Multi-Label Classification of Patient Notes a Case Study on ICD Code Assignment

Tal Baumel talbau@cs.bgu.ac.il
Jumana Nassour-Kassis jumanan@cs.bgu.ac.il
Michael Elhadad elhadad@cs.bgu.ac.il
Department of Computer Science
Ben-Gurion University
Beersheba, Israel

Noémie Elhadad noemie.elhadad@columbia.edu
Department of Biomedical Informatics
Columbia University
New York, NY

Abstract

In the context of the Electronic Health Record, automated diagnosis coding of patient notes is a useful task, but a challenging one due to the large number of codes and the length of patient notes. We investigate four models for assigning multiple ICD codes to discharge summaries taken from both MIMIC II and III. We present Hierarchical Attention-GRU (HA-GRU), a hierarchical approach to tag a document by identifying the sentences relevant for each label. HA-GRU achieves state-of-the art results. Furthermore, the learned sentence-level attention layer highlights the model decision process, allows easier error analysis, and suggests future directions for improvement.

1. Introduction

In Electronic Health Records (EHR), there is often a need to assign multiple labels to a patient record, choosing from a large number of potential labels. Diagnosis code assignment is such a task, with a massive amount of labels to chose from (14,000 ICD9 codes and 68,000 ICD10 codes). Large-scale multiple phenotyping assignment, problem list identification, or even intermediate patient representation can all be cast as a multi-label classification over a large label set. More recently, in the context of predictive modeling, approaches to predict multiple future healthcare outcomes, such as future diagnosis codes or medication orders have been proposed in the literature. There again, the same setup occurs where patient-record data is fed to a multi-label classification over a large label set.

In this paper, we investigate how to leverage the unstructured portion of the EHR, the patient notes, along a novel application of neural architectures. We focus on three characteristics: **(i) a very large label set** (8,000 unique ICD9 codes and 1,047 3-digit unique codes); **(ii) a multi-label setting** (up to 20 labels per instance); **(iii) instances are long documents** (discharge summaries on average 1500 word long); and **(iv)** furthermore, because we work on long documents, one critical aspect of the multi-label classification is **transparency**—to highlight the elements in the documents that explain and support the predicted labels. While there has been much work on each of these characteristics, there has been limited work to tackle all at once, particularly in the clinical domain.

We experiment with four approaches to classification: an **SVM-based one-vs-all** model, **a continuous bag-of-words** (CBOW) model, **a convolutional neural network**

(CNN) model, and **a Gated Recurrent Unit model with a Hierarchical Attention mechanism** (HA-GRU). Among them, the attention mechanism of the HA-GRU model provides full **transparency for classification decisions**. We rely on the publicly available MIMIC datasets to validate our experiments. A characteristic of the healthcare domain is long documents with a large number of technical words and typos/misspellings. We experiment with simple yet effective preprocessing of the input texts.

Our results show that careful tokenization of the input texts, and hierarchical segmentation of the original document allow our Hierarchical Attention GRU architecture to yield the most promising results, over the SVM, CBOW, and CNN models, while preserving the full input text and providing effective transparency.

2. Previous Work

We review previous work in the healthcare domain as well as recent approaches to extreme multi-label classification, which take place in a range of domains and tasks.

2.1 Multi-label Patient Classifications

Approaches to classification of patient records against multiple labels fall into three types of tasks: diagnosis code assignment, patient record labeling, and predictive modeling.

Diagnosis Code Assignment. Automated ICD coding is a well established task, with several methods proposed in the literature, ranging from rule based (Crammer et al., 2007; Farkas and Szarvas, 2008) to machine learning such as support vector machines, Bayesian ridge regression, and K-nearest neighbor (Larkey and Croft, 1995; Lita et al., 2008). Some methods exploit the hierarchical structure of the ICD taxonomy (Perotte et al., 2011, 2014), while others incorporated explicit co-occurrence relations between codes (Kavuluru et al., 2015). In many cases, to handle the sheer amount of labels, the different approaches focus on rolled-up ICD codes (i.e., 3-digit version of the codes and their descendants in the ICD taxonomy) or on a subset of the codes, like in the shared community task for radiology code assignment (Pestian et al., 2007).

It is difficult to compare the different methods proposed, since each relies on different (and usually not publicly available) datasets. We decided to use MIMIC dataset, since it is publicly available to the research community. Methods-wise, our approach departs from previous work in two important ways: we experiment with both massively large and very large label sets (all ICD9 code and rolled-up ICD9 codes), and we experiment with transparent models that highlight portions of the input text that support the assigned codes.

Patient Record Labeling. Other than automated diagnosis coding, most multi-label patient record classifiers fall in the tasks of phenotyping across multiple conditions at once. For instance, the UPhenome model takes a probabilistic generative approach to assign 750 latent variables (Pivovarov et al., 2015). More recently, in the context of multi-task learning, Harutyunyan and colleagues experimented with phenotyping over 25 critical care conditions (Harutyunyan et al., 2017).

Predictive Modeling. Previous work in EHR multi-label classification has mostly focused on predictive scenarios. The size of the label set varies from one approach to another, and most limit the label set size however: DeepPatient (Miotto et al., 2016) predicts over a set of 78 condition codes. Lipton et al. (2015) leverage an LSTM model to predict over a vocabulary of 128 diagnosis codes. DoctorAI (Choi et al., 2015) predicts over a set of 1,183 3-digit ICD codes and 595 medication groups. The Survival Filter (Ranganath et al., 2015) predicts a series of future ICD codes across approximately 8,000 ICD codes.

Inputs to Multi-Label Classifications. Most work in multi-label classification takes structured input. For instance, the Survival Filter expects ICD codes as input to predict the future ICD codes. DoctorAI takes as input medication orders, ICD codes, problem list, and procedure orders at a given visit. Deep Patient does take the content of notes as input, but the content is heavily preprocessed into a structured input to their neural network, by tagging all texts with medical named entities. In contrast, our approach is to leverage the entire content of the input texts. Our work contributes to clinical natural language processing (Demner Fushman and Elhadad, 2016), which only recently investigated neural representations and architectures for traditional tasks such as named entity recognition (Jagannatha and Yu, 2016).

2.2 Multi-label Extreme Classification

In extreme multi-label learning, the objective is to annotate each data point with the most relevant subset of labels from an extremely large label set. Much work has been carried outside of the healthcare domain on tasks such as image classification (Tsoumakas and Katakis, 2006; Weston et al., 2011), question answering (Choi et al., 2016), and advertising (Jain et al., 2016). In (Weston et al., 2011), the task of annotating a very large dataset of images ($> 10M$) with a very large label set ($> 100K$) was first addressed. The authors introduced the WSABIE method which relies on two main features: (i) records (images) and labels are embedded in a shared low-dimension vector space; and (ii) the multi-label classification task is modeled as a ranking problem, evaluated with a Hamming Loss on a P@k metric. The proposed online approximate WARP loss allowed the algorithm to perform fast enough on the scale of the dataset. We found that in our case, the standard Macro-F measure is more appropriate as we do not tolerate approximate annotations to the same extent as in the image annotation task.

The SLEEC method (Bhatia et al., 2015) also relies on learning an embedding transformation to map label vectors into a low-dimensional representation. SLEEC learns an ensemble of local distance preserving embeddings to accurately predict infrequently occurring labels. This approach attempts to exploit the similarity among labels to improve classification, and learns different representations for clusters of similar labels. Other approaches attempt to reduce the cost of training over very large datasets by considering only part of the labels for each classification decision (Yen et al., 2016). SLEEC was later improved in (Jain et al., 2016) with the PfastreXML method which also adopted P@k loss functions aiming at predicting tail labels.

In (Joulin et al., 2016), the FastText method was introduced as a simple and scalable neural bag of words approach for assigning multiple labels to text. We test a similar model (CBOW) in our experiments as one of our baselines.

	MIMIC II	MIMIC III	Test Set
#Records	20,533	49,857	2,282
#Unique Tokens	69,248	119,171	33,958
Avg. #Tokens per Record	1528.88	1947.32	1893.39
Avg. #Sentences per Record	90.5	111.7	103.63
#Labels	4,847	6,527	2,451
Label Cardinality	9.24	11.48	11.42
Label Density	0.0019	0.0018	0.0047
% of Labels with at least 50 records	11.33%	18.19%	4.08%

Table 1: Datasets descriptive statistics.

3. Dataset and Preprocessing

We use the publicly available de-identified MIMIC dataset of ICU stays from Beth Israel Deaconess Medical Center (Saeed et al., 2011; Johnson et al., 2016).

3.1 MIMIC Datasets

To test the impact of training size, we relied on both the MIMIC II (v2.6) and MIMIC III (v1.4) datasets. MIMIC III comprises records collected between 2001 and 2012, and can be described as an expansion of MIMIC II (which comprises records collected between 2001 and 2008), along with some edits to the dataset (including de-identification procedures).

To compare our experiments to previous work in ICD coding, we used the publicly available split of MIMIC II from Perotte et al. (2014). It contains 22,815 discharge summaries divided into a training set (20,533 summaries) and a test-set of unseen patients (2,282 summaries). We thus kept the same train and the test-set from MIMIC II, and constructed an additional training set from MIMIC III. We made sure that the test-set patients remained unseen in this training set as well. Overall, we have two training sets, which we refer to as MIMIC II and MIMIC III, and a common test-set comprising summaries of unseen patients.

While there is a large overlap between MIMIC II and MIMIC III, there are also marked differences. We found many cases where discharge summaries from 2001-2008 are found in one dataset but not in the other. In addition, MIMIC III contains addenda to the discharge summaries that were not part of MIMIC II. After examining the summaries and their addenda, we noticed that the addenda contain vital information for ICD coding that is missing from the main discharge summaries; therefore, we decided to concatenate the summaries with their addenda.

Table 1 reports some descriptives statistics regarding the datasets. Overall, MIMIC III is larger than MIMIC II from all standpoints, including amounts of training data, vocabulary size, and overall number of labels.

3.2 ICD9 Codes

Our label set comes from the ICD9 taxonomy. The International Classification of Diseases (ICD) is a repository maintained by the World Health Organization (WHO) to provide a standardized system of diagnostic codes for classifying diseases. It has a hierarchical structure, connecting specific diagnostic codes through is-a relations. The hierarchy has eight levels, from less specific to more specific. ICD codes contain both diagnosis and procedure codes, though the discharge summaries of both MIMIC II and III contained only diagnosis codes. ICD codes are typically conveyed as 5 digits, with 3 primary digits and 2 secondary ones.

Table 1 provides the ICD9 label cardinality and density as defined by Tsoumakas and Katakis (2006). Cardinality is the average number of codes assigned to records in the dataset. Density is the cardinality divided by the total number of codes. For both training sets, the number of labels is of the same order as the number of records, and the label density is extremely low. This confirms that the task of code assignment belongs to the family of extreme multi-label classification.

We did not filter any ICD code based on their frequency. We note, however that there are approximately 1,000 frequent labels (defined as assigned to at least 50 records) (Table 1). We experimented with two versions of the label set: one with all the labels, and one with the labels rolled up to their 3-digit equivalent. On the MIMIC III dataset, this resulted in 1,047 rolled-up ICD codes (compared to the 6,527 5-digit codes).

3.3 Input Texts

Tokenization. Preprocessing of the input records comprised the following steps: (i) tokenize all input texts using spaCy library [1]; (ii) convert all non-alphabetical characters to pseudo-tokens (e.g., “11/2/1986” was mapped to “dd/d/dddd”); (iii) build the vocabulary as tokens that appear at least 5 times in the training set; and (iv) map any out-of-vocabulary word to its nearest word in the vocabulary (using the edit distance). This step is simple, yet particularly useful in reducing the number of misspellings of medical terms. These preprocessing steps has a strong impact on the vocabulary. For instance, there were 1,005,489 unique tokens in MIMIC III and test set before preprocessing, and only 121,595 remaining in the vocabulary after preprocessing (an 88% drop).

Hierarchical Segmentation. Besides tokenization of the input texts, we carried one more level of segmentation, at the sentence level (using the spaCy library as well). There are two reasons for preprocessing the input texts with sentence segmentation. First, because we deal with long documents, it is impossible and ineffective to train a sequence model like an GRU on such long sequences. In previous approaches in document classification, this problem was resolved by truncating the input documents. In the case of discharge summaries, however, this is not an acceptable solution: we want to preserve the entire document for transparency. Second, we are inspired by the moving windows of Johnson and Zhang (2014) and posit that sentences form linguistically inspired windows of word sequences.

Beyond tokens and sentences, discharge summaries exhibit strong discourse-level structure (e.g., history of present illness and past medical history, followed by hospital course, and discharge plans) Li et al. (2010). This presents an exciting opportunity for future work to exploit discourse segments as an additional representation layer of input texts.

[1] https://spacy.io/

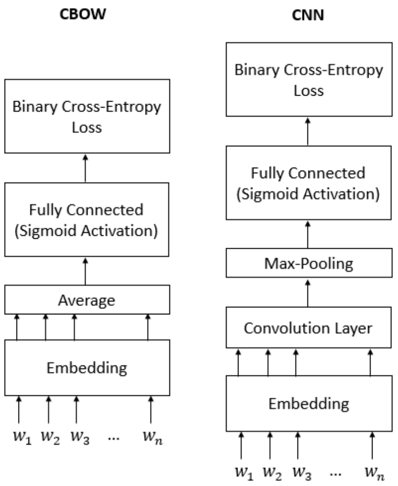
CBOW
Binary Cross-Entropy Loss
Fully Connected (Sigmoid Activation)
Average
Embedding
w_1 w_2 w_3 ... w_n
CNN
Binary Cross-Entropy Loss
Fully Connected (Sigmoid Activation)
Max-Pooling
Convolution Layer
Embedding
w_1 w_2 w_3 ... w_n

Figure 1: (a) CBOW architecture and (b) CNN model architecture.

4. Methods

We describe the four models we experimented with. ICD coding has been evaluated in the literature according to different metrics: Micro-F, Macro-F, a variant of Macro-F that takes into account the hierarchy of the codes (Perotte et al., 2014), Hamming and ranking loss (Wang et al., 2016), and a modified version of mean reciprocal rank (MRR) (Subotin and Davis, 2014). We evaluate performance using the Macro-F metric, since it is the most commonly used metric.

SVM. We used Scikit Learn (Pedregosa et al., 2011) to implement a one-vs-all, multi-label binary SVM classifier. Features were bag of words, with tf*idf weights for each label. Stop words were removed using Scikit Learn default English stop-word list. The model fits a binary SVM classifier for each label (ICD code) against the rest of the labels. We also experimented with χ^2 feature filtering to select the top-N words according to their mutual information with each label, but this did not improve performance.

CBOW. The continuous-bag-of-words (CBOW) model is inspired by the word2vec CBOW model (Mikolov et al., 2013) and FastText (Joulin et al., 2016). Both methods use a simple neural-network to create a dense representation of words and use the average of this representation for prediction. The word2vec CBOW tries to predict a word from the words that appear around it, while our CBOW model for ICD classification predicts ICD9 codes from the words of its input discharge summary.

The model architecture consists of an embedding layer applied to all the words in a given input text $[w_1, w_2, ..., w_n]$, where w_i is a one-hot encoding vector of the vocabulary. E is the embedding matrix with dimension $n_{emb} \times V$, where V is the size of the vocabulary and n_{emb} is the embedding size (set to 100).

The embedded words are averaged into a fixed-size vector and are fed to a fully connected layer with a matrix W and bias b, where the output dimension is the number of labels. We use a sigmoid activation on the output layer so all values are in the range of $[0-1]$ and use a fixed threshold to determine whether to assign a particular label. To train the model, we used binary cross-entropy loss ($loss(target, output) = -(target \cdot \log(output) + (1 - target) \cdot \log(1 - output))$.

$$\begin{aligned} Embedding &= E \cdot [w_1, w_2, ..., w_n] \\ Averaged &= 1/n\Sigma_{e \in Embedding}(e) \\ Prob &= sigmoid(W \cdot Averaged + b) \end{aligned}$$

While the model is extremely lightweight and fast it suffers from known bag-of-words issues: (i) it ignores word order; i.e., if negation will appear before a diagnosis mention, the model would not be able to learn this; (ii) multi-word-expressions cannot be identified by the model, so different diagnoses that share lexical words will not be distinguished by the model.

CNN. To address the problems of the CBOW model, the next model we investigate is a convolutional neural network (CNN). A one dimensional convolution applied on list of embedded words could be considered as a type of n-gram model, where n is the convolution filter size.

The architecture of this model is very similar to the CBOW model, but instead of averaging the embedded words we apply a one dimensional convolution layer with filter f, followed by a max pooling layer. On the output of the max pool layered a fully connected layer was applied, like in the CBOW model. We also experimented with deeper convolution networks and inception module (LeCun, 2015), but they did not yield improved results.

$$\begin{aligned} Embedding &= E \cdot [w_1, w_2, ..., w_n] \\ Conved &= \max_{i \in channels}(Embedding * f) \\ Prob &= sigmoid(W \cdot Conved + b) \end{aligned}$$

In our experiments, we used the same embedding parameter as in the CBOW model. In addition, we set the number of channels to 300, and the filter size to 3.

HA-GRU. We now introduce the Hierarchical-Attention GRU model (HA-GRU) an adaptation of a Hierarchical Attention Networks (Yang et al., 2016) to be able to handle multi-label classification. A Gated Recurrent Unit (GRU) is a type of Recurrent Neural Network. Since the documents are long (up to 13,590 words per doc), a regular GRU applied over the entire document is too slow as it requires number layers of the document length. Instead we apply a hierarchal model with two levels of GRU encoding. The first GRU operates over tokens and encodes sentences. The second GRU encodes the document, applied over all the encoded sentences. In this architecture, each GRU is applied to a much shorter sequence compared with a single GRU.

To take advantage of the property that each label is invoked from different parts of the text, we use an attention mechanism over the second GRU with different weights for each label. This allows the model to focus on the relevant sentences for each label (Choi et al., 2016). To allow clarity into what the model learns and enable error analysis attention is also applied over the first GRU with the same weights for all the labels.

Each sentence in the input text is encoded to a fixed length vector by applying an embedding layer over all the inputs, applying a GRU layer on the embedded words, and using a neural attention mechanism to encode the GRU outputs. After the sentences are

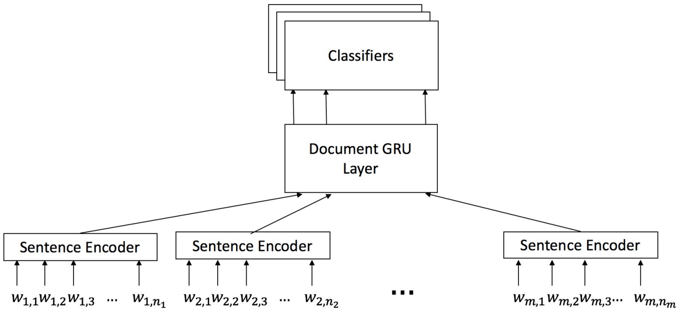
Classifiers
Document GRU Layer
Sentence Encoder
Sentence Encoder
Sentence Encoder
$w_{1,1} w_{1,2} w_{1,3}$ ... w_{1,n_1}
$w_{2,1} w_{2,2} w_{2,3}$ ... w_{2,n_2}
...
$w_{m,1} w_{m,2} w_{m,3}$... w_{m,n_m}

Figure 2: HA-GRU model architecture overview.

encoded into a fixed length vector, we apply a second GRU layer over the sentences using different attention layers to generate an encoding specified to each class. Finally we applied a fully connected layer with softmax for each classifier to determine if the label should be assigned to the document. Training is achieved by using categorical cross-entropy on every classifier separately $(loss(target, output) = -\sum_x ouput(x) \cdot log(target(x)))$

$$\begin{aligned}
AttentionWeight(input_i, v, w) &= v \cdot tanh(w \cdot (input_i)) \\
\overline{AttentionWeight}(input_i, v, w) &= \frac{e^{AttentionWeight(input_i, v, w)}}{e^{\sum_j AttentionWeight_j(v, w)}} \\
attend(input, v, w) &= sum(input_i \cdot \overline{AttentionWeight}(input_i, v, w)) \\
Embedding &= E \cdot [w_1, w_2, ..., w_n] \\
EncodedSents_j &= Attend(GRUwords(Embedding), v_{words}, w_{words}) \\
EncodedDoc_{label} &= Attend(GRU_{sents}(EncodedSents, v_{label}, w_{label}),) \\
Prob_{label} &= softmax(pw_{label} \cdot EncodedDoc_{label} + pb_{label})
\end{aligned}$$

Where w_i is a one-hot encoding vector of the vocabulary size V, E is an embedding matrix size of $n_{emb} \times V$, GRU_{words} is a GRU layer with state size h_{state}, w_{words} is a square matrix $(h_{state} \times h_{state})$ and v_{words} is a vector (h_{state}) for the sentence level attention. GRU_{sents} is a GRU layer with state size of h_{state}. w_{label} is a square matrix $(h_{state} \times h_{state})$ and v_{label} is a vector (h_{state}) for the document level attention for each class, pw_{label} is a matrix $(h_{state} \times 2)$ and pb_{label} is a bias vector with a size of 2 for each label. We implemented the model using DyNetNeubig et al. (2017)[2]

In our experiments, we learned the HA-GRU model for the smaller label set of 1,047 rolled-up 3-digit ICD9 codes only - yielding 1,047 independent binary classifiers. The embedding size n_{emb} was set to 50 and both GRUs state size h_{state} was set to 50.

5. Results

5.1 Model Comparison

To evaluate the proposed methods on the MIMIC datasets, we conducted the following experiments. In the first setting we considered all ICD9 codes as our label set. We trained the SVM, CBOW, and CNN on the MIMIC II and on the MIMIC III training sets separately. All models were evaluated on the same test set according to Macro-F. In the second setting, we only considered the rolled-up ICD9 codes to their 3-digit codes. There, we trained all

[2] Code found here: `https://github.com/talbaumel/MIMIC`

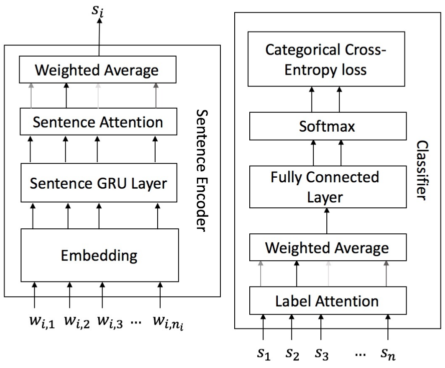

Figure 3: Zoom-in of the sentence encoder and classifier.

	ICD9 codes		Rolled-up ICD9 codes	
	MIMIC II	MIMIC III	MIMIC II	MIMIC III
SVM	28.13%	22.25%	32.50%	53.02%
CBOW	30.60%	30.02%	42.06%	43.30%
CNN	33.25%	**40.72%**	46.40%	52.64%
HA-biGRU	**36.60%**	40.52%	**53.86%**	**55.86%**

Table 2: Macro-F on two settings (full and rolled-up ICDs) for both training sets.

models, including the HA-GRU model (Table 2). The HA-GRU was not tested on the full ICD9 due to lack of resources (time and memory) to train such a model.

HA-GRU gives the best results in the rolled-up ICD9 setting, with a 3.6% and 3% improvement over the SVM, the second best method, in MIMIC II and MIMIC III respectively. In the full ICD-9 scenario, all methods yield better results when trained on MIMIC III rather than on MIMIC II. This is expected considering the larger size of MIMIC III over II. We note that our SVM yields the best Macro-F when trained on MIMIC II, while CNN surpasses it by 2.13% when trained on MIMIC III.

In comparison to the previous work of Perotte et al. (2014), our one-vs-all SVM yielded better results than their flat and hierarchy classifiers. This trend was confirmed when training on the new MIMIC III set, as well as when using the same evaluation metrics of Perotte et al. (2014). We attribute these improved results both to the one-vs-all approach as well as our tokenization approach.

5.2 Model Explaining Power

We discuss how the CNN and HA-GRU architectures can support model explaining power.

CNN. To analyze the CNN prediction we can test which n-grams triggered the max-pooling layer. Given a sentence with n words we can feed it forward through the embedding layer and the convolution layer. The output of the convolution a list of vectors each the size of the number of channels of the convolution layer where vector corresponds to an n-gram.

We can identify what triggered the max pooling layer by finding the maximum value of each channel. Thus, for predicted labels, one of the activated n-grams does include information relevant for that label (whether correct for true positive labels or incorrect for false positive labels). For example in our experiments, for the label: *"682.6-Cellulitis and abscess of leg, except foot"* one of the activated n-gram detected was *"extremity cellulitis prior"*.

This transparence process can also be useful for error analysis while building a model, as it can highlight True Positive and False Positive labels. However, it is difficult in the CNN to trace back the decisions for False Negatives predictions.

HA-GRU For the HA-GRU model we can use attention weights to better understand what sentences and what words in that sentence contributed the most to each decision. We can find which sentence had the highest attention score for each label, and given the most important sentence, we can find what word received the highest attention score. For example, in our experiments for label *"428-Heart failure"* we found that the sentence with the highest attention score was *"d . congestive heart failure (with an ejection fraction of dd % to dd %) ."*, while the token *"failure"* was found most relevant across all labels.

Like in the CNN, we can use this process for error analysis. In fact, the HA-GRU model explains prediction with greater precision, at the sentence level. For instance, we could explore the following False Positive prediction: the model assigned the label *"331-Other cerebral degenerations"* to the sentence:*"alzheimer 's dementia ."*. We can see that the condition was relevant to the medical note, but was mentioned under the patient's past medical history (and not a current problem). In fact, many of the False Positive labels under the HA-GRU model were due to mentions belonging to the past medical history section. This suggests that the coding task would benefit from a deeper architecture, with attention to discourse-level structure.

In contrast to the CNN, the HA-GRU model can also help analyze False Negative label assignments. When we explored the False Negative labels, we found that in many cases the model found a relevant sentence, but failed to classify correctly. This suggests the document-level attention mechanism is successful. For instance, for the False Negative *"682-Other cellulitis and abscess"*, the most attended sentence was *"... for right lower extremity cellulitis prior to admission ..."*. The false positive codes for this sentence included *"250-Diabetes mellitus"* and *"414-Other forms of chronic ischemic heart disease"*. We note that in the case of cellulitis, it is reasonable that the classifier preferred other, more frequent codes, as it is a common comorbid condition in the ICU.[3]

6. Conclusion

We investigate four modern models for the task of extreme multi-label classification on the MIMIC datasets. Unlike previous work, we evaluate our models on all ICD9 codes thus making sure our models could be used for real world ICD9 tagging. The tokenization step, mapping rare variants using edit distance, improved results for all models by 0.5%, highlighting the importance of preprocessing data noise problems in real-world settings. The HA-GRU model not only achieves the best performance on the task (53% $F1$ on MIMIC III, 1% absolute improvement on the best SVM baseline) but is able to provide insight on the

[3] Visualizer example: `https://www.cs.bgu.ac.il/~talbau/mimicexample.html`

task for future work such as using discourse-level structure available in medical notes yet never used before. The ability to highlight the decision process of the model is important for adoption of such models by medical experts. On the sub-task of MIMIC II, which includes a smaller training dataset, HA-GRU achieved 5% absolute $F1$ improvement, suggesting it requires less training data to achieve top performance, which is important for domain adaptation efforts when applying such models to patient records from other sources (such as different hospitals).

acknowledgments

This work is supported by National Institute of General Medical Sciences Grant R01GM114355.

References

Kush Bhatia, Himanshu Jain, Purushottam Kar, Manik Varma, and Prateek Jain. Sparse local embeddings for extreme multi-label classification. In *Advances in Neural Information Processing Systems (NIPS)*, pages 730–738, 2015.

Edward Choi, Mohammad Taha Bahadori, and Jimeng Sun. Doctor AI: Predicting clinical events via recurrent neural networks. *arXiv preprint arXiv:1511.05942*, 2015.

Eunsol Choi, Daniel Hewlett, Alexandre Lacoste, Illia Polosukhin, Jakob Uszkoreit, and Jonathan Berant. Hierarchical question answering for long documents. *arXiv preprint arXiv:1611.01839*, 2016.

Koby Crammer, Mark Dredze, Kuzman Ganchev, Partha Pratim Talukdar, and Steven Carroll. Automatic code assignment to medical text. In *Proceedings of the ACL Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, pages 129–136, 2007.

Dina Demner Fushman and Noemie Elhadad. Aspiring to unintended consequences of natural language processing: A review of recent developments in clinical and consumer-generated text processing. *Yearbook of Medical Informatics*, 10(1):224–233, 2016.

Richárd Farkas and György Szarvas. Automatic construction of rule-based ICD-9-CM coding systems. *BMC bioinformatics*, 9(3):S10, 2008.

Hrayr Harutyunyan, Hrant Khachatrian, David C Kale, and Aram Galstyan. Multitask learning and benchmarking with clinical time series data. *arXiv preprint arXiv:1703.07771*, 2017.

Abhyuday N Jagannatha and Hong Yu. Structured prediction models for RNN based sequence labeling in clinical text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, page 856, 2016.

Himanshu Jain, Yashoteja Prabhu, and Manik Varma. Extreme multi-label loss functions for recommendation, tagging, ranking & other missing label applications. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 935–944, 2016.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3, 2016.

Rie Johnson and Tong Zhang. Effective use of word order for text categorization with convolutional neural networks. *arXiv preprint arXiv:1412.1058*, 2014.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*, 2016.

Ramakanth Kavuluru, Anthony Rios, and Yuan Lu. An empirical evaluation of supervised learning approaches in assigning diagnosis codes to electronic medical records. *Artificial Intelligence in Medicine*, 65(2):155–166, 2015.

Leah S Larkey and W Bruce Croft. Automatic assignment of ICD9 codes to discharge summaries. Technical report, University of Massachusetts at Amherst, Amherst, MA, 1995.

Yann LeCun. LeNet-5, convolutional neural networks. `http://yann.lecun.com/exdb/lenet`, 2015.

Ying Li, Sharon Lipsky Gorman, and Noemie Elhadad. Section classification in clinical notes using supervised hidden Markov model. In *Proceedings of the 1st ACM International Health Informatics Symposium*, pages 744–750. ACM, 2010.

Zachary C Lipton, David C Kale, Charles Elkan, and Randall Wetzell. Learning to diagnose with LSTM recurrent neural networks. *arXiv preprint arXiv:1511.03677*, 2015.

Lucian Vlad Lita, Shipeng Yu, Radu Stefan Niculescu, and Jinbo Bi. Large scale diagnostic code classification for medical patient records. In *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP)*, pages 877–882, 2008.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Riccardo Miotto, Li Li, Brian A Kidd, and Joel T Dudley. Deep Patient: An unsupervised representation to predict the future of patients from the electronic health records. *Scientific Reports*, 6, 2016.

Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, et al. Dynet: The dynamic neural network toolkit. *arXiv preprint arXiv:1701.03980*, 2017.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

Adler Perotte, Rimma Pivovarov, Karthik Natarajan, Nicole Weiskopf, Frank Wood, and Noémie Elhadad. Diagnosis code assignment: models and evaluation metrics. *Journal of the American Medical Informatics Association*, 21(2):231–237, 2014.

Adler J Perotte, Frank Wood, Noemie Elhadad, and Nicholas Bartlett. Hierarchically supervised latent dirichlet allocation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2609–2617, 2011.

John P Pestian, Christopher Brew, Paweł Matykiewicz, Dj J Hovermale, Neil Johnson, K Bretonnel Cohen, and Włodzisław Duch. A shared task involving multi-label classification of clinical free text. In *Proceedings of the ACL Workshop on BioNLP: Biological, Translational, and Clinical Language Processing*, pages 97–104, 2007.

Rimma Pivovarov, Adler J Perotte, Edouard Grave, John Angiolillo, Chris H Wiggins, and Noémie Elhadad. Learning probabilistic phenotypes from heterogeneous EHR data. *Journal of Biomedical Informatics*, 58:156–165, 2015.

Rajesh Ranganath, Adler J Perotte, Noémie Elhadad, and David M Blei. The Survival Filter: joint survival analysis with a latent time series. In *UAI*, pages 742–751, 2015.

Mohammed Saeed, Mauricio Villarroel, Andrew T Reisner, Gari Clifford, Li-Wei Lehman, George Moody, Thomas Heldt, Tin H Kyaw, Benjamin Moody, and Roger G Mark. Multiparameter intelligent monitoring in intensive care II (MIMIC-II): a public-access intensive care unit database. *Critical Care Medicine*, 39(5):952, 2011.

Michael Subotin and Anthony R Davis. A system for predicting ICD-10-PCS codes from electronic health records. In *Proceedings of the ACL Workshop on Biomedical Natural Language Processing (BioNLP)*, pages 59–67. Citeseer, 2014.

Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3), 2006.

Sen Wang, Xiaojun Chang, Xue Li, Guodong Long, Lina Yao, and Quan Z Sheng. Diagnosis code assignment using sparsity-based disease correlation embedding. *IEEE Transactions on Knowledge and Data Engineering*, 28(12):3191–3202, 2016.

Jason Weston, Samy Bengio, and Nicolas Usunier. WSABIE: Scaling up to large vocabulary image annotation. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, volume 11, pages 2764–2770, 2011.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of NAACL-HLT*, pages 1480–1489, 2016.

Ian EH Yen, Xiangru Huang, Kai Zhong, Pradeep Ravikumar, and Inderjit S Dhillon. PD-Sparse: A primal and dual sparse approach to extreme multiclass and multilabel classification. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2016.