JAMIA Open, 3(3), 2020, 431–438
doi: 10.1093/jamiaopen/ooaa029
Advance Access Publication Date: 14 October 2020
Research and Applications

Research and Applications

Natural language processing systems for pathology parsing in limited data environments with uncertainty estimation

Anobel Y. Odisho [1], Briton Park[2], Nicholas Altieri[2], John DeNero[3], Matthew R. Cooperberg[1,4], Peter R. Carroll[1], and Bin Yu[2,3,5]

[1]Department of Urology, UCSF Helen Diller Family Comprehensive Cancer Center, San Francisco, California, USA, [2]Department of Statistics, University of California, Berkeley, California, USA, [3]Department of Electrical Engineering and Computer Science, University of California, Berkeley, California, USA, [4]Department of Epidemiology & Biostatistics, University of California, San Francisco, California, USA, [5]Chan-Zuckerberg Biohub, San Francisco, California, USA

Corresponding Author: Bin Yu, PhD, Department of Statistics, University of California, 367 Evans Hall, #3860 Berkeley, CA 94720, USA; binyu@berkeley.edu

Received 21 April 2020; Revised 9 June 2020; Editorial Decision 12 June 2020; Accepted 13 July 2020

ABSTRACT

Objective: Cancer is a leading cause of death, but much of the diagnostic information is stored as unstructured data in pathology reports. We aim to improve uncertainty estimates of machine learning-based pathology parsers and evaluate performance in low data settings.
Materials and methods: Our data comes from the Urologic Outcomes Database at UCSF which includes 3232 annotated prostate cancer pathology reports from 2001 to 2018. We approach 17 separate information extraction tasks, involving a wide range of pathologic features. To handle the diverse range of fields, we required 2 statistical models, a document classification method for pathologic features with a small set of possible values and a token extraction method for pathologic features with a large set of values. For each model, we used isotonic calibration to improve the model's estimates of its likelihood of being correct.
Results: Our best document classifier method, a convolutional neural network, achieves a weighted F1 score of 0.97 averaged over 12 fields and our best extraction method achieves an accuracy of 0.93 averaged over 5 fields. The performance saturates as a function of dataset size with as few as 128 data points. Furthermore, while our document classifier methods have reliable uncertainty estimates, our extraction-based methods do not, but after isotonic calibration, expected calibration error drops to below 0.03 for all extraction fields.
Conclusions: We find that when applying machine learning to pathology parsing, large datasets may not always be needed, and that calibration methods can improve the reliability of uncertainty estimates.

Key words: pathology, natural language processing, information extraction, cancer, prostate cancer, machine learning

INTRODUCTION

An estimated 1.8 million Americans will be diagnosed with cancer in 2020.[1] 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

LAY SUMMARY

When a patient has a tumor removed, the details of the diagnosis are written in an unstructured free-text pathology report. The unstructured nature of such reports renders this information unusable for automated methods that can recommend personalized treatments or facilitate clinical trial enrollment. To address this, natural language processing systems have been created to extract information such as the tumor grade from these reports. However, due to the expertise required, annotating reports is expensive and time-consuming. Second, errors from these systems can lead to incorrect clinical research conclusions and negative outcomes for patients. Both of these issues are major obstacles to deployment. In this article, we develop a system for classifying tumor attributes and extracting values from reports. Then we analyze how many labeled reports are needed across a range of tasks. We find that with only 64 reports we achieve high accuracy, a much smaller number than many existing datasets, which are in the several hundreds or thousands. Furthermore, we analyze our system's ability to estimate the probability of correctness of its outputs. For some tasks, it can reliably estimate this probability but for others, it's generally overconfident. However, by rescaling these estimates, we greatly improve their reliability.

screening, and surveillance.[2,3] 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 processing (NLP) algorithms to extract these details from pathology reports.[4,5] However, only a limited number of categorical data elements are typically extracted and model outputs often lack reliable uncertainty estimates, limiting the clinical applicability of these systems, only 10% of which have been reported to be in real-world use.[5]

Parsing pathology reports has traditionally been approached using rule-based methods.[6–10] 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.[11] NLP has been applied to pathology report information extraction with promising results, using both classic NLP (boosting over a bag-of-*n*-grams representation of the document) and deep learning approaches (convolutional, recurrent, and hierarchical attention networks).[6,12] While most work focuses on classification tasks involving fields with a small number of labels (such as histology or margin status), Li and Martinez[13] investigate categorical fields as well as numeric fields such as the tumor size and the number of lymph nodes examined. Furthermore, many other information extraction tasks and methods have been applied to pathology reports, such as Coden et al[14] which creates a knowledge representation model to represent cancer disease characteristics; Si and Roberts[15] 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[16] which considers attribute detection as a sequence labeling problem; and Oliwa et al[17] 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 2 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.[18–22] 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 2 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 as 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 3232 free-text pathology reports for patients that had undergone radical prostatectomy for prostate cancer at the University of California, San Francisco (UCSF) from 2001 to 2018, which were extracted from UCSF's electronic health record (Epic Systems, Verona, WI, United States). For each document, annotations for 17 pathologic features, such as Gleason scores, margin status, extracapsular extension, and seminal vesicle invasion were extracted (Table 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.

Table 1. Data elements extracted from pathology reports

Data elements	Description
Document classifier algorithm fields	
Gleason GradePrimary, 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 to 5
Tumor histologic type	Primary histologic type, such as acinar adenocarcinoma, ductal adenocarcinoma, and small cell neuro-endocrine 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	Based on American Joint Committee on Cancer TNM staging system for prostate cancer.
T (primary tumor)	Based on the edition used in each report (5th–8th edition)
N (regional lymph nodes)	
M (distant metastasis)	
Tumor volume	Amount of tumor identified in prostate specimen (cubic centimeters)
Prostate weight	Overall weight of the prostate (g)

The full corpus was divided into 4 parts, 64% training, 16% 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 2 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.[6,12] 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.[23] 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.[24,25] 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 2 and 6 possible classes (Table 1). We apply classical methods, such as logistic regression, random forests, support-vector machines (SVMs), and adaptive boosting (AdaBoost) on bag-of-*n*-gram features, as well as deep learning methods, such as convolutional neural networks and long short-term memory networks.

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 patient's medical record number, Table 1). For these fields, we take each token's 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 2. 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 S1–S4, 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 $\leq$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 model's 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

$$P_x(f(x) = y \mid P^*(x) = p) = p \text{for all} p \in [0,1]$$

One common definition of the discrepancy between the model's predicted probability of correctness and its true probability of correctness is given by the expected calibration error (ECE) which is the expected difference between the models confidence and its true probability of being correct.[26]

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

where $f(x)$is the model's prediction for a datapoint point x, $Y$is the true value, and $\widehat{P}(x)$ is the model's predicted probability of being correct for point x. However, this is typically not able to be measured in practice, for example if $\widehat{P}(x)$takes on a continuous set of values, so instead $\widehat{P}(x)$is discretized into bins and the ECE[27] 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 $\widehat{P}(x)$ of the model in bin m.

To improve the calibration of our system, we apply isotonic regression.[26] 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 model's 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 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 model's 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 the token "2-cm" for example.

RESULTS

Document classifier performance

We calculated the weighted F1-score for each data field using the true test set (Table 2). When working with the full training corpus ($n = 2066$), 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). 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

We see in (Table 3) 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

Table 2. Weighted F1 scores for classification fields and mean accuracy for token extractor fields on full training data sample ($n = 2066$)

Data elements	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 across classification models	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	–	–	–	–

CNN, convolutional neural network; LSTM, long short-term memory neural network; SVM, support vector machine.

Table 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

CNN, convolutional neural network; LSTM, long short-term memory neural network; SVM, support vector machine.

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 2 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 ECE on the test set, binning our uncertainty estimates $\widehat{P}(x)$ into bins of width 0.1 (Table 4). Additional experiments investigating the ECE as a function of the bin size, which we include in Supplementary Figures S1 and S2, show that while the average ECE increased, the difference in the average ECE between the smallest bin size (4) and the largest (64) was less than 0.02 for both classification and extraction tasks.[27]

We find that for most classifications fields, the model had expected calibration scores less than 0.1 and that isotonic regression generally improves upon this. Since for each class the one-vs-all probabilities are calibrated, the calibrated model's predictions may differ from the original model if it is not a binary classification problem, so in addition to the ECE 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 0.05 ECEs.

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

Table 4. *Upper*: classifier accuracy and expected calibration error for boosting before and after isotonic calibration and *Lower*: expected calibration error for random forest model before and after isotonic calibration

Data elements Classification calibration	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

Data elements	ECE	Isotonic ECE
Extractor calibration		
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

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 (Supplementary Table S6). For example, if the label for the estimated volume of tumor was 2 (in cm) and the model extracted the token "2-cm", 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 Supplementary Table S5. 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[6] applied boosting across twenty binary fields on 17 000 labeled breast cancer reports and observed strong performance with F1 scores above 0.9 for many fields. Gao et al[12] 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 2 fields (primary site and histologic grade) for a dataset of lung and breast cancer pathology reports. 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.[13] Furthermore, while our 2 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,[28–30] 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 2 values.[6] 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 2066 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 authors' 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 conclusions. Through 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.

FUNDING

Partial support is gratefully acknowledged from ARO (W911NF1710005), NSF (DMS-1613002 and IIS 1741340), the Center for Science of Information (CSoI), a US NSF Science and Technology Center, under grant agreement CCF-0939370, and the Bakar Computational Health Sciences Institute, University of California, San Francisco.

AUTHOR CONTRIBUTIONS

BP and NA implemented, tested, and validated the experiments. All authors were involved in designing and developing the study and writing the paper.

SUPPLEMENTARY MATERIAL

Supplementary material is available at *Journal of the American Medical Informatics Association* online.

CONFLICT OF INTEREST

There are no competing interests.

REFERENCES

1. Siegel RL, Miller KD, Jemal A. Cancer statistics, 2020. *CA Cancer J Clin* 2020; 70 (1): 7–30.
2. Schroeck FR, Patterson OV, Alba PR, *et al*. Development of a natural language processing engine to generate bladder cancer pathology data for health services research. *Urology* 2017; 110: 84–91.
3. Yim W, Yetisgen M, Harris WP, *et al*. Natural language processing in oncology: a review. *JAMA Oncol* 2016; 2 (6): 797.
4. Kreimeyer K, Foster M, Pandey A, *et al*. Natural language processing systems for capturing and standardizing unstructured clinical information: a systematic review. *J Biomed Inform* 2017; 73: 14–29.
5. Burger G, Abu-Hanna A, de Keizer N, *et al*. Natural language processing in pathology: a scoping review. *J Clin Pathol* 2016; 69 (11): 949–55.
6. Yala A, Barzilay R, Salama L, *et al*. Using machine learning to parse breast pathology reports. *Breast Cancer Res Treat* 2017; 161 (2): 203–11.
7. Napolitano G, Fox C, Middleton R, *et al*. Pattern-based information extraction from pathology reports for cancer registration. *Cancer Causes Control* 2010; 21 (11): 1887–94.
8. Nguyen AN, Lawley MJ, Hansen DP, *et al*. Symbolic rule-based classification of lung cancer stages from free-text pathology reports. *J Am Med Inform Assoc* 2010; 17 (4): 440–5.
9. Glaser AP, Jordan BJ, Cohen J, *et al*. Automated extraction of grade, stage, and quality information from transurethral resection of bladder tumor pathology reports using natural language processing. *JCO Clin Cancer Inform* 2018; (2): 1–8.
10. Odisho AY, Bridge M, Webb M, *et al*. Automating the capture of structured pathology data for prostate cancer clinical care and research. *JCO Clin Cancer Inform* 2019; (3): 1–8.
11. Edwards GA. Expert systems for clinical pathology reporting. *Clin Biochem Rev* 2008; 29 (1): S105–109.
12. Gao S, Young MT, Qiu JX, *et al*. Hierarchical attention networks for information extraction from cancer pathology reports. *J Am Med Inform Assoc* 2018; 25 (3): 321–30.
13. Li Y, Martinez D. Information extraction of multiple categories from pathology reports. In: *Proceedings of the Australasian Language Technology Association Workshop 2010*. Melbourne, Australia; 2010. 41–48. https://www.aclweb.org/anthology/U10-1008 (Accessed December 15, 2019).
14. Coden A, Savova G, Sominsky I, *et al*. Automatically extracting cancer disease characteristics from pathology reports into a Disease Knowledge Representation Model. *J Biomed Inform* 2009; 42 (5): 937–49.
15. Si Y, Roberts K. A frame-based NLP system for cancer-related information extraction. *AMIA Annu Symp Proc* 2018; 2018: 1524–33.
16. Xu J, Li Z, Wei Q, *et al*. Applying a deep learning-based sequence labeling approach to detect attributes of medical concepts in clinical text. *BMC Med Inform Decis Mak* 2019; 19 (S5): 236.
17. Oliwa T, Maron SB, Chase LM, *et al*. Obtaining knowledge in pathology reports through a natural language processing approach with classification, named-entity recognition, and relation-extraction heuristics. *JCO Clin Cancer Inform* 2019; (3): 1–8.
18. Deleger L, Li Q, Lingren T, *et al*. Building gold standard corpora for medical natural language processing tasks. *AMIA Annu Symp Proc* 2012; 2012: 144–53.
19. Roberts A, Gaizauskas R, Hepple M, *et al*. The CLEF corpus: semantic annotation of clinical text. *AMIA Annu Symp Proc* 2007; 2007: 625–9.
20. Ogren PV, Savova G, Buntrock JD, *et al*. Building and evaluating annotated corpora for medical NLP systems. *AMIA Annu Symp Proc* 2006; 2006: 1050.

21. South BR, Shen S, Jones M, *et al*. Developing a manually annotated clinical document corpus to identify phenotypic information for inflammatory bowel disease. *BMC Bioinformatics* 2009; 10: S12.
22. Fong A, Adams K, Samarth A, *et al*. Assessment of automating safety surveillance from electronic health records: analysis for the quality and safety review system [published online ahead of print, June 30, 2017]. *J Patient Saf* 2017; doi:10.1097/PTS.0000000000000402.
23. Jurafsky D, Martin JH. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. 2nd ed. Upper Saddle River, NJ: Pearson Prentice Hall; 2009.
24. Pedregosa F, Varoquaux G, Gramfort A, *et al*. Scikit-learn: Machine Learning in Python. *JMLR* 2011; 12: 2825–30.
25. Paszke A, Gross S, Massa F, *et al*. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Proc Adv Neural Inf Process Syst* 2019: 8026–37.
26. Zadrozny B, Elkan C. *Transforming Classifier Scores into Accurate Multiclass Probability Estimates*. New York, NY: Association for Computing Machinery; 2002: 694–99.
27. Degroot MH, Fienberg SE. The comparison and evaluation of forecasters. *J R Stat Soc Ser Stat* 1983; 32 (1/2): 12–22.
28. Alt C, Hübner M, Hennig L. Improving relation extraction by pre-trained language representations. Published Online First: 7 June 2019. https://arxiv.org/abs/1906.03088v1 (Accessed May 22, 2020).
29. Howard J, Ruder S. Universal language model fine-tuning for text classification. Published Online First: 18 January 2018. https://arxiv.org/abs/1801.06146v5 (Accessed May 22, 2020).
30. Shen Y, Yun H, Lipton ZC, *et al*. Deep active learning for named entity recognition. Published Online First: 19 July 2017. https://arxiv.org/abs/1707.05928v3 (Accessed May 22, 2020).