Artificial Intelligence In Medicine 96 (2019) 116–122

Contents lists available at ScienceDirect

Artificial Intelligence In Medicine

journal homepage: www.elsevier.com/locate/artmed

Neural transfer learning for assigning diagnosis codes to EMRs

Anthony Rios[a], Ramakanth Kavuluru[a,b,*]

[a] Department of Computer Science, University of Kentucky, Lexington, KY, United States
[b] Division of Biomedical Informatics, Dept. of Internal Medicine, University of Kentucky, Lexington, KY, United States

ARTICLE INFO

Keywords:
Medical coding
Convolutional neural networks
Transfer learning
Multi-label classification

ABSTRACT

Objective: Electronic medical records (EMRs) are manually annotated by healthcare professionals and specialized medical coders with a standardized set of alphanumeric diagnosis and procedure codes, specifically from the International Classification of Diseases (ICD). Annotating EMRs with ICD codes is important for medical billing and downstream epidemiological studies. However, manually annotating EMRs is both time-consuming and error prone. In this paper, we explore the use of convolutional neural networks (CNNs) for automatic ICD coding. Because many codes occur infrequently, CNN performance is inhibited. Therefore, we propose supplementing EMR data with PubMed indexed biomedical research abstracts through neural transfer learning.

Materials and methods: Transfer learning is the process of "transferring" knowledge acquired from one task (the source task) to a different (target) task. For the source task, we train a CNN to predict medical subject headings (MeSH) using 1.6 million PubMed indexed biomedical abstracts. For the target task, we train a CNN on 71,463 real-world EMRs collected from the University of Kentucky (UKY) medical center to predict ICD diagnosis codes. We introduce a simple, yet effective, transfer learning methodology which avoids forgetting knowledge gained from the source task.

Results: Compared to our prior work using EMRs from the UKY medical center, we improve both the micro and macro *F*-scores by more than 8%. Likewise, compared to other transfer learning methods, our approach results in nearly 2% improvement in macro *F*-score.

Conclusion: We show that transfer learning can improve CNN performance for EMR coding in the presence of data sparsity issues. Furthermore, we find that our proposed transfer learning approach outperforms other methods with respect to macro *F*-score. Finally, we analyze how transfer learning impacts codes with respect to code frequency. We find that we achieve greater improvement on infrequent codes compared to improvements in most frequent codes.

1. Introduction

The transition to electronic medical records (EMRs) in the healthcare field has numerous benefits including facilitating the collection of accurate, up-to-date, and complete information about patients. EMRs are manually annotated by healthcare professionals and specialized medical coders with the International Classification of Diseases (ICD) codes, a standardized set of alphanumeric diagnosis and procedure codes. Annotating EMRs with ICD codes is important for medical billing. If a diagnosis code cannot be justified, then the doctor/hospital may not be paid by the insurers,[1] or worse, cause unfair financial burden to the patient. Therefore, developing automated medical coding systems and tools for human coders to become more efficient and accurate is vital.

There are two major difficulties that should be addressed when developing automated medical coding systems. First, we must develop methods which can be efficiently trained on long documents. Public EMR datasets such as MIMIC II [1] and MIMIC III [2] contain discharge summaries with around 1000 words per instance. In this paper, we use EMRs from the University of Kentucky (UKY) medical center. On average, our real-world dataset contains over 5000 words per EMR. Developing methods that can be trained efficiently on large documents is critical given this is the realistic situation for in-patient EMRs. Second, medical coding datasets are plagued with "big-small data" (data sparsity). EMR datasets may contain tens of thousands of records. However, given the large number of diagnosis and procedure codes, only a few training examples may be available for each code. It is common for many codes to never appear in the training dataset. In this

* Corresponding author at: Division of Biomedical Informatics, Department of Internal Medicine, University of Kentucky, Lexington, KY, United States.
E-mail address: rvkavu2@uky.edu (R. Kavuluru).
[1] https://www.aafp.org/fpm/2015/0900/p7.html.

https://doi.org/10.1016/j.artmed.2019.04.002
Received 19 July 2018; Received in revised form 20 December 2018; Accepted 10 April 2019
0933-3657/

paper, we introduce a transfer learning training methodology which improves the performance of convolutional neural networks (CNNs) on both frequently and infrequent occurring codes. The method described in this paper does not handle the extreme tail codes in the dataset — codes that occur only a few times or codes that never appear in the training dataset. However, we show that transfer learning can substantially improve codes that occur frequently enough for traditional supervised learning techniques.

Much of the prior work on automated ICD coding has trained models from scratch, which means the models assume zero prior knowledge about the domain. However, expert domain knowledge is abundant for various medical applications. To build models that can predict infrequent codes, it is essential to take advantage of all available information we have about the problem. Some of the available knowledge sources are in structured form. For example, the Unified Medical Language System (UMLS) [4] is a comprehensive thesaurus and ontology of biomedical concepts. However, much of the available information is in the form of unstructured text. PubMed indexes more than 27 million biomedical research articles and provides Medline citations (abstracts, titles, and other metadata) as an available resource with a free license. Users can search titles, abstracts, and all metadata (authors, affiliations, journal name) via the PubMed search interface. Some of the research articles exposed through PubMed contain relevant information about treating specific diseases or illnesses. Moreover, many of the indexed articles are "case reports" which describe the symptoms, diagnosis, and treatment of individual patients. We show an example abstract indexed by PubMed in Fig. 1a, and an example discharge report in Fig. 1b. If we compare the abstract to the "History of Present Illness" section in Fig. 1b, then we can see how this auxiliary data may be useful. For example, we observe that the patient experienced atypical headaches which should have been a sign of a serious illness (i.e., meningioma). Likewise, the EMR also reports headaches as a symptom. How can we use PubMed abstracts (including titles) to improve ICD-9 code prediction? State-of-the-art results have been achieved in text classification using CNNs with neural word embeddings. However, traditional CNN models require a large amount of training data, and using them for multi-label datasets becomes problematic for large label spaces because many labels occur infrequently. To overcome the data sparsity issue, we use *transfer learning* [5] to take advantage of the biomedical articles indexed by PubMed. Each article indexed by PubMed is annotated with a set of indexing terms called Medical Subject Headings (MeSH) terms. For example, there is a specific MeSH term for "meningioma" (D008579). In this case, there is a 1-to-many match from the MeSH term D008579 to the ICD-10-CM codes C70.0 and D32.0. Given the textual similarities observed in Fig. 1, if we pass the EMR in Fig. 1b to a model trained to predict MeSH terms, then the model may be able to predict D008579. Transfer learning is a machine learning technique which improves the predictive performance on a new task by transferring knowledge from a different but related task. We use transfer learning to improve the performance of automated medical coding systems (target task) by "transferring" knowledge acquired from learning to predict MeSH terms for biomedical articles indexed by PubMed (source task). Intuitively, instead of forcing our model to learn how to represent documents for diagnosis code prediction with a limited dataset, we pretrain a CNN on a larger dataset of PubMed abstracts to compute intermediate document representations useful for assigning diagnosis codes to EMRs. Furthermore, we introduce a simple, yet effective, method to fine-tune the document representations to the target task without forgetting the information learned from the source task.

Overall, the goal of this paper is to study the effect of transfer learning for ICD code prediction. We want to answer the following questions: Can transfer learning improve CNNs for medical coding? If transfer learning helps, then what is the best transfer learning method which achieves the largest increase in performance?

We summarize the contributions of this paper below:

- Our method uses CNNs [6,7] to efficiently train on EMRs with more than 5000 words. We also propose a new simple, yet effective, transfer learning approach to improve the performance of CNNs for ICD classification without forgetting information learned on the source task.
- We provide a comprehensive analysis comparing our method with our prior work on extracting diagnosis codes from UKY medical center's EMRs. Furthermore, besides our proposed transfer learning approach, we compare several other transfer learning methodologies to understand what works best for ICD coding.

The rest of this paper is organized as follows: In Section 2, we discuss related work in transfer learning. Section 3 presents the dataset used for our study and discusses the various transfer learning methods we use in our experiments. Next, in Section 4 we compare our method to prior work and present a detailed analysis of different transfer learning approaches. Finally, in Section 5, we summarize our contributions presented in this paper and discuss future avenues of research.

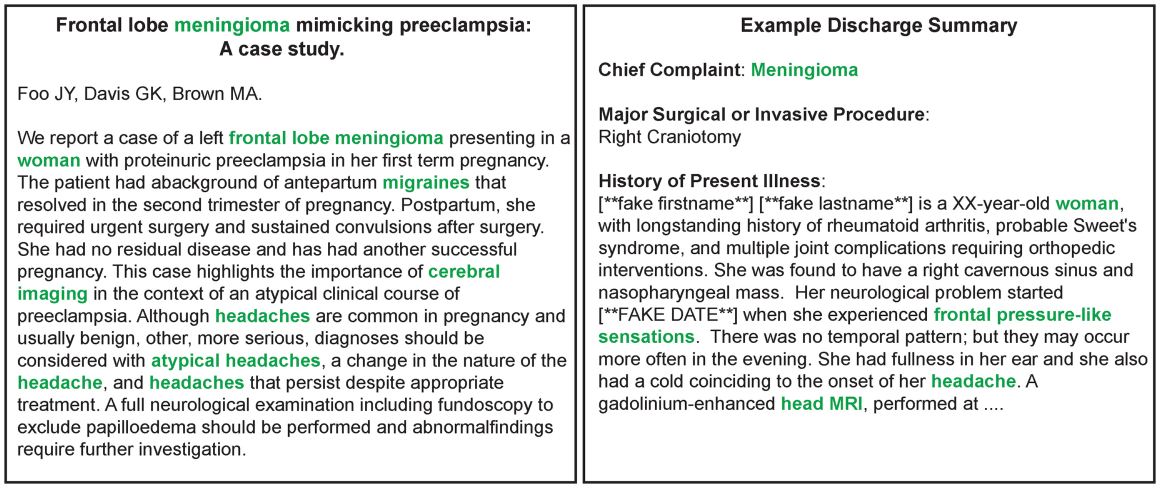

Fig. 1. In Fig. 1a, we show an example title and abstract from the PubMed indexed paper by Foo et al. [3]. In Fig. 1b, we show an example snippet from a discharge summary in the MIMIC III dataset [2].

2. Related work

2.1. Computational medical coding

Researchers have curated many datasets to support the development of machine learning-based medical coding methods. For example, the CMC [8] dataset, a corpus of radiology reports where each example is annotated with a set of 45 ICD-9-CM codes, was introduced in 2007. While it is possible to prototype methods using the CMC dataset, given the small label space, it does not provide a realistic benchmark. The Medical Information Mart for Intensive Care (MIMIC) dataset, containing around 8k ICD-9-CM codes, provides a real-world test bed for medical coding systems [1,2]. MIMIC contains discharge reports that have 1k words per report. However, in general, EMRs may contain supplementary textual information besides the discharge summary such as diagnostic reports, progress reports, and other lab reports, all of which are required to be taken into consideration when coding an EMR. In this paper, following our prior work [9], we experiment on real-world EMRs from the UKY medical center that contain more than five times more information (words) per EMR compared to the MIMIC datasets on average.

Linear models have proven to be strong baselines for extracting diagnosis and procedure codes from EMRs. For example, Perotte et al. [10] developed a hierarchical support vector machine-based method [11] which takes advantage of relationships between ICD codes using the ICD hierarchy. In our prior work, we also developed linear models and simple ensembles [12,9] for medical coding. However, recently, many researchers have shown that neural network-based methods outperform linear methods with respect to medical coding [13,14]. Baumel et al. [7] compare both recurrent neural networks (RNNs) and CNNs to assign diagnosis and procedure codes to EMRs. In Vani et al. [15] the authors modify traditional RNN architectures to ground word embeddings for multi-label classification. Mullenbach et al. [16] introduce an CNN with attention for each label (ICD-9 code). Shi et al. [17] also show that attention can improve neural ICD coding methods. Finally, in our recent work [18], we introduce a neural network architecture that incorporates a matching (a k-NN-like component) to better handle infrequent codes.

2.2. Transfer learning

Like neural networks, transfer learning has shown impressive improvements in classification applications for computer vision. Oquab et al. [5] show that parts of neural networks trained on large datasets can be used to generate features for datasets with a small number of training examples. More recently, Mou et al. [19] explored the application of transfer learning to NLP tasks. In a similar manner to Oquab et al. [5], Mou et al. [19] show that transferred neural network features are useful for prediction. Al-Stouhi and Reddy [20] show that transfer learning can improve classification performance in the presence of label imbalance. This result is promising given EMR power-law datasets generally contain large imbalances between different ICD codes.

Our method addresses similar concepts as Mou et al. [19], where they study how to apply transfer learning to various NLP tasks to understand two questions. First, does transfer learning help in NLP? Second, what is the best way to implement transfer learning? Besides the different application domains, we also introduce a different transfer learning method not explored by Mou et al. [19]. Al-Stouhi and Reddy [20] also emphasize the use of transfer learning-like methods to improve problems with label imbalance. Compared to our work, Al-Stouhi and Reddy [20] do not take advantage of recent advances in neural networks, and instead use a boosting-based classifier. Howard and Ruder [21] show that transfer learning approaches produce significant improvements by training only a language model on the source domains.

Recently, transfer learning has been shown to be useful in biomedical research. Wiens et al. [22] discuss issues with model generalization across different hospitals. For example, different hospitals may have different norms, or even EMR structures, which cause models to perform poorly if the model is trained on data from a different hospital. Choi et al. [23] use transfer learning to improve model performance across different hospitals; however they model disease progression rather than performing text classification. Transfer learning has also been shown to improve biomedical relation extraction [24]. In Rios et al. [25], we propose a transfer learning-like technique using domain adaptation for biomedical relation extraction. This method assumes no labeled training data is available for the target dataset. Finally, transfer learning has recently been applied to CNNs for ICD-9 coding by Zeng et al. [26]. Our work differs from that of Zeng et al. in three ways. First, we introduce a simple, yet effective, novel transfer learning method. Second, we provide a fine-grained analysis of different transfer learning approaches. Third, we experiment on a real set of EMRs from the UKY medical center and compare against our prior work using this dataset.

3. Materials and methods

Our dataset contains 71,463 EMRs and a total of 7485 unique diagnosis codes based on in-patient visits to the UKY hospital between 2011 to 2012. We refer to this dataset as UKLarge inline with the naming convention used in our prior paper using the exact same dataset [9]. Each EMR is annotated with a set of ICD-9 codes.[2] Following our prior work [9], we preprocess our data by truncating all ICD-9 codes of the form $abc.xy$ to $abc.x$ and we remove all codes that occur in less than 50 EMRs. Truncating and removing the most infrequent ICD-9 diagnosis codes results in a total of 1231 codes which we use for classification. Intuitively, by truncating the labels and removing codes that occur infrequently, we reduce the extreme tail of the frequency distribution — codes that only occur a few times in the dataset. Traditional neural network-based methods will not be able to predict labels that occur only a few times, even with transfer learning. The models presented in this paper are only trained on the 1231 codes. From the full dataset, 2000 EMRs are randomly removed to create a validation dataset and 3000 EMRs are held-out for final testing; the remaining EMRs (over 65,000) are used for training. These splits are identical to those in our prior work [9] to be able to compare the results appropriately. The frequency distribution over all codes is available in Fig. 2. We find that one diagnosis code occurs in more than 27,000 EMRs which is nearly 38% of the entire dataset. 400 diagnosis codes occur in no more than 100 EMRs. Basic statistics about the datasets are shown in Table 1.

3.1. Overview

Fig. 3 provides a high-level overview of our method. Transfer learning involves training two models, one model for a source task (stage 1) and the other for a target task (stage 2). Each model is trained on a different dataset. For the source task (stage 1), we collected 1.6 million PubMed citations (title and abstracts) and trained a CNN model to predict MeSH terms. All of the parameters of the source task model, except for the output layer, are used to initialize the parameters for the target task model. Finally, for stage 2, the target model is trained on the UKLarge EMR dataset to predict ICD codes.

3.2. Convolutional neural networks for text classification

In this section, we provide a brief summary and intuition of the base

[2] We realize that US health care facilities have moved to ICD-10-CM as of October 1, 2015. Given this is a recent move, it has limited the availability of training data with ICD-10 codes. Hence as proof of concept for transfer learning, we experimented with ICD-9 codes.

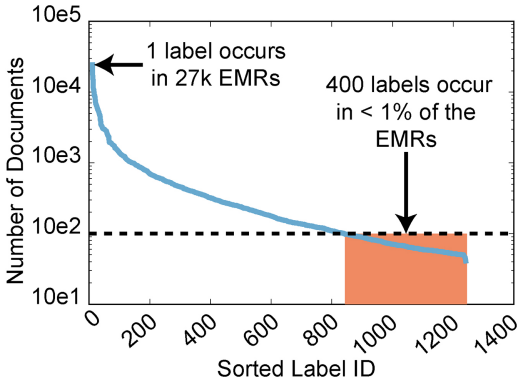

Fig. 2. ICD-9 code frequency distribution of the UKLarge EMR dataset.

Table 1
Transfer learning dataset statistics.

	PubMed	UKLarge
# Instances	1,600,000	71,463
# Labels	27,150	1231
Label cardinality	12.62	7.4
Avg # words per instance	147	5303
# Code combinations	–	60,238

CNN architecture we use to represent each EMR [6,27][6, 27, p. 386]. Intuitively, CNNs learn to find informative ngrams in the input instance. We generate ngram scores by creating a ngram feature vector which is the concatenation of every s successive word embeddings in the document

$$\mathbf{m}_j = \mathbf{w}_{i-s+1} || \cdots || \mathbf{w}_i$$

where $\mathbf{m}_j \in \mathbb{R}^{sd}$ is the jth ngram feature representation. To score how informative $\mathbf{m}_j$ is, $\mathbf{m}_j$ is passed through a non-linear function

$$\hat{\mathbf{m}}_j = \text{ReLU}(\mathbf{W}\,\mathbf{m}_j + \mathbf{b}),$$

where $\mathbf{W} \in \mathbb{R}^{v \times sd}$, $\mathbf{b} \in \mathbb{R}^v$, and *ReLU* is a rectified linear unit [28,29]. Each row of $\mathbf{W}$ forms a single convolutional filter and v is the number of scores we generate for each ngram. Thus, to form a fixed size feature vector of the document, we use max-over-time pooling

$$g(\mathbf{x}) = [\hat{\mathbf{m}}'_1, \hat{\mathbf{m}}'_2, \ldots, \hat{\mathbf{m}}'_v], \quad \text{where}$$

$$\hat{\mathbf{m}}'_i = \max(\hat{\mathbf{m}}^j_1, \hat{\mathbf{m}}^j_2, \ldots, \hat{\mathbf{m}}^j_{n-s+1}),$$

$g(\mathbf{x}) \in \mathbb{R}^v$, and $\hat{\mathbf{m}}^j_i$ represents the jth element of the ith ngram vector. For convenience, in the remainder of this paper, we will refer to the convolution filters $\mathbf{W}$, which form the convolution layer, as "CV" and the embedding layer (all the word vectors) as "EM".

3.3. Stage 1: Training on source

To use transfer learning techniques, we first train our model on the source data. Given $g(\mathbf{x})$, we pass it through multiple sigmoid outputs, one for each label

$$\hat{\mathbf{y}}_S = \text{sigmoid}(\mathbf{W}_S g(\mathbf{x}) + \mathbf{b}_S)$$

where $\mathbf{W}_S \in \mathbb{R}^{L_S \times k}$, $\mathbf{b}_S \in \mathbb{R}^{L_S}$, L_S is number of source labels, and the sigmoid function is defined as

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}}.$$

For multi-label classification, sigmoid units are required rather than the softmax layer used for multi-class classification. Each element, $\hat{\mathbf{y}}_i$, produces a score for each label using the sigmoid squashing function which constrains the score to the range [0, 1].

We train over all labels jointly by minimizing the multi-label binary cross-entropy loss [30] parameterized by $\boldsymbol{\theta}$, inputs $\mathbf{x}$, and outputs $\mathbf{y}$ as

$$J_{\text{CE}}(\theta; \mathbf{x}; \mathbf{y}) = -\sum_{l}^{L_S} y_l \log(\hat{y}_l) + (1 - y_l)\log(1 - \hat{y}_l).$$

where L_s is the number of labels in the source task, y_l is the Boolean 1/0 ground truth and $\hat{y}_l$ is the probability estimate for the lth label using our model. The loss, J_{CE}, can be optimized using stochastic gradient descent (SGD).

3.4. Stage 2: Transfer learning

We experiment with three traditional transfer learning approaches and introduce a new method. The three traditional methods take the model trained on the source dataset and replace the output layer with two additional layers. First, given $g(\mathbf{x})$, the max-pooled feature vectors, we pass it through a full-connected layer

$$\mathbf{h} = \text{ReLU}(\mathbf{W}_a g(\mathbf{x}) + \mathbf{b}_a) \tag{1}$$

where $\mathbf{W}_a \in \mathbb{R}^{\alpha \times k}$ and $\mathbf{b}_a \in \mathbb{R}^{\alpha}$. In transfer learning literature, this layer is known as as an "adaptation layer" [5]. The adaptation layer learns to transform the mid-level features optimized on the source dataset to better represent the target data.

Next, $\mathbf{h}$ is passed to a target specific output layer

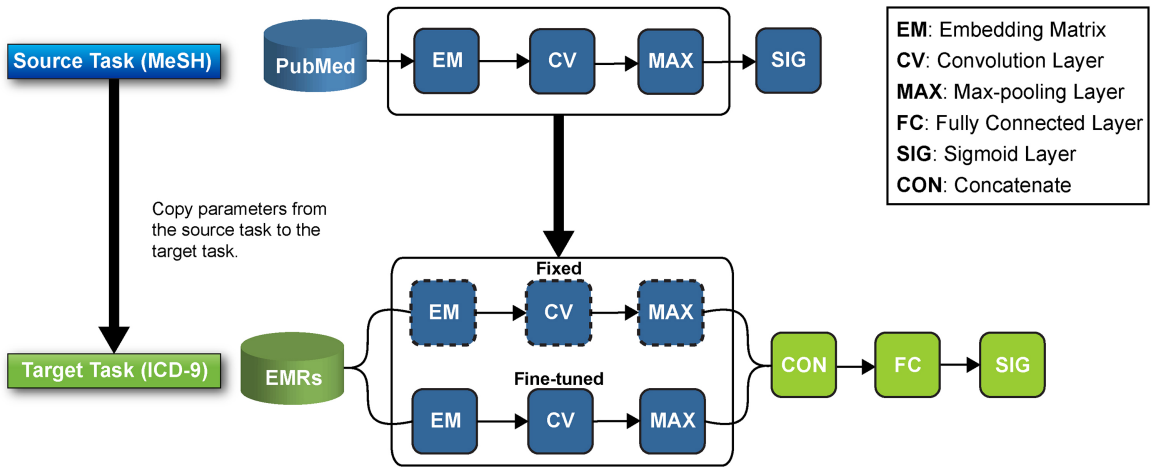

Fig. 3. The parameters learned in the source task are transferred to the target task model and fixed while the target task specific model parameters are updated during training.

$$\hat{\mathbf{y}} = \text{sigmoid}(\mathbf{W}_o \mathbf{h} + \mathbf{b}_o) \quad (2)$$

where $\mathbf{W}_o \in \mathbb{R}^{L \times \alpha}$, $\mathbf{b}_o \in \mathbb{R}^L$, and L is the total number of target labels.

As previously stated, we experiment with three recently proposed transfer learning methods. Each method shares the same overall CNN architecture. However, they vary based on which parameters are updated while training on the target dataset. We describe the different variations below:

- EM[✗] CV[✗] – For this variation, all parameters used during stage 1 including the word vectors EM and convolution weights CV, are not updated during the stage 2 training process. However, The adaptation layer parameters, $\mathbf{W}_a$ and $\mathbf{b}_a$, and the target output layer parameters, $\mathbf{W}_o$ and $\mathbf{b}_o$, are updated.
- EM[✗] CV[✓] – This method is initialized with the CNN weights after stage 1. Similar to the previous method, we keep the word embeddings fixed. However, the convolution parameters CV are fine-tuned during stage 2.
- EM[✓] CV[✓] – The third method expands on EM[✗] CV[✓] by fine-tuning both the word embeddings and convolution parameters while training on the target dataset.

We also introduce a simple, yet effective, transfer learning method. Transfer learning methods that fine-tune the weights transferred from the source task tend to forget what they have learned from the source dataset [31,32]. Generally, this issue is measured by testing how well the fine-tuned NNs perform on the original source task after fine-tuning. In our case, we are only concerned about predictive performance on the target task, assigning ICD diagnosis codes to EMRs. Therefore, we are not concerned with how well the model performs on the source dataset. However, we hypothesize if we forget information about the source dataset, our model will not generalize as well to the target task. We believe this given the high-level of similarity between the two domains. To overcome the issue of catastrophic forgetting, we propose the method EM[✓] CV[✓] + EM[✗] CV[✗]. Specifically, we make two copies of the word embeddings and convolution parameters learned during stage 1. The two copies are used to generate two mid-level representations, $g(\mathbf{x})$ and $g'(\mathbf{x})$, of each document x. Both representations are concatenated

$$\mathbf{h}_2 = g(\mathbf{x}) \| g'(\mathbf{x})$$

where $\mathbf{h}_2 \in \mathbb{R}^{2k}$. $\mathbf{h}_2$ is then passed to the adaptation layer defined in Eq. (1), then to the output layer defined by Eq. (2). During training, we only optimize the word embedding and convolution parameters used to generate $g(\mathbf{x})$. The parameters that create $g'(\mathbf{x})$ are not updated.

3.5. Word dropout

EMRs in the UKLarge dataset contain more than 5000 words per instance on average. A few examples in the dataset contain more than 10,000 words. Training on lengthy instances can take a long time and uses a lot of memory on the GPU. To improve training efficiency, we use word dropout [33]. Similar to dropout which randomly sets some unit weights to zero to avoid overfitting, during training, word dropout completely removes words from an EMR at random. Besides reducing the overall training time, word dropout also reduces overfitting by acting as a regularizer that perturbs documents slightly. For long documents, we assume that removing words from the document will not substantially change the overall meaning of what the EMRs describe.

3.6. Ensemble

It is possible for our model to overfit to infrequently occurring labels. Wallace et al. [34] show that bagging multiple oversampled classifiers improve the performance of infrequent labels in the multi-class setting. However, oversampling is not trivial in the multi-label setting. Averaging multiple NNs trained with different seeds is a well known way to improve performance [35] of NNs in general. Therefore, we train Γ different models, each initialized with a different random seed. At test time, the predictions for each model are averaged

$$\hat{\mathbf{y}}_e = \frac{1}{\Gamma} \sum_{i=1}^{\Gamma} \hat{\mathbf{y}}^i$$

where $\hat{\mathbf{y}}_e \in \mathbb{R}^L$ and $\hat{\mathbf{y}}^i$ represents the predictions for the ith model.

4. Results

In this section, we compare our work with prior medical coding methods on the UKLarge dataset. We also analyze how our transfer learning model compares to related methods.

4.1. Implementation details

The CNN architecture used in this work [6] uses convolution filter widths that span 3, 4, and 5 words. We train 300 filters for each filter width. Therefore, the size of the max-pooled feature vectors $g(\mathbf{x})$ will have a dimensionality of 900. Each filter, for each width, will produce a feature map that is proportional to the size of the sentence. However, a fixed-size vector is required by the output layer. As shown in Fig. 3, max-over-time pooling is used to convert the output of the convolution layer – the feature maps – to a single fixed size vector. Specifically, max-pooling will return the largest real number in each feature map. The word embedding dimensionality is set 300. The adaptation layer dimensionality parameter α is set to 512 for the transfer learning stage (Section 3.4). We use standard dropout before the final output layer with a dropout probability of 0.5. The dropout probability for word dropout is set to 0.3. Furthermore, we truncate all documents to a max length of 6000 words by simply removing all words after the 6000th word. However, given the average length of an EMR only has 5303 words, only a small subset of the EMRs are actually truncated. The model is optimized using the SGD variant AdaDelta [36] with a learning learning rate of 0.001 and a minibatch size of 50.

4.2. Baseline methods

We compare against three different methods:

1. A logistic regression (LR) model (one per label) trained on tf-idf weighted n-grams.
2. A more complex model, LR + L2R + NERC, which uses label scores from LR, the k-nearest neighbor similarity scores, and named entity recognition based codes (NERC) extracted using NLM's MetaMap [37] as features, to a second-level stacking-like learning-to-rank (L2R) method [38]. This was the best model from our prior work [9] for the UKLarge dataset used here.
3. A model averaging ensemble with three CNNs without transfer learning.

We also experiment with two versions of each transfer learning method, an ensemble model that averages 3 models trained with different seeds, and a single model with out model averaging.

4.3. Evaluation measures

For evaluation, we use two evaluation measures: micro and macro F-score. Both F-score measures have been widely adopted for multi-label classification [39]. For each label l_j, we define the label-based precision $P(l_j)$, recall $R(l_j)$, and F-score $F(l_j)$ as

$$P(T_j) = \frac{\mathrm{TP}_j}{\mathrm{TP}_j + \mathrm{FP}_j}, \quad R(T_j) = \frac{\mathrm{TP}_j}{\mathrm{TP}_j + \mathrm{FN}_j},$$

$$\text{and } F(l_j) = \frac{2P(l_j)R(l_j)}{P(l_j) + R(l_j)},$$

where TP_j, FP_j, and FN_j are true positives, false positives, and false negatives, respectively, of label l_j. Given the *F*-score for each label, the label-based macro *F*-score is defined as

$$\text{Macro} - \text{F} = \frac{1}{L}\sum_{j=1}^{L} F(l_j).$$

The label-based micro precision, recall, and *F*-score are defined as

$$P^{\text{mic}} = \frac{\sum_{j=1}^{L} \mathrm{TP}_j}{\sum_{j=1}^{L} (\mathrm{TP}_j + \mathrm{FP}_j)}, \quad R^{\text{mic}} = \frac{\sum_{j=1}^{L} \mathrm{TP}_j}{\sum_{j=1}^{L} (\mathrm{TP}_j + \mathrm{FN}_j)},$$

$$\text{and Micro} - \text{F} = \frac{2P^{\text{mic}}.R^{\text{mic}}}{P^{\text{mic}} + R^{\text{mic}}},$$

respectively. Intuitively, the macro measures give equal importance to all labels independent of the label frequency, while the micro measures give more weight to frequently occurring labels.

4.4. Layer by layer analysis

In Table 2 we compare the different transfer learning variations. We find that updating parameters always outperforms transfer learning methods that keep parameters fixed. For example, EM[✗] CV[✓] outperforms EM[✗] CV[✗] by more than 3% with respect to micro *F*-Score. Likewise, EM[✓] CV[✓] improves by more than 3% over the micro *F*-score obtained by EM[✗] CV[✓]. Without ensembling, we find only a small improvement in micro *F*-Score using the EM[✓] CV[✓] + EM[✗] CV[✗] method. However, we find nearly a 2% improvement with respect to macro *F*-score. If updating all the parameters outperforms methods which fixes the weights, then does this imply that catastrophic forgetting is not an issue for our task? The difference in macro *F*-score between EM[✓] CV[✓] and EM[✗] CV[✗] is only 0.6%. Yet, EM[✓] CV[✓] + EM[✗] CV[✗] improves the macro *F*-score over EM[✓] CV[✓] by nearly 2%. This result suggests that forgetting source task information may not negatively impact infrequent codes when we update the parameters on the target task. However, it also does not improve the performance of infrequent codes either. When we update the weights and store an extra copy of the source copy paramters (EM[✓] CV[✓] + EM[✗] CV[✗]), then it generalizes better across all ICD-9 diagnosis codes regardless of the code frequency in the dataset.

4.5. Comparison with prior work

In Table 3 we compare our proposed transfer learning method with prior work on the UKLarge EMR dataset. We improve over LR by more than 8% for both the micro and macro *F*-Scores. Our ensemble method improves on "LR + L2R + NERC" by nearly 7% micro *F*-Score which suggests that NNs can better predict frequent labels. Likewise, the ensemble approach improves on the prior best macro *F*-score by more than 5%. Even without ensembling, we improve over LR + L2R + NERC by 3% with respect to the macro *F*-score. Overall, we find that even in the presence of data sparsity, NNs can outperform traditional text classification methods when we use transfer learning.

Table 2
Layer-by-layer results for various transfer learning methodologies.

	Micro *F*-score	Macro *F*-score
EM[✗] CV[✗]	46.5	23.6
EM[✗] CV[✓]	49.8	23.8
EM[✓] CV[✓]	53.1	24.2
EM[✓] CV[✓] + EM[✗] CV[✗]	53.5	26.0
EM[✗] CV[✗] AVG	48.3	25.5
EM[✗] CV[✓] AVG	51.3	25.5
EM[✓] CV[✓] AVG	54.1	25.8
EM[✓] CV[✓] + EM[✗] CV[✗] AVG	**56.7**	**28.6**

The bold values signifies top scores for precision, recall, and F-score. The higher the score the better the performance as per these metrics.

Table 3
Results comparing conventional approaches (from Table 4 in Kavuluru et al. [9]), CNNs, and CNNs with transfer learning.

	Micro *F*-score	%-Increase	Macro *F*-score	%-Increase
LR [9]	48.2	–	19.8	–
LR + L2R [9]	49.5	1.3%	21.2	1.4%
LR + L2R + NERC [9]	49.9	1.7%	23.0	3.2%
EM[✓] CV[✓] + EM[✗] CV[✗]	53.5	5.3%	26.0	6.2%
EM[✓] CV[✓] + EM[✗] CV[✗] AVG	**56.7**	**8.5%**	**28.6**	**8.8%**

The bold values signifies top scores for precision, recall, and F-score. The higher the score the better the performance as per these metrics.

4.6. Label frequency analysis

In Fig. 4, we analyze the macro *F*-Scores of the 10% least frequent and 10% most frequent diagnosis codes in the UKLarge dataset. While calculating the macro *F*-score over all labels gives some insight about how our method performs on infrequent labels, if the frequent and infrequent codes are jointly compared, then it confounds its interpretation. We find that our proposed method improves infrequent label performance by 5%. The macro-averaged performance improves by 2% for frequent classes. Compared to EM[✓] CV[✓], these results suggests that the source information EM[✓] CV[✓] + EM[✗] CV[✗] avoids forgetting has a greater impact on infrequent codes. Moreover, we find that the linear model performs similar to EM[✓] CV[✓] on infrequent codes. For the top 10% most frequent codes, the linear model is substantially worse than the two neural network methods.

5. Conclusion

In this paper, we demonstrate the potential of transfer learning using CNNs for biomedical text classification. Furthermore, we introduce a simple transfer learning methodology (EM[✓] CV[✓] + EM[✗] CV[✗]) that improves on prior transfer learning approaches. Our method also improves on our prior methods for the UKLarge dataset. The major weakness of this line of work is similar to the weaknesses of

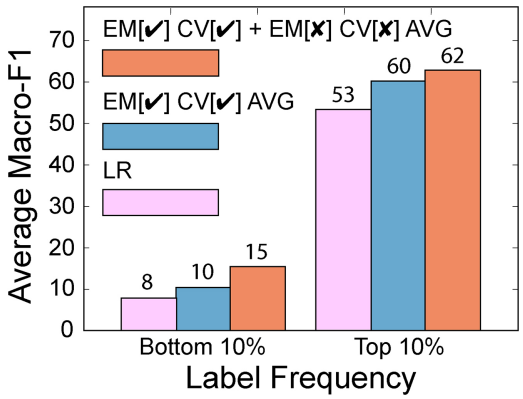

Fig. 4. Macro *F*-scores on the top 10% least frequent codes to the macro *F*-score on the top 10% most frequent ICD-9 diagnosis codes.

other transfer learning methodologies – we must train our model on two different datasets. However, we believe this is an acceptable weakness because only the training time is increased.

There are three major avenues for future work:

- We use 1.6 million abstracts indexed by PubMed as our source data for transfer learning. However, PubMed indexes more the 27 million research articles. To handle an order of magnitude more data, we need to develop methods which can scale well. If we have more data, then we will be able to experiment with more sophisticated models.
- We only transfer knowledge through the learned textual representations formed by the CNN. A subset of ICD-9 codes are contained in the MeSH terminology (e.g., meningioma). We may be able to predict certain ICD-9 codes using the MeSH CNN directly. Unfortunately, issues such as differences in the data distribution may affect the predictive performance achieved using the MeSH model directly. If we can jointly take advantage of the textual knowledge available in the PubMed indexed research articles via transfer learning and the label overlap between MeSH and ICD-9-CM, then we may be able to overcome the data sparsity problem.
- The focus of this paper is to study the impact of transfer learning on ICD-9-CM code extraction from real-world EMRs. For our experiments, we train our source model to predict MeSH terms. Besides studying MeSH to ICD-9-CM, it is also important to explore transfer learning between hospitals (ICD to ICD) [22]. In future work, we plan to explore the use of the MIMIC EMR dataset [2] as our source task which will model hospital-to-hospital transfer learning.

Conflicts of interest

None declared.

Acknowledgment

This research is supported by the U.S. National Library of Medicine through grant R21LM012274. We also gratefully acknowledge the support of the NVIDIA Corporation for its donation of the Titan X Pascal GPU used for this research.

References

[1] Lee J, Scott DJ, Villarroel M, Clifford GD, Saeed M, Mark RG. Open-access mimic-ii database for intensive care research. IEEE annual international conference engineering in medicine and biology society (EMBC) 2011:8315–8.
[2] Johnson A, Pollard T, Shen L, Lehman L, Feng M, Ghassemi M, et al. Mimic-iii, a freely accessible critical care database. Sci Data 2016;3.
[3] Foo JY, Davis GK, Brown MA. Frontal lobe meningioma mimicking preeclampsia: a case study. Obstet Med 2017;10(4):192–4.
[4] Bodenreider O. The unified medical language system (umls): integrating biomedical terminology. Nucl Acids Res 2004;32(suppl_1):D267–70.
[5] Oquab M, Bottou L, Laptev I, Sivic J. Learning and transferring mid-level image representations using convolutional neural networks. IEEE conference on computer vision and pattern recognition (CVPR) 2014:1717–24.
[6] Kim Y. Convolutional neural networks for sentence classification. Empirical methods in natural language processing (EMNLP) 2014:1746–51.
[7] Baumel T, Nassour-Kassis J, Elhadad M, Elhadad N. Multi-label classification of patient notes a case study on ICD code assignment. 2017arXiv preprint arXiv:1709.09587.
[8] Pestian JP, Brew C, Matykiewicz P, Hovermale DJ, Johnson N, Cohen KB, et al. A shared task involving multi-label classification of clinical free text. Proceedings of the workshop on BioNLP: biological, translational, and clinical language processing 2007:97–104.
[9] Kavuluru R, Rios A, Lu Y. An empirical evaluation of supervised learning approaches in assigning diagnosis codes to electronic medical records. Artif Intell Med 2015;65(2):155–66.
[10] Perotte A, Pivovarov R, Natarajan K, Weiskopf N, Wood F, Elhadad N. Diagnosis code assignment: models and evaluation metrics. J Am Med Inform Assoc 2013;21(2):231–7.
[11] Zhang D, He D, Zhao S, Li L. Enhancing automatic icd-9-cm code assignment for medical texts with pubmed. BioNLP 2017;2017:263–71.
[12] Rios A, Kavuluru R. Supervised extraction of diagnosis codes from EMRs: role of feature selection, data selection, and probabilistic thresholding. IEEE international conference on healthcare informatics (ICHI) 2013:66–73.
[13] Karimi S, Dai X, Hassanzadeh H, Nguyen A. Automatic diagnosis coding of radiology reports: a comparison of deep learning and conventional classification methods. BioNLP 2017;2017:328–32.
[14] Duarte F, Martins B, Pinto CS, Silva MJ. Deep neural models for icd-10 coding of death certificates and autopsy reports in free-text. J Biomed Inform 2018;80:64–77.
[15] Vani A, Jernite Y, Sontag D. Grounded recurrent neural networks. 2017. . arXiv preprint arXiv:1705.08557.
[16] Mullenbach J, Wiegreffe S, Duke J, Sun J, Eisenstein J. Explainable prediction of medical codes from clinical text. North American chapter of the association for computational linguistics (NAACL) 2018.
[17] Shi H, Xie P, Hu Z, Zhang M, Xing EP. Towards automated ICD coding using deep learning. 2017arXiv preprint arXiv:1711.04075.
[18] Rios A, Kavuluru R. EMRcoding with semi-parametric multi-head matching networks. Proceedings of the North American chapter of the association for computational linguistics (NAACL) 2018.
[19] Mou L, Meng Z, Yan R, Li G, Xu Y, Zhang L, et al. How transferable are neural networks in NLP applications? Conference on empirical methods in natural language processing (EMNLP) 2016:479–89.
[20] Al-Stouhi S, Reddy CK. Transfer learning for class imbalance problems with inadequate data. Knowl Inf Syst 2016;48(1):201–28.
[21] Howard J, Ruder S. Universal language model fine-tuning for text classification. Proceedings of the 56th annual meeting of the association for computational linguistics (Volume 1: Long Papers), vol. 1 2018:328–39.
[22] Wiens J, Guttag J, Horvitz E. A study in transfer learning: leveraging data from multiple hospitals to enhance hospital-specific predictions. J Am Med Inform Assoc 2014;21(4):699–706.
[23] Choi E, Bahadori MT, Schuetz A, Stewart WF, Sun J. Doctor AI: predicting clinical events via recurrent neural networks. Machine learning for healthcare conference 2016:301–18.
[24] Sahu SK, Anand A. What matters in a transferable neural network model for relation classification in the biomedical domain? Artif Intell Med 2018;87:60–6.
[25] Rios A, Kavuluru R, Lu Z. Generalizing biomedical relation classification with neural adversarial domain adaptation. Bioinformatics 2018;1:9.
[26] Zeng M, Li M, Fei Z, Yu Y, Pan Y, Wang J. Automatic icd-9 coding via deep transfer learning. Neurocomputing 2019;324:43–50.
[27] Goldberg Y. A primer on neural network models for natural language processing. J Artif Intell Res 2016;57:345–420.
[28] Glorot X, Bordes A, Bengio Y. Deep sparse rectifier networks. International conference on artificial intelligence and statistics. JMLR W&CP Volume, vol. 15 2011:315–23.
[29] Nair V, Hinton GE. Rectified linear units improve restricted Boltzmann machines. International conference on machine learning (ICML) 2010:807–14.
[30] Nam J, Kim J, Loza Mencía E, Gurevych I, Fürnkranz J. Large-scale multi-label text classification – revisiting neural networks. European conference on machine learning and knowledge discovery in databases – (ECML PKDD) 2014:437–52.
[31] Li Z, Hoiem D. Learning without forgetting. IEEE Trans Pattern Anal Mach Intell 2018;40(12):2935–47.
[32] Kirkpatrick J, Pascanu R, Rabinowitz N, Veness J, Desjardins G, Rusu AA, et al. Overcoming catastrophic forgetting in neural networks. Proc Natl Acad Sci 2017;114(13):3521–6.
[33] Iyyer M, Manjunatha V, Boyd-Graber J, Daumé III H. Deep unordered composition rivals syntactic methods for text classification. Annual meeting of the association for computational linguistics (ACL) 2015:1681–91.
[34] Wallace BC, Small K, Brodley CE, Trikalinos TA. Class imbalance, redux. IEEE international conference on data mining (ICDM) 2011:754–63.
[35] Huang G, Li Y, Pleiss G, Liu Z, Hopcroft JE, Weinberger KQ. Snapshot ensembles: Train 1, get m for free. International conference on learning representations (ICLR) 2017.
[36] Zeiler MD. ADADELTA: an adaptive learning rate method. 2012arXiv preprint arXiv:1212.5701.
[37] Aronson AR, Lang F-M. An overview of metamap: historical perspective and recent advances. J Am Med Inform Assoc 2010;17(3):229–36.
[38] Liu T-Y. Learning to rank for information retrieval. Found Trends Inf Retriev 2009;3(3):225–331.
[39] Tsoumakas G, Katakis I, Vlahavas IP. Mining multi-label data. Data mining and knowledge discovery handbook. 2010. p. 667–85.