Published in final edited form as:

ACM BCB. 2021 August ; 2021: . doi:10.1145/3459930.3469541.

Assigning ICD-O-3 Codes to Pathology Reports using Neural Multi-Task Training with Hierarchical Regularization

Anthony Rios,

Dept. of Information Systems & Cyber Security, Cyber Center for Security & Analytics, University of Texas at San Antonio, San Antonio, Texas, USA

Eric B. Durbin,

Division of Biomedical Informatics (Internal Medicine), Kentucky Cancer Registry, University of Kentucky, Lexington, Kentucky, USA

Isaac Hands,

Kentucky Cancer Registry, Lexington, Kentucky, USA

Ramakanth Kavuluru

Division of Biomedical Informatics (Internal Medicine), University of Kentucky, Lexington, Kentucky, USA

Abstract

Tracking population-level cancer information is essential for researchers, clinicians, policymakers, and the public. Unfortunately, much of the information is stored as unstructured data in pathology reports. Thus, to process the information, we require either automated extraction techniques or manual curation. Moreover, many of the cancer-related concepts appear infrequently in real-world training datasets. Automated extraction is difficult because of the limited data. This study introduces a novel technique that incorporates structured expert knowledge to improve histology and topography code classification models. Using pathology reports collected from the Kentucky Cancer Registry, we introduce a novel multi-task training approach with hierarchical regularization that incorporates structured information about the International Classification of Diseases for Oncology, 3rd Edition classes to improve predictive performance. Overall, we find that our method improves both micro and macro F1. For macro F1, we achieve up to a 6% absolute improvement for topography codes and up to 4% absolute improvement for histology codes.

Keywords

neural networks; natural language processing; biomedical informatics

1 INTRODUCTION

The National Cancer Institute's (NCI) mission is to conduct and support cancer research and, more importantly, to help all people live longer, healthier lives. The NCI Surveillance, Epidemiology, and End Results (SEER) Program collects and publishes population-based information, including incidence and survival data. The Centers for Disease Control and Prevention (CDC) also collects population-based cancer incidence data through the National Program of Cancer of Cancer Registries (NPCR). Much of the data collected through these programs are manually extracted from pathology reports and other clinical documents [18]. Cancer Registries use the data to develop and use systems designed to collect, store, and manage cancer patients' information to provide high quality cancer surveillance data that serves as the foundation of many types of cancer research used to plan and evaluate cancer control and prevention interventions. Currently, SEER and NPCR registries collect and report population-based cancer information for 100 percent of the U.S. population. It is essential to develop tools to scale data curation efforts for each registry. In this paper, we are studying the use of natural language processing (NLP) techniques to automatically extract ICD-O-3¹ codes from pathology reports [11].

Cancer registries primarily use ICD-O-3 codes to code cancer site (topography) and morphology (histology/behavior) based on the pathology report for histologically confirmed tumors. In general, human workers (Certified Tumor Registrars) manually extract ICD-O-3 codes. For instance, in a pathology report, if a coder was provided the sentence

Poorly differentiated hepato-cellular carcinoma of right lobe of liver,

they can code the sentence with the morphology code 8170/3 (Hepatocellular carcinoma, NOS) because the substring "*hepato-cellular carcinoma*" appears in the text. Each morphology code is composed of a histology (cell type) code indicated by the first four digits (e.g., 8170). The fifth digit of the code (e.g., /3) describes neoplasm's behavior (e.g., Benign or Malignant). Next, the topography code could be annotated as C22.0 (Liver, NOS) because of the substring "*right lobe of liver*". For this study, we focus on all topography codes and the major histology classes (e.g., 8170). Unfortunately, there are many unique topography and histology classes and many codes appear infrequently in the dataset. We term codes that occur rarely as "tail codes". Our goal is to develop techniques that can accurately extract tail codes without adversely affecting each model's performance on frequent classes.

Many automated NLP-based techniques have been developed for ICD-O-3 coding. In one of the first efforts in the field, we used ngram-based linear models to extract primary site topography codes from pathology reports [21]. Our experiments at the time were limited to codes that appeared at least 50 times in the training dataset. Thus, many codes were ignored. In Jouhet et al. (2012), they also developed linear ngram-based models but focused on French text and trained on classes that appeared at least 25 times in the training dataset [19]. We note that ICD-O-3 classification is similar to the task of coding electronic medical records with diagnosis and procedures (i.e., ICD-10 codes) [37]. There is even some ICD-10

¹International Classification of Diseases for Oncology Third Revision

oncology-related concepts. Thus, Jouhet et al. (2017) recently developed a system that integrates both ICD-O-3 and ICD-10 coding systems to improve disease classification in oncology [20].

Neural network-based methodologies have also been applied to extract ICD-O-3 codes from pathology reports [2, 13, 35]. Neural networks have produced state-of-the-art results for a wide variety of text classification tasks [22]. Gao et al. (2017) developed a hierarchical attention method to extract ICD-O-3 codes [13]. However, their work is limited to 12 topography codes and four histology grades (e.g., “well differentiated”). Similarly, Dubey et al. (2019) experiment with deep filters to extract 14 different ICD-O-3 codes [9]. They also explore grade, locality, and behavior classification, but they are still limited to 29 total classes. Recently, there have been a few large-scale studies of ICD-O-3 classification [3, 8, 40].

In this paper, we introduce a method that combines multi-task learning [27, 41], concept embeddings [38], and hierarchical regularization. The main idea of multi-task learning is that training a model on two or more similar but different tasks can improve performance. Moreover, neural network models share parameters across tasks, reducing memory usage and improving model efficiency. Multi-task learning has been used to improve the classification performance across a wide variety of tasks [2, 17, 27, 28]. Recently, Alward et al. (2020) [1] introduced a multi-task learning method for ICD-O-3 coding. Specifically, Alward et al. show that learning both topography and histology codes jointly improves both groups’ performance. However, their work combines infrequent labels into a single “other” class, thus, not directly training all classes. Furthermore, Yoon et al. (2020) [40] introduced a large-scale ICD-O-3 classification method using massive ensembles trained on a high performance computing (HPC) system. Our work focuses on improving the predictive performance of tail codes while training a large-scale ICD-O-3 classification system on standard hardware (i.e., a single GPU).

Concept embeddings have recently been studied to improve ICD-10 classification of electronic medical records [39]. Intuitively, first, vector representations are learned for each class, then a model is learned to match a report to the code embedding rather than learning a traditional classifier. In this work, rather than matching directly to the concept embeddings, which has been shown to adversely impact frequent classes [39], we introduce a novel hierarchical regularization term that makes use of the embeddings. Hierarchical information has been shown to improve both the efficiency and accuracy of large classification problems [6]. In this work, we take advantage of the implicit hierarchy in the ICD-O-3 code structure and the hierarchy in the SEER Site/Histology Validation List to improve the model’s performance on tail codes.

Toward addressing the potential societal impacts of ICD-O-3 code extraction, this paper makes the following contributions:

1. To the best of our knowledge, this is the first extensive study of machine learning techniques for large-scale ICD-O-3 code detection in pathology reports using a single neural network model on commodity hardware. Prior work has focused on smaller code sets, grouped infrequent classes into an “other” class [1], focused

on developing techniques more suited for HPC systems [40], or assumed access to large number of pathology reports (e.g., > 900k pathology reports) [40] (which is not accessible to all cancer registries).

2. We introduce a novel neural network-based hierarchical regularization technique that integrates concept embeddings and the ICD-O-3 hierarchy. This study is the first work to explore incorporating both structured and unstructured ICD-O-3 information into neural networks for automated coding of pathology reports. Furthermore, we show that the technique generalizes across two common neural architectures: Convolutional Neural Networks (CNNs) and Long Short Term Memory Networks (LSTMs), showing how our method can generalize across different neural architectures.

The rest of this paper is structured as follows: In Section 2, we discuss related biomedical text classification research and incorporating structure into neural models and representations. Section 3 discusses the pathology dataset we use for our study and Section 4 discusses the overall methodology of our approach. Finally, in Section 6 we present the major findings of our study and in Section 7 we provide an analysis of tail label performance and provide a discussion of the broad implications of our work.

2 RELATED WORK

In this section, we discuss two main areas of related work: Biomedical text classification and structured text classification methods.

2.1 Biomedical Text Classification

Neural Networks have produced state-of-the-art results across a wide range of biomedical tasks [24, 25]. Mullenbach et al. (2018) introduced a label-wise attention mechanism for medical coding [32]. In Rios and Kavuluru (2018), we introduced a matching network-based method to improve medical coding results [37]. Similar to this paper, Qiu et al. (2018) apply convolutional neural networks (CNNs) to pathology reports, showing strong predictive performance for histology and topography code classification [36]. Moreover, as previously stated, neural network-based methodologies have also been applied to extract ICD-O-3 codes from pathology reports [2, 13, 35]. Dubey et al. (2019) experiment with deep filters to extract ICD-O-3 codes [9]. They also explore grade, locality, and behavior classification, but they are still limited to 29 total classes. Alawad et al. (2019), explored the use of deep transfer learning to classify 306 topography codes [3]. Overall, they show that a model trained on pathology reports collected by a cancer registry in Kentucky can improve a model trained on data from Louisiana. Finally, Yoon et al. (2020) recently performed on of the largest ICD-O-3 classification studies at this time [40]. Specifically, Yoon et al. proposed a novel ensemble method which involved training 40,000 models. While this model performs well, it is more suited for HPC settings.

2.2 Structured Biomedical Text Classification.

Generally, there is auxiliary knowledge regarding biomedical concepts. Auxiliary information can come in the form descriptions of each class as well as relationships

between classes. For example, the Unified Medical Language System (UMLS) contains many controlled vocabularies in the biomedical sciences including, but not limited to, ICD-10-CM, SNOMED-CT, and ICD-O-3 [5]. Recently, there has been a number of papers that have explored the use of auxiliary knowledge sources to improve natural language processing tasks classification methods[7], where many papers show improvements for few-shot and zero-shot classification settings. For instance, Liu et al. (2021) [26] introduced a reinforcement learning-based strategy to reason of class hierarchies. Meng et al. (2020) [29] show that class names can be used as a weak supervision signal to improve text classification models. Specifically, they make use of the contextual embeddings to associate semantically similar words to the label names. Zhou et al. (2020) [43] created a hierarchially aware decoder for multi-label classification, showing substantial improvement over prior hierarchical classification methods. Related to the biomedical domain, in our prior work [39], we show that both label descriptions and the hierarchy can be used to improve the performance of ICD-9 code detection models in the few- and zero-shot settings. Overall, contrary to prior work, this paper differs in two major ways. First, we focus on a new domain of ICD-O-3 classification, which has not been exhaustively explored in prior work introducing methods that use auxiliary information. Second, many of the prior papers have focused on introducing highly specialized neural architectures. In this work, we introduce a method that generalizes across multiple architectures. So, it could potentially be combined with many of the prior approaches.

Overall, our technique is most similar to recent word embedding [30] retrofitting methods which was introduced by Faruqi et al. (2015) [10]. Retrofitting incorporates structured information into word embeddings from valuable information contained in semantic lexicons such as WordNet [31], FrameNet [4], and the Paraphrase Database [12]. Retrofitting has been explored adapted to other domains. For instance, Hovy and Purschke (2018) [16] use retrofitting-based methodology to incorporate geospatial information into document embeddings. Our approach is most similar to Yu et al. (2016) [42]. Specifically, Yu et al. (2016) use retrofitting to improve semantic similarity among medical subject heading word vector representations by incorporating structured information from UMLS. Rather than focusing on semantic similarity between medical concepts, we expand on the use of retrofitting by including a retrofitting-like regularization term while training neural networks for ICD-O-3 classification.

3 DATASET

For this study, we use 82,106 pathology reports collected from the Kentucky Cancer Registry. The dataset contains a total of 298 topography codes and 493 histology codes. The reports are split into a training/validation dataset and a test set, with 65,705 and 16,401 examples in each set, respectively. We list some of the basic statistics in Table 1. Overall, the dataset has two main unique characteristics. First, each report contains 1.4 thousand tokens (words) on average. Second, the dataset contains a combined total of 770 and 617 unique codes (including topography and histology codes) in the training and test datasets, respectively. In Figure 1, we plot the Log-Frequency distribution of the topography and histology codes. We see a large imbalance for both code sets where some of the codes

appear in more than 14,000 pathology reports, while other codes appear in less than ten reports.

4 METHOD

In this section we provide brief explanations of the structured data sources we use for our hierarchical regularization loss, the base neural network architectures we use in our experiments, the actual hierarchical regularization technique, and provide training details for our experiments.

4.1 Method overview.

In Figure 2, we provide an overview of our model. While the figure shows a Convolutional Neural Network (CNN) as the primary technique used to represent each report, we note that the CNN component is interchangeable with any neural network-based model that generates a fixed size vector representation of the text. Thus we also experiment with a bi-directional long short term memory (LSTM) network. Overall, our method can be broken down into three major components. First, a neural network, CNN or LSTM, represents the text of each pathology report. Second, vector representations of the ICD-O-3 topography and histology names are created. Third, given the ICD-O-3 vector representation and the CNN/LSTM pathology report vector, we train our model using a novel regularization loss that makes use of the ICD-O-3 hierarchy.

The remainder of this section is organized as follows: First, we introduce the hierarchical information we use in our model. Second, we summarize the CNN and LSTM neural network architecture we use in our experiments. Third, we discuss the classification loss and multi-task learning procedure. Finally, we discuss our proposed hierarchical regularization loss.

4.2 ICD-O-3 Hierarchy using Code Structure and Validation Lists

In our experiments—for our hierarchical regularization technique—we empirically test two types of structured knowledge: The “SEER Site/Histology Validation List”² and basic ICD-O-3 hierarchical code structure. Moreover, we make use of unstructured knowledge in the form of ICD-O-3 code descriptions. Our regularization approach uses both the hierarchical information and the code descriptions. The “Site/Histology Validation List” records all of the “*valid*” topography and histology code combinations. Combinations that occur rarely or never are not included. Note that topography-histology combinations that do not appear in the validation list are not necessarily “invalid”, but they generally require review by the physician before unknown combinations are used. In fact, in our dataset, a substantial percentage of the code combinations were “*invalid*” (as per SEER validation lists) even though they are deemed appropriate as per certified tumor registrars; thus simply pruning options during inference based on SEER validation lists results in sub-par performance. We provide an example of the SEER Site/Histology Validation List hierarchy (*Hier. Valid.*) below:

² <https://seer.cancer.gov/icd-o-3/>

- LUNG & BRONCHUS C34.0-C34.3,C34.8-C34.9
 - SQUAMOUS CELL CARCINOMA, NOS 807
 - ◆ 8075/3 Squamous cell carcinoma, adenoid
 - ◆ 8076/2 Sq. cell carc. in situ with question. stromal invas.
 - ◆ 8076/3 Sq. cell carcinoma, micro-invasive
 - ◆ 8078/3 Squamous cell carcinoma with horn formation
 - LYMPHOEPITHELIAL CARCINOMA 808
 - ◆ 8083/3 Basaloid squamous cell carcinoma

The list starts with a collection of topography codes (e.g., C34.0-C34.3 and C34.8-C34.9), followed by a generic histology group code (e.g., 807), then, a list of specific histology codes (e.g., 8075, 8076, and 8078). We define links between all top-level topography codes (e.g., C34.0 $\leftrightarrow$ C34.1), between the generic histology code and all top-level topography codes (e.g., C34.0 $\leftrightarrow$ 807), and between the generic histology code and all specific histology codes (e.g., 807 $\leftrightarrow$ 8075).

For the base hierarchy (*Base Hier.*), there is an implicit relation between the topography codes (e.g., C34.0) using the whole and fractional digit structure. Specifically, for each topography group, we create a new node in the tree (e.g., C34) that relates all codes sharing the same prefix (e.g., C34 $\leftrightarrow$ C34.0, C34 $\leftrightarrow$ C34.1, and C34 $\leftrightarrow$ C34.2). We also make use of a basic hierarchical structure over the histology codes. Particularly, we create a new node in the tree using the first three digits of the histology code (e.g., 807) that links to each child code (e.g., 807 $\leftrightarrow$ 8075, 807 $\leftrightarrow$ 8076, and 807 $\leftrightarrow$ 8078).

Again, we also use the textual descriptions along with the hierarchies when generating concept embeddings. For example, the topography code C34.1 has the description “Upper lobe, lung”. Likewise, the parent of C34.1 would be C34 with the corresponding description “LUNG & BRONCHUS”. We use similar descriptors for histology codes. For instance, the descriptor for code 807 is “SQUAMOUS CELL CARCINOMA, NOS”, and its child’s descriptor (code 8075) is “Squamous cell carcinoma, adenoid”. Because unique descriptions are not available for all codes (e.g., for all mid-level histology codes), we also include the unique code itself as part of the description (e.g., “807 SQUAMOUS CELL CARCINOMA, NOS”).

4.3 Convolutional neural networks for text classification.

For the CNN component, we use a standard model from Kim (2014) [22]. Word vectors form the base element of the model. Given a pathology report, let $\mathbf{w}_i^j \in \mathbb{R}^d$ represent the j -th word’s vector in the i -th document and d is the size of the word vectors. As shown in Figure 2, each pathology report is represented as a matrix $\mathbf{X}_i = [\mathbf{w}_i^1; \dots; \mathbf{w}_i^{N_i}]$ by concatenating the word vectors, where N_i is the number of words in the i -th document and $\mathbf{X}_i \in \mathbb{R}^{N_i \times d}$. Given the document matrix $\mathbf{X}_i$, the CNN produces a fixed size feature representation

$$\mathbf{c} = CNN(\mathbf{X}_i)$$

where $\mathbf{c} \in \mathbb{R}^{f \cdot s}$, where s is the number of convolution filter sizes and f is the number of filters per size.

4.4 Recurrent neural networks for text classification.

While CNNs only extract informative n-grams from text, RNNs are expected to capture long term dependencies between words. For our RNN method, we use LSTMs [14], specifically a variant introduced by Graves (2012) [15]. With this setup, we first use a BiLSTM layer at the word level to capture the contextual information of the sentence with respect to each word. Concretely,

$$\begin{aligned} \mathbf{h}_i &= \text{LSTM}^{\rightarrow}(\mathbf{w}_i), \\ \overleftarrow{\mathbf{h}}_i &= \text{LSTM}^{\leftarrow}(\mathbf{w}_i), \text{ and} \\ \mathbf{h}_i &= \overrightarrow{\mathbf{h}}_i \parallel \overleftarrow{\mathbf{h}}_i \text{ for } i = 1, \dots, N \end{aligned}$$

where $\overleftarrow{\mathbf{h}}_i \in \mathbb{R}^d$ and $\mathbf{h}_i \in \mathbb{R}^d$ such that d (a network hyperparameter) is the number of output units at each LSTM time step. Also, Word-LSTM is functionally identical to a traditional LSTM cell (without peepholes) and the prefix “Word” is used only to indicate that its input is based on word vectors. Next, we perform a max-pooling operation over all $\mathbf{h}_i$ vectors to produce the feature vector $\mathbf{c} = [h_{\max}^1, \dots, h_{\max}^d]$ where $h_{\max}^i = \max(h_1^i, \dots, h_N^i)$ represents the maximum value across all N BiLSTM word representations for the dimension i .

4.5 Classification loss.

Finally, we use a fully-connected output layer with m outputs, where m corresponds to the number of classes. The output is computed as

$$\mathbf{q} = \mathbf{W}_q \cdot \mathbf{c} + \mathbf{b}_q$$

where $\mathbf{q} \in \mathbb{R}^m$, $\mathbf{W}_q \in \mathbb{R}^{m \times z}$, and $\mathbf{b}_q \in \mathbb{R}^m$ are additional network parameters. z is the size of the mid-level neural network representation $\mathbf{c}$, where $z = f \cdot s$ for the CNN and $z = d$ for the Bi-LSTM model. In order to get a categorical distribution, we apply the softmax function to the vector $\mathbf{q}$ to obtain

$$C(\mathbf{X}_i) = p_j = \frac{e^{\mathbf{q}_j}}{\sum_{i=1}^m e^{\mathbf{q}_i}}, \quad (1)$$

where $p_j \in [0, 1]$ is the probability estimate of the label at index j and where $C(\mathbf{X}_i) \in \mathbb{R}^m$ is the vector of all p_j s. Next, we can train the classifier $C()$, and the CNN/LSTM $F()$, using a multi-class cross-entropy loss

$$CE(\mathcal{X}, \mathcal{Y}; \Theta) = -\mathbb{E}_{(i, y_i) \in T} \left[\sum_{j=1}^m y_{i,j} \log(C(\mathbf{X}_i)_j) \right]$$

over training dataset T , training examples $\mathcal{X}$, labels $\mathcal{Y}$, and parameters Θ .

In the multi-task classification scenario, we have two output layers: $C(\mathbf{X}_i)_{top}$ and $C(\mathbf{X}_i)_{hist}$ for topography and histology, respectively. Likewise, to train the multi-task model, two versions of the cross-entropy loss are combined into a single loss function as

$$L_{MT} = CE_{top} + CE_{hist} \quad (2)$$

where CE_{top} is the loss for the topography codes and CE_{hist} is the loss function for the histology codes. Essentially, this is similar to the hard-parameter sharing multi-task model described in Alawad et al. (2020) [1]. We refer to the multi-task variants of the CNN and LSTM as MTCNN and MTLSTM, respectively.

4.6 Hierarchical Loss.

A high-level diagram of the information our hierarchical loss incorporates is shown in Figure 3. We make use of two forms of information in the loss function: label descriptions and code hierarchy. Examples of the code descriptions and the hierarchies are shown in Section 4.2. Intuitively, our goal is to incorporate both the label descriptions and hierarchical knowledge into the classification layer. Moreover, we want to do this without impacting the performance of frequent classes. Thus, the loss function is made of two major components—see Part 1 and Part 2 in Figure 3. Part 1 will take the mid-level neural network features $\mathbf{c}$ and try to match it with *relevant* label embeddings, where the label embeddings are created using the label descriptions. We define the relevant labels as the union of the gold-standard labels (histology and topography codes) assigned to a given instance and all labels connected to the gold standard codes in the chosen hierarchy. Next, for Part 2, we integrate the label embeddings for each code and each code linked to it in the chosen hierarchy with the output layer defined in Equation 1. To integrate the supplementary knowledge with the output layer, we embed them into the same “output embedding space.”

For Part 1, we generate embeddings for each code by averaging word embeddings of the site/histology descriptions. For example, the site code C34.1 has the description “Upper lobe, lung.” We also add each code as a unique token to each description, such that the description for C34 would “C34 LUNG & BRONCHUS”. Likewise, the histology code 800 with the description “800 NEOPLASM”. Adding the codes to the descriptions helps the model learn unique aspects between the codes that may not be easily captured when multiple codes have similar descriptions. The averaged embeddings are then defined as

$$\mathbf{h}_i = \mathbf{e}_i \mathbf{W}$$

where $\mathbf{e}_i$ is the averaged word embeddings for one of the classes linked to the i -th code (e.g., C34). $\mathbf{W} \in \mathbb{R}^{b \times d}$ is a transformation matrix that matches the dimensions of the size of

the *CNN/LSTM* output. It is important to note that there is an embedding $\mathbf{e}_i$ for every class in the hierarchy we use, including leaf and non-leaf nodes in the hierarchy (e.g., C34 and C34.1) Next, we learn to match the neural features $\mathbf{c}$ to the *relevant* class embeddings

$$\hat{\mathbf{y}} = \text{sigmoid}(\mathbf{c}\mathbf{H} + \mathbf{b}_h) \quad (3)$$

where $\mathbf{H} \in \mathbb{R}^{d \times K}$ is a matrix of the code description embeddings, K is the number of embeddings (this includes the mid-tier classes in the hierarchy such as C34), and $\mathbf{b}_h$ are biases for each of the hierarchy classes. We define relevant embeddings as the gold-standard histology and topography codes for that example and all codes connected to the gold-standard codes in the chosen hierarchy. Intuitively, Equation 3 represents the probability of each class, similar to Equation 1. However, in this equation the label embeddings generated from the descriptions are being used in-place of the instead output matrix in a full-connected neural network layer. Furthermore, the a sigmoid squashing function is used instead of the softmax because we want to match more than one related class. Next, let given the scores from Equation 3, we train the model using a binary cross-entropy loss

$$CE_2(\mathcal{X}, \mathcal{Y}; \Theta) = -\mathbb{E}_{(i, y_i) \in T} \left[\sum_{j=1}^m y_{i,j} \log(C(\mathbf{X}_i)_j) + (1 - y_{i,j}) \log(1 - C(\mathbf{X}_i)_j) \right]$$

over training dataset T , training examples $\mathcal{X}$, labels $\mathcal{Y}$, and parameters Θ .

Finally, we directly connect Equations 2 and 3 using a regularization term

$$R = \sum_{i=1}^m \sum_{n \in \mathcal{N}_i \cup i} \|W_q^i - \mathbf{h}_n\|_2^2$$

where W_q^i represent the i -th row in $\mathbf{W}_q$ representing the i -th class. Similarly, let $\mathbf{h}_n$ represent the n -th column of H , where n returns the index of one of the neighbors of the i -th class in $\mathbf{H}$, and $\mathcal{N}_i$ is the set of all classes linked to code i . Also, the class i is included in the neighborhood of i via a self edge. This formulation shares similarities with prior work on embedding retrofitting [10]. The distance is minimized between each of the parameters of the output to match the parent embeddings with their children. Intuitively, we are telling the model that the parameters for each site/histology code should be similar to its self and parent.

We define the final loss as

$$L = L_{MT} + CE_2 + \lambda R$$

where L_{MT} represents the multi-task cross-entropy loss from Equation 2. CE_2 represents the binary cross-entropy between the predictions in Equation 3 and ground-truth linked codes in each report based on one of the hierarchies. λ is a hyperparameter to enforce the similarity strength between L_{MT} and CE_2 . At inference time, predictions are made using predictions

from Equation 1. The parent predictions are ignored. By weighting R we can control how much relevant information is transferred about the hierarchy and label descriptions to the standard output layer.

Why are Equations 3 and CE_2 needed in as part regularization function? When we regularize by simply forcing the output parameters $\mathbf{W}_q$ to be similar to the label embeddings (the averaged word embeddings for words appearing in the description), our preliminary analysis shown that this adversely effected the performance. The word embeddings used to generate the word embeddings are initialized using pre-trained embeddings. Thus, they have no knowledge about how the neural network features $\mathbf{c}$ relate to them. Thus, if we force the label embeddings to be similar to $\mathbf{W}_q$, the output layer parameters will change substantially in a direction that will reduce the performance of the model because we would be telling the model they should be close in the word embedding space. This intuition is shown in Figure 3 where the descriptor embeddings and the output layer embeddings are in different embeddings spaces. Therefore, we learn to relate the embeddings with $\mathbf{c}$ (or to map the label embeddings into the “output layer space”) by learning to map them with the loss function CE_2 and secondary output layer described in Equation 3.

4.7 Training Details.

We train both the CNN and LSTM models using the Adam optimizer [23]. Also, both models use 300 dimensional pre-trained 42B GloVe [34] embeddings³. For the CNN, we train the model with filters that span three, four, and five words. For each filter size, we train 300 filters. For the LSTM model, we set the hidden state size to 512. The hierarchical regularization weight λ was chosen via a simple grid-search procedure on the validation dataset across the values $\{1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 1\}$.

5 EVALUATION METRICS AND BASELINES

In our experiments, we compare the performance of our proposed method to two groups of models listed below:

5.1 Linear Models

We train three traditional linear text classification methods: Naive Bayes, Logistic Regression (LR), and Linear Support Vector Machines (SVM). All methods are trained with the scikit-learn package [33]. For the LR and SVM methods, we grid-search over C values $\{0.0001, 0.001, 0.01, 0.1, 1., 10, 100\}$ and both the L1 and L2 regularization losses. We also grid-search of the NB alpha parameter values $\{0.0001, 0.001, 0.01, 0.1, 1., 10, 100\}$.

5.2 Neural Networks Baselines

As baselines, we experiment with the standard CNN and LSTM models without multi-task learning and without hierarchical regularization. For the CNN model, we use the architecture proposed by Kim (2014) [22] and the LSTM uses the bidirectional variant of the architecture proposed by Graves (2012) [15]. We experiment without both single-task (CNN and LSTM)

³ <http://nlp.stanford.edu/data/glove.42B.300d.zip>

and multi-task variants (MTCNN and MTLSTM) of the models. Given the size of the pathology reports, BERT-based models were not evaluated (i.e., BERT allows a maximum of 512 tokens, while the average pathology report has 1.4 thousand tokens).

5.3 Evaluation Metrics.

We use the macro and micro *F1* evaluation metrics in this paper. Macro *F1* score is defined as

$$P_i = \frac{TP_i}{TP_i + FP_i}, R_i = \frac{TP_i}{TP_i + FN_i}, F1_i = \frac{2 \cdot P_i \cdot R_i}{P_i + R_i} \text{ and Macro-F1} = \frac{1}{C} \sum_{i=1}^C F1_i$$

where TP_i , FP_i , and FN_i are true positives, false positives, and false negatives for the i -th class, respectively. C is the total number of classes. P_i , R_i , and $F1_i$ represent the precision, recall and F1 score of the i -th class, respectively. Intuitively, the F1 score is calculated for each class independently, then each of the scores are averaged to generate the Macro-F1 measure. Macro-F1 treats each class with equal weight. Therefore, tail (infrequent) classes are treated equally to frequent classes. On the other hand, the Micro-F1 measure gives more weight to the frequent classes. The Micro-F1 measure is defined as a harmonic mean between the Micro-Precision and Micro-Recall metrics, defined as

$$\text{Micro-P} = \frac{\sum_{i=1}^C TP_i}{\sum_{i=1}^C TP_i + FP_i}, \text{Micro-R} = \frac{\sum_{i=1}^C TP_i}{\sum_{i=1}^C TP_i + FN_i}, \text{and Micro-F1} = \frac{2 \cdot \text{Micro-P} \cdot \text{Micro-R}}{\text{Micro-P} + \text{Micro-R}}$$

where Micro-P and Micro-R represent Micro-Precision and Micro-Recall scores, respectively. In summary, our goal is to use Macro F1 to measure how well our model performs across all classes irrespective of the class frequency. Likewise, we use the Micro F1 measure to evaluate how well our proposed method performs on frequent classes.

6 RESULTS

The main results of our experiments are shown in Table 2. Overall, we make three major findings. First, we find that incorporating hierarchical regularization improves both the LSTM and CNN models. The MTLSTM model achieves a Micro F1 of 68.32 and an Macro F1 of 17.41. Incorporating the hierarchical F1 score increases the Topography Macro F1 by nearly 7%. Similarly, the MTLSTM's Macro F1 for the Histology classes increases by nearly 4% from 23.82 to 27.48 with the addition of the hierarchical loss. Likewise, the MTCNN model's Micro F1 improves from 78.43 to 78.67—only an improvement of 0.24—whereas the Macro F1 improves by more than 4% from 30.85 to 35.14. On the contrary, the multi-task framework does not have as much of an impact on the CNN and LSTM models as the hierarchical regularization loss, with a Macro F1 improvement of 1% or less for both Topography and Histology codes caused by using multi-task learning. Second, the MTCNN model with hierarchical regularization outperforms all other tested solutions with regard to both Micro and Macro-F1. This result is expected since the CNN model outperforms all other standalone methods without multi-task learning or hierarchical regularization (i.e., the

LSTM and linear models). Our results indicate that the hierarchical regularization method is applicable even if the model already performs well. Third, we do find that the linear models are still strong baselines. For instance, the SVM is the best performing linear model achieving a Micro F1 of 74.93 and a Macro F1 of 28.86, outperforming the Naïve Bayes, Logistic Regression, and LSTM models. Finally, we find that the base hierarchy (Base Hier.) outperforms the SEER validation lists (Hier. Valid.)

A careful ablation study of our method is shown in Table 3. We look at the impact of both the multi-task learning and hierarchical regularization component of our model. Overall, for both the LSTM and CNN models, we see that the hierarchical regularization component has the largest impact on the Macro F1 score. For instance, with the CNN model, removing the hierarchical regularization component reduces the Micro F1 by only 0.24 for Topography classes. But, the Macro F1 reduces by three points. Likewise, for the LSTM model, we obtain a six point reduction in Macro F1 for topography codes, while the micro F1 reduces by less than one point. We see similar results for the histology codes, i.e., the improvement is largely in the Macro F1 scores.

7 DISCUSSION

In Section 6, Macro F1 is used to measure the improvement of tail codes. However, is Macro F1 really measuring tail code performance or are improvements in (semi-)frequent ICD-O-3 codes causing the majority of the score increases? We report the macro scores for classes appearing no more than ten times in the training data in Table 4. There were 262 histology and 94 topography codes that appeared no more than ten times, respectively. Overall, for topography, we find that the Macro F1 on the infrequent classes improves by more than 7% for the CNN and LSTM models when hierarchical regularization is used. Moreover, compared the Macro F1 scores for all codes in Table 2, we find that the improvement is slightly better on the infrequent class Macro F1 than the overall macro F1 after including our regularization method. This result indicates the overall improvement is not simply occurring for frequent or even semi-frequent classes, but also improvements in performance occur for classes with very small numbers of training examples. We make similar findings for Histology codes.

8 CONCLUSION

In this paper, we introduced a novel method that improves the predictive performance of tail ICD-O-3 codes that generalizes across different neural network architectures. Moreover, our method shows the usefulness of incorporating expert domain knowledge into neural networks in the biomedical domain. Specifically, structured knowledge improves the predictive performance of frequent and tail codes. By improving the predictive performance of tail codes on commodity hardware, this should lead to more confident real-time surveillance of cancer cases by cancer registries. Moreover, by focusing on relatively simple base models that can be trained on a single GPU, this model can more easily be used by registries with limited resources and compute power.

Overall, there are two major avenues for future work:

1. First, there are other sources of domain knowledge that we have not yet explored. For instance, pretraining the model on large sources of data (e.g., PubMed) can help add additional information to the model.
2. Second, our model does not make predictions for classes that never appear in the training data. Thus, incorporating recent advances in zero-shot learning learning using structured knowledge can help improve ICD-O-3 classification performance [38].

ACKNOWLEDGMENTS

This research was supported by the Shared Resource Facilities of the University of Kentucky Markey Cancer Center (P30CA177558). Kavuluru's effort was also supported by the U.S. National Library of Medicine under award number R01LM013240.

REFERENCES

- [1]. Alawad Mohammed, Gao Shang, Qiu John X, Yoon Hong Jun, Christian J Blair, Penberthy Lynne, Mumphy Brent, Wu Xiao-Cheng, Coyle Linda, and Tourassi Georgia. 2020. Automatic extraction of cancer registry reportable information from free-text pathology reports using multitask convolutional neural networks. *Journal of the American Medical Informatics Association* 27, 1 (2020), 89–98. [PubMed: 31710668]
- [2]. Alawad Mohammed, Yoon Hong-Jun, and Tourassi Georgia D. 2018. Coarse-to-fine multi-task training of convolutional neural networks for automated information extraction from cancer pathology reports. In *2018 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*. IEEE, 218–221.
- [3]. Alawad Mohammed M, Gao Shang, Qiu John X, Schaefferkoetter Noah T, Hinkle Jacob, Yoon Hong-Jun, Christian Blair, Wu Xiao-Cheng, Durbin Eric B, Jeong Jong Cheol, et al. 2019. Deep Transfer Learning Across Cancer Registries for Information Extraction from Pathology Reports. Technical Report. Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States).
- [4]. Baker Collin F, Fillmore Charles J, and Lowe John B. 1998. The Berkeley framenet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, Volume 1. 86–90.
- [5]. Bodenreider Olivier. 2004. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic acids research* 32, suppl_1 (2004), D267–D270. [PubMed: 14681409]
- [6]. Cesa-Bianchi Nicolò, Gentile Claudio, and Zaniboni Luca. 2006. Hierarchical classification: combining bayes with svm. In *Proceedings of the 23rd international conference on Machine learning*. ACM, 177–184.
- [7]. Chen Jiaoyan, Geng Yuxia, Chen Zhuo, Horrocks Ian, Pan Jeff Z, and Chen Huajun. 2021. Knowledge-aware Zero-Shot Learning: Survey and Perspective. *arXiv preprint arXiv:2103.00070* (2021).
- [8]. De Angeli Kevin, Gao Shang, Alawad Mohammed, Yoon Hong-Jun, Schaefferkoetter Noah, Wu Xiao-Cheng, Durbin Eric B, Doherty Jennifer, Stroup Antoinette, Coyle Linda, et al. 2021. Deep active learning for classifying cancer pathology reports. *BMC bioinformatics* 22, 1 (2021), 1–25. [PubMed: 33388027]
- [9]. Dubey Abhishek K, Hinkle Jacob, Christian J Blair, and Tourassi Georgia. 2019. Extraction of Tumor Site from Cancer Pathology Reports using Deep Filters. In *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*. ACM, 320–327.
- [10]. Faruqi Manaal, Dodge Jesse, Jauhar Sujay Kumar, Dyer Chris, Hovy Eduard, and Smith Noah A. 2015. Retrofitting Word Vectors to Semantic Lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 1606–1615.

- [11]. Jack Shanmugaratnam Sobin Parkin Whelan Fritzt, Percy. 2001. International Classification of Diseases for Oncology; Third Edition. (2001).
- [12]. Ganitkevitch Juri, Van Durme Benjamin, and Callison-Burch Chris. 2013. PPDB: The paraphrase database. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 758–764.
- [13]. Gao Shang, Young Michael T, Qiu John X, Yoon Hong-Jun, Christian James B, Fearn Paul A, Tourassi Georgia D, and Ramanathan Arvind. 2017. Hierarchical attention networks for information extraction from cancer pathology reports. *Journal of the American Medical Informatics Association* 25, 3 (2017), 321–330.
- [14]. Gers Felix A, Schmidhuber Jürgen, and Cummins Fred. 2000. Learning to forget: Continual prediction with LSTM. *Neural computation* 12, 10 (2000), 2451–2471. [PubMed: 11032042]
- [15]. Graves Alex. 2012. Supervised sequence labelling with recurrent neural networks. Vol. 385. Springer.
- [16]. Hovy Dirk and Purschke Christoph. 2018. Capturing regional variation with distributed place representations and geographic retrofitting. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. 4383–4394.
- [17]. Howard Jeremy and Ruder Sebastian. 2018. Universal Language Model Fine-tuning for Text Classification. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 328–339.
- [18]. National Cancer Institute. 2017. Overview of the SEER Program. <https://seer.cancer.gov/about/overview.html>.
- [19]. Jouhet Vianney, Defossez Georges, Burgun Anita, Beux Pierre Le, Levillain P, Ingrand Pierre, and Claveau Vincent. 2012. Automated classification of free-text pathology reports for registration of incident cases of cancer. *Methods of information in medicine* 51, 03 (2012), 242–251. [PubMed: 21792466]
- [20]. Jouhet Vianney, Mougin Fleur, Bérénice Bréchat, and Frantz Thiessard. 2017. Building a model for disease classification integration in oncology, an approach based on the national cancer institute thesaurus. *Journal of biomedical semantics* 8, 1 (2017), 6. [PubMed: 28173841]
- [21]. Kavuluru Ramakanth, Hands Isaac, Durbin Eric B, and Witt Lisa. 2013. Automatic extraction of ICD-O-3 primary sites from cancer pathology reports. *AMIA Summits on Translational Science Proceedings* 2013 (2013), 112.
- [22]. Kim Yoon. 2014. Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Doha, Qatar, 1746–1751. <http://www.aclweb.org/anthology/D14-1181>
- [23]. Kingma Diederik P. and Ba Jimmy. 2014. Adam: A Method for Stochastic Optimization. *CoRR abs/1412.6980* (2014). <http://arxiv.org/abs/1412.6980>
- [24]. Lan Kun, Wang Dan-tong, Fong Simon, Liu Lian-sheng, Wong Kelvin KL, and Dey Nilanjan. 2018. A survey of data mining and deep learning in bioinformatics. *Journal of medical systems* 42, 8 (2018), 139. [PubMed: 29956014]
- [25]. Li Yu, Huang Chao, Ding Lizhong, Li Zhongxiao, Pan Yijie, and Gao Xin. 2019. Deep learning in bioinformatics: introduction, application, and perspective in big data era. *arXiv preprint arXiv:1903.00342* (2019).
- [26]. Liu Hui, Zhang Danqing, Yin Bing, and Zhu Xiaodan. 2021. Improving Pretrained Models for Zero-shot Multi-label Text Classification through Reinforced Label Hierarchy Reasoning. *arXiv:2104.01666 [cs.CL]*
- [27]. Liu Pengfei, Qiu Xipeng, and Huang Xuanjing. 2016. Recurrent neural network for text classification with multi-task learning. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence. AAAI Press, 2873–2879.
- [28]. Liu Pengfei, Qiu Xipeng, and Huang Xuanjing. 2017. Adversarial Multi-task Learning for Text Classification. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 1–10.

- [29]. Meng Yu, Zhang Yunyi, Huang Jiaxin, Xiong Chenyan, Ji Heng, Zhang Chao, and Han Jiawei. 2020. Weakly-Supervised Text Classification Using Label Names Only. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). 9006–9017.
- [30]. Mikolov Tomas, Sutskever Ilya, Chen Kai, Corrado Greg S, and Dean Jeff. 2013. Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems. 3111–3119.
- [31]. Miller George A, Beckwith Richard, Fellbaum Christiane, Gross Derek, and Miller Katherine J. 1990. Introduction to WordNet: An on-line lexical database. International journal of lexicography 3, 4 (1990), 235–244.
- [32]. Mullenbach James, Wiegrefe Sarah, Duke Jon, Sun Jimeng, and Eisenstein Jacob. 2018. Explainable Prediction of Medical Codes from Clinical Text. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Vol. 1. 1101–1111.
- [33]. Pedregosa Fabian, Varoquaux Gaël, Gramfort Alexandre, Michel Vincent, Thirion Bertrand, Grisel Olivier, Blondel Mathieu, Prettenhofer Peter, Weiss Ron, Dubourg Vincent, et al. 2011. Scikit-learn: Machine learning in Python. the Journal of machine Learning research 12 (2011), 2825–2830.
- [34]. Pennington Jeffrey, Socher Richard, and Manning Christopher. 2014. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 1532–1543.
- [35]. Qiu John X, Yoon Hong-Jun, Fearn Paul A, and Tourassi Georgia D. 2017. Deep learning for automated extraction of primary sites from cancer pathology reports. IEEE journal of biomedical and health informatics 22, 1 (2017), 244–251. [PubMed: 28475069]
- [36]. Qiu John X, Yoon Hong-Jun, Fearn Paul A, and Tourassi Georgia D. 2018. Deep learning for automated extraction of primary sites from cancer pathology reports. IEEE journal of biomedical and health informatics 22, 1 (2018), 244–251. [PubMed: 28475069]
- [37]. Rios Anthony and Kavuluru Ramakanth. 2018. EMR Coding with Semi-Parametric Multi-Head Matching Networks. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Vol. 1. 2081–2091.
- [38]. Rios Anthony and Kavuluru Ramakanth. 2018. Few-Shot and Zero-Shot Multi-Label Learning for Structured Label Spaces. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '18). Association for Computational Linguistics.
- [39]. Rios Anthony and Kavuluru Ramakanth. 2018. Few-shot and zero-shot multi-label learning for structured label spaces. In Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing, Vol. 2018. NIH Public Access, 3132.
- [40]. Yoon Hong-Jun, Klasky Hilda B, Gounley John P, Alawad Mohammed, Gao Shang, Durbin Eric B, Wu Xiao-Cheng, Stroup Antoinette, Doherty Jennifer, Coyle Linda, et al. 2020. Accelerated training of bootstrap aggregation-based deep information extraction systems from cancer pathology reports. Journal of Biomedical Informatics 110 (2020), 103564. [PubMed: 32919043]
- [41]. Yoon Hong-Jun, Ramanathan Arvind, and Tourassi Georgia. 2016. Multi-task deep neural networks for automated extraction of primary site and laterality information from cancer pathology reports. In INNS Conference on Big Data. Springer, 195–204.
- [42]. Yu Zhiguo, Cohn Trevor, Wallace Byron C, Bernstam Elmer, and Johnson Todd. 2016. Retrofitting word vectors of mesh terms to improve semantic similarity measures. In Proceedings of the seventh international workshop on health text mining and information analysis. 43–51.
- [43]. Zhou Jie, Ma Chunping, Long Dingkun, Xu Guangwei, Ding Ning, Zhang Haoyu, Xie Pengjun, and Liu Gongshen. 2020. Hierarchy-Aware Global Model for Hierarchical Text Classification. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 1106–1117.

CCS CONCEPTS

• **Applied computing** → **Health informatics**; • **Computing methodologies** → **Neural networks**; *Regularization*.

(a) Topography Code Distribution

(b) Histology Code Distribution

Figure 1:

The Log-frequency distributions in the training dataset for both the Topography (Figure 1a) and Histology codes (Figure 1b). The long tail of both distributions is indicative of large imbalances in the number of training examples for each code set.

Figure 2:
High-level diagram of our method.

Figure 3:
Intuitive representation of the hierarchical regularization method.

Table 1:

Summary statistics of the pathology report dataset, including the total number of pathology reports (# Path. Reports), average/median number of tokens/words per report (Avg. Tokens and Med. Tokens), and the number of unique topography/Histology codes (Unique Topo./Hist. Codes).

	# Path. Reports	Avg. Tokens	Med. Tokens	Unique Topo. Codes	Unique Hist. Codes
Train	65,705	1,463	869	296	478
Test	16,401	1,472	863	256	361

Table 2:

Overall Micro and Macro F1 results for Topography and Histology codes. The best scores for each neural network base model (LSTM and CNN) are bolded.

	Topography		Histology	
	Micro F1	Macro F1	Micro F1	Macro F1
Traditional Machine Learning Methods				
Naive Bayes	51.81	18.18	54.72	06.67
Logistic Regression	73.02	26.01	63.11	26.28
SVM	74.93	26.25	63.54	28.86
Neural Network Methods				
LSTM	68.21	16.28	58.19	22.37
MTLSTM	68.32	17.41	58.71	23.82
MTLSTM + Base Hier. Regularization (Ours)	69.13	23.94	58.29	27.48
ENS. LSTM + Base Hier. Regularization (Ours)	69.64	24.23	58.23	27.35
MTLSTM + Hier. Valid. Regularization (Ours)	69.01	23.64	58.31	27.18
ENS. LSTM + Hier. Valid. Regularization (Ours)	69.64	24.13	58.47	27.21
CNN	78.39	29.12	64.77	30.11
MTCNN	78.43	30.85	65.40	30.93
MTCNN + Base Hier. Regularization (Ours)	78.67	35.14	64.58	34.13
ENS. CNN + Base Hier. Regularization (Ours)	79.03	36.11	65.39	34.67
MTCNN + Hier. Valid. Regularization (Ours)	78.44	32.37	65.02	33.95
ENS. CNN + Hier. Valid. Regularization (Ours)	78.95	33.89	65.68	34.14

Table 3:

Ablation of the Multi-task CNN and LSTM models using Base Hier. Regularization.

	Topography		Histology	
	Micro F1	Macro F1	Micro F1	Macro F1
Multi-task CNN + Base. Hier. Regularization	78.67	35.14	64.58	34.13
- without Multi-task	78.17	32.34	64.41	31.56
- without Base. Hier. Regularization	78.43	30.85	65.40	30.93
Multi-task LSTM + Base. Hier. Regularization	69.13	23.94	58.29	27.48
- without Multi-task	68.85	20.96	58.28	24.33
- without Base. Hier. Regularization	68.32	17.41	58.71	23.82

Table 4:

Macro F1 for Topography and Histology codes appearing no more than ten times in the training dataset.

	Topography Macro F1	Histology Macro F1
MTCNN	08.32	06.85
MTCNN + Base. Hier. Regularization	16.09	11.26
MTLSTM	07.61	06.12
MTLSTM + Base. Hier. Regularization	15.17	11.18