Deep Learning for Automated Extraction of Primary Sites From Cancer Pathology Reports

John X. Qiu, Hong-Jun Yoon, Paul A. Fearn, and Georgia D. Tourassi, *Senior Member, IEEE*

***Abstract*—Pathology reports are a primary source of information for cancer registries which process high volumes of free-text reports annually. Information extraction and coding is a manual, labor-intensive process. In this study, we investigated deep learning and a convolutional neural network (CNN), for extracting ICD-O-3 topographic codes from a corpus of breast and lung cancer pathology reports. We performed two experiments, using a CNN and a more conventional term frequency vector approach, to assess the effects of class prevalence and inter-class transfer learning. The experiments were based on a set of 942 pathology reports with human expert annotations as the gold standard. CNN performance was compared against a more conventional term frequency vector space approach. We observed that the deep learning models consistently outperformed the conventional approaches in the class prevalence experiment, resulting in micro- and macro-F score increases of up to 0.132 and 0.226, respectively, when class labels were well populated. Specifically, the best performing CNN achieved a micro-F score of 0.722 over 12 ICD-O-3 topography codes. Transfer learning provided a consistent but modest performance boost for the deep learning methods but trends were contingent on the CNN method and cancer site. These encouraging results demonstrate the potential of deep learning for automated abstraction of pathology reports.**

***Index Terms*—Convolutional neural network, deep learning, information extraction, natural language processing, pathology reports, primary cancer site.**

I. Introduction

CANCERis the second leading cause of death in the United States [1]. Over the course of a cancer patient's diagnosis and treatment, pathologists record highly descriptive and specific observations of cells and tissues in pathology reports. Because these individualized yet mass produced clinical reports are encoded in mostly unstructured text, the potential accessibility of vast amounts of data is contingent on the performance of natural language processing (NLP) tools for automated information extraction [2].

Cancer registries process a very high volume of pathology reports, hundreds of thousands in the Surveillance, Epidemiology and End Results (SEER) registries alone, which cover 30% of US population [3]. These reports are highly variable, as they come from hundreds of healthcare providers and laboratories. There are also data quality issues due to human fatigue, differences in interpretation and application of coding rules, etc. Moreover, the clinical details from pathology and other reports that are needed to characterize cancer patient trajectories are increasing as patients live longer and have more complex treatments. At this scale, manual information extraction and coding is expensive to sustain, and registries are unlikely to address issues of volume, variability, and timeliness of reporting without some automation.

Since the registries have access to high volumes of electronic pathology reports and coded variables extracted from those reports, a machine learning approach to feature and classifier development could offer an effective path for registries to implement automation using artificial intelligence for information extraction and coding. For cancer registries, an important piece of information in a pathology report is the corresponding ICD-O-3 topographical code, which describes the specific anatomical site of a tumor's origin [4]. Although multiple sites may be mentioned in a pathology report, for the most part, only one primary site is discussed per report. Extraction and coding of primary sites by ICD-O-3 topographical codes provides a well-documented starting point for exploration of deep learning NLP techniques to support cancer surveillance.

Within the biomedical informatics literature, NLP approaches for information extraction range from strictly defined, manually tuned rule-based systems [5], to generalized and domain-specific systems relying on feature-based machine learning classifiers [6], to recent deep learning approaches for both automated feature development [7] and supervised classifier models [8]. Though these prior techniques have individually demonstrated effective performance within their own experiments, a systematic comparison of methods across varying distributions of clinical documents has yet to be studied.

Since 1971, Vector Space Models (VSM) utilizing term and document frequency counts as automatic document feature extractors have been used to retrieve information from a corpus of

Manuscript received December 6, 2016; revised March 28, 2017; accepted April 26, 2017. Date of publication May 3, 2017; date of current version January 3, 2018. This work has been supported in part by the Joint Design of Advanced Computing Solutions for Cancer program established by the U.S. Department of Energy and in part by the National Cancer Institute of the National Institutes of Health. *(Corresponding author: G.D. Tourassi.)*

J. X. Qiu is with the University of Tennessee, Knoxville, TN 37996 USA, and the Health Data Sciences Institute, Oak Ridge National Laboratory, Oak Ridge, TN 37831 USA (e-mail: jqiu1@utk.edu).

P. A. Fearn is with the National Cancer Institute, Surveillance Research Program, Bethesda, MD 20850 USA (e-mail: paul.fearn@nih.gov).

H. Yoon and G. D. Tourassi are with the Biomedical Sciences, Engineering, and Computing Group, Computational Sciences and Engineering Division and the Health Data Sciences Institute, Oak Ridge National Laboratory, Oak Ridge, TN 37831 USA (e-mail: yoonh@ornl.gov; tourassig@ornl.govs).

Digital Object Identifier 10.1109/JBHI.2017.2700722

documents [9], [10]. Recent deep learning applications for NLP have resulted in great progress for both automated feature extraction and classification tasks. Multilayered neural networks extend automated feature extraction techniques beyond document-level representation to a machine learned latent vector space representation for words. These word-level learned features do not need to be task specific, as word training has been performed on unlabeled but vast corpora such as Wikipedia or a national periodical such as the Wall Street Journal [11]. These learned word features are called "word embeddings" and like document VSMs, attempt to capture semantic information via observed similarities in word contexts. This completely automated and generalized approach nonetheless resulted in state-of-the-art performance in NLP benchmark subtasks against even highly engineered task-specific prior techniques [11]. Researchers have investigated techniques for making this feature extraction more task-specific with either word pre-training on a domain specific corpus [7], [12] or by simultaneously learning latent word features alongside a particular class task [11]. Previous deep learning approaches focused on the sequential nature of text data with recurrent networks [13] or encoding documents into individual latent feature vectors [14]. Such approaches focus mainly on documents-wide attributes, such as form and structure. Convolutional neural networks (CNN), initially developed for computer vision [15] and subsequently applied to NLP applications [16], demonstrated superior performance for document-level information extraction and classification utilizing word embeddings [17]. Specifically, the CNN's convolving filters and max-over time pooling scheme result in a highly effective document feature extraction and selection [17], which greatly improve upon the traditional VSM's limitations of a sparse high dimensionality feature space and inability to directly utilize word order.

In this paper, we investigate the performance of these recent deep learning NLP techniques for extracting ICDs-O-3 codes by conducting two experiments. First, we examine the effect of dataset size and class imbalance on classifier performance. Therefore, our first experiment uses a dataset with only well-populated classes and an expanded dataset including both well and minimally populated classes. In the second experiment, we explore the learning tradeoffs of increased data specificity versus increased data size. We partition our data by primary organ site and evaluate classifier performance against classifiers trained across primary sites.

The manuscript is organized as follows: Section II discusses our dataset used in the information extraction task and describes our model and our experimental procedure in detail. Section III presents our experimental results which we discuss in Section IV.

II. Materials and Methods

A. Pathology Dataset

Our analysis used a corpus of 942 de-identified pathology reports matched to 12 ICD-O-3 topography codes corresponding to 7 breast and 5 lung primary sites. For 6 ICD-O-3 codes the dataset included at least 50 observations per code but the remaining 6 ICD-O-3 codes were minimally populated with at least 10 but less than 50 observations per code. Table I describes the 12 topography codes and the corresponding observation count per code included in the database.

TABLE I
ICD-O-3 Topographical Codes

Code	Count	Description
C34.0	26	Main Bronchus
C34.1	139	Upper lobe, lung
C34.2	11	Middle lobe, lung
C34.3	78	Lower lobe, lung
C34.9	191	Lung, NOS
C50.1	13	Central portion of breast
C50.2	36	Upper-inner quadrant of breast
C50.3	10	Lower-inner quadrant of breast
C50.4	63	Upper-outer quadrant of breast
C50.5	21	Lower-outer quadrant of breast
C50.8	62	Overlapping lesion of breast
C50.9	292	Breast NOS

TABLE II
Pathology Report Origin Repository

Repository	Breast (C50)	Lung (C34)
HI	109	102
KY	96	105
NM	131	113
CT	44	32
Seattle	117	93

The pathology reports were provided from five different SEER cancer registries (CT, HI, KY, NM, Seattle) with the proper IRB-approved protocol. Cancer registry experts manually annotated all pathology reports based on standard guidelines and coding instructions used in cancer surveillance. Their annotations served as the gold standard. For label consistency in our training set, we only used pathology reports with a single topography code sourced only from the "Final Diagnosis" section of the report to minimize variation in our training data, though we investigated further the robustness of our findings when including reports with ground truth labels sourced from other report sections (i.e., e.g. Clinical History, Microscopic Description, Clinical Information). Since some report sections available in each pathology report 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 II includes the number and types of pathology reports provided by each SEER registry.

B. Vector Space Models and Word-Feature Training

With VSMs, documents within a corpus are encoded with a feature vector based on word counts, also known as the 'bag of words' feature representation [10]. For decades, this conceptually simple and relatively effective document representation was the basis for many NLP tasks, particularly information retrieval and extraction [8], [9]. Likewise, word embedding approaches use observed similarities to derive meaning from word occurrence. Unlike VSMs, word embeddings are learned

representations of words rather than observed representations of documents, and are trained with temporal context windows utilizing deep learning feature development techniques. The word vector representation is expressed as the following: given a sequence of words $x_1, x_2, \ldots x_t$, there are corresponding word vectors $w_1, w_2, \ldots, w_T$ where $w_i \in \mathbb{R}^k$ [16]. One recent approach to training word vectors is Mikolov's word2vec skip-gram method, in which vectors are initialized randomly and a corpus's word vectors values are trained by generating contexts for the corpus words and maximizing the following objective function for a context containing word w_t:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c, j\neq 0}\log\ p\left(w_{t+j}|w_t\right)$$

where c is the size of the training context $w_1, w_2, \ldots, w_T$ and p is the hierarchical f function [18] In essence, word vectors are trained to predict the text context in which they are likely to appear. In practice, word embeddings are trained by stochastically traversing a corpora's contexts, building an embedding matrix $\boldsymbol{W} \in \mathbb{R}^{v\times k}$, where k is the pre-specified dimensionality for the vector space for learning embeddings, and v being the number of learned vectors, effectively corresponding to a vocabulary size.

Since trained word vectors trained to represent a word's locality within a corpus, training word vectors with a different corpus can capture varying word meanings Using a more domain specific corpus has been observed to result in improved performance across various biomedical information extraction tasks [7]. Therefore, in our experiments, we compared the performance of a deep learning document classifier using word2vec skip-gram learned embeddings from i) a general purpose corpus, trained with articles collected by Google News, ii) a domain specific corpus in the form of word embeddings trained with biomedical publications hosted on PubMed, as well as iii) untrained randomly initialized embeddings. Using only randomly initialized embeddings, is effectively equivalent to performing word feature learning on the available corpus of clinical documents. We used Kim's [17] word vector space of 300 and trained the PubMed embeddings using the Gensim python package [19] but imported pre-trained Google News embeddings.

1) Preprocessing: After extracting the pathology report text content, we tokenized each document by removing all non-alphanumeric symbols and setting all alphabetical characters to lowercase. Though non-alphanumeric characters can hold semantic meaning, in our dataset these symbols are often ambiguous, such as a period appearing in both end of a sentence and in a decimal number. For the two pre-trained word embeddings, tokenized words were matched with their pre-trained embedding's word vector index; if no embedding existed in the embeddings vocabulary but the word's pathology report document frequency was at least 2, a new word embedding vector would be initialized, otherwise the word would be mapped to an embedding vector corresponding to all words lacking a unique mapping. The decision to remove all non-alphanumeric characters as well as the minimum document frequency was made after considering the size and text content of our dataset and was informed with experimental evidence, though we found these preprocessing decisions did not have a statistically significant effect on performance. As our network is trained, these embedding vectors can be updated via back-propagated errors, thus making these latent features more task-specific. A pathology report with n words corresponding to word vectors $w_i \in \mathbb{R}^k$ can be represented by a document matrix $\boldsymbol{A} \in \mathbb{R}^{n\times k}$. To accommodate for documents of multiple lengths in our network, we specify a document length parameter n so that all documents longer than n will be truncated and documents shorter than n will be padded with corresponding tokens [16], [17].

2) Network Architecture: In machine learning, a convolution is an automatic feature generation technique which processes an input with a trainable regional filter. Similar to applications for image processing, we can apply a convolutional filter to the document matrix with a linear filter with region size h that corresponds to a context length of h word vectors. Specifically, we can parameterize a linear filter as a weight matrix $\boldsymbol{w}$ with dimensions $k \times h$. A context window for matrix $\boldsymbol{A}$ starting from the ith word vector of $\boldsymbol{A}$ with length h can be represented by submatrix $\boldsymbol{A}[i : i + h - 1]$ [16]. A single convolution on the document's ith word with context length h can be denoted as $o_i = w \cdot A[i : i + h - 1]$ where $o_i \in \mathbb{R}^{n-h}$. Finally, we apply an activation function f with a bias term $b \in \mathbb{R}$ to o_i, inducing a single feature map $c_i = f(o_i + b)$. The final output of a convolutional filter over document matrix $\boldsymbol{A}$ can be expressed as feature mapping $\boldsymbol{c} = c_1, c_2 \ldots c_n$, which is a representation of every context window of length h over document matrix $\boldsymbol{A}$ [16], [17].

Through the extraction of all contexts of a particular length, our CNN learns to consider contexts of multiple sizes simultaneously when classifying a pathology report, producing a "feature mapping" to represent each context size. The subsequent pooling layer trains the convolutional filter to extract a single feature map scalar from each mapping, aggregating the selected contexts by concatenation. This pooling is a highly efficient form of feature selection which allows our model to classify documents by learning for every convolutional filter which is the most useful context instead of learning some direct document abstraction which can be difficult to fully develop with datasets of our scale [17]. The pooled features connect to a fully connected hidden layer, where we aggregate the selected separate convolutional features and implement our regularization, including dropout. Our final layer is the soft-max (multinomial logistic regression) classifier itself. Further regularization was employed on the hidden layer with an l_2-normalization of weight vectors. Our network weights were trained with the Adadelta adaptive gradient descent algorithm [17]. The CNN architecture is illustrated in Fig. 1.

3) Hyper Parameter Tuning: For guidance in tuning the model's hyper-parameters, we used Zhang et al's CNN sensitivity analysis [16] as a starting point. Further, we incrementally adjusted model parameters to maximize average class performance (Macro-F). By optimizing for Macro-F, we attempted to optimize for algorithm generalizability across different labels with varying features and numbers of training observations.

To accommodate the wide lengths of pathology reports while maintaining reasonable network train times, we defined the

document length input as 1500 words vectors. In our convolutional layer, we used context window sizes of 3, 4 and 5 along with a Rectified Linear Unit activation function. For our pooling strategy, we used max-over time pooling, which selects a single context from the document-spanning convolutions to present to our fully connected hidden layer and the final soft-max classifier. Finally, to account for class imbalance, we weighed our error costs inversely proportional to a class prevalence in the dataset.

B. Experimental Design

Experiment 1 - Class Prevalence: We first compared performance when the model is developed using data from the well populated classes (6 ICD-O-3 classes with a minimum of 50 pathology reports per class) vs. a model developed using data combining the six well-populated classes with six minimally-populated classes containing 10-49 pathology reports per class.

Experiment 2: Transfer Learning: We investigated the cross-label learning capabilities of each model by evaluating classification performance for each primary site (lung, breast) with models trained with only one primary site's data to models trained with the entire dataset.

C. Comparative Analysis

In our experiments, we compared the proposed word embeddings CNN approach with the baseline NLP approach of unigram and bigram term frequency-inverse document frequency (TF-IDF) vector space model combined with conventional classifiers. This document feature representation is considered a standard approach in both the NLP [10], [20] and the biomedical informatics information extraction literature [21], [22]. Similar to the deep-learning approach, we pre-processed our pathology report data by removing, all non-alphabetical and numeric characters, and stop words. We then tokenized the processed pathology reports into *n*-grams of up to length 2 and generated an *n*-gram based term-frequency vector for each report while aggregating a training corpus document-frequency dictionary for each *n*-gram. Through model development experimentation, we concluded that utilizing an *n*-gram feature space of 400 resulted in optimal performance given our dataset and information extraction task. Therefore, we removed all but the top 400 document-occurring *n*-grams from our corpus vocabulary and finalized the pathology report feature vector by dividing each report term frequency vectors by the corpus document frequency vector. We utilized multinomial Naive Bayes (NB), Logistic Regression (LR), and Linear Support Vector Machines (SVM) as our baseline ICD-O-3 topography code classifiers as the literature suggests [20]. To address ICD-O-3 topography code class imbalance, we trained model parameters weighted inversely proportional to an observation's class representation. For network regularization, the optimal dropout rate appeared to be .5, but our l_2 normalization appeared to have little impact on performance metrics.

D. Performance Evaluation

Each of our two experiments contains two classification tasks; 6-class vs 12-class. We utilize a balanced tenfold cross validation scheme by randomly partitioning the dataset into ten parts with near balanced label distributions. For each fold we use one partition once for testing and combine the rest for our training set, evaluating model performance on the combined predicted-actual results from each fold. We evaluated each model primarily using the standard NLP metrics of micro and macro averaged F-scores, the harmonic mean of related metrics precision and recall. For each ICD-O-3 topography code C_j from a set of possible code classes in the subtask $\boldsymbol{C}$, the number of class true positives is denoted TP_j, class false positives are FP_j, and class false negatives are positives FN_j.

We use class-based metrics precision $P(C_i)$, recall $R(C_i)$, and F-score $F(C_i)$ defined as

$$\begin{aligned} P\left(C_i\right) &= \frac{TP_j}{TP_j s \ + \ FP_j} \\ R\left(C_i\right) &= \frac{TP_j}{TP_j \ + \ FN_j} \\ F\left(C_i\right) &= \frac{2P\left(C_i\right)R\left(C_i\right)}{P\left(C_i\right) \ + \ R\left(C_i\right)} \end{aligned} \tag{1}$$

Micro-averaged metrics are defined as

$$\begin{aligned} P^{\text{micro}} &= \frac{\sum_{C_j}^{\mathbf{C}} TP_j}{\sum_{C_j}^{\mathbf{C}} \left(TP_j + \ FP_j\right)} \\ R^{\text{micro}} &= \frac{\sum_{C_j}^{\mathbf{C}} TP_j}{\sum_{C_j}^{\mathbf{C}} \left(TP_j + \ FN_j\right)} \\ F^{\text{micro}} &= \frac{2P^{\text{micro}}R^{\text{micro}}}{P^{\text{micro}} \ + \ R^{\text{micro}}} \end{aligned} \tag{2}$$

Macro-averaged metrics are defined as

$$\begin{aligned} P^{\text{macro}} &= \frac{1}{|\mathbf{C}|} \cdot \sum_{C_j}^{\mathbf{C}} P\left(C_j\right) \\ R^{\text{macro}} &= \frac{1}{|\mathbf{C}|} \cdot \sum_{C_j}^{\mathbf{C}} R\left(C_j\right) \\ F^{\text{macro}} &= \frac{1}{|\mathbf{C}|} \cdot \sum_{C_j}^{\mathbf{C}} F\left(C_j\right) \end{aligned} \tag{3}$$

In summary, micro-averaged metrics have class representation roughly proportional to their test set representation, whereas macro-averaged metrics are averaged by class without weighing by class prevalence [21]. Since micro averaged precision, recall, and F-scores are equivalent for multiclass single-label tasks [23], we will use both the micro and macro F-score metric and only the macro average for the recall and precision metrics.

III. Results

A. Class Prevalence

Table III shows the performance metrics of the first experiment comparing model performance with a minimally-populated 12 class task compared to a well-populated 6 class task. The Table includes 95% confidence intervals for each performance metric, derived using bootstrapping. The proposed

TABLE III
EXPERIMENT I – CLASS PREVELANCE RESULTS

	Approach	Micro-F	Avg	Macro-F	Avg	Macro-P	Macro-R
Minimally Populated – Class Count > 10 – 12 ICD-O-3 topography code classes							
	Naïve Bayes (NB)	0.591 (0.561, 0.622)	0.639	0.222 (0.207, 0.239)	0.259	0.234 (0.217, 0.251)	0.225 (0.207, 0.239)
TF-IDF	Logistic Regression (LR)	0.654 (0.627, 0.688)		0.263 (0.250, 0.281)		0.278 (0.255, 0.294)	0.280 (0.266, 0.294)
	Support Vector Machine (SVM)	0.672 (0.639, 0.700)		0.291 (0.273, 0.303)		0.291 (0.276, 0.300)	0.302 (0.286, 0.315)
	Google News pre-trained	0.705 (0.674, 0.735)	0.713	0.342 (0.324, 0.372)	0.368	0.353 (0.323, 0.372)	0.358 (0.334, 0.374)
CNN	PubMed pre-trained	0.713 (0.676, 0.737)		**0.389** (0.338, 0.411)		**0.399** (0.359, 0.421)	**0.407** (0.360, 0.432)
	No pre-training	**0.722** (0.682, 0.740)		0.372 (0.334, 0.409)		0.372 (0.347, 0.415)	0.393 (0.359, 0.422)
Well Populated – Class Count > 50 - 6 ICD-O-3 topography code classes							
	Approach	Micro-F	Avg	Macro-F	Avg	Macro-P	Macro-R
	Naïve Bayes (NB)	0.679 (0.655, 0.718)	0.728	0.475 (0.452, 0.525)	0.554	0.500 (0.471, 0.523)	0.500 (0.472, 0.525)
TF-IDF	Logistic Regression (LR)	0.744 (0.722, 0.774)		0.546 (0.539, 0.612)		0.587 (0.540, 0.612)	0.552 (0.521, 0.570)
	Support Vector Machine (SVM)	0.760 (0.745, 0.802)		0.640 (0.600, 0.672)		0.684 (0.633, 0.723)	0.625 (0.580, 0.651)
	Google News pre-trained	0.794 (0.756, 0.815)	0.804	0.687 (0.637, 0.708)	0.687	0.727 (0.671, 0.772)	0.692 (0.667, 0.738)
CNN	PubMed pre-trained	0.797 (0.761, 0.817)		0.688 (0.640, 0.712)		0.733 (0.672, 0.779)	0.695 (0.656, 0.723)
	No pre-training	**0.811** (0.762, 0.829)		**0.701** (0.642, 0.723)		**0.737** (0.679, 0.781)	**0.707** (0.649, 0.733)

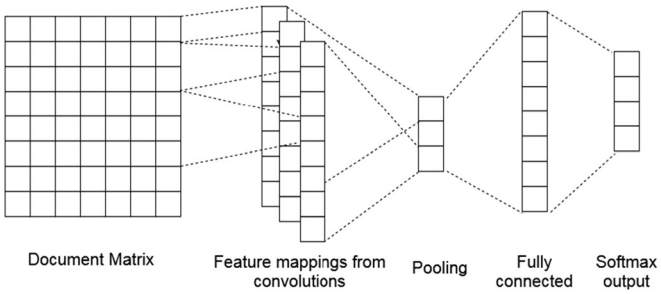

Fig. 1. Convolutional neural network architecture.

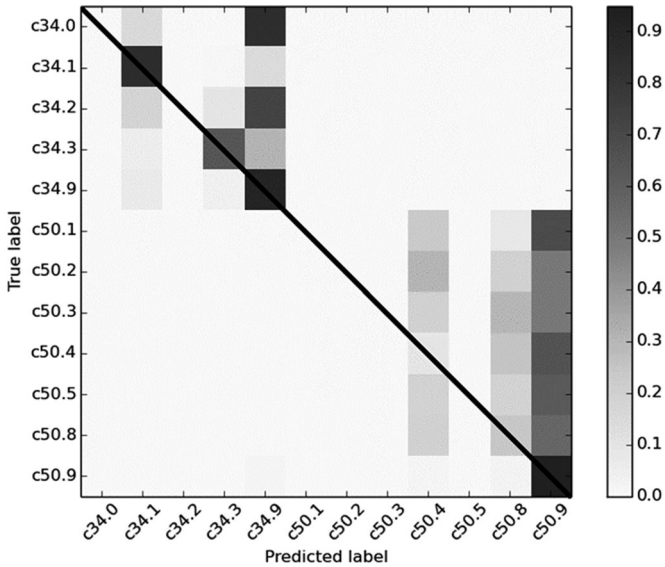

Fig. 2. Normalized confusion matrix for support vector machine minimally populated task.

CNN classifier consistently performed better than the vector-space models in both the minimally populated classification subtask and the well-populated classification subtask. The TF-IDF approaches averaged micro- and macro-F scores at 0.639 and 0.259 respectively and the CNN approaches averaged micro- and macro-F at 0.713 and 0.368 for the minimally populated task. The TF-IDF approaches averaged micro- and macro-F scores at 0.728 and 0.554 respectively and the CNN approaches averaged micro- and macro-F scores at 0.804 and 0.692 for the well populated task. CNNs consistently outperformed all TF-IDF approaches. Although, the CNN performance metrics were consistently higher than those achieved by SVM, most differences were not statistically significant with the exception of Macro-F.

Figs. 2–5 compare the aggregated normalized confusion matrices for the highest Macro-F scoring model from each approach in the minimally populated classification subtask and the well-populated classification subtask. The true-positive diagonal has been identified with a black line. False-positives are indicated on the vertical axis and false-negatives on the horizontal. From this, we see that the CNN approaches predict fewer false positives for the C34.9 and C50.9 classes, particularly for the breast classes in the well populated task.

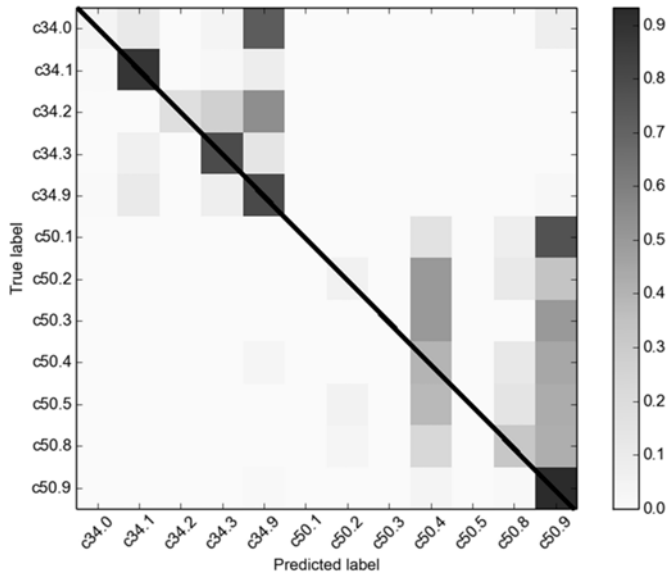

Fig. 3. Normalized confusion matrix for CNN with PubMed word vectors minimally populated task

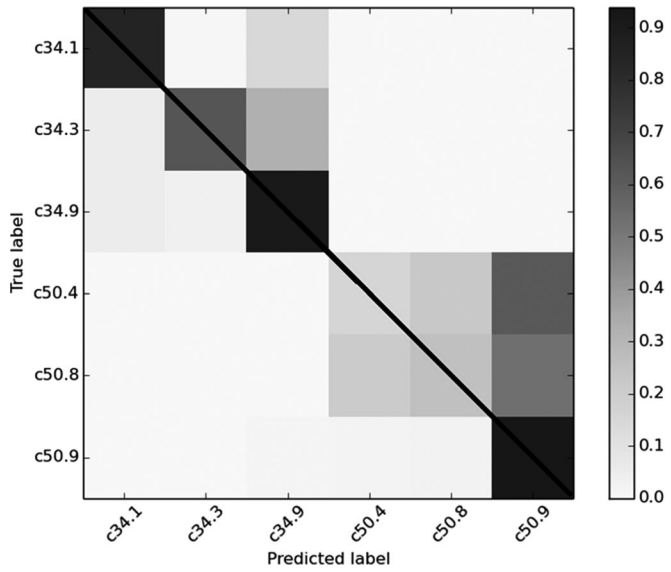

Fig. 4. Normalized confusion matrix for support vector machine well populated task

B. Transfer Learning

Table IV shows the performance metrics of the second experiment comparing separate breast and lung classification performance for primary site models trained on data split by primary site compared to a joint primary site classifier. For the TF-IDF methods, joint training appears to slightly decrease the micro- and macro-F scores. The opposite trend was observed for CNN with joint training resulting in a slight but consistent improvement across embeddings methods and cancer sites. Notable improvement was observed for Breast ICD-O-3 codes; the embeddings without pre-training resulted in substantial performance increases with micro- and macro-F scores of 0.685 and 0.265 respectively compared to 0.631 and 0.193 for split training. For Lung ICD-O-3 codes the PubMed corpus appeared to be superior with joint training resulting in micro- and macro-F scores of 0.782 and 0.606 respectively. and 0.501 and 0.606 for split and joint training respectively. For the breast codes with split training, the Google News embeddings performed best with micro-F scores of 0.644 and macro-F scores of 0.213; the embeddings without pre-training resulted in substantial performance increases in breast class with joint training resulting in micro-F scores of .685 and macro-F scores of 0.265. Overall, this experiment demonstrated the potential benefits of CNN joint training by transferring learning across primary cancer sites although the general trend appeared to be contingent upon the cancer site and the embeddings training method.

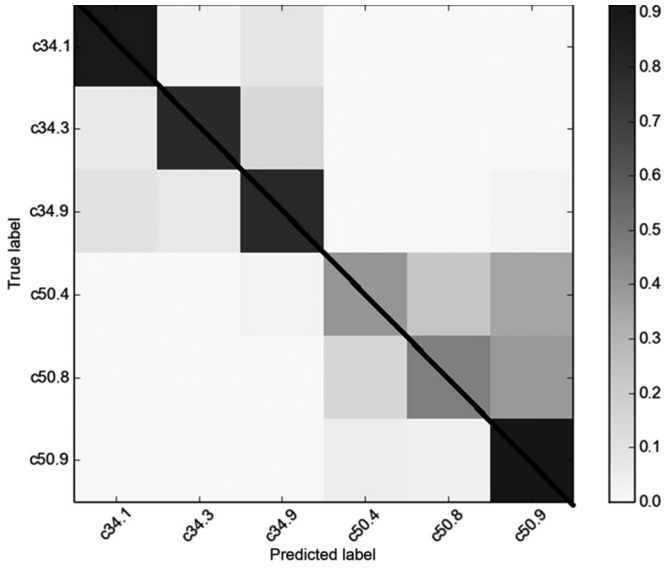

Fig. 5. Normalized confusion matrix for CNN with untrained word vectors well populated task

IV. Discussion

Based on the performance metrics shown in Tables III and IV, we see that the word-vector representation with CNN classifier performs consistently better than conventional TF-IDF approaches on every metric, particularly macro-F. It is notable that the most commonly occurring ICD-O-3 topography labels C50.9 and C34.9 correspond to a breast or lung tumor in location 'Not Otherwise Specified'. The normalized confusion matrices show how class imbalance and label ambiguity result in frequent false-positives for the TF-IDF feature representation, even for the well populated classes. Although the CNN improvements over macro-F scores resulted from improved performance in classifying well populated codes with less than 50 examples (as illustrated in Figs. 2–5), poor performance in classifying rare topographic codes in the minimally populated subtask demonstrates the well-known need for sufficient labeled training data when using deep learning.

In our experiments, we observed that none of the word vector pre-training approaches consistently outperformed the others, though in some specific tasks, one pre-training approach may perform best within that task. For example, compared

TABLE IV
EXPERIMENT 2 – TRANSFER LEARNING RESULTS

Split Trained – 7 Breast and 5 Lung ICD-O-3 Topography Code Classes									
		Breast Generic Site - 7 ICD-O-3 Classes				Lung Generic Site - 5 ICD-O-3 Classes			
	Approach	Micro-F	Macro-F	Macro-P	Macro-R	Micro-F	Macro-F	Macro-P	Macro-R
TF-IDF	NB	0.573	0.182	0.170	0.203	0.638	0.339	0.415	0.357
	LR	0.590	0.143	0.174	0.163	0.737	0.438	0.477	0.443
	SVM	0.618	0.191	0.186	0.204	0.774	**0.509**	**0.500**	**0.590**
CNN	GNews	**0.644**	**0.213**	0.218	**0.236**	0.765	0.473	0.461	0.494
	PubMed	0.631	0.205	**0.221**	0.226	**0.781**	0.501	0.488	0.531
	No pre-training	0.631	0.193	0.185	0.217	0.772	0.481	0.475	0.503
Joint Trained – 12 ICD-O-3 Topography Code Classes									
		Breast Generic Site - 7 ICD-O-3 Classes				Lung Generic Site - 5 ICD-O-3 Classes			
	Approach	Micro-F	Macro-F	Macro-P	Macro-R	Micro-F	Macro-F	Macro-P	Macro-R
TF-IDF	NB	0.564	0.168	0.162	0.184	0.620	0.301	0.411	0.340
	LR	0.586	0.143	0.161	0.161	0.727	0.440	0.461	0.446
	SVM	0.602	0.168	0.160	0.184	0.751	0.472	0.478	0.479
CNN	GNews	0.643	0.241	**0.266**	0.255	0.774	0.485	0.476	0.502
	PubMed	0.653	0.234	0.254	0.256	**0.782**	**0.606**	**0.602**	**0.619**
	No pre-training	**0.685**	**0.265**	**0.266**	**0.291**	0.764	0.522	0.520	0.535

to the other pre-training approaches, the PubMed embeddings achieved the highest performance metrics for both the joint-trained and split-trained lung classification tasks, but performed second-best in the breast classification task to the Google News embeddings for split training and the no-pretraining, embeddings for joint training. Though other researchers identified performance increases from using domain specific or pre-trained embeddings over randomized embeddings [4], [15]. Our labeled dataset's comparatively small size, class distribution, or merely the pathology report's unique language style may be the cause. To explore this, we attempted to utilize *n*-grams derived from a PubMed corpus as features [24]. All our TF-IDF classifiers performed equally poorly for both classification tasks, with dramatically low micro-F and macro-F scores, indicating an inability for this representation to meaningfully differentiate between our pathology reports (detailed results not shown here due to inferior performance). Finally, although we did not observe consistent performance superiority among the pre-training CNN approaches, we can distinguish word embeddings' ability to transfer information across corpuses while maintaining adequate representation of the data.

In our experiments, the various implementation parameters such as the CNN optimal filter window size, the minimum number of word occurrences required for a unique embedding vector, and the exclusion of non-alphanumeric characters were set based literature recommendations that seemed intuitively appropriate for the specific task [16]. For example, despite using entire pathology reports as network inputs, we found the CNN filter window sizes of 3, 4 and 5 used in sentence classification were still effective. This may be task related, as it seems probable that topographical code may be inferred from such a small context. To set the minimum number of word occurrences required for a unique embedding vector, we empirically investigated a range of 1 to 5. We observed that using a minimum document frequency parameter of 2 worked well across all tasks and word embeddings methods, although the differences were not statistically significant. Similarly, with respect to the inclusion or exclusion of non-alphanumeric characters, there was no statistical significance among the various performance metrics. Previous deep learning NLP experiments using less domain-specific training sets have retained non-alphanumeric characters as individual punctuation tokens with individual embedding vectors [16]. However, we chose to exclude the non-alphanumeric characters since they tend to be ambiguous in pathology reports. Although the above implementation decisions had marginal effects on classifier performance, it remains unclear if our observations are a result of the writing style of the pathology reports, the particular information extraction task, or merely just because of the relatively low number of training observations.

A possible limitation in our study is that our corpus included pathology reports for which the ground truth was sourced only using the final diagnosis section of the report. To better understand the impact of our case selection criterion, we performed a set of additional experiments comparing the performance of our original final-diagnosis-only trained CNNs against networks trained with an additional 144 pathology reports with ICD-O-3 labels sourced from outside the final diagnosis section (e.g., synoptic report, clinical history). The 12-class prevalence was similar to the one in the original dataset. In a series of cross-validation experiments, we observed that the reported trends remained consistent. Although several differences were noted, none of them turn out to be statistically significant. This finding suggests that the CNN approaches are fairly robust.

Although the training data used for these experiments was small relative to the total volume and variety of pathology reports used in state and national cancer surveillance, and the primary site variable is relatively well-characterized, this work was an

important first step in applying deep learning NLP methods to assist the work of cancer registries cancer. Work is underway to obtain a larger set of training and validation data from cancer registries to expand the scale and scope of this approach.

V. Conclusion

We performed a set of experiments comparing the ICD-O-3 topographical information extraction performance of a temporal convolutional network against a more traditional TF-IDF classifier approach over varying datasets of pathology reports. We observed that consistently the CNN outperformed the more traditional classifier, peaking at a micro-F score of 0.811 and a macro-F score of 0.701 for the 6-class task and achieving an overall micro-F score of 0.722 and a macro-F score of 0.389 for the 12-class task. We also tested how pre-trained word embeddings features on differing corpora can influence performance on certain subtasks. Our results suggested that for the information extraction problem with well-populated classes randomized embeddings lead to superior performance. However, when the classification task included classes with low prevalence, pre-trained embeddings achieved better performance.

Acknowledgment

This manuscript has been authored by UT-Battelle, LLC under Contract No. DE-AC05-00O R22725 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 this 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). 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-AC52-07 NA27344, Los Alamos National Laboratory under Contract DE-AC5206 NA25396, and Oak Ridge National Laboratory under Contract DE-AC05-00 OR22725

The authors wish to thank Valentina Petkov of the Surveillance Research Program from the National Cancer Institute and the SEER registries at HI, KY, CT, NM, and Seattle for the de-identified 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-AC05-00 OR22725.

References

[1] D. R. Lowy and F. S. Collins, "Aiming High—Changing the trajectory for cancer," *New England J. Med.*, vol. 374, no. 20, pp. 1901–1904, May 2016.
[2] Y. Li and D. Martinez, "Information extraction of multiple categories from pathology reports," in *Australasian Lang. Technol. Assoc. Workshop*, Melbourne, Australia, 2010, pp. 41–49.
[3] L. Penberthy *et al.*, "Cancer surveillance informatics," *Oncol. Informat.*, B. Hesse, D. Ahern, and E. Beckjord, Eds., Amsterdam, The Netherlands: Elsevier, 2016, pp. 277–285.
[4] A. Fritz *et al.*, *International Classification of Diseases for Oncology*. Geneva, Switzerland: World Health Organization, 2000, pp. 31–52.
[5] D. S. Carrell *et al.*, "Using natural language processing to improve efficiency of manual chart abstraction in research: The case of breast cancer recurrence," *Amer. J. Epidemiol.*, vol. 179, no. 6, pp. 749–758, Mar. 2014.
[6] D. Martinez and Y. Li, "Information extraction from pathology reports in a hospital setting," in *Proc. 20th ACM Int. Conf. Inf. Knowl. Manage.*, Glasgow, U. K., 2011, pp. 1877–1882.
[7] Z. Jiang *et al.*, "Training word embeddings for deep learning in biomedical text mining tasks," in *Proc. 2015 IEEE Int. Conf. Bioinformat. Biomed.*, 2015, pp. 625–628.
[8] P. Li and H. Huang, "Clinical information extraction via convolutional neural network," May 27, 2016. [Online]. Available: https://arxiv.org/abs/1603.09381
[9] G. Salton, *The SMART Retrieval System: Experiments in Automatic Document Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, 1971.
[10] P. D. Turney and P. Pantel, "From frequency to meaning: Vector space models of semantics," *J. Artif. Intell. Res.*, vol. 37, no. 1, pp. 141–188, Feb. 2010.
[11] R. Collobert *et al.*, "Natural language processing (Almost) from scratch," May 16, 2016. [Online]. Available: https://arxiv.org/abs/1103.0398
[12] K. M. Nam *et al.*, "Detection of alternative ovarian cancer biomarker via word embedding," *Int. J. Softw. Eng. Appl.*, vol. 10, no. 4, pp. 1–12, May 2016.
[13] T. Mikolov *et al.*, "Recurrent neural network based language model," presented at the International Conference on Spoken Language Processing, Makuhari, Japan, 2010.
[14] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. 31st Int. Conf. Mach. Learn.*, Beijing, China, 2014, pp. 1188–1196.
[15] Y. LeCun *et al.*, "Gradient-Based learning applied to document recognition," in *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[16] Y. Zhang and B. Wallace, "A sensitivity analysis of (and Practitioners' Guide to) convolutional neural networks for sentence classification," Jun. 2, 2016. [Online]. Available: https://arxiv.org/abs/1510.03820v4
[17] Y. Kim, "Convolutional neural networks for sentence classification," Jun. 16, 2016, [Online]. Available: https://arxiv.org/abs/1408.5882v2
[18] T. Mikolov *et al.*, "Distributed representations of words and phrases and their compositionality," in *Adv. Neural Inf. Process. Syst.*, Stateline, NV, 2013, pp. 3111–3119.
[19] R. Rehurek and P. Sojka, "Software framework for topic modelling with large corpora," presented at the Conference on Language Resources and Evaluation, Floriana, Malta, 2010.
[20] J. Piskorski and R. Yangarber, "Information Extraction: Past, present and future," *Theory and Applications of Natural Language Processing*, T. Poibeau *et al.*, Eds., Berlin, Germany: Springer, 2012, pp. 23–49.
[21] E. Ford *et al.*, "Extracting information from the text of electronic medical records to improve case detection: A systematic review," *J. Amer. Med. Inf. Assoc.*, vol. 23, no. 5, pp. 1007–1015, Feb. 2016.
[22] R. Kavuluru *et al.*, "Automatic extraction of ICD-O-3 Primary sites from cancer pathology reports," in *AMIA Joint Summits Translational Sci.*, San Francisco, CA, 2013, pp. 112–116.
[23] D. Zhang *et al.*, "Estimating the uncertainty of average F1 scores," in *Proc.2015 Int. Conf. Theory Inf. Retrieval*, New York, NY, 2015, pp. 317–320
[24] [Online]. Available: http://bio.nlplab.org/ Accessed on: Mar. 1, 2017..