HHS Public Access

Author manuscript

J Biomed Inform. Author manuscript; available in PMC 2021 July 13.

Published in final edited form as:

J Biomed Inform. 2020 October ; 110: 103564. doi:10.1016/j.jbi.2020.103564.

Accelerated training of bootstrap aggregation-based deep information extraction systems from cancer pathology reports

Hong-Jun Yoon[a,*], **Hilda B. Klasky**[a], **John P. Gounley**[a], **Mohammed Alawad**[a], **Shang Gao**[a], **Eric B. Durbin**[b], **Xiao-Cheng Wu**[c], **Antoinette Stroup**[d], **Jennifer Doherty**[e], **Linda Coyle**[f], **Lynne Penberthy**[g], **J. Blair Christian**[a], **Georgia D. Tourassi**[h]

[a]Computational Sciences and Engineering Division, Oak Ridge National Laboratory, Oak Ridge, TN 37830, United States of America

[b]College of Medicine, University of Kentucky, Lexington, KY 40536, United States of America

[c]Louisiana Tumor Registry, Louisiana State University Health Sciences Center, School of Public Health, New Orleans, LA 70112, United States of America

[d]New Jersey State Cancer Registry, Rutgers Cancer Institute of New Jersey, New Brunswick, NJ, 08901, United States of America

[e]Utah Cancer Registry, University of Utah School of Medicine, Salt Lake City, UT 84132, United States of America

[f]Information Management Services Inc., Calverton, MD 20705, United States of America

[g]Surveillance Research Program, Division of Cancer Control and Population Sciences, National Cancer Institute, Bethesda, MD 20814, United States of America

[h]National Center for Computational Sciences, Oak Ridge National Laboratory, Oak Ridge, TN 37830, United States of America

Abstract

Objective: In machine learning, it is evident that the classification of the task performance increases if bootstrap aggregation (bagging) is applied. However, the bagging of deep neural networks takes tremendous amounts of computational resources and training time. The research question that we aimed to answer in this research is whether we could achieve higher task performance scores and accelerate the training by dividing a problem into sub-problems.

Materials and Methods: The data used in this study consist of free text from electronic cancer pathology reports. We applied bagging and partitioned data training using Multi-Task

[*]Corresponding author. yoonh@ornl.gov (H.-J. Yoon).

CRediT authorship contribution statement

Hong-Jun Yoon: Conceptualization, Methodology, Software, Writing - original draft. **Hilda B. Klasky:** Writing - original draft, Writing - review & editing. **John P. Gounley:** Methodology. **Mohammed Alawad:** Software. **Shang Gao:** Software. **Eric B. Durbin:** Data curation. **Xiao-Cheng Wu:** Data curation, Writing - review & editing. **Antoinette Stroup:** Data curation. **Jennifer Doherty:** Data curation. **Linda Coyle:** Data curation. **Lynne Penberthy:** Supervision. **J. Blair Christian:** Data curation, Supervision, Project administration. **Georgia D. Tourassi:** Methodology, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Convolutional Neural Network (MT-CNN) and Multi-Task Hierarchical Convolutional Attention Network (MT-HCAN) classifiers. We split a big problem into 20 sub-problems, resampled the training cases 2,000 times, and trained the deep learning model for each bootstrap sample and each sub-problem—thus, generating up to 40,000 models. We performed the training of many models concurrently in a high-performance computing environment at Oak Ridge National Laboratory (ORNL).

Results: We demonstrated that aggregation of the models improves task performance compared with the single-model approach, which is consistent with other research studies; and we demonstrated that the two proposed partitioned bagging methods achieved higher classification accuracy scores on four tasks. Notably, the improvements were significant for the extraction of cancer histology data, which had more than 500 class labels in the task; these results show that data partition may alleviate the complexity of the task. On the contrary, the methods did not achieve superior scores for the tasks of site and subsite classification. Intrinsically, since data partitioning was based on the primary cancer site, the accuracy depended on the determination of the partitions, which needs further investigation and improvement.

Conclusion: Results in this research demonstrate that 1. The data partitioning and bagging strategy achieved higher performance scores. 2. We achieved faster training leveraged by the high-performance Summit supercomputer at ORNL.

Keywords

Bootstrap aggregation; Data partitioning; Natural language processing; Convolutional neural networks; Hierarchical self-attention networks; Deep learning; High-performance computing

1. Introduction

Cancer constitutes a major public health concern, and its impact upon society cannot be overestimated. In 2018, global cancer statistics roughly calculated that every year, about 3.8 million and 18.1 million people are diagnosed with cancers in the United States and across the world, respectively [1]. Accurate, timely, and comprehensive cancer surveillance are critical tasks, not only for assessing the world's progress in the war against cancer, but also, for guiding the development of effective population cancer control policies and interventions.

Population-based cancer registries in the United States of America provide a reliable surveillance source because they collect case-level data from regional sources like hospitals, doctors' offices, and diagnostic laboratories. To obtain timely, complete, and accurate data, cancer registries rely heavily on pathology reports received from those institutions to record histological evidence and the characteristics of the detected cancers, such as tumor type, histology, grade, stage at diagnosis, and type of surgery received. Such critical information resides in narrative text that not only is ungrammatical, fragmented, and marred with typos and abbreviations, but also exhibits tremendous linguistic variability even when pathologists are describing the same cancer type. Some samples can be found at [2–4].

Because of these challenges, information extraction from unstructured pathology reports still constitutes a heavily manual effort performed, by trained cancer abstractors/registrars, to

ensure high quality in the extracted information. However, with the growing complexity of cancer diagnoses, treatments, and key features such as biomarkers, cancer registries face challenges in scaling the manual effort to handle the rapidly increasing volumes of clinical reports that must be processed and the amount of essential information that needs to be captured per report [5]. Professional organizations and researchers are aware of these challenges. Currently, several clinical language processing and deep learning (DL) research efforts are under way to provide solutions to automate, simplify, and improve this complex task [6].

We studied and presented novel machine learning (ML) and DL-based approaches for extracting information by designing and training classifiers to read, extract features, and understand the contents of the documents in a clinical corpus [7–10]. Those studies demonstrated that artificial intelligence (AI) is an effective means of carrying out such tasks.

Bootstrap aggregation (bagging) is an ML algorithm for obtaining an ensemble of models trained by resampled cases so as to achieve stability and avoid overfitting of models and thus boosting the task performance score. However, previous work [7,9] has shown that training such DL models with a large-volume training corpus takes both time and powerful hardware accelerators, which makes it challenging to apply bagging to DL models.

In this work, we address those challenges by experimenting with a data partitioning method along with bagging, wherein the partitions are determined based on primary cancer site categories. By doing so, the original scientific contributions we present in this paper are the following: 1. We reduced the training time of the DL models and mitigated the complexity of the information extraction tasks. 2. We demonstrate that bagging and partitioned data training with Multi-Task Convolutional Neural Network (MT-CNN) and Multi-Task Hierarchical Convolutional Attention Network (MT-HCAN) classifiers improve accuracy and classification performance. 3. Our results demonstrate that the use of high-performance computing decreased the time to build the bagging classifier and that the partitioned bagging model is well-suited for HPCs and supercomputers.

This paper is organized as follows. Section 2 presents an overview of related work. Section 3 describes the data, methods, and approach. Section 4 presents our experimental results, the verification process, a discussion of our approach, pros and cons, and possible improvements. The conclusion is presented in Section 5.

2. Related work

Bagging [11] has been adopted to improve performance and accuracy during classification in DL applied to different types of data. It is often used either as part of an ensemble of classifiers or alone. Examples of the diversity of its application include the following: multi-view vehicle surveillance [12], credit-risk management of payment data [13], online visual tracking [14], sentiment analysis [15], detection of artistic styles in painting [16], crude oil price forecasting [17], pattern recognition for binary classifiers [18], optimization of ensemble strategies for convolutional neural networks (CNNs) [19], and denoising of auto-encoding-based deep neural networks [20], to name a few.

Specifically, in published DL studies of health care data, bagging has been applied successfully to improve classification performance for imaging, electroencephalogram (EEG) signals and other areas, examples include the following:

- Studies applied to imaging include the estimation of glomerular filtration rates and chronic kidney disease [21], an automated detection of metastates in hematoxylin and eosin-stained whole-slide images of lymph node sections [22], the classification of prostate cancer lesions using 3D multiparametric magnetic resonance imaging data [23], implementation of an augmented image-enhanced bagging ensemble learning to tackle challenges in deficiency training samples and minor visual differences [24], and classification of histopathological biopsy images [25].
- Studies applied to electroencephalogram (EEG) signals include: epileptic seizure detection using EEG signals [26], emotion recognition in the human brain using feature selection of EEG signals [27].
- Other health care related studies include: the development of an enhanced implementation of bagging for the prediction and analysis of heart disease [28], the use of bagging for feature reduction for in-silico drug design [29], identification of personalized medicine problems in an outcome-weighted learning framework [30], the study of human activity recognition in a smart health-care environment that monitors patients using wearable sensor technology [31].

All of these studies reported performance improvements by using bagging in their approaches.

However, none of the examples cited mentioned the application of bagging in health care data within a high-performance computing (HPC) environment. A recent mini-track for Big-Data on Health Care Applications, the proceedings of the 53rd Hawaii International Conference on System Sciences [32], compiles six papers [33–38], two of which described the use of bootstrap training to analyze big health care data [37,38] Buettner et al. developed ensemble approaches that include the utilization of random forests to study the data frequencies of EEG recording snippets to accurately diagnose sleep disorders and schizophrenia, respectively. Other models published recently that are related to natural language processing applied to biomedical data in HPC are ClinicalXLNet [39], ClinicalBert [40], and BioBERT, [41]. ClinicalXLNet, ClinicalBert, and BioBERT based their models on XLNet and BERT over a training layer based on specific bioclinical corpora. These models have demonstrated improvements in biomedical text mining, modeling, and prediction accuracy.

In an attempt to boost and advance insights, we conducted a study with the objectives of documenting our results and observations related to task performance scores and accelerating training time by dividing a problem into sub-problems using the Summit supercomputer at the Oak Ridge Leadership Computing Facility (OLCF).

3. Materials and methods

3.1. Datasets

The dataset for this study consists of unstructured text in pathology reports from four cancer registries: the Louisiana Tumor Registry (LTR), Kentucky Cancer Registry (KCR), Utah Cancer Registry (UCR), and New Jersey State Cancer Registry (NJSCR). These registries belong to the National Cancer Institute's (NCI) Surveillance, Epidemiology, and End Results (SEER) program. The study was executed in accordance with the institutional review board protocol DOE000152.

Certified tumor registrars manually coded the ground truth labels associated with each unique case based on free text from the corresponding pathology reports, according to the SEER program coding and staging manual. We consider the International Classification of Diseases for Oncology, Third Edition (ICD-O-3) coding convention for labeling the cases. We extracted the following six data fields from the cancer reports: (1) cancer site (70 classes), (2) subsite (320 classes), (3) laterality (7 classes), (4) histology (571 classes), (5) behavior (4 classes), and (6) tumor grade (9 classes). Table 1 lists the number of pathology reports from the four registries. Note that we renamed the registries in the table for security purposes.

We chose reports with specimens collected in or after 2017 as testing data, and specimens collected in or before 2016 as training data. We randomly selected and reserved 10% of the training data for validation of the model training. We determined truth labels of the pathology reports based on the Cancer/Tumor/Case (CTC), which stores all diagnostic, staging, and treatment data for a reportable neoplasm in the SEER Data Management System (SEER*DMS). Note that there are possible differences between the information in the cancer pathology reports and CTC because various diagnoses and surgery determine the topography, histology, and behavior codes in the CTC. To mitigate the discrepancy, we only considered cases for which there was less than a 10-day difference between the date of diagnosis and either the specimen collection date or the date of surgery. The 10-day time difference was determined based on an analysis of the pathology report submissions. The vast majority of reports and addenda fell within that time frame.

We applied standard text preprocessing techniques to clean the corpus, as described in previous studies. Cancer pathology reports were represented as one-dimensional vectors of word token indices in integer numbers. Different lengths of cancer pathology reports were accommodated by specifying a fixed length of $L = 1500$ words for all reports. Documents longer than L were truncated, and documents shorter than L were padded. Note that 95% of the pathology reports in our dataset contained fewer than 1500 words.

3.2. Multi-task learning classifiers

Multi-task learning (MTL) is a training mechanism in which one classification model trains multiple related tasks simultaneously to leverage knowledge across the tasks. These related tasks can be learned using the same or different datasets. MTL was successfully used to train a word-level CNN model to simultaneously extract five different data elements from cancer

pathology reports. In this approach, each data element of interest was modeled as a separate output layer for each task.

3.2.1. Multi-task convolutional neural network—The MT-CNN [10] is an extension of the CNN for sentence classification [7,42], tailored to extract information from a cancer pathology data corpus. The model consists of three parts: word embedding, 1D convolution, and a task-specific fully-connected layer. Word embedding is a learned representation of terms to map a set of words onto vectors of numerical values that have the same semantic meaning and have a similar observation. The convolution layer has a series of one-dimensional convolution filters that have latent representations to capture the features from the word vectors of documents. These features are passed to the fully connected softmax layer to decide on the tasks. We added six independent fully connected layers for extracting six tasks. Details of the model and the determination of hyper-parameters can be found at [10]. Note that, in our experiments, we did not use word embedding training techniques such as word2vec [43]. We recognized that such learned vectors are meant to capture the syntactic meaning of the words, but they were not always beneficial to our information extraction tasks. Instead, we randomly initialized the word vectors and let the training of the model determine the meaningful representations. Source code of the MT-CNN model is available at https://github.com/ECP-CANDLE/Benchmarks/tree/master/Pilot3/P3B3.

3.2.2. Multi-task-hierarchical convolutional attention network—The MT-HCAN [8,44] is a hierarchical classification model that uses a self-attention mechanism to overcome challenges associated with cancer pathology report classification. In the model, document embedding is generated in a hierarchical fashion. In the lower level of the hierarchy, we considered words composing lines and, in the upper level, lines composing the complete document. The decision to separate the document into lines rather than sentences was based on the typical syntax of cancer pathology reports, which may employ phrases and non-standard punctuation in place of complete sentences.

The self-attention mechanism within each level of the hierarchy compares a sequence of embeddings with itself to find a relationship between the components of the sequence. For the lower hierarchy, line embedding represents the content of each line in terms of the most important word embeddings included in that line. Similarly, the upper hierarchy constructs document embedding, which represents a complete pathology report based on the line embeddings, which are most important in the lower level of the hierarchy. In sum, the hierarchical model maintains the crucial advantage of self-attention—being able to find relationships among the entries in a sequence regardless of how far apart they are in that sequence—while still leveraging the syntactic organization of the document. From this shared document embedding, we carried out classification for the six information extraction tasks using the multi-task approach network. Fig. 1 illustrates hierarchical structure of attention mechanism. Source code of the MT-HCAN model is available at https://github.com/ECP-CANDLE/Benchmarks/tree/master/Pilot3/P3B4.

3.3. Data partitioning

Since the amount of the training data increased and the number of tasks and number of classes of the task was enormously large (e.g., histology had 300+ labels), the training time for the MTL classifiers was significantly longer. Consequently, we proposed a data partitioning technique: we split the problem into multiple independent sub-problems, developed the classifiers for each sub-problem, and combined answers from the multiple classifiers of sub-problems to conclude a final solution. We expected two advantages from this approach. First, we could reduce the training time for developing a classification system. The size of the training set for each classifier would be decreased by the factor of the number of partitions, so we could reduce the time by training multiple classifiers simultaneously. Second, since the sub-problems contained fewer class labels of cancer sites and subsites, we could mitigate the complexity of problems. However, this approach raised a contradiction—somehow, we defeated the purpose of the MTL mechanism, which we expected to increase the accuracy and stability of the classifier by introducing multiple tasks in a single training to achieve generalizability of the features captured by the DL classifiers. We examined how this discrepancy might affect the classification accuracy by a series of experiments.

In this study, we applied a simple rule for partitioning data into groups of sub-problems as follows: our rule determines group association in incremental order of the cancer main site codes, and it settles to keep the number of training samples in the group not to exceed 1/20 of the training samples' total number. We divided one problem into 20 sub-problems having a similar number of cases per each problem, but we kept the same label for the main cancer site in one group; for example, we kept the C50 (breast cancer) cases in one bucket, even though the number of C50 cases was greater than the size of the partition. Table 2 lists the main site labels for each group and the number of cases of the associated group.

Note that we trained 20 classifiers for 20 groups of sub-problems, so we obtained 20 answers from the classifiers. Therefore, we needed a way to combine those 20 answers to obtain one final answer. We studied two approaches. First, we introduced one preclassifier to predict which sub-problem a given case was associated with. We applied a single-task CNN with 20 softmax output. This was a straightforward and intuitive approach, but it implied that the accuracy of the cancer site classification greatly depended on the quality of the preclassifier. Also, to a certain extent, it defeated the purpose of partitioning the data to save training time, for training the preclassifier involves the entire dataset, which takes longer than training a preclassifier with a sub-problem. Second, we applied an abstention mechanism to the sub-problem classifiers. We introduced an "other" class label, which consisted of 10,000 randomly selected training cases from other sub-problems. Thus, we expected the classifier would predict whether the input sample was associated with the sub-problem. The class association of the problem is determined by the majority vote of every classifier in the model. If the multiple classes received the same amount of votes, we chose the class association randomly between the highest votes. The additional "other" class might increase the training time of the sub-problem classifiers, but it would increase the time less than preparing a preclassifier would.

3.4. Bootstrap aggregation

Even though bagging has a long history [11], the concept has not been actively applied to DL-related training and evaluation, because it requires a tremendous amount of training time. We ran the experiments on the Summit supercomputer at the OLCF to accelerate the training time by taking advantage of its parallel computing capabilities. Because the training of each bootstrap sample is independent, the training of the bagging DL classifier is embarrassingly parallel, maximizing the utility of the multi-node, multi-GPU capacity. We developed a bagging training workflow under the CANcer Distributed Learning Environment (CANDLE) software stack. The fundamental question remains: how many bootstrap samples would be sufficient to yield reliable results? We present experimental results for this issue in Section 4.2.1.

3.5. Implementation

The MT-CNN classifier was implemented with the Keras and TensorFlow backend, and the MT-HCAN classifier was implemented using the TensorFlow platform. Development environments and packages are within the IBM Watson Machine Learning (WML) framework, the community edition version 1.6.2. Note that the partitioned and tokenized pathology report dataset occupied about 300 megabytes (=1500 [number of tokens] × 4 (32-bit integer[× 50,000 [average number of ePath Report documents]). That size was sufficiently handled by the AC922 node on the Summit supercomputer with 512 GB. We set the stopping criteria for the classifier training so that it waited for up to ten subsequent epochs to evaluate if it approached the minimum validation loss and restored the best model. We set the maximum number of iterations for the training classifiers as 100 epochs.

4. Experimental results

We designed a series of experiments for evaluating the feasibility of the bagging technique for information extraction tasks based on the DL-based MTL algorithms, demonstrating whether data partitioning reduces the training time in HPC environments, and exploring the proper way of implementing bagging and data partitioning methods to achieve better classification task performance.

4.1. Experimental setup

We examined the following four models described down below. Illustration of those four models are in Fig. 2.

4.1.1. Single model—The single model is the traditional approach to MT-CNN and MT-HCAN classifiers: one model extracts all the tasks and labels. We applied all the training and validation samples to train a model. For MT-CNN, an average of 843 s was required for the first training epoch, and an average 835 s per epoch for the rest of the training. For MT-HCAN, 5299.2 s was required for the first epoch and 5294.2 s for the remaining epochs. Note that the training and optimization of DL models are stochastic, which implies that uncertainty and randomness are involved in the trained model. In this experiment, we repeated the training 2000 times and obtained the average scores.

4.1.2. Bagging model—We performed training of independent MT-CNN and MT-HCAN models with 2000 bootstrap samples for the training and validation sets. The decision was made with an aggregation of decisions from 2000 DL models. We tested three options: (1) MT-CNN only, (2) MT-HCAN only, and (3) a combination of MT-CNN and MT-HCAN.

4.1.3. Partitioned bagging model A - abstention classifiers—We conducted training of 2000 DL models for each data partition. Therefore, we trained up to 40,000 MT-CNN models and 40,000 MT-HCAN models to set up the bagging model. We introduced an abstention mechanism for each model by adding 10,000 samples to a class named “other”, which we randomly chose from other partitions. As we did for the bagging model, we tested the following: (1) MT-CNN only, (2) MT-HCAN only, and (3) an aggregation of the MT-CNN and MT-HCAN models.

4.1.4. Partitioned bagging model B - additive preclassification—We included an additional bagging CNN model as a preclassification to determine to which data partition the given input would be assigned. The preclassifier took 1500 s per epoch for training. It was the most expensive model in this study. As before, we also tested (1) MT-CNN, (2) MT-HCAN, and (3) an aggregation of MT-CNN and MT-HCAN.

4.2. Results

4.2.1. Number of bootstrapping samples—The number of bootstrap samples that is sufficient to achieve reliable accuracy is highly dependent on the complexity of the problem and the number of training samples; therefore, it was determined by trials. Training the models for information extraction for six tasks and many class labels with a 1 million natural language text corpus requires several hours with a decent DL accelerator. With NVIDIA’s V100 GPU installed on an AC922 node, which is the Summit compute node, an average of 184 min is required to train a MT-CNN model with one bootstrap sample. Properly choosing the number of bootstrap samples is critical to the realization of the bagging classification system.

We experimented with various numbers of bootstrap samples and observed micro- and macro-averaged F1 scores of the classifications of cancer subsites and histologies from the testing sets; micro- and macro-F1 scores are illustrated in Fig. 3. We chose to observe cancer subsite and histology out of six tasks because those tasks are the most difficult ones, possessing more than 100 class labels. We observed that the F1 scores fluctuated if we applied a small number of bootstrap samples, and scores became stable if more than 1000 bootstrap samples were used. With one exception in Fig. 3(d) MT-CNN histology macro-F1, bagging classifiers achieved optimal scores when trained by more than 1000 bootstrap samples. Thus we concluded that the application of 2000 bootstrap samples was sufficiently stable to achieve optimal clinical task performance scores from all six tasks.

4.2.2. Scalability—As we elaborated in Section 3.5, the training of one MT-CNN model or one MT-HCAN model can be done with a single GPU accelerator. The memory requirement for holding 1 million electronic pathology reports as a tokenized dataset is 6 GB

(= 1500 [length of a document] × 4 [32-bit integer] × 1,000,000). Thus, a tokenized dataset can reside in the memory of a Summit compute node; consequently, it does not cause overhead to the storage devices, except during the initialization and saving checkpoints for each training epoch. This model is appropriate to execute training jobs in an embarrassingly parallel manner.

It took 2.32 h to train one MT-CNN classifier. To build up a bagging classifier with 2000 bootstrap samples, we had to spend 4658 h with one V100 GPU accelerator. However, using the Summit supercomputer, which provides 4608 nodes and 27,648 V100 GPUs, we needed only 2.32 h to finish training the bagging model. If we divided the training set into 20 sub-problems, it took only 14 min to finish training a model for one sub-problem; and the total training time for the partitioned bagging model took only 30 min. The partitioned bagging model is well-suited for HPCs and supercomputers. Table 4 lists the total node hours that we consumed to train the models and the actual wall-time hours spent to train the models on the Summit supercomputer.

4.2.3. Clinical task performance—Clinical task performance scores in micro- and macro-averaged F1 scores are listed in Table 3. The Bagging Model resulted in higher performance scores than the single Model in every task, which accords with many studies of bagging techniques [16,21,25]. However, the aggregation of MT-CNN and MT-HCAN (reported as Combo in Table 3) of the Bagging Model produced little improvement. We also observed that, in the Bagging Model, the MT-HCAN classifiers awarded slightly higher scores than the MT-CNN ones, but not decisively.

Comparing the Partitioned Bagging Models A and B with the Bagging Model yielded mixed results. The MT-CNN's of Partitioned Bagging Models A and B achieved higher task performance scores than MT-CNN's of the Bagging Model in every task except the main cancer site classification. Notably, the substantially higher macro-F1 scores for the subsite and histology classifications. This result hints that the data partitioning approach mitigated the complexity of the problem, thus helping boost the clinical task performance. However, the results from MT-HCAN's of Partitioned Bagging Models A and B were not higher than the MT-HCAN's from the Bagging Model. That is, the data partitioning approach may not have benefited from the richness and generalizability of features, whereas the MT-HCAN approach may have taken advantage of those characteristics.

Also, note that the primary cancer site classification scores of the Partitioned Bagging Models A and B were substantially lower than those of the Bagging Model. Those results were due to the weaker performance of abstention classifiers in the Partitioned Bagging Model A and the additive preclassifiers in the Partitioned Bagging Model B, which needs further investigation to improve the accuracy. Note also that the aggregation of MT-CNN and MT-HCAN classifiers for the Partitioned Bagging Models A and B did not always yield higher scores in most cases.

Applying preclassifiers in the Partitioned Bagging Model B resulted in higher primary cancer site classification scores than were obtained with the Partitioned Bagging Model A, which also resulted in boosting other task performance scores. The results suggest that

determining the associated partition is the first and essential step of the partitioned bagging classifiers.

5. Conclusions

In this paper, we applied the bootstrap aggregation to the DL models for the information extraction tasks with one million cancer pathology reports on HPC systems. We introduced a data partitioning scheme to the bagging model to maximize the utility of compute nodes on HPC systems, thus reducing the training time required as well as boosting performance. We evaluated the feasibility of the proposed system in both scalability and clinical task performance. The experimental results demonstrated that the proposed data partitioning alleviated the complexity of the information extraction tasks and improved both micro- and macro-averaged F1 scores. The bagging model with data partitioning completed the training of the model as quickly as or more quickly than the single model did.

Specifically, the classification task of cancer histology is the one that showed superior performance by the bagging and the partitioned bagging models. The histology task included more than 500 class labels, and many of the labels were underrepresented classes. Data partitioning made it possible to divide this single large, imbalanced problem into several easier sub-problems. Higher scores on the macro-F1 score are an indication that the model was useful in classifying minor labels. Such a trend was consistent with the classification tasks of laterality, behavior, and grade.

Relatively low clinical task performances on primary cancer site and subsite classifications remain as outstanding research questions. We have experimented with an ensemble of models: first, abstention mechanisms, and second, preclassification layers. The former needs further investigation to determine an adequate number of samples for the “other” class. Intuitively, more samples may result in higher task performance, but they also will increase training time, which is not desirable. The latter model scored higher in site and subsite classification than did the former, but its scores were nonetheless lower than the scores of the standard bagging model. Our next research steps will be the implementation of better ensembling of partitioned models; optimal data partitioning for both higher scalability and task performance; and evaluation of portability to the next generation of OLCF supercomputers, Frontier.

Acknowledgments

This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the US Department of Energy (DOE) Office of Science and the National Nuclear Security Administration. This work has been supported in part by the Joint Design of Advanced Computing Solutions for Cancer (JDACS4C) program established by DOE and the National Cancer Institute of the National Institutes of Health. This work was performed under the auspices of DOE by Argonne National Laboratory under Contract DE-AC02-06-CH11357, Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344, Los Alamos National Laboratory under Contract DE-AC5206NA25396, and ORNL under Contract DE-AC05-00OR22725.

KCR data were collected with funding from the NCI SEER Program (HHSN261201800013I), the CDC National Program of Cancer Registries (NPCR) (U58DP00003907) and the Commonwealth of Kentucky.

LTR data were collected using funding from NCI and the SEER Program (HHSN2612018000071), the NPCR (NU58DP006332-02-00), ands the State of Louisiana.

NJSCR data were collected using funding from NCI and the SEER) Program (HHSN261201300021I, the (NPCR (NU58DP006279-02-00), and the State of New Jersey and the Rutgers Cancer Institute of New Jersey.

The Utah Cancer Registry is funded by the NCI's SEER Program, Contract No. HHSN261201800016I, and the NPCR, Cooperative Agreement No. NU58DP0063200, with additional support from the University of Utah and Huntsman Cancer Foundation.

This research used resources of the OLCF at ORNL, which is supported by the DOE Office of Science under Contract No. DE-AC05-00OR22725.

References

[1]. Bray F, Ferlay J, Soerjomataram I, Siegel RL, Torre LA, Jemal A, Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries, CA: Cancer J. Clin 68 (6) (2018) 394–424. [PubMed: 30207593]

[2]. Yala A, Barzilay R, Salama L, Griffin M, Sollender G, Bardia A, Lehman C, Buckley JM, Coopey SB, Polubriaginof F, et al., Using machine learning to parse breast pathology reports, Breast Cancer Res. Treat 161 (2) (2017) 203–211. [PubMed: 27826755]

[3]. Wu Y, Denny JC, Rosenbloom ST, Miller RA, Giuse DA, Xu H, A comparative study of current clinical natural language processing systems on handling abbreviations in discharge summaries, in: AMIA Annual Symposium Proceedings, Vol. 2012, American Medical Informatics Association, 2012, p. 997. [PubMed: 23304375]

[4]. Buckley JM, Coopey SB, Sharko J, Polubriaginof F, Drohan B, Belli AK, Kim EM, Garber JE, Smith BL, Gadd MA, et al., The feasibility of using natural language processing to extract clinical information from breast pathology reports, J. Pathol. Inform 3 (2012).

[5]. Hasan SA, Farri O, Clinical natural language processing with deep learning, in: Data Science for Healthcare: Methodologies and Applications, 2019.

[6]. Wang Y, Xu H, Uzuner O, Editorial: The second international workshop on health natural language processing (healthnlp 2019), BMC Med. Inform. Decis. Mak 19 (5) (2019) 233, 10.1186/s12911-019-0930-9. [PubMed: 31801516]

[7]. Qiu JX, Yoon H-J, Fearn PA, Tourassi GD, Deep learning for automated extraction of primary sites from cancer pathology reports, IEEE J. Biomed. Health Inform 22 (1) (2017) 244–251. [PubMed: 28475069]

[8]. Gao S, Ramanathan A, Tourassi G, Hierarchical convolutional attention networks for text classification, in: Proceedings of the Third Workshop on Representation Learning for NLP, 2018, pp. 11–23.

[9]. Yoon H-J, Gounley J, Gao S, Alawad M, Ramanathan A, Tourassi G, Model-based hyperparameter optimization of convolutional neural networks for information extraction from cancer pathology reports on HPC, in: 2019 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI), IEEE, 2019, pp. 1–4.

[10]. Alawad M, Gao S, Qiu JX, Yoon HJ, Blair Christian J, Penberthy L, Mumphrey B, Wu X-C, Coyle L, Tourassi G, Automatic extraction of cancer registry reportable information from free-text pathology reports using multitask convolutional neural networks, J. Am. Med. Inform. Assoc 27 (1) (2020) 89–98. [PubMed: 31710668]

[11]. Breiman L, Bagging predictors, Mach. Learn 24 (2) (1996) 123–140.

[12]. Kim P-K, Lim K-T, Vehicle type classification using bagging and convolutional neural network on multi view surveillance image, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 41–46.

[13]. Hamori S, Kawai M, Kume T, Murakami Y, Watanabe C, Ensemble learning or deep learning? Application to default risk analysis, J. Risk Financ. Manage 11 (1) (2018) 12.

[14]. Li H, Li Y, Porikli F, Convolutional neural net bagging for online visual tracking, Comput. Vis. Image Underst 153 (2016) 120–129.

[15]. Rong W, Nie Y, Ouyang Y, Peng B, Xiong Z, Auto-encoder based bagging architecture for sentiment analysis, J. Vis. Lang. Comput 25 (6) (2014) 840–849.

[16]. Lecoutre A, Negrevergne B, Yger F, Recognizing art style automatically in painting with deep learning, in: Asian Conference on Machine Learning, 2017, pp. 327–342.

[17]. Zhao Y, Li J, Yu L, A deep learning ensemble approach for crude oil price forecasting, Energy Econ. 66 (2017) 9–16.

[18]. Mordelet F, Vert J-P, A bagging SVM to learn from positive and unlabeled examples, Pattern Recognit. Lett 37 (2014) 201–209.

[19]. Lee S, Purushwalkam S, Cogswell M, Crandall D, Batra D, Why m heads are better than one: Training a diverse ensemble of deep networks, 2015, arXiv preprint arXiv:1511.06314.

[20]. Alvear-Sandoval R, Figueiras-Vidal AR, Does diversity improve deep learning? in: 2015 23rd European Signal Processing Conference (EUSIPCO), IEEE, pp. 2496–2500.

[21]. Kuo C-C, Chang C-M, Liu K-T, Lin W-K, Chiang H-Y, Chung C-W, Ho M-R, Sun P-R, Yang R-L, Chen K-T, Automation of the kidney function prediction and classification through ultrasound-based kidney imaging using deep learning, npj Digit. Med 2 (1) (2019) 29. [PubMed: 31304376]

[22]. Fernández-Carrobles MM, Serrano I, Bueno G, Déniz O, Bagging tree classifier and texture features for tumor identification in histological images, Procedia Comput. Sci 90 (2016) 99–106.

[23]. Liu S, Zheng H, Feng Y, Li W, Prostate cancer diagnosis using deep learning with 3D multiparametric MRI, in: Medical Imaging 2017: Computer-Aided Diagnosis, Vol. 10134, International Society for Optics and Photonics, 2017, 1013428.

[24]. Liu Y, Long F, Acute lymphoblastic leukemia cells image analysis with deep bagging ensemble learning, bioRxiv (2019) 580852.

[25]. Kassani SH, Kassani PH, Wesolowski MJ, Schneider KA, Deters R, Classification of histopathological biopsy images using ensemble of deep learning networks, 2019, arXiv preprint arXiv:1909.11870.

[26]. Hassan AR, Siuly S, Zhang Y, Epileptic seizure detection in EEG signals using tunable-q factor wavelet transform and bootstrap aggregating, Comput. Methods Programs Biomed 137 (2016) 247–259. [PubMed: 28110729]

[27]. Mehmood RM, Du R, Lee HJ, Optimal feature selection and deep learning ensembles method for emotion recognition from human brain EEG sensors, IEEE Access 5 (2017) 14797–14806.

[28]. Bashir S, Qamar U, Khan FH, BagMOOV: A novel ensemble for heart disease prediction bootstrap aggregation with multi-objective optimized voting, Australas. Phys. Eng. Sci. Med 38 (2) (2015) 305–323, 10.1007/s13246-015-0337-6, URL https://www.ncbi.nlm.nih.gov/pubmed/25750025. [PubMed: 25750025]

[29]. Embrechts MJ, Arciniegas F, Ozdemir M, Breneman CM, Bennett K, Lockwood L, Bagging neural network sensitivity analysis for feature reduction for in-silico drug design, in: IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222), Vol. 4, IEEE, 2001, pp. 2478–2482.

[30]. Mi X, Zou F, Zhu R, Bagging and deep learning in optimal individualized treatment rules, Biometrics 75 (2) (2019) 674–684. [PubMed: 30365175]

[31]. Subasi A, Khateeb K, Brahimi T, Sarirete A, Human activity recognition using machine learning methods in a smart healthcare environment, in: Innovation in Health Informatics, Elsevier, 2020, pp. 123–144.

[32]. Hung P, Poon S, Tsoi K, Introduction to the minitrack on big data on healthcare application, in: Proceedings of the 53rd Hawaii International Conference on System Sciences, 2020.

[33]. Wang T, Li W, Lewis D, Blood glucose forecasting using LSTM variants under the context of open source artificial pancreas system, in: Proceedings of the 53rd Hawaii International Conference on System Sciences, 2020.

[34]. Redd D, Goulet J, Zeng-Treitler Q, Using explainable deep learning and logistic regression to evaluate complementary and integrative health treatments in patients with musculoskeletal disorders, in: Proceedings of the 53rd Hawaii International Conference on System Sciences, 2020.

[35]. Gupta J, Poon S, Configurational approach to identify concept networks in selected clinical safety incident classes, in: Proceedings of the 53rd Hawaii International Conference on System Sciences, 2020.

[36]. Dashtban M, Li W, Predicting risk of hospital readmission for comorbidity patients through a novel deep learning framework, in: Proceedings of the 53rd Hawaii International Conference on System Sciences, 2020.

[37]. Buettner R, Beil D, Scholtz S, Djemai A, Development of a machine learning based algorithm to accurately detect schizophrenia based on one-minute EEG recordings, in: Proceedings of the 53rd Hawaii International Conference on System Sciences, 2020.

[38]. Buettner R, Grimmeisen A, Gotschlich A, High-performance diagnosis of sleep disorders: A novel, accurate and fast machine learning approach using electroencephalographic data, in: Proceedings of the 53rd Hawaii International Conference on System Sciences, 2020.

[39]. Huang K, Singh A, Chen S, Moseley ET, Deng C.-y., George N, Lindvall C, Clinical XLNet: Modeling sequential clinical notes and predicting prolonged mechanical ventilation, 2019, arXiv preprint arXiv:1912.11975.

[40]. Alsentzer E, Murphy JR, Boag W, Weng W-H, Jin D, Naumann T, McDermott M, Publicly available clinical BERT embeddings, 2019, arXiv preprint arXiv:1904.03323.

[41]. Lee J, Yoon W, Kim S, Kim D, Kim S, So CH, Kang J, BioBERT: pre-trained biomedical language representation model for biomedical text mining, 2019, arXiv preprint arXiv:1901.08746.

[42]. Kim Y, Convolutional neural networks for sentence classification, 2014, arXiv preprint arXiv:1408.5882.

[43]. Goldberg Y, Levy O, Word2vec explained: deriving Mikolov et al.'s negative-sampling word-embedding method, 2014, arXiv preprint arXiv:1402.3722.

[44]. Gao S, Qiu JX, Alawad M, Hinkle JD, Schaefferkoetter N, Yoon H-J, Christian B, Fearn PA, Penberthy L, Wu X-C, et al., Classifying cancer pathology reports with hierarchical self-attention networks, Artif. Intell. Med 101 (2019) 101726. [PubMed: 31813492]

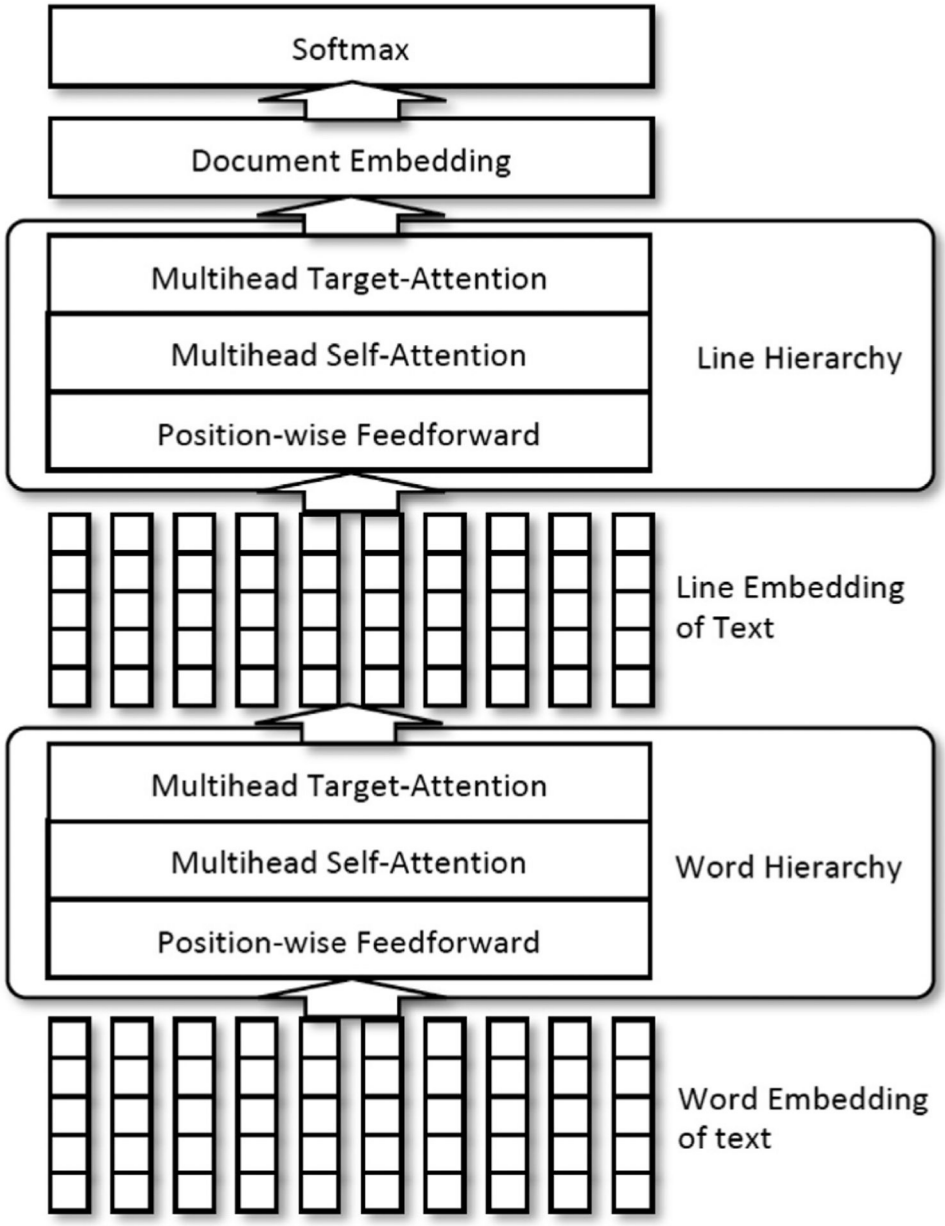

Fig. 1.
Architecture of hierarchical convolutional attention network.

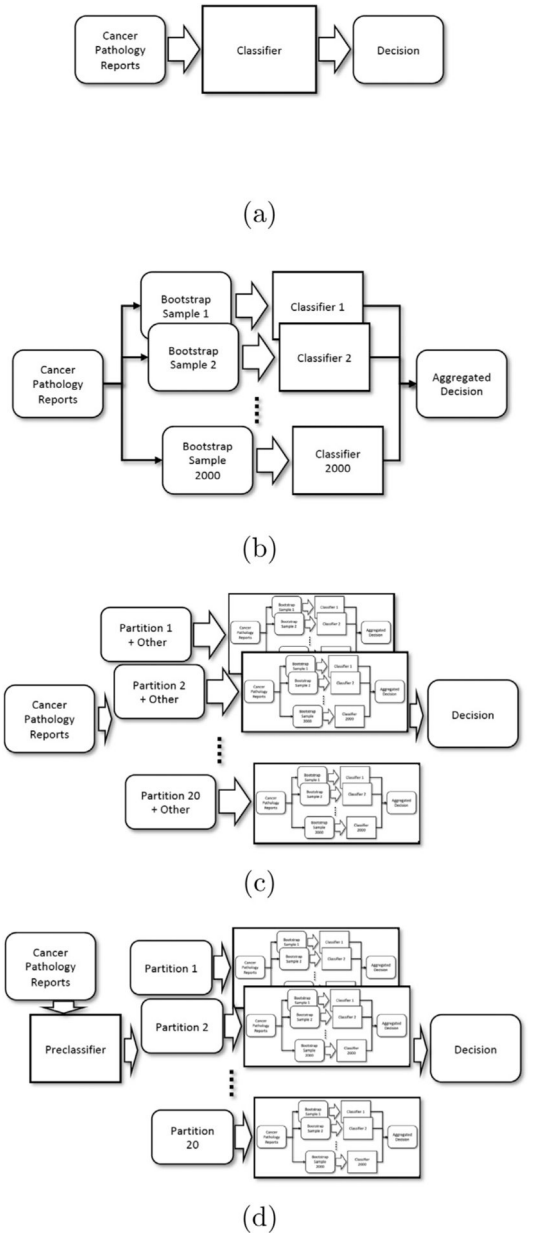

Fig. 2.

Experimental setup of four models (a) Single Model, (b) Bagging Model, (c) Partitioned Bagging Model A - abstention classifiers, and (d) Partitioned Bagging Model B - additive preclassification.

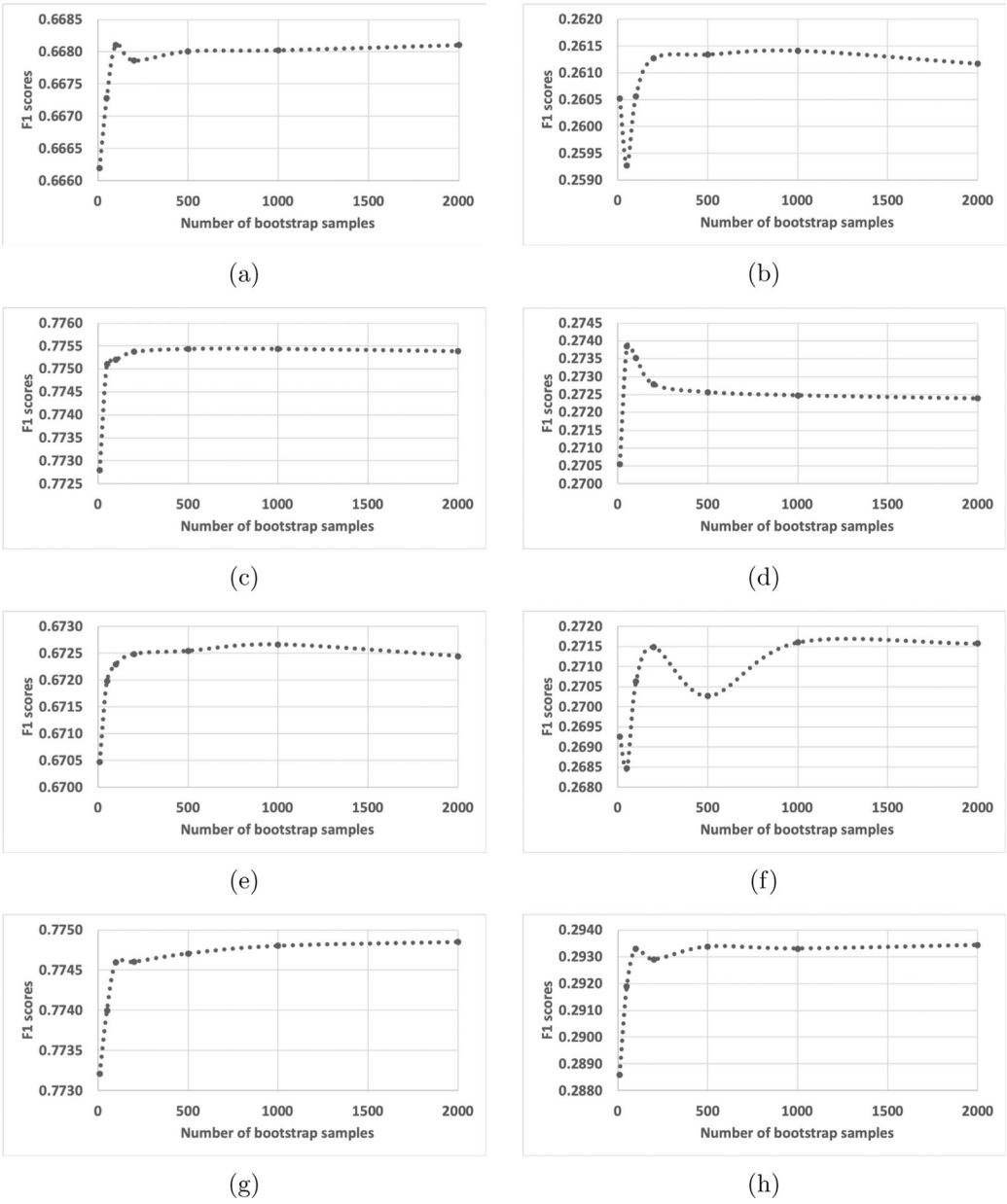

Fig. 3.

Micro and macro-averaged F1 scores per number of bootstrap samples being aggregated. (a) MT-CNN subsite micro-F1, (b) MT-CNN subsite macro-F1, (c) MT-CNN histology micro-F1, (d) MT-CNN histology macro-F1, (e) MT-HCAN subsite micro-F1, (f) MT-HCAN subsite macro-F1, (g) MT-HCAN histology micro-F1, and (h) MT-HCAN histology macro F1. Except in (d), we observed that the F1-scores were stable if more than 1000 bootstrap samples were applied.

Table 1

Number of training and test cases of cancer pathology reports from the four SEER registries for which the number of days between the specimen collection and diagnosis or surgery dates was less than or equal to 10.

	A	B	C	D	Total
Train	135,995	87,230	279,222	311,783	814,230
Test	2468	21,594	60,118	53,434	137,614

Table 2

List of data partitions, associated primary cancer sites per group, and the number of training, validation (Val), and testing samples for each primary site code. Note that the partitioning was done by the algorithm described in Section 3.3; thus, the group may not represent the cancer topography defined in the ICD-O-3 coding manual.

Group	Site	Train	Val	Test	Group	Site	Train	Val	Test
0	C00	642	62	73	6	C40	799	103	141
0	C01	3 377	362	538	6	C41	1 393	144	240
0	C02	2 491	248	383	7	C42	68,274	7 584	9 881
0	C03	503	55	104	8	C44	44,535	4 864	10,625
0	C04	959	97	94	9	C47	227	17	17
0	C05	810	84	142	9	C48	1 574	156	225
0	C06	1 004	115	140	9	C49	5 324	538	881
0	C07	1 585	161	264	10	C50	144,230	16,089	24,746
0	C08	445	43	71	11	C51	4 881	541	630
0	C09	4 332	495	650	11	C52	1 005	104	117
0	C10	800	86	159	11	C53	6 362	713	919
0	C11	1 065	102	144	12	C54	22,083	2 428	3 747
0	C12	466	44	26	12	C55	626	57	110
0	C13	580	51	88	12	C56	9 080	990	1 333
0	C14	610	65	83	12	C57	1 229	128	281
0	C15	6 257	673	1 043	12	C58	37	4	1
1	C16	12,374	1 443	2 367	13	C60	843	104	119
1	C17	4 241	480	655	14	C61	54,136	5 979	11,878
2	C18	44,198	4 949	7 275	15	C62	2 651	307	417
3	C19	4 544	469	689	15	C63	149	18	21
3	C20	13,392	1 498	2 346	15	C64	16,033	1 686	2 731
3	C21	2 856	349	494	15	C65	1 781	197	272
3	C22	6 350	734	1 135	15	C66	1 228	136	177
3	C23	1 439	161	240	16	C67	27,165	3 028	3 902
3	C24	2 044	219	347	16	C68	770	110	128
4	C25	13,652	1 532	2 608	16	C69	867	123	112

Group	Site	Train	Val	Test	Group	Site	Train	Val	Test
4	C26	1 097	134	217	16	C70	3 460	353	661
4	C30	700	74	89	17	C71	9 210	1 023	1 593
4	C31	576	76	111	17	C72	1 332	159	215
4	C32	6 049	730	726	17	C73	16,866	1 940	2 447
4	C33	116	16	7	17	C74	534	59	39
5	C34	94,089	10,590	16,276	17	C75	2 296	244	437
6	C37	344	35	46	17	C76	419	44	52
6	C38	1 954	222	322	18	C77	36,900	4 191	5 196
6	C39	4	0	7	19	C80	8 428	943	1 412

Table 3

Classification task performance of the information extraction models: simple model, bagging model, and two partitioned bagging models. We adopted both micro- and macro-averaged F1 scores as a performance metric; micro F1 weighs on individual cases and macro F1 considers a balance of the classes.

	Site		**Subsite**		**Laterality**	
	Micro F1	**Macro F1**	**Micro F1**	**Macro F1**	**Micro F1**	**Macro F1**
Single model						
MT-CNN	0.9082	0.6312	0.6523	0.2483	0.8923	0.5087
MT-HCAN	0.9138	0.6533	0.6632	0.2614	0.8983	0.5006
Bagging model						
MT-CNN	0.9127	0.6402	0.6681	0.2612	0.8980	0.5172
MT-HCAN	0.9168	0.6618	0.6724	0.2716	0.9015	0.5089
Combo	0.9177	0.6660	0.6774	0.2701	0.9022	0.5176
Partitioned bagging model A - abstention classifiers						
MT-CNN	0.8992	0.6104	0.6690	0.2976	0.8982	0.5223
MT-HCAN	0.8992	0.6048	0.6595	0.2565	0.8967	0.5119
Combo	0.9034	0.6117	0.6708	0.2814	0.8992	0.5211
Partitioned bagging model B - additive preclassification						
MT-CNN	0.9042	0.6301	0.6750	0.3098	0.9004	0.5349
MT-HCAN	0.9036	0.6385	0.6656	0.2677	0.8997	0.5239
Combo	0.9047	0.6317	0.6745	0.2946	0.9007	0.5326
	Histology		**Behavior**		**Grade**	
Single model						
MT-CNN	0.7645	0.2601	0.9753	0.8660	0.7599	0.6274
MT-HCAN	0.7684	0.2824	0.9765	0.8713	0.7637	0.6637
Bagging model						
MT-CNN	0.7754	0.2724	0.9776	0.8684	0.7727	0.6390
MT-HCAN	0.7748	0.2934	0.9779	0.8692	0.7717	0.6726
Combo	0.7815	0.2925	0.9791	0.8807	0.7787	0.6677
Partitioned bagging model A - abstention classifiers						
MT-CNN	0.7793	0.3651	0.9807	0.9096	0.7857	0.6384
MT-HCAN	0.7686	0.2935	0.9777	0.8872	0.7752	0.6491
Combo	0.7821	0.3474	0.9806	0.9129	0.7869	0.6488
Partitioned bagging model B - additive preclassification						
MT-CNN	0.7817	0.3664	0.9809	0.9130	0.7864	0.6482
MT-HCAN	0.7704	0.2989	0.9777	0.8852	0.7752	0.6305
Combo	0.7828	0.3488	0.9806	0.9127	0.7870	0.6419

Table 4

Hours required to train MT-CNN and MT-HCAN models. A node hour is the total time budget spent to train the models on Summit nodes, and Wall time is the actual time from the start of training the models to the end.

		Single	Bagging	Partitioned 1	Partitioned 2
MT-CNN	Node hours	2.32	776.39	1601.85	1361.39
	Wall time Hours	2.32	2.33	0.48	2.08
MT-HCAN	Node hours	11.01	3668.98	12,120.37	11,514.35
	Wall time hours	11.01	11.01	3.64	12.45