Grounded Recurrent Neural Networks

Ankit Vani* **Yacine Jernite†** **David Sontag‡**
*†CIMS, New York University, New York, NY 10012
‡CSAIL & IMES, Massachusetts Institute of Technology, Cambridge, MA 02139
*ankit.vani@nyu.edu, †jernite@cs.nyu.edu, ‡dsontag@csail.mit.edu

Abstract

In this work, we present the Grounded Recurrent Neural Network (GRNN), a recurrent neural network architecture for multi-label prediction which explicitly ties labels to specific dimensions of the recurrent hidden state (we call this process "grounding"). Additionally, structural constraints on the transition matrices help the model remain tractable as the label space grows. This approach is particularly well-suited for extracting large numbers of concepts from text in the presence of limited data. We apply the new model to address an important problem in healthcare of understanding what medical concepts are discussed in clinical text. Our evaluation shows an advantage in performance, data efficiency and interpretability to using our proposed architecture over a variety of strong baselines.

1 Introduction

The ability of recurrent neural networks to model sequential data and capture long-term dependencies makes them powerful tools for natural language processing. These models maintain a state at each time step, representing the relevant history and task-specific beliefs. Based on the current value of this recurrent state and a new input, the state is updated at each time step. Recurrent models have become a popular choice for a variety of natural language processing tasks such as language modeling [Mikolov et al., 2010], text classification [Graves, 2012], or machine translation [Cho et al., 2014a]. The success of this paradigm has been driven in great part by a number of structural innovations since the original version of Elman [1990]. Recurrent cells such as the Long Short Term Memory (LSTM) [Hochreiter and Schmidhuber, 1997] or Gated Recurrent Units (GRU) [Cho et al., 2014b], for example, alleviate the problem of vanishing gradients [Bengio et al., 1994]. Attention mechanisms [Bahdanau et al., 2014] and Memory Networks [Sukhbaatar et al., 2015] have also significantly increased the expressiveness of recurrent architectures, revealing their potential to tackle more complex tasks such as question answering [Rajpurkar et al., 2016]. One notable property of these models, however, is that they often require significant amounts of training data to perform at their best [Bajgar et al., 2016], which can limit their application domain.

In this work, we focus on developing recurrent models for the task of extracting medical concepts from Intensive Care Unit discharge summaries. This is a multi-class, multi-label text classification task with a target vocabulary of several thousand concepts. Given the difficulty to obtain very large medical datasets, there is a need to come up with new, more data-efficient architectures. To this end, we introduce the Grounded Recurrent Neural Network (GRNN). At a high level, we *ground* the model's hidden state by introducing dimensions whose sole purpose is to track the model's belief in the presence of specific labels for the current example. We find that this addition aids optimization, and outperforms standard recurrent models for text classification. Although each new label adds to the hidden state size, we impose a semi diagonal constraint on the recurrent transition matrices, so that the size of our model grows linearly with the number of labels. We show that this not only lets the model scale with the number of labels, but also helps with optimization. Furthermore, our approach leads to increased interpretability, which is especially appreciated in medical applications. Indeed, even though we do not provide our model with the location of phrases of interest for a label

at training time, we can track changes in the grounded dimensions tied to a specific concept as a document is read, indicating evidence in text for or against its presence when the model's belief changes drastically.

We evaluate our model on the publicly available MIMIC datasets [Saeed et al., 2011, Johnson et al., 2016] to predict ICD9 (International Classification of Diseases [Organization et al., 1978]) codes given a patient's discharge summary text [Perotte et al., 2014, Lita et al., 2008]. These codes are usually determined by humans perusing health records and selecting relevant codes from very long lists. Due to the high number of ICD9 codes, there is significant human error, arising from the cognitive load of such a task [Birman-Deych et al., 2005, Hsia et al., 1988] and differences in human judgment [Pestian et al., 2007]. The effort needed, the errors in the coding process and the inconsistency of labeling can be mitigated through automatically detecting concepts in text or offering suggestions as smarter auto-complete, which motivates our contribution. We also show our model's performance on a tag prediction dataset built from StackOverflow data.

Section 2 presents relevant previous work, Section 3 describes the model, and we present experimental results in Section 4. Section 5 concludes and outlines possible future research directions.

2 Related Work

Entity Tracking The idea to keep information relating to specific concepts in dedicated cells was inspired in part by the work of Henaff et al. [2016] on recurrent entity networks. They propose to define one full RNN per entity of interest which is tasked with keeping track of all of the relevant information as a story is read, and their model is able to answer questions about the state of the world by consulting all of their final hidden states. Such an approach is unfortunately impractical for our case, which presents thousands of concepts of interest. Instead, we use one dimension in the recurrent hidden state for each of them to track one specific piece of information: the likelihood that they are present in the current example.

Sparse Recurrent Units Subakan and Smaragdis [2017] use diagonal recurrent transition matrices for music modeling and show that their model outperforms the the one with full transition matrices. The upper-left block of the transition matrix in our model is diagonal. Narang et al. [2017] explored sparsity in recurrent models, and showed that sparse weight matrices result in much smaller models without significant loss in accuracy. The weight sparsity introduced in our model through a semi diagonal weight matrix additionally limits over-fitting, as discussed in Section 3.

Interpretable RNNs Lei et al. [2016] propose a method for encouraging interpretability in the learning objective by selecting predictive phrases as the first step. Additionally, Lample et al. [2016] discuss recurrent models followed by a pairwise conditional random field (CRF) for named entity recognition. Their LSTM+CRF model is used to predict word tags that determine the span a word belongs to and the entity the span represents. They also present a transition-based chunking model which uses a stacked LSTM where words are pushed onto a stack and popped with an entity label.

Medical Concept Extraction Joshi et al. [2016] use non-negative matrix factorization for simultaneous phenotyping of co-occurring medical conditions from clinical text. They obtain identifiable sparse latent factors by grounding them to have a one-to-one mapping with a fixed set of chronic conditions. Our model grounds its recurrent hidden state dimensions to have a one-to-one mapping with the labels for a task. Automatic ICD9 coding has been explored in clinical machine learning literature, such as Perotte et al. [2014]. Their work presents a tree structured Support Vector Machine (SVM) which takes advantage of the hierarchical nature of ICD9 codes to outperform a flat baseline.

3 Grounded Recurrent Neural Networks

In this section, we show how to derive a Grounded Recurrent Neural Network (GRNN) architecture given a label set $\mathcal{L}$. Note that while we decide to build our version of GRNN on top of a Gated Recurrent Unit, similar ideas can be applied to add grounding to other types of recurrence functions, such as the Elman or LSTM unit.

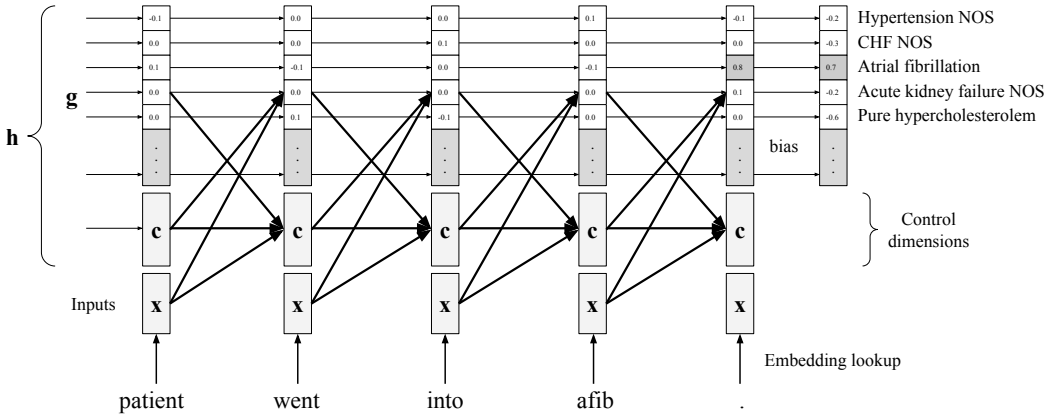

Figure 1: The Grounded RNN architecture. At each time step, the model's belief in the presence of a concept of interest (stored in the corresponding dimension of $\mathbf{g}$) is updated based on the control part of the recurrent state $\mathbf{c}$ and the inputs $\mathbf{x}$.

Sequence Labeling with GRUs Let $\mathcal{L}$ be a set of labels of interest. Consider a text sequence $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$ and a label assignment for the sequence denoted by $\mathbf{y} \in \{0,1\}^{|\mathcal{L}|}$. For ease of exposition, we identify the words $(\mathbf{x}_1, \ldots, \mathbf{x}_T)$ with their embeddings of dimension D_e. The task of sequence labeling consists in predicting $\mathbf{y}$ given $\mathbf{x}$. To that end, we can use a Recurrent Neural Network to obtain a vector representation of the text of dimension D_h, then compute the likelihood of each label being present given that representation. More specifically, the recurrent model starts with a representation $\mathbf{h}_0$, and updates it at each timestep by using a recurrence function f to obtain the global sequence representation $\mathbf{h}_T$:

$$\forall t \in \{1, \ldots, T\}, \quad \mathbf{h}_t = f(\mathbf{h}_{t-1}, \mathbf{x}_t) \tag{1}$$

In the case of the Gated Recurrent Unit, which we build upon in this work, the recurrence function f is parameterized by the $D_h \times (D_e + D_h)$ matrices $\mathbf{Z}$, $\mathbf{R}$ and $\mathbf{W}$ (and dimension D_h bias vectors $\mathbf{b}_z$, $\mathbf{b}_r$ and $\mathbf{b}_w$), and computed as follows. Let $[\mathbf{a}, \mathbf{b}]$ denote the concatenation of vectors $\mathbf{a}$ and $\mathbf{b}$, and $\odot$ be the element-wise product. Then, $\mathbf{h}_t = f(\mathbf{h}_{t-1}, \mathbf{x}_t)$ is given by:

$$\mathbf{z}_t = \sigma\left(\mathbf{Z}[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_z\right) \tag{2}$$

$$\mathbf{r}_t = \sigma\left(\mathbf{R}[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_r\right) \tag{3}$$

$$\tilde{\mathbf{h}}_t = \tanh\left(\mathbf{W}[\mathbf{r}_t \odot \mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_w\right) \tag{4}$$

$$\mathbf{h}_t = (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t \tag{5}$$

From the final text representation $\mathbf{h}_T$, we can obtain a prediction vector $\tilde{\mathbf{y}} \in [0,1]^{|\mathcal{L}|}$ by applying an affine transformation followed by a sigmoid function, parameterized by a $|\mathcal{L}| \times D_h$ matrix $\mathbf{P}$ and bias vector $\mathbf{b}_p$. The model parameters Θ can then be learned by minimizing the expected sum of the binary cross-entropy loss for each (text, label set) pair, denoted as L:

$$\forall (\mathbf{x}, \mathbf{y}), \quad \tilde{\mathbf{y}} = \sigma(\mathbf{P}\mathbf{h}_T + \mathbf{b}_p) \qquad \text{and} \qquad L(\mathbf{x}, \mathbf{y}; \Theta) = -\sum_{l=1}^{|\mathcal{L}|} y_l \log(\tilde{y}_l) \tag{6}$$

Grounded Dimensions At a high level, the above approach corresponds to having a model summarize all of the relevant information from a text sequence in the final recurrent state $\mathbf{h}_T$, such that each label corresponds to a different sub-space. This presents several challenges. On the one hand, if the dimension of the recurrent space is much smaller than the label space size, the model capacity might be too small to store the required information about all of the labels in the target set. On the other hand, too large a recurrent space, while making the model more expressive, can make it prone

to over-fitting, rendering optimization difficult when training data is limited. In addition, even though GRUs and LSTMs are better than standard Elman units at modeling long term dependencies, it can still be challenging for them to maintain relevant information from the very beginning of longer sequences (up to a few thousand words in many applications). Our goal in this work is to alleviate the aforementioned problems by adding grounded dimensions to the model's recurrent space.

We split the recurrent state $\mathbf{h}$ into $|\mathcal{L}|$ *grounded* dimensions $\mathbf{g}$ and D_c *control* dimensions $\mathbf{c}$. At each time step, the value stored in a grounded dimension g_l corresponds to the model's current belief that $y_l = 1$. Since by construction we have $\mathbf{g} \in [-1, 1]^{|\mathcal{L}|}$, the label predictions $\tilde{\mathbf{y}}$ can simply be obtained by scaling and shifting the final grounded state $\mathbf{g}_T$. However, we also found it useful to use a bias term to allow grounded dimensions to be centered on 0 regardless of the corresponding label frequency in the data. With this rescaling, and keeping with the notations introduced in the previous paragraph, the model predictions are then given by:

$$\tilde{\mathbf{y}} = \sigma\big(\sigma^{-1}\left(\frac{\mathbf{g}_T + 1}{2}\right) + \mathbf{b}_p\big) \tag{7}$$

Figure 1 illustrates this process as the model reads the end of a medical note: after reading the phrase "patient went into afib", the model increases its belief that this person was diagnosed with Atrial Fibrillation. This formulation already presents some advantages. For example, dedicated dimensions can make learning of long-term dynamics easier. However, simply applying a GRU update to the complete $\mathbf{h} = [\mathbf{g}, \mathbf{c}]$ recurrent state at each time step yields a model with very large capacity ($D_h = |\mathcal{L}| + D_c$), which, as stated previously, can make optimization difficult with limited data. We address this issue in the next paragraph.

Semi Diagonal Updates When given liberty to use the $|\mathcal{L}|$ grounded dimensions without constraints, the model can choose to use them to store other information than label-specific beliefs, especially in the case of dimensions corresponding to rarer concepts. To avoid this potential issue, we propose to restrict the model dynamics by making the $\mathbf{Z}$, $\mathbf{R}$ and $\mathbf{W}$ matrices in Equations 2 to 4 semi-diagonal, as illustrated in Figure 2.

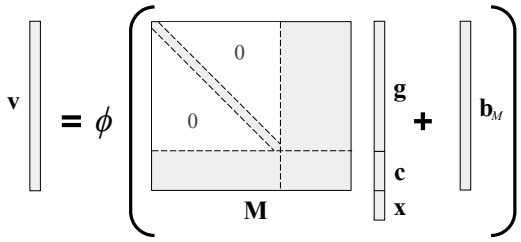

Figure 2: The semi diagonal transition operation used to update $\mathbf{h} = [\mathbf{g}, \mathbf{c}]$. $\mathbf{v}$ corresponds to $\mathbf{z}$, $\mathbf{r}$ or $\tilde{\mathbf{h}}$ from Equations 2, 3 and 4 respectively. $\mathbf{M} \in \{\mathbf{Z}, \mathbf{R}, \mathbf{W}\}$ is a weight matrix and $\mathbf{b}_M$ is the corresponding bias vector.

With this architecture, the value of any g_l is updated at each timestep based solely on its previous state, the current input and control state $\mathbf{c}$, which is then made responsible for modeling correlations. Moreover, the number of parameters of the model and computational cost of each time step with semi diagonal transitions grows linearly in the size of the label space, which allows learning to remain tractable even when $|\mathcal{L}|$ is large.

Bidirectional GRNN Finally, in many tasks, it is common for bidirectional recurrent models to outperform the unidirectional recurrent models, as they allow for context from the future to be taken into consideration at each timestep along with the history. We extend the GRNN to the bidirectional setting by running a standard GRU in the reverse direction on the document, and concatenating the outputs of this GRU to the inputs of the GRNN, allowing modifications of the grounded dimensions to be based on future context as well as past.

In this Section, we presented the Grounded Recurrent Neural Network: a recurrent architecture designed to have significantly higher capacity than comparable models while making optimization easier and improving interpretability. Section 4 analyzes the model's behavior on real world data, and demonstrates these properties experimentally.

4 Experiments

4.1 Experimental Setting

Datasets We evaluate our model on the task of multi-label classification on three datasets: two versions of the MIMIC medical dataset and Stack Overflow questions.

MIMIC-II [Saeed et al., 2011] is a dataset of Intensive Care Unit medical records. Each patient admission ends with a free text discharge summary describing the patient's stay, diagnoses, and procedures performed. Here, we consider the problem of predicting diagnosis codes from the text, learning from tags manually provided by humans after going through the admission records. We follow Perotte et al. [2014] in extending the label set to also consider parents of the gold label codes in the ICD9 hierarchy, which yields a total vocabulary of 7042 ICD9 codes that we use as labels, with 36.7 labels per note on average. We split the data into the same training and test sets as Perotte et al. [2014], and further split their training set into training and validation. This gives us $18{,}822$ training, 1711 validation, and 2282 testing notes. The average sequence length was 1409.6, and the maximum length was truncated at 4000. MIMIC-III [Johnson et al., 2016] is an updated version of the MIMIC-II dataset with thousands more patients and admissions. For this dataset, we predict ICD9 diagnosis as well as procedure codes. We consider the most frequent 4000 diagnoses and 1000 procedures as our label space, giving us about 11.5 diagnoses and 4.4 procedures per note on average. We used a training set of $36{,}998$ notes with 1356 and 2755 notes for validation and testing, respectively. The average length for MIMIC-III discharge summaries is 1720.3.

Stack Overflow[1] is a website which features a large number of computer programming questions and answers. Every question on Stack Overflow has tags defined by the asker. We used a subset of the Stack Overflow data, downloaded from the Kaggle website[2] to evaluate the GRNN on the task of predicting tags from question text. We pre-process the data to remove all code blocks from the questions, and select questions which have more than 100 words left. This gave us $365{,}192$ training, $13{,}390$ validation and $27{,}187$ testing samples. We chose the 4000 most frequent tags as our label set, with 2.9 tags per sample on average. The average sentence length is much shorter at 190.5 words, and we truncate the maximum length to 600.

Baselines We first compare the GRNN with a Bag-of-Words baseline, where independent binary classifiers are trained via $L1$-regularized logistic regression for each label, with early stopping based on validation loss. The regularization parameters are tuned independently for each label using the validation sets, making this an especially strong baseline for datasets where certain words or sets of words are highly predictive of the labels.

Considering recent advances in convolutional models for text [Dauphin et al., 2017, Gehring et al., 2017] and attention-based models [Bahdanau et al., 2014], we also devise a convolutional approach that uses soft attention per embedding dimension over the words in a note. This method, called the Attention Bag-of-Words, uses the attention scores for each word to add their embeddings to obtain the note representation $\mathbf{h}$. The attention scores are based on the local neighborhoods of the words, and are shared by all labels. We obtain predictions and learn the model parameters as described in Section 3 (replacing $\mathbf{h}_T$ with $\mathbf{h}$ in Equation 6).

The GRU baseline is as defined in Section 3, with a hidden size of either 128, or a larger dimension corresponding to a GRU with the same number of parameters as the corresponding GRNN (846 for 5000 labels, 793 for 4000). This allows us to test whether the difference in our model's performance is due to grounding or simply comes from an increased capacity. We also consider bidirectional GRUs with a dimension 64 hidden state. For all our neural approaches, we used a word embedding size of 192.

Furthermore, for the MIMIC-II dataset, we compare our results to those of Perotte et al. [2014], which uses a flat and a hierarchical SVM for the same task. For each example, the hierarchical SVM node decision for an ICD9 code is trained only if its parent code is positive, and a child code is evaluated only if its parent is classified as positive during testing. The flat SVM predicts all the leaf ICD9 codes independently and builds the extended predictions according to the hierarchy. Unlike Perotte et al. [2014], we learn and predict on the entire extended label set without considering the ICD9 hierarchy,

[1] `https://stackoverflow.com/`

[2] `https://www.kaggle.com/stackoverflow/stacksample`

Model	F1		AUC(PR)		AUC(ROC)		P@n		R@n	
	Micro	Macro	Micro	Macro	Micro	Macro	8	40	8	40
Flat SVM*	0.276	-	-	-	-	-	-	-	-	-
Hier. SVM*	0.395	-	-	-	-	-	-	-	-	-
Logistic	0.523	0.042	0.541	0.125	0.919	0.704	0.913	0.572	0.169	0.528
Attn BoW	0.520	0.027	0.521	0.079	0.975	0.807	0.912	0.549	0.169	0.508
GRU-128	0.512	0.027	0.523	0.082	**0.976**	**0.827**	0.906	0.541	0.168	0.501
BiGRU-64	0.485	0.021	0.493	0.071	0.973	0.811	0.892	0.522	0.165	0.483
GRNN-128	**0.580**	0.052	0.587	0.126	**0.976**	0.815	**0.930**	0.592	**0.172**	0.548
BiGRNN-64	0.578	**0.054**	**0.589**	**0.131**	0.975	0.798	0.925	**0.596**	**0.172**	**0.552**

Table 1: Results on MIMIC-II, 7042 labels. * Lines are taken from [Perotte et al., 2014].

letting our learning algorithm infer relevant label correlations. However, we also note that the authors of the baseline SVM models could have obtained better results by fine-tuning their regularization parameters.

Other Comparisons We ran early experiments with grounded models without a semi diagonal constraint on the recurrent transition matrices, and found that such models failed to generalize for large label spaces due to over-fitting. We also tried grounded recurrent models without any control dimensions, which always performed worse, since we deny the model the capacity to track history beyond current beliefs in labels. In addition, we investigated a variant of the bidirectional GRU closer to our formulation of the bidirectional GRNN, wherein the outputs of a reverse GRU are concatenated to the inputs for a forward GRU (denoted as BiGRU-l in the supplementary material tables), but we found that it always performed worse than the standard bidirectional GRU described earlier.

Finally, we ran the entity network of [Henaff et al., 2016] on MIMIC-III text with 6 entity RNNs. The keys for the entities are learned globally, enabling the network to learn clusters of related labels, with each entity tracking one such cluster. For predictions, each label performed attention over the entity network blocks to determine its value, enabling each label to focus on the cluster it is a part of. This network took several days to train while performing similar to or worse than the GRU baselines.

Curriculum Learning To maintain the invariance of the grounded dimensions representing the likelihoods of labels at intermediate timesteps, we train the model on truncated documents. A convenient way to do this is to start with small sentence lengths and increase the maximum document length as training progresses. At smaller lengths, this helps learn the overall statistics of labels and any early evidence in text. As the maximum document length is increased, the model learns to attribute labels with evidence in text that appears later on. Since we never go back to a smaller length, this enables the model to fine-tune its predictions as more useful information becomes available. We can also view this strategy as a way of doing curriculum learning, enabling the model to perform well on long documents by initially learning on shorter documents. The initial document length was set to 50, and increased by a factor of 1.35 on every training epoch. Initial experiments without curriculum learning took much longer to train, and performed worse than models trained with curriculum learning. We found that this approach got better results and trained faster for all recurrent models, and thus we decided to use the same strategy for the GRU baselines as well.

4.2 Concept Extraction Results

Quantitative Evaluation Tables 1, 2 and 3 report quantitative results of our model and baselines on MIMIC-II, MIMIC-III, and StackOverflow data respectively. We report the F1 measure, and Area Under the Precision/Recall (AUC(PR)) and ROC (AUC(ROC)) curves. The Micro-averaged versions of the measures correspond to considering any (text, label) pair as an independent prediction, either true or false, and computing the statistics on all of those together. To obtain macro-averaged measures, the statistics are computed independently on each label, then averaged uniformly, regardless of the label frequency in the data. Compared to the micro-averaged versions, macro-averaging puts much more of a weight on the model's ability to accurately predict rare labels. We also consider our model's performance in actual health care applications. Given the specific requirements of the domain, one

Model	F1		AUC(PR)		AUC(ROC)		P@n		R@n	
	Micro	Macro	Micro	Macro	Micro	Macro	8	40	8	40
Logistic	0.431	0.061	**0.451**	**0.151**	0.934	0.739	0.614	**0.254**	0.311	0.646
Attn BoW	0.406	0.057	0.357	0.096	0.970	0.877	0.572	0.234	0.290	0.594
GRU-64	0.359	0.044	0.361	0.106	0.971	0.886	0.537	0.225	0.273	0.571
GRU-128	0.399	0.058	0.374	0.108	0.971	0.880	0.563	0.230	0.286	0.585
GRU-846	0.382	0.057	0.329	0.094	0.963	0.850	0.529	0.216	0.269	0.550
BiGRU-64	0.375	0.052	0.352	0.096	0.968	0.873	0.537	0.221	0.273	0.561
GRNN-128	0.410	0.049	0.409	0.111	**0.974**	**0.889**	0.599	0.251	0.304	0.638
BiGRNN-64	**0.467**	**0.078**	0.437	0.132	0.972	0.874	**0.628**	**0.254**	**0.319**	**0.647**

Table 2: Results on MIMIC-III, 5000 labels.

Model	F1		AUC(PR)		AUC(ROC)		P@n		R@n	
	Micro	Macro	Micro	Macro	Micro	Macro	8	40	8	40
Logistic	0.495	0.157	0.497	0.206	0.951	0.808	0.253	0.059	0.707	0.827
Attn BoW	0.549	0.169	0.517	0.196	**0.990**	0.975	**0.271**	**0.064**	**0.760**	0.897
GRU-128	0.552	0.191	0.532	**0.231**	0.989	0.972	0.263	0.063	0.738	0.880
GRU-793	0.545	**0.204**	0.505	0.215	0.989	0.971	0.263	0.063	0.738	0.882
BiGRU-64	0.554	0.189	0.531	0.221	0.991	**0.976**	0.269	**0.064**	0.753	0.893
GRNN-128	0.539	0.153	0.514	0.190	0.985	0.957	0.265	0.063	0.742	0.884
BiGRNN-64	**0.560**	0.179	**0.536**	0.214	0.989	0.973	**0.271**	**0.064**	**0.760**	**0.902**

Table 3: Results on StackOverflow, 4000 labels.

successful human-in-the-loop strategy consists in using the model scores to show a user the n highest scored labels in a highly multi-class application, or to use these scores to improve an auto-complete system as in Jernite et al. [2013] and Greenbaum et al. [2017]. In that case, it is important to know how many of the proposals are correct (precision at n: P@n in the tables). Indeed, low precision can significantly hurt a user's confidence in the system. Additionally, we want to know how many of the example's labels are covered by the proposed predictions (recall at n: R@n). We provide these measures for $n = 8$ and $n = 40$.

Tables 1 and 2 show that the GRNN performs significantly better on medical data all measures combined. The gap is greater for MIMIC-II, which agrees with our intuition: MIMIC-II has less data, which makes having a data efficient model more important, and the target label space has a hierarchical structure, which makes being able to take advantage of correlations all the more useful. In particular, the advantage of the grounded architecture is most noticeable on the precision and recall at n measures, which correspond to our proposed use case. Finally, we give results on StackOverflow data in Table 3 to show that our model also performs well on a different domain and setting: the dataset is much larger, while the individual example text sequences are significantly shorter. The grounded architectures are on par with the best baselines, and still consistently out-perform them on the P@n and R@n measures.

Model Introspection Figure 3 provides more insight into the properties of the model architecture. The left plot shows that the GRNN AUC(PR) outperforms most baselines regardless of label frequency. We note that the Logistic curve is close to the grounded architectures, which corresponds to the similar macro-averaged PR AUC, as shown in Table 1. Compared to the other models, the GRNN has most of an advantage on concepts in the middle of the frequency spectrum: labels which appear between a few dozen and a few hundred times in the training data. We also investigate our model's data efficiency by training both a GRU and GRNN on subsets of MIMIC-III of increasing size: using 20%, 40% and 60% of the data. While the GRNN always outperforms the standard GRU, the gap is larger the less training data the model has access to, which implies that grounding does indeed allow a recurrent neural network to learn from less data.

Interpretable Predictions We also want our model to provide interpretable predictions. Indeed, better interpretability of the model decisions is as important as improved quantitative performances in a medical setting, where practitioners need to be able to trust the system they use, and to easily query

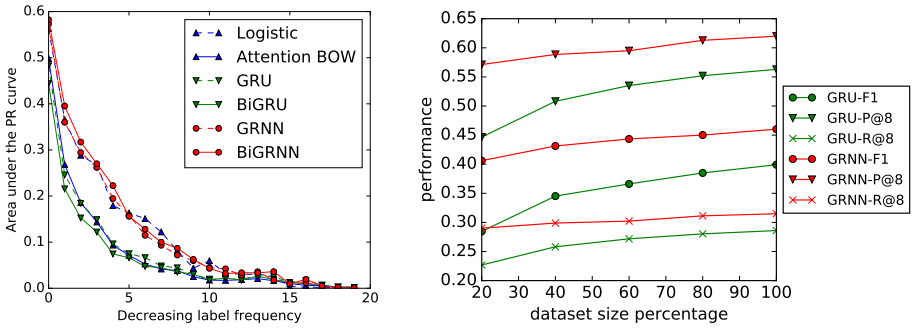

Figure 3: **Left:** Per-label AUC(PR) as a function of the label frequency on MIMIC-II. The labels are ordered from most to least frequent. **Right:** Comparing the performance of the GRU and GRNN by varying the amount of training data available on MIMIC-III.

Dorsopathies	
GRNN: 0.743	**GRU: 0.464**
...back and leg pain major surgical or invasive procedure : posterior lumbar fusion <unk> history of present illness : ms. known patient lastname # has herniated the disc above her previous l4-5 fusion . she has failed conservative therapy and now presents for surgical intervention . past medical history : htn , lumbar spondylosis , stenosis , <unk> herniation s/p <unk> fusion social history : denies family history : n/c physical exam	...back and leg pain major surgical or invasive procedure : posterior lumbar fusion <unk> history of present illness : ms. known patient lastname # has herniated the disc above her previous l4-5 fusion . she has failed conservative therapy and now presents for surgical intervention . past medical history : htn , lumbar spondylosis , stenosis , <unk> herniation s/p <unk> fusion social history : denies family history : n/c physical exam

Coagulation Defects	
GRNN: 0.404	**GRU: 0.421**
...postoperatively the patient required multiple transfusions of crystalloid , <unk> , and blood products including ffp , platelets , and prbc . despite very aggressive electrolyte , blood , and fluid resuscitation the patient continued to become increasingly coagulopathic , anemic , and with further <unk> abnormalities . at approximately the # the patient went into ventricular fibrillation and rapidly converted to asystole .	...postoperatively the patient required multiple transfusions of crystalloid , <unk> , and blood products including ffp , platelets , and prbc . despite very aggressive electrolyte , blood , and fluid resuscitation the patient continued to become increasingly coagulopathic , anemic , and with further <unk> abnormalities . at approximately the # the patient went into ventricular fibrillation and rapidly converted to asystole .

Figure 4: Evolution of the network belief state while reading a note. The color scale from blue to red indicates whether the belief decreases or increases respectively at each time step.

the decision process for predictions that are more surprising to them. It should be noted that one can obtain some limited insight into the decision process of the attention-based baseline, for instance, by looking at the global scores. However, for both the GRU and GRNN, we can actually track the model's belief in the presence of a *specific label* as a note is read. As mentioned in Section 3, for the GRNN, one simply needs to look at the evolution of the corresponding grounded dimension between -1 and 1. It is possible to obtain similar information from the GRU, by applying the projection defined in Equation 6 at each time step, which gives a value between 0 and 1.

Figure 4 presents this visualization for extracts from two discharge summaries, outlining the time steps which either increase (red, orange, yellow) or decrease (green or blue) the model's belief in the presence of a concept. In both cases, the Grounded architecture provides a sharper, more interpretable signal, focusing on clinically meaningful passages ("herniated disc" or "lumbar spondylosis" for the patient with dorsopathies, "transfusions of [...] ffp" and "become increasingly coagulopathic" for the coagulation defect diagnosis). On the other hand, while the GRU's belief does increase somewhat on those same phrases, this effect is similar to that of other more distantly related phrases ("fibrillation") as well as some that do not seem especially relevant ("patient").

5 Conclusion

In this work, we introduce the Grounded Recurrent Neural Network, a recurrent network architecture which learns to perform multi-label text classification in a data efficient way by tying concepts of interest to specific dimensions of its hidden state. At the same time structural constraints on the recurrence matrices allow the model to remain tractable even in the presence of a large number of labels. Thus, the model is able to combine the data efficiency of simple Bag-of-Word text classification methods with an RNN's ability to model linguistic structures by tying labels of interest to specific dimensions of its hidden state.

We show that our model is especially suited to a medical setting where its ability to learn from limited data, model concept correlations, and provide interpretable predictions lead to both better performance and improved trustworthiness for practitioners over several strong baselines. We also demonstrate our network's ability to match or outperform these baselines even in a case where data efficiency is less crucial. We hope to improve our model further in future work by introducing structured prediction objectives so as to take better advantage of our proposed architecture's ability to represent interactions between concepts.

Acknowledgments

YJ and DS gratefully acknowledge the support of the Defense Advanced Research Projects Agency (DARPA) Probabilistic Programming for Advancing Machine Learning (PPAML) Program under Air Force Research Laboratory (AFRL) prime contract no. `FA8750-14-C-0005`. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the view of DARPA, AFRL, or the US government.

References

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

O. Bajgar, R. Kadlec, and J. Kleindienst. Embracing data abundance: Booktest dataset for reading comprehension. *CoRR*, abs/1610.00956, 2016.

Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

E. Birman-Deych, A. D. Waterman, Y. Yan, D. S. Nilasena, M. J. Radford, and B. F. Gage. Accuracy of icd-9-cm codes for identifying cardiovascular and stroke risk factors. *Medical care*, 43(5):480–485, 2005.

K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. In *EMNLP 2014, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, 25 October 2014*, pages 103–111, 2014a.

K. Cho, B. van Merriënboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Oct. 2014b.

Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier. Language modeling with gated convolutional networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 933–941, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL `http://proceedings.mlr.press/v70/dauphin17a.html`.

J. L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990.

J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin. Convolutional sequence to sequence learning. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL `http://proceedings.mlr.press/v70/gehring17a.html`.

A. Graves. *Supervised Sequence Labelling with Recurrent Neural Networks*, volume 385 of *Studies in Computational Intelligence*. Springer, 2012.

N. R. Greenbaum, Y. Jernite, Y. Halpern, S. Calder, L. A. Nathanson, D. A. Sontag, and S. Horng. Contextual autocomplete: A novel user interface using machine learning to improve ontology usage and structured data capture for presenting problems in the emergency department. *bioRxiv*, page 127092, 2017.

M. Henaff, J. Weston, A. Szlam, A. Bordes, and Y. LeCun. Tracking the world state with recurrent entity networks. *arXiv preprint arXiv:1612.03969*, 2016.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

D. C. Hsia, W. M. Krushat, A. B. Fagan, J. A. Tebbutt, and R. P. Kusserow. Accuracy of diagnostic coding for medicare patients under the prospective-payment system. *New England Journal of Medicine*, 318(6):352–355, 1988.

Y. Jernite, Y. Halpern, S. Horng, and D. Sontag. Predicting chief complaints at triage time in the emergency department. In *NIPS 2013 Workshop on Machine Learning for Clinical Data Analysis and Healthcare*, 2013.

A. E. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3, 2016.

S. Joshi, S. Gunasekar, D. Sontag, and J. Ghosh. Identifiable phenotyping using constrained non–negative matrix factorization. *arXiv preprint arXiv:1608.00704*, 2016.

G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*, 2016.

T. Lei, R. Barzilay, and T. Jaakkola. Rationalizing neural predictions. *arXiv preprint arXiv:1606.04155*, 2016.

L. V. Lita, S. Yu, R. S. Niculescu, and J. Bi. Large scale diagnostic code classification for medical patient records. In *IJCNLP*, pages 877–882. Citeseer, 2008.

T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur. Recurrent neural network based language model. In *INTERSPEECH 2010, Makuhari, Chiba, Japan, September 2010*, pages 1045–1048, 2010.

S. Narang, G. Diamos, S. Sengupta, and E. Elsen. Exploring sparsity in recurrent neural networks. *arXiv preprint arXiv:1704.05119*, 2017.

W. H. Organization et al. International classification of diseases:[9th] ninth revision, basic tabulation list with alphabetic index. 1978.

A. Perotte, R. Pivovarov, K. Natarajan, N. Weiskopf, F. Wood, and N. Elhadad. Diagnosis code assignment: models and evaluation metrics. *Journal of the American Medical Informatics Association*, 21(2):231–237, 2014.

J. P. Pestian, C. Brew, P. Matykiewicz, D. J. Hovermale, N. Johnson, K. B. Cohen, and W. Duch. A shared task involving multi-label classification of clinical free text. In *BioNLP 2007: Biological, Translational, and Clinical Language Processing*, BioNLP '07, 2007.

P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392, 2016.

M. Saeed, M. Villarroel, A. T. Reisner, G. Clifford, L.-W. Lehman, G. Moody, T. Heldt, T. H. Kyaw, B. Moody, and R. 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.

Y. C. Subakan and P. Smaragdis. Diagonal rnns in symbolic music modeling. *arXiv preprint arXiv:1704.05420*, 2017.

S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus. End-to-end memory networks. In *NIPS 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2440–2448, 2015.