deaths from April 11 to May 10 of the worst-affected 24 counties. Many of these counties exhibit a sharp uptick in the number of recorded deaths similar to that which we encountered in New York, possibly due to reporting lag. For instance, Philadelphia (top row, first column from left) only recorded 2 new deaths between April 28 and May 3, but recorded 201 new deaths on May 4, which brought the cumulative deaths on May 4 to 625.

MEPI vs conformal interence

Recall that the MEPI (equation 5.10a) can be viewed as a special case of conformal prediction interval vovk2005algorithmic,shafer2008tutorial. Here, we provide further discussion on this connection and discuss the assumptions under which the MEPI should achieve good coverage. A general recipe in conformal inference with streaming data is to compute the past several

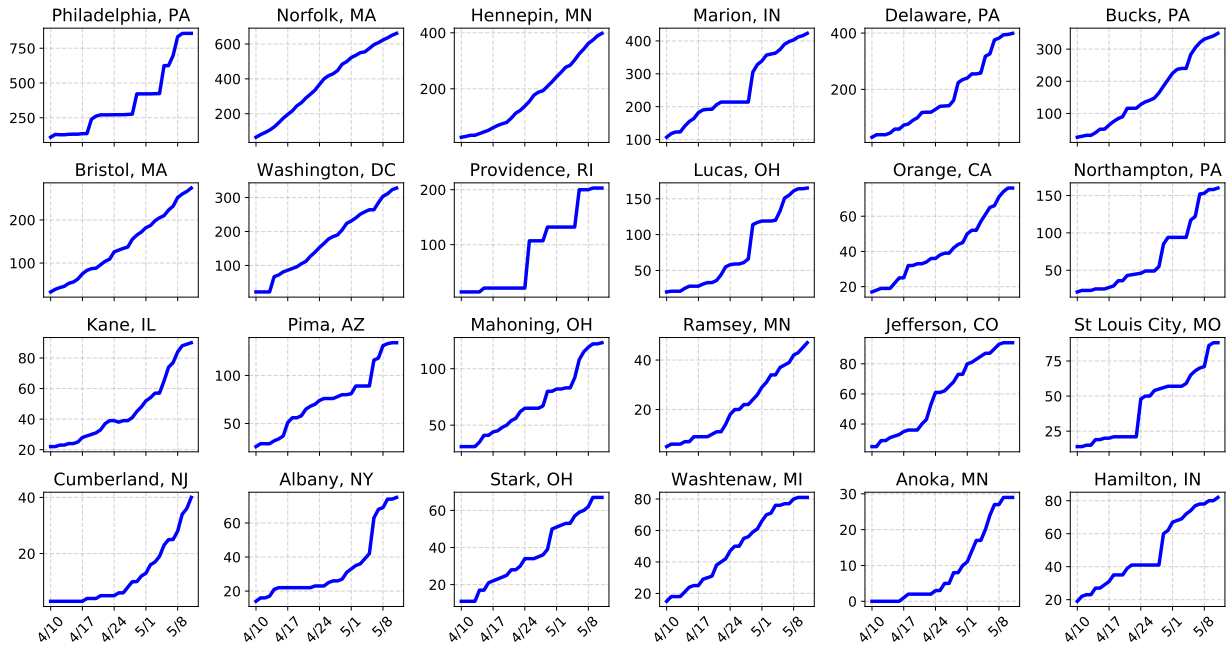


Figure .4: The cumulative death count data from the 24 worst affected counties where the coverage of the *7-day-ahead* MEPIs is below 0.8 (in Figure 5.13(a)).

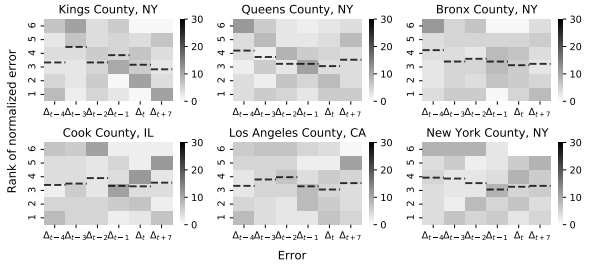
errors of the prediction model and use an s-percentile value for some suitable s (e.g., $s = 95$) to construct the prediction interval for the future observations. At a high-level, theoretical guarantees for conformal prediction intervals rely on the assumption that the sequence of errors is exchangeable. Roughly, the proof proceeds as follows: the exchangeability of the residuals ensures that the rankings of future residuals are uniformly distributed. Hence, the probability of the future residuals being in the top s-percentile is no larger than s, thereby obtaing the promised s%-coverage. For more details, we refer the reader to the excellent tutorial shafer2008tutorial and the book vovk2005algorithmic.

Given the dynamic nature of COVID-19, it is unrealistic to assume that the prediction errors are exchangeable over a long period. As the cumulative death count grows, so too will the magnitude of the errors. Thus our MEPI scheme deviates from the general conformal recipe in two ways. We compute a *maximum error over the past 5 days*, and we *normalize* the errors. Each of these choices—of normalized errors and looking at only past 5 errors— is designed to make the errors more exchangeable. Moreover, given that we take 5 data points to bound the future error, computing a maximum over them is a more conservative choice (e.g., when compared to taking median or a percentile-based cut-off). Furthermore, in order to compute a 95-percentile value, we need to consider errors for at least the past 20 days. Exchangeability is not likely to hold for such a long horizon.

Normalized vs. unnormalized errors: We now provide some numerical evidence to support our choice of normalized errors to define the MEPI. Figure .5(a) shows the rank distribution of normalized errors of our 7-day-ahead CLEP predictions for the six worst-affected counties over over an earlier period (March 26–April 25), and Figure .5(b) shows the (unnormalized) ℓ_1 errors $|\widehat{y}_t - y_t|$ over the same period. We found that in Figure .5(b), the ℓ_1 errors on days $t-4, t-3, t-2, t-1, t$ and $t+7$ do not appear to be exchangeable. Recall that under exchangeability conditions, the expected average rank of each of these six ℓ_1 errors would be 3.5. However, for all six counties, the average rank of the absolute error on day $t+7$ is larger than 4. This indicates that the future absolute error tends to be higher than past errors, and using the ℓ_1 error $|\widehat{y}_t - y_t|$ in place of the normalized error Δ_t can lead to substantial underestimation of future prediction uncertainty.

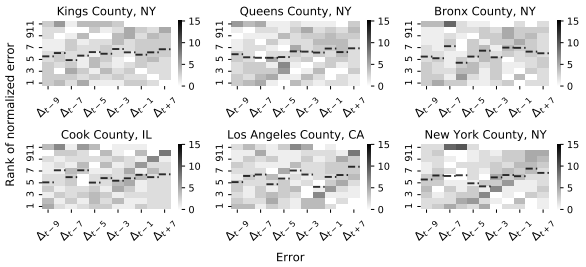
Longer time window: In Figure .5(c), we show the rank distribution of normalized errors over a longer window of 10 days. We found that due to the highly dynamic nature of COVID-19, these errors appear to be even less exchangeable. Under exchangeability conditions, the expected average rank of each of these 11 errors would be 6. However, we found that the average rank substantially deviates from this expected value for many days in this longer window for all displayed counties.

Overall, we believe that putting together the observations from Figures 5.5 and .5 yield reasonable justification for the two choices we made to define MEPI (equation (5.10a)), namely, the 5-day window (versus the entire past) and the choice of normalized errors (versus the unnormalized absolute errors).



(a) Rank distribution of normalized errors over 5 days

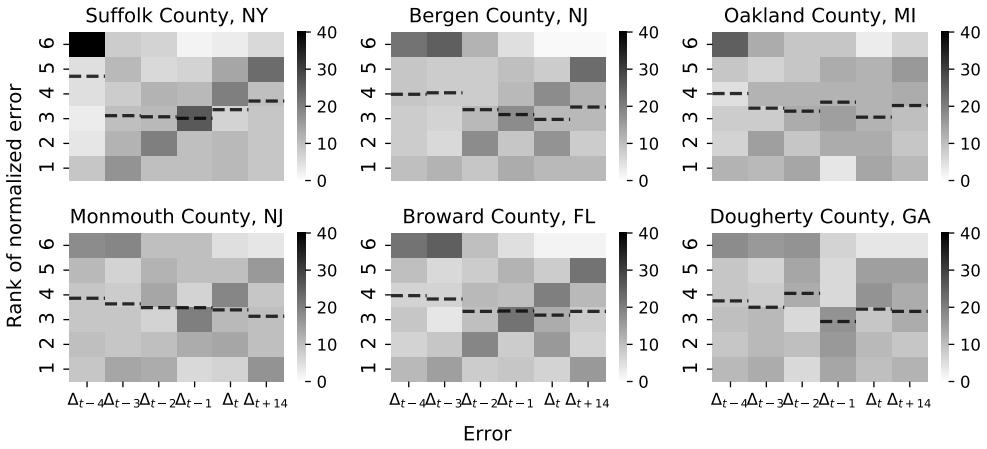
(b) Rank distribution of absolute unnormalized errors over 5 days



(c) Rank distribution of normalized errors over a longer window of 10 days

Figure .5: Exploratory data analysis (EDA) plots with unnormalized and normalized errors for *7-day-ahead* predictions made by CLEP, computed over t = March 26, ..., April 25. **(a)** The rank distribution of normalized errors of our CLEP (with the expanded shared and linear predictors) for the six worst affected counties; **(b)** the rank distribution of the absolute unnormalized errors of our CLEP for the six worst affected counties and **(c)** the rank distribution of the normalized errors over a longer window.

(a) Six worst-affected counties



(b) Six randomly-selected counties

Figure .6: Exploratory data analysis (EDA) plots for investigating exchangeability of normalized errors of *14-day-ahead* CLEP predictions with its last 5 errors made at time t, over the period t = April 2, ..., Jun 6. These plots are obtained using a similar procedure as Figure 5.5.