Filter Pruning of Convolutional Neural Networks for Text Classification: A Case Study of Cancer Pathology Report Comprehension

Hong-Jun Yoon[1], Sarah Robinson[2], J. Blair Christian[1], John X. Qiu[1] and Georgia D. Tourassi[1]

***Abstract*— Convolutional Neural Networks (CNN) have recently demonstrated effective performance in many Natural Language Processing tasks. In this study, we explore a novel approach for pruning a CNN's convolution filters using our new data-driven utility score. We have applied this technique to an information extraction task of classifying a dataset of cancer pathology reports by cancer type, a highly imbalanced dataset. Compared to standard CNN training, our new algorithm resulted in a nearly .07 increase in the micro-averaged F1-score and a strong .22 increase in the macro-averaged F1-score using a model with nearly a third fewer network weights. We show how directly utilizing a network's interpretation of data can result in strong performance gains, particularly with severely imbalanced datasets.**

I. Introduction

Cancer surveillance is the ongoing, timely, systematic collection and analysis of data relating to new cancer cases, the disease's extent, screening tests, treatment, survival, and cancer deaths [1]. The resulting intelligence can be used to identify trends over time, cancer patterns in certain regions or populations, and evaluate the impact of screening and other prevention measures. In 2016, an estimated 1,685,210 new cases of cancer were diagnosed and 595,690 people died from the disease in the United States; equivalent to 454.8 new cases per 100,000 people, and 171.2 cancer deaths per 100,000 people [2]. Extracting structured data at this scale with only manual human efforts is labor intensive, costly, and error prone. Therefore, the cancer surveillance research communities have expended much effort automatically extracting information from natural language data, an essential application of big data science.

Pathology reports are unstructured text documents containing specific and detailed descriptions of human tissue specimens. They are a standard component of the clinical reporting and management of cancer patients. In addition, cancer pathology reports are a primary source of information for the Surveillance, Epidemiology, and End Results (SEER) program. The SEER program is the premier population cancer surveillance program covering 30% of the US population [3]. It is an important national resource for monitoring cancer outcomes across demographic groups, geographic regions, and time. Furthermore, SEER provides unique insights into the impact of oncology practice outside the clinical trial setting. Collected data includes patient demographics, primary tumor site, tumor morphology and stage at diagnosis, the first course of treatment, and follow-up for vital status.

Deep Learning (DL) approaches based on Convolutional Neural Networks (CNN) have shown success in many areas of machine learning and pattern recognition research [4]. Originating in computer vision applications, CNN techniques have been adapted to natural language text classification [5]. The feasibility of such a model was studied in the Bioinformatics domain [6], which reported that the CNN models outperformed conventional machine learning techniques. Our previous work [7] demonstrated primary cancer subsite identification from cancer pathology reports using the CNN, which outperformed popular traditional learning techniques such as a Support Vector Machine (SVM) classifier [8] and text feature extraction using Term Frequency-Inverse Document Frequency (TF-IDF) [9].

DL models are characterized by the ability to produce effective abstract representations of data using automatic latent feature extraction [10]. To accomplish this, deep neural networks are substantially more complex compared to more traditional machine learning techniques, and require many orders of magnitude more parameters to be trained. To better understand how a network interprets data, researchers have recently examined how individual components or intermediate layers of a deep network respond to an input [11]. Some recent efforts expanded this further by using a deep neural network's intermediate responses to identify and remove superfluous sections of a network and reduce its size, known in the literature as node or filter pruning [12].

In this paper, we investigated the learning mechanism of the CNN for text classification, suggested a utility measurement that represents the power of distinguishing relevant and non-relevant contents, and implemented an algorithm of iterative processes of convolution filter pruning followed by fine-tuning with the remaining filters. This work demonstrates that the algorithm maintains or increases task performance while reducing unnecessary filtering, while preserving classification performance of the underrepresented classes. Cancer types and their subtypes on the pathology report corpus,

This manuscript has been authored by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of the manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

[1]Hong-Jun Yoon, J. Blair Christian, John X. Qiu and Georgia D. Tourassi are with the Biomedical Sciences, Engineering and Computing Group, Oak Ridge National Laboratory, Oak Ridge, TN 37931 USA yoonh@ornl.gov, christianjb@ornl.gov, qiujx@ornl.gov, tourassig@ornl.gov

[2]Sarah Robinson is with Rice University, Houston, TX 77005 USA srobinson@rice.edu

TABLE I
TWELVE PRIMARY CANCER SUBSITES OF BREAST AND LUNG CANCERS AND NUMBER OF CASES IN THE DATASET

Code	Description	# cases
C34.0	Main Bronchus	26
C34.1	Upper Lobe of Lung	139
C34.2	Middle Lobe of Lung	11
C34.3	Lower Lobe of Lung	78
C34.9	Lung, NOS	191
C50.1	Central Portion of Breast	13
C50.2	Upper-Inner Quadrant of Breast	36
C50.3	Lower-Inner Quadrant of Breast	10
C50.4	Upper-Outer Quadrant of Breast	63
C50.5	Lower-Outer Quadrant of Breast	21
C50.8	Overlapping Lesion of Breast	62
C50.9	Breast, NOS	292

review of the word-based CNN model and its application to the filter pruning, and performance evaluation metrics are described in Section II, experimental results of the proposed filter pruning algorithm and evaluations are in Section III, and, lastly, the discussion of the results and future research are addressed in Section IV.

II. METHODS

A. Cancer Pathology Report Data

Our analysis used a corpus of 942 de-identified pathology reports annotated with 12 ICD-O-3 topography codes corresponding to 7 breast and 5 lung primary sites. The pathology reports were provided from five different SEER cancer registries (CT, HI, KY, NM, Seattle) with the proper IRB-approved protocol.

For our gold standard, cancer registry experts manually annotated using the standardized SEER coding guidelines [13]. For label consistency in our training set, we used only pathology reports with a single topography code sourced only from the Final Diagnosis section of the report to minimize variation in our training data. Since pathology report sections vary across pathology labs and registries, we aggregated the text content of every section in the pre-processing phase. The average length of the reports was 469 words. Table I includes the number and types of pathology reports provided by each SEER registry in our dataset.

Of particular note is the heavy class imbalance of the "Not Otherwise Specified" or NOS codes (C34.9, C50.9). This is because some reports were produced from a patient's post diagnostic follow-up visit, which can produce a less detailed report which coding guidelines denote as NOS. This allows our experiments to be more robust and demonstrates the ability to handle imprecise codes.

B. Convolutional Neural Networks for Text Classification

Yoon Kim's word-based CNN model [5] is a popular model for natural language text comprehension. In this approach, documents are represented as 2 dimensional matrices by mapping each word token to a word embedding vector with dimensionality k. These word embeddings can be pre-trained by the training corpus, pre-trained by a general-purpose text database, or initialized by uniformly distributed random numbers. Specifically, a tokenized document of length l, is transformed into a $l \times k$ matrix. However, if a tokenized document's length is shorter than l, the sequence will be zero padded; if longer than l, the document token sequence will be truncated. For our domain-specific classification task, we truncate the beginning of the document to preserve the end of the document's final diagnosis section. For our dataset, we set l as 1500 tokens, approximately the 98th percentile of our tokenized dataset.

For our convolution layer, we applied convolution filters of size $h \times k$ a 1-dimensional stride to the document matrix followed by the Rectified Linear Unit (ReLU) activation function. We then applied global max-pooling, resulting in a scalar value corresponding to the maximum filter response given a particular document matrix input. Therefore, in our word-based CNN, the convolution filters are trained to parameterize the relative discriminative power of a h-word phrase and the global max-pooling selects the most discriminative phrase.

C. Pruning Convolution Filters on the Convolutional Neural Networks for Text Classification

Pruning of neural networks has a long history [14]. Traditionally, network node pruning algorithms utilized saliency scores based on weight values between nodes. We can assume those nodes are tightly coupled if the weight between the nodes is high.

We propose an analogous approach based on the feature mapping property of convolution filters in CNNs [12]. An individual filter contributes to the overall classification task by feeding forward only a scalar activation. Training multiple filters in parallel enables individual filters to become more specialized in their purpose as feature extractors. We can then determine the utility of a particular filter given by its max-pooled activation variability given some set of input data. A filter with small activation variability implies that filter does not convey much discriminative ability for the network's classification task. Our implementation used the training set to calculate the variance of each filter's globally max-pooled activation, which we used as our filter utility score. We can use this utility score to identify which filters to prune. Our algorithm iteratively prunes convolution filters in the lowest x percentile of utility scores, then fine-tunes the CNN by adjusting the remaining convolution filters. The pseudo-code of the algorithm is described in Algorithm 1.

D. Performance Metrics

For our experiments we utilize 10 fold cross validation by randomly partitioning our dataset into 10 sets using each set once as our test set and the others as our train set. Classification performance on the primary cancer subsite was evaluated with each fold's test sets. We evaluated performance with the F1 score, which is the harmonic mean of precision and recall, as it is widely accepted in the natural language processing community [15]. Due to our dataset's heavy class imbalance, we report both micro-averaged and macro-averaged F1 scores. Micro-averaging

Algorithm 1 Filter pruning for word-based CNN

Input: M training set of pathology reports represented as token sequences

repeat
  $N \leftarrow GetConvFilters(CNN)$
  for each n in N **do**
    for each m in M **do**
      $w \leftarrow WordVectorMatrix(m)$
      $c \leftarrow ReLU(1DConvolution(n, w))$
      $P(n, m) \leftarrow GlobalMaxPooling(c)$
    end for
    $U(n) \leftarrow Var(P(n))$
  end for
  $\theta \leftarrow percentile(U, x)$
  for each n in N **do**
    if $U(n) > \theta$ **then**
      $N_{opt}(n) \leftarrow N(n)$
    end if
  end for
  $CNN_{opt} \leftarrow FineTuning(CNN(N_{opt}, M))$
until stopping criteria has not been met

effectively weights each class's performance by prevalence using class-agnostic precision and recall. Macro-averaged F1 scores use a simple average of each class's F1 scores and more effectively describes performance in under-represented and difficult classes.

III. Experimental Results

We classified primary cancer subsites from pathology reports of breast and lung cancers using a word-based CNN classifier with word vector length $k = 300$, and convolution filters of sizes $h = 3, 4, 5$. The maximum word length of documents was set to $l = 1500$. We did not set any fully collected hidden layers.

The training started with 900 convolution filters – 300 from each filter length. We trained the CNN model using the Adam optimizer [16] with stopping criteria based on validation loss. Once our model validation loss converged, we identified and eliminated convolution filters with low utility scores, which we calculated as the variance of the filter's max-pooled activation over each training sample. For each cross validation fold's training set, we removed the convolution filters with utility scores in the lowest decile. We repeated 20 iterations of this train - prune procedure to observe how the task performance is influenced by the remaining convolution filters. We implemented the algorithm using Keras [17] with a TensorFlow [18] backend. We performed the experiments on Summitdev at the Oak Ridge Leadership Computing Facility with NVIDIA P100 Graphics Processing Units. Micro and Macro-F1 scores are shown in Figure 1 for different amounts of pruning.

The results in Figure 1 illustrate that the proposed convolution filter pruning algorithm boosts Micro-F1 scores significantly from 0.72 to 0.78 as well as Macro-F1 scores from 0.35 to 0.55 using fewer convolution filters. This

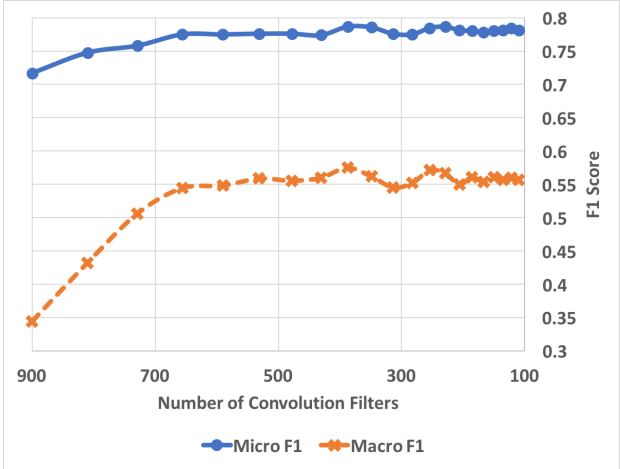

Fig. 1. Micro and Macro-F1 scores of 10-fold cross validation results of CNN's from the 900 convolution filters followed by 20 iteration of filter pruning the utility scores in the lowest decile.

TABLE II
Comparison of Classification Performances between Convolution Filter Pruning Algorithm and Traditional CNN Training by Number of Convolution Filters

# filters	Pruning		Traditional	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1
430	0.774	0.560	0.717	0.344
387	0.787	0.575	0.719	0.355
282	0.775	0.552	0.732	0.373
150	0.780	0.561	0.725	0.349
108	0.781	0.556	0.717	0.338

suggests that the proposed algorithm efficiently eliminates unnecessary convolution filters. It is even evident that the classification power holds with just 100 convolution filters.

For comparison purposes, we trained a conventional word-based CNN with the same number of convolution filters as the pruned CNN. Comparisons of task performance between the pruning algorithm and the traditional CNN training algorithm are listed in Table II. The suggested algorithm consistently yielded much higher task performance than the traditional method, even with a much smaller number of convolution filters. This suggests the filter pruning algorithm is an efficient approach to cut down the low-utility filters.

Table III lists our confusion matrix for evaluating predictions from the CNN with our filter pruning algorithm with 387 convolution filters. Our algorithm correctly classified minor classes (e.g.: C34.0 and C50.5) that the traditionally trained CNN often failed to classify correctly [7]. However, some of the severely underrepresented classes were not classified correctly (e.g.: C50.1 and C50.3), suggesting the need for further research and refinement.

IV. Discussion

In this paper, we introduced a new word-based CNN training and filter pruning algorithm to achieve both better task performance and increased computational efficiency. We evaluated convolution filters using the variance of their global

TABLE III

CONFUSION MATRIX OF THE RESULT FROM THE CONVOLUTION FILTER PRUNING ALGORITHM WITH 387 CONVOLUTION FILTERS, LABELS IN THE TABLE ARE 1: C34.0, 2: C34.1, 3: C34.2, 4: C34.3, 5: C34.9, 6: C50.1, 7: C50.2, 8: C50.3, 9: C50.4, 10: C50.5, 11: C50.8, 12: C50.9

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	12	2	1	1	10	0	0	0	0	0	0	0
2	0	123	1	6	7	0	0	0	0	0	0	2
3	0	0	7	2	2	0	0	0	0	0	0	0
4	1	1	1	64	10	0	0	0	0	0	0	1
5	2	19	0	9	159	0	0	0	1	0	0	1
6	0	0	0	0	0	0	0	0	0	2	2	9
7	0	0	0	0	0	0	12	0	10	0	4	10
8	0	0	0	0	0	0	1	0	0	6	1	2
9	0	0	0	0	0	0	10	0	36	0	0	17
10	0	0	0	0	0	0	0	0	1	13	1	6
11	0	0	0	0	1	0	2	0	3	0	42	14
12	0	0	0	0	1	0	0	0	9	2	7	273

max pooling outputs from the training set. Experimental results support the effectiveness of our utility score in eliminating unnecessary convolution filters that are either too general or too specific. Therefore, we achieved substantially higher Micro and Macro-F1 scores than the traditional word-based CNN training method while using a similar number of convolution filters. Moreover, we can maintain high task performance even with a small fraction of the convolution filters. Substantial improvement of the Macro-F1 score is a strong indication that the optimal convolution filters are efficiently describing cases of underrepresented classes, which is a highly desirable characteristic for classifying a severely imbalanced text corpus such as cancer pathology reports.

Our algorithm improves a trained CNN model by observing the network's responses to data and intelligently modifying the model's architecture. Identifying optimal network structure and other network hyper-parameters has recently received much focus from researchers since current standard practice is to use default configurations described in literature or utilizing costly search algorithms. Our approach improves upon convention by utilizing actual data and observing intermediate interactions to efficiently adapt the model beyond the initialization phase. Identifying additional utility scores could further improve performance or allow for additional model compression.

This study was currently focused solely on word-based CNN models, assessing only the utility of the convolution filters at a single layer of the network. Further research will be conducted to develop more utility measurements capable of evaluating multiple layers of convolution filters or multiple layer types, such as the fully-connected layer.

ACKNOWLEDGMENT

This work has been supported in part by the Joint Design of Advanced Computing Solutions for Cancer (JDACS4C) program established by the U.S. Department of Energy (DOE) and the National Cancer Institute (NCI) of National Institutes of Health. This work was performed under the auspices of the U.S. Department of Energy 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 Oak Ridge National Laboratory under Contract DE-AC05-00OR22725.

The authors wish to thank Valentina Petkov of the Surveillance Research Program from the National Cancer Institute and the SEER registry at Connecticut, Hawaii, Kentucky, New Mexico and Seattle for the pathology reports used in this investigation.

This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S., Department of Energy under Contract No. DE-AC05-00OR22725.

REFERENCES

[1] Thomas J Smith, Nancy E Davidson, David V Schapira, Eva Grunfeld, Hyman B Muss, Victor G Vogel III, and Mark R Somerfield. American society of clinical oncology 1998 update of recommended breast cancer surveillance guidelines. *Journal of Clinical Oncology*, 17(3):1080–1080, 1999.

[2] Rebecca L Siegel, Kimberly D Miller, and Ahmedin Jemal. Cancer statistics, 2016. *CA: a cancer journal for clinicians*, 66(1):7–30, 2016.

[3] Bradford W Hesse, David Ahern, and Ellen Beckjord. *Oncology informatics: using health information technology to improve processes and outcomes in cancer*. Academic Press, 2016.

[4] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[5] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[6] Sebastian Gehrmann, Franck Dernoncourt, Yeran Li, Eric T Carlson, Joy T Wu, Jonathan Welt, John Foote Jr, Edward T Moseley, David W Grant, Patrick D Tyler, et al. Comparing rule-based and deep learning models for patient phenotyping. *arXiv preprint arXiv:1703.08705*, 2017.

[7] John Qiu, Hong-Jun Yoon, Paul A Fearn, and Georgia D Tourassi. Deep learning for automated extraction of primary sites from cancer pathology reports. *IEEE Journal of Biomedical and Health Informatics*, 2017.

[8] Johan AK Suykens and Joos Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999.

[9] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[11] Sajid Anwar, Kyuyeon Hwang, and Wonyong Sung. Structured pruning of deep convolutional neural networks. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 13(3):32, 2017.

[12] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient transfer learning. *arXiv preprint arXiv:1611.06440*, 2016.

[13] MB Adamo, LA Dickie, and JL Ruhl. SEER Program Coding and Staging Manual 2016. *Bethesda, MD: National Cancer Institute*, 2016.

[14] Russell Reed. Pruning algorithms-a survey. *IEEE transactions on Neural Networks*, 4(5):740–747, 1993.

[15] Jakub Piskorski and Roman Yangarber. Information extraction: past, present and future. In *Multi-source, multilingual information extraction and summarization*, pages 23–49. Springer, 2013.

[16] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[17] François Chollet et al. Keras. `https://github.com/fchollet/keras`, 2015.

[18] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.