Use of Conventional Machine Learning to Optimize Deep Learning Hyper-parameters for NLP Labeling Tasks

Yang Gu
University of Arizona
ygu@email.arizona.edu

Gondy Leroy
University of Arizona
gondyleroy@email.arizona.edu

Abstract

Deep learning delivers good performance in classification tasks, but is suboptimal with small and unbalanced datasets, which are common in many domains. To address this limitation, we use conventional machine learning, i.e., support vector machines (SVM) to tune deep learning hyper-parameters. We evaluated our approach using mental health electronic health records in which diagnostic criteria needed to be extracted. A bidirectional Long Short-Term Memory network (BI-LSTM) could not learn the labels for the seven scarcest classes, but saw an increase in performance after training with optimal weights learned from tuning SVMs. With these customized class weights, the F1 scores for rare classes rose from 0 to values ranging from 18% to 57%. Overall, the BI-LSTM with SVM-customized class weights achieved a micro-average of 47.1% for F1 across all classes, an improvement over the regular BI-LSTM's 45.9%. The main contribution lies in avoiding null performance for rare classes.

1. Introduction

Autism spectrum disorder (ASD) is a developmental disorder that affects 1 in 59 children in the US [1]. ASD can cause serious impairment in the social, verbal, intellectual, and/or behavioral development of its patients. The economic cost of ASD is estimated to be $66 billion per year in the US, from medical care, specialist care, and lost productivity [2]. Better understanding of this condition has the potential to impact the lives of the large population of patients and families affected by ASD. However, a deeper understanding of the condition and further study on treatments and their different effects on subgroups among the patients would benefit from much larger datasets than are customarily available.

Since 2000, the Center for Disease Control and Prevention (CDC) has been actively monitoring and counting cases of ASD in the US through the Autism and Development Disabilities Monitoring Network (ADDM). ADDM selected eleven sites from eleven different states and collect data on all four- and eight-year-olds in their catchment area every two years [1]. The process has two phases. In the first phase, ADDM identifies children demonstrating ASD-like behaviors and collects medical, specialist, and school records on these children. Data on each child is abstracted into a single case record, which contains large amounts of free text to describe the behaviors of the patients in detail. In the second phase, trained clinicians review and analyze these records to determine the ASD status for each case. Through this study, the CDC has been able to track changes in the prevalence of ASD over time and across different regions and ethnicities [1]. Our goal is to design an artifact that can assist in the surveillance effort.

One of the main challenges of studying ASD is that it is a mental disorder, which is diagnosed based on observable behaviors. Currently, there is no physiological "ground truth" that can be captured by, for example, a pathology report or an MRI scan. Instead, ASD cases are defined by a set of high-level diagnostic criteria described in the Statistical and Diagnostic Manual of Mental Disorders (DSM) [3]. These diagnostic criteria focus on the patients' interactions and behaviors with other people and with their environments. The distinction between a peculiar behavior and a diagnostic criterion can be very subtle, and there exists some inherent ambiguity in the language used to describe human behavior. Overall, ASD case assignment is a difficult task for which humans experts, specifically trained for the task, achieve around 90% agreement [1].

In this work, we developed and evaluated two machine learning (ML) approaches to automatically identify ASD diagnostic criteria and associated features using annotated training data. A machine learning approach can be rapidly updated when the

URI: https://hdl.handle.net/10125/63867
978-0-9981331-3-3

diagnostic standards in the domain evolves, as it did when the DSM updated to the Fifth Edition in 2013. However, the complexity of the class definitions, scarcity of expert-annotated training data, and unbalanced classes pose a challenge for applying state-of-the-art deep learning models. In this work, we first used Support Vector Machines (SVMs) to serve as our baseline and compared them with Bidirectional Long Short-Term Memory (BI-LSTM) networks, a popular deep learning model for text data. We then leveraged SVMs to alleviate weaknesses displayed by the BI-LSTM. The SVM can be trained more quickly than a BI-LSTM, allowing us to conduct grid search to optimize model hyperparameters. We searched through hyperparameters that controlled the shape of the separation plane and class weights to account for unbalanced data, the latter of which can be directly adapted for training an LSTM on the same dataset.

We found that overall, fine-tuned SVMs perform nearly as well as a BI-LSTM in classifying most classes in our data. Our dataset is highly unbalanced, with positive instances appearing in only 0.1% to 3% of all training sentences. Between the small class ratio and limited number of training instances overall, machine learning with this dataset is very difficult. In addition, the best class-weighting scheme found during tuning the SVM can be leveraged during deep learning to improve the performance of the BI-LSTM for extremely sparse classes. Our best system, a BI-LSTM with custom class-weights informed by tuning SVM, achieved a micro-average of 47.1% for F1 across all classes. This work demonstrates two contributions. First, we make a clinical contribution: while this result is insufficient for automated clinical deployment, the system would already be helpful as an assistive tool for clinicians. Second, this study demonstrates a method for selecting machine learning algorithms and model hyperparameters for future work with limited, real-world text data.

2. Related work

2.1 Design Science Research

Hevner's framework [4] for information systems (IS) design science research describes the connecting role IS research plays between a business environment and knowledge base. Business needs of the environment should drive the design of the artifact, and technical foundations from the knowledge base are drawn upon to create the artifact itself. The environment for our work is the ASD surveillance and research community, as well as the community of ASD patients and service providers in a broader sense. This environment needs efficient and accurate analysis of ASD electronic health records (EHR). Since healthcare is a high-stakes domain, practitioners are wary to adopt on black-box solutions with uninterpretable decision processes [5, 6]. Furthermore, the domain's primary duty is to provide care to patients, and thus has only a limited amount of resources devoted to the development of technical artifacts.

From the knowledge base, we draw on two technologies: natural language processing (NLP) and ML. NLP aims "to get computers to perform useful tasks involving human language" [7]. In our use case, we apply NLP to identify clinically relevant information from free text in the ASD EHR. ML algorithms can analyze information and create classification models to infer a class label based on the input data. Evaluation of the artifact is guided by well-established methodology for evaluating ML models and standard evaluation metrics.

2.2 Environment: ASD surveillance

The ADDM defines ASD case status using diagnostic criteria from the DSM. When surveillance started in 2000, the data were analyzed with DSM Fourth Edition, Text Revision (DSM-IV-TR) [8], but the field has since updated diagnostic practices and criteria and since 2014 uses the fifth edition of the DSM (DSM-V). DSM-IV-TR defined four diagnostic criteria for each of three dimensions: social interaction, communication, and stereotyped behaviors. An ASD case must meet six or more diagnostic criteria; with at least two from the social dimension and at least one each from the other two. DSM-V uses seven diagnostic criteria across two categories. A positive case must exhibit all three criteria under category A (A1 – A3) and at least two under category B (B1 – B4). Since the domain undergoes such drastic changes over time, it is worthwhile to develop a fully automated approach that can also adjust to such changes. In addition to the diagnostic criteria, clinicians also make note of associated features (AFs), which are behaviors commonly seen in children with ASD but do not contribute to the diagnosis. Table 1 briefly summarizes the DSM-V diagnostic criteria and relevant AFs (AF1a – AF14) [1].

Identifying the DSM diagnostic criteria from text is very challenging because there is a high level of

[1] AF9 has been defined for an earlier round of surveillance but dropped in the current iteration with DSM-V

variety in the textual features associated with each criterion. There are two reasons for this diversity in expression: the definition of a criterion can encompass a wide range of phenotypes, or observable behaviors, and the linguistic variation involved in describing human behaviors. For example, criteria A3 under DSM-V is defined as "deficits in developing, maintaining, and understanding relationships". This includes impairments in adjusting to social contexts, playing with children, being aware of others, among other characteristics. From our EHR data, we have seen phrases such as "he often seems confused and unaware of others around him" and "seem to be out of touch with the world around him" to describe impairment in adjusting to social contexts. To describe a child not playing with peers, the records have noted "he prefers to play alone rather than with others" and "he sometimes avoids playing with peers". The heterogeneity in the language and semantics associated with each criterion makes this a challenging task for automation.

Table 1. Description of DSM-V Diagnostic Criteria and Associated Features for ASD

Diagnostic Criteria or Associated Features	Description
A1	Social-emotional reciprocity
A2	Nonverbal communicative behaviors
A3	Developing and understanding relationships
B1	Stereotyped movements or speech
B2	Insistence on sameness or routines
B3	Highly restricted or intense interests
B4	Unusual sensory reactivity
AF1a	Abnormalities in eating or drinking
AF1b	Abnormalities in sleeping
AF2	Abnormalities in mood or affect
AF3	Abnormalities in cognitive skill development
AF4	Aggression
AF5	Argumentative, oppositional, defiant, destructive
AF6	Delayed motor milestones
AF7	Hyperactivity
AF8a	Lack of fear in response to real dangers
AF8b	Excessive fear of harmless events
AF10	Self-injuring behavior
AF11a	Staring
AF11b	Seizure-like activity
AF12	Temper tantrums
AF13a	Language delay/disorder
AF13b	Nonverbal child
AF14	Play delay

2.3 NLP and healthcare applications

To apply technology to text, we must first represent text in a way suitable for computation. The classical approach is a bag of words (BOW), which represents a collection of documents as a large, sparse matrix. Every row in the matrix presents one document and every column represents a word in the entire vocabulary. BOW has three weaknesses: it mostly contains zeros, it cannot encode the sequence of words, and it does not encode similarity between words. Even so, this representation has worked well and it requires much more sophisticated approaches with longer development time to improve significantly from this baseline [9].

An alternative representation is word embedding, which represents each word as a dense vector of a pre-determined size (usually 50 to 300 dimensions). Based on the principle that similar words appear in similar contexts, numerical values in the word vectors can capture similarity between words based on their co-occurrence in a large, unlabeled corpus. Word2Vec with skip-gram is an efficient embedding algorithm [10]. It scans sentences in a corpus and learns to predict a word's context within a given window. Word embeddings are commonly used to compute a measure of similarity [11] or to automatically identify similar words related to concepts of interest [12].

NLP technologies have been applied to a variety of tasks in the medical domain. Named entity extraction (NER), a common task in medicine, refers to identifying entities such as diseases or body parts from medical literature and EHR [13-15]. Text classification assigns a label or a class to a document. In the medical domain, this can be used to determine if a text refers to a positive instance of a particular medical condition [16, 17]. Support vector machines (SVM) is a popular algorithm that generally performs well in a variety of tasks, including clinical applications [13, 14, 18].

ML requires labeled training data. However, many clinical applications that require expert knowledge, including ours, face a shortage of expert-labeled training data. In addition, the decision process for most machine learning algorithms is not interpretable by humans. In domains with high stakes and high expectations of transparency, such as healthcare, it is impractical to expect that the domain

will adopt a black-box that cannot explain or justify its decisions.

2.3 Learning with class imbalance

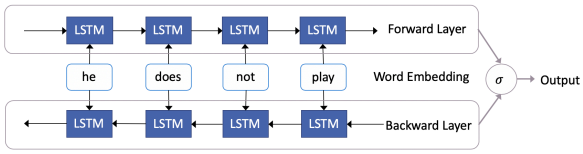
Data imbalance has been known to cause challenges with machine learning and classification. However, unbalance in the ratio between classes are inherent in some domains, such as disease or fraud detection, in which the phenomenon of interest naturally occurs infrequently. There is extensive research on this topic in machine learning, but most methods come down to one of two approaches: adjustments at the sample level or the algorithmic level [19]. At the sample level, the data can be forcibly balanced by under-sampling, over-sampling, or data generation. At the algorithmic level, the cost of different classes can be adjusted. The cost of different classes is a hyper-parameter in many algorithms.

2.4 Deep Learning for NLP

Deep learning uses deep neural networks with multiple layers and specialized architectures to capture specific types of information from data. For example, Convolutional Neural Network (CNN) is a type of deep learning model that specializes in identifying important features that occur in a fixed-size region, such as a curve or an edge in an image, or an n-gram in text [20]. CNNs are especially useful for image recognition but have also be applied successfully to NLP tasks [21, 22].

Long Short Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) designed to capture long-term dependencies in a sequence and has been shown to work well for language tasks [23, 24]. RNNs take in the output from the previous state in the sequence ($state_{t-1}$) as an additional input while processing the current state ($state_t$), allowing it to retain information in the entire sequence. However, information from earlier states decay exponentially as the RNN processes additional elements in the sequence. LSTM uses additive rather than multiplicative operations to avoid exponential decay of information, and uses logical gates to "forget" irrelevant data [23]. The Bidirectional RNN architecture adds a backward RNN layer, in which output from the next element in the sequence ($state_{t+1}$) captures information from upcoming element in the sequence [25]. The bidirectional LSTM (BI-LSTM) can theoretically capture long term dependences in both directions, making it a very powerful architecture for NLP tasks [26-29].

Figure 1. Simplified illustration of BI-LSTM



Both CNNs and RNNs are widely used in NLP. However, there is no clear winner when it comes to their comparative performance – the most suitable model selection depends on the nature of the task [20].

There are some practical challenges for adopting a deep learning approach. Firstly, deep learning models are complex, usually containing millions of trainable parameters. Training these models require a large amount of data and computational resources. Secondly, the performance of deep learning algorithms is sensitive to model hyperparameters. Optimizing the hyperparameter search process is an active area of deep learning research [30, 31]. Hyperparameters are typically optimized through a long and expensive search process that trains and tests the network with multiple combinations of potential hyperparameters. Overall, while deep learning models have the potential to deliver good performance, compared to traditional machine learning algorithms, they also require a higher level of resources in data, computing power, and training time.

3. Research Questions

Our goal is to leverage NLP and ML technologies to provide decision support for ASD diagnosis by automatically identifying ASD diagnostic criteria from EHR. We frame the task as a multi-label sentence classification problem, to determine which sentences contain a positive instance of a diagnostic criterion. The clinician can quickly verify if each sentence identified by the system contains diagnostic criteria for ASD. Then, the clinician can decide if such a set of diagnostic criteria constitute a positive ASD case, and use the sentences identified by the system as evidence to explain their decision to patients or other providers. This setup is designed with the business environment in mind, and the goal is a semi-automated decision support system that aims to facilitate and expedite the diagnostic process while keeping a human clinician in the loop. The automated classification can improve work efficiency by filtering out irrelevant sentences that do not include diagnostic criteria, and leaving the final diagnosis to the human in the loop is more acceptable by the high-stakes healthcare domain.

Our domain reflects challenges for NLP and ML found in many real-world applications. First, our data set is small. For training, we have approximately

26,000 sentences from 120 EHR records. Some labels have fewer than 100 positive examples (0.5% of sentences). This is a small dataset for deep learning. Moreover, our data demonstrates a high level of diversity in the training examples. The lexical features and semantics for the diagnostic criteria, i.e., the labels we want to assign, differ widely per criterion. Given these challenges, our labeling task is a difficult classification problem.

While deep learning models can make sophisticated classifications, our dataset may be too small and unsuited for employing such complex models. Complex models can easily overfit the small number of training examples, or conversely, there may not be enough information in the data to inform the large number of model parameters. Theories and empirical results from the literature point to a general, "out-of-the-box" architecture for this type of problem. For example, gated RNNs, such as LSTM or GRU (Gated Recurrent Units), are commonly used for text data. However, deep learning is highly sensitive to network architecture and model hyperparameters, which are time-consuming and computationally expensive to optimize. Compared to deep learning, classic ML algorithms are faster to tune and train, so we can search over a larger parameter space during tuning and more likely to find a fine-tuned model for a particular dataset. Usually, the hyperparameters for a model depend on characteristics of the data: complexity or dimension of the data, separability between classes, impact of scaling, and imbalance. Therefore, we pose the following research questions:

RQ 1: Can fine-tuned classic ML models outperform "out-of-the-box" deep learning on NLP classification tasks with relatively small training data?

RQ 2: Can we use insights from tuning traditional ML models to inform training and parameter tuning for deep learning models?

4. System description

4.1 Classifiers

We choose two classification approaches for the task. SVM is a reliable, classic ML algorithm that has shown superior performance in a variety of text classification applications and competitions [13, 14, 18, 24]. BI-LSTM is a popular deep learning architecture that can model variable-length sequences such as text, and underlies various state-of-the-art models for NLP tasks [26-29].

SVM. The SVM classifier draws a hyperplane through the high dimensional space in which data is embedded, to separate data into different classes. The algorithm first identifies "support vectors", or edge-cases that exist on the boundary between classes. Then, it finds a separation hyperplane which maximizes the margin between the hyperplane and the support vectors on both sides. Parameters in the algorithm, C and gamma, can be adjusted to be more or less "forgiving" of training data that fall on the other side of the hyperplane, which can be very useful in modeling some datasets. SVM traditionally create a linear separation, but kernel functions can be used for data with non-linear separation between classes. In this work, we use the radial basis function (rbf) kernel. We used scikit-learn's implementation of the SVM in Python [32]. Since the SVM naturally has a two-class formulation, we used the "one-vs-all" training approach to detect the presence of each diagnostic criteria. Our BOW features are the 5000 most frequent tokens from the training data. Since this is a sentence classification task, each row in the BOW matrix is a sentence instead of a full EHR document.

BI-LSTM. We used a BI-LSTM with tunable pre-trained embeddings. The input into the BI-LSTM is 200-dimensional pretrained word embeddings from 4480 ASD EHR from 2000-2010, the complete set of unlabeled EHR text from one ADDM surveillance site during that time. Each LSTM Layer has an internal layer size of 350 and was trained with a dropout ratio of 0.5. We use a sigmoid output layer with one unit for each label. The model is set to train for up to 50 epochs with early stopping. In practice, most models in our experiment trained for less than 25 epochs. In this study, we used Keras (2.1.5) [33] to implement the BI-LSTM and Deeplearning4J's word2vec implementation [34] to train the word embeddings.

4.2 Tuning process

On a personal computer, it takes a few minutes to train an SVM on our dataset, compared to approximately two hours needed to train a BI-LSTM. Therefore, we can conduct fairly thorough parameter tuning for the SVM through-grid search. We validated the parameters on 20% of our training examples, and retrained the final model using the entire dataset based on the best set of parameters.

It is less feasible to exhaustively tune the BI-LSTM through grid-search. We selected the baseline architecture based on a manual search, guided by our

previous experience working with text data and common values seen in literature.

Of the training parameter we tuned for the SVMs, class weights are the set of values most suitable to be adapted for training the BI-LSTM. Since we have a highly imbalanced dataset, we can increase the weights of the minority class to increase their impact on the model. We evaluate the impact of these weights by comparing two BI-LSTM models. The first BI-LSTM uses only naïve under-sampling: half the cases without any positive label were removed from training. Then, we also tested a version of BI-LSTM which, in addition to under-sampling, the classes are weighted by the best values found by tuning the SVMs. We will discuss the optimal weights in more detail in the Results section.

In summary, we compare the following three systems:

- SVMs: a set of highly tuned SVMs, one for each class (uses optimal class weights found through grid search)
- BI-LSTM-1: a regular BI-LSTM (uses naive under-sampling that removes half of all negative training examples)
- BI-LSTM-W: a BI-LSTM (uses naive under-sampling that removes half of all negative training examples, then trained with class weights learned from tuned SVMs)

5. Evaluation

5.1 Dataset

Our dataset consists of 170 EHR records containing 38,934 sentences collected for ADDM from 2012 to 2014. A clinical expert working for ADDM tagged texts in the record with applicable DSM-V features. We used 26,013 sentences (from 120 EHRs) for training and 12,921 sentences (from 50 EHRs) for testing. Table 2 below summarizes the counts and distributions of the classes or labels for the classification task. For this study, we included Associated Features (AF) as well as diagnostic criteria. Associated Features are behaviors commonly found in children with ASD but are not (yet) included in diagnostic criteria.

5.2 Evaluation metrics

Since we have an extremely unbalanced dataset, we use precision, recall, and F1 to assess system performance instead of overall accuracy. The metrics are defined as follows (TP = True Positive; TN = True Negative; FP = False Positive; FN = False Negative).

Table 2. Statistics of training and testing Data

Diagnostic Criteria or Associated Features	Train (N = 26,013)		Test (N = 12,921)	
-	Count	%	Count	%
A1	697	2.7	371	2.9
A2	340	1.3	166	1.3
A3	457	1.8	202	1.6
B1	394	1.5	182	1.4
B2	249	1.0	162	1.3
B3	125	0.5	85	0.7
B4	496	1.9	321	2.5
AF1a	84	0.3	48	0.4
AF1b	41	0.2	17	0.1
AF2	257	1.0	172	1.3
AF3	31	0.1	7	0.1
AF4	195	0.7	89	0.7
AF5	273	1.0	110	0.9
AF6	215	0.8	204	1.6
AF7	462	1.8	216	1.7
AF8a	57	0.2	21	0.2
AF8b	19	0.1	8	0.1
AF10	45	0.2	22	0.2
AF11a	14	0.1	8	0.1
AF11b	20	0.1	20	0.2
AF12	123	0.5	75	0.6
AF13a	606	2.3	296	2.3
AF13b	11	0.0	22	0.2
AF14	72	0.3	40	0.3

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

5.3 Results

The results of the classification system are summarized in Table 3. The highest F1 value for each class is shown in bold, and the null values in italics. The SVMs achieved a micro-averaged F1 value of 46.7% across all classes, and outperforms the LSTM in 10 of the 24 classes in this study. This algorithm performed best for class AF12, reaching F1 of 80.5%, the best F1 out of all classes in this study. It performed worst for classes AF11a and AF11b, reaching F1 of 2.5% and 7.5% respectively.

Table 3. Classification results

Class	SVMs			BI-LSTM-1			BI-LSTM-W		
-	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
A1	0.606	0.407	0.487	0.497	0.499	**0.498**	0.450	0.437	0.443
A2	0.577	0.723	**0.642**	0.599	0.621	0.610	0.450	0.813	0.579
A3	0.454	0.460	0.457	0.695	0.327	0.444	0.522	0.475	**0.497**
B1	0.597	0.456	**0.517**	0.446	0.495	0.469	0.462	0.528	0.492
B2	0.733	0.525	**0.612**	0.648	0.488	0.556	0.525	0.593	0.557
B3	0.509	0.329	**0.400**	0.500	0.012	0.023	0.274	0.271	0.272
B4	0.506	0.517	0.512	0.605	0.586	**0.595**	0.497	0.695	0.579
AF1a	0.418	0.583	0.487	0.560	0.292	0.384	0.520	0.542	**0.531**
AF1b	1.000	0.235	0.381	*0.000*	*0.000*	*0.000*	0.435	0.588	**0.500**
AF2	0.618	0.366	**0.460**	0.806	0.314	0.452	0.667	0.349	0.458
AF3	0.167	0.143	0.154	*0.000*	*0.000*	*0.000*	0.143	0.571	**0.229**
AF4	0.532	0.652	**0.586**	0.765	0.146	0.245	0.516	0.562	0.538
AF5	0.343	0.427	**0.381**	0.352	0.173	0.232	0.310	0.246	0.274
AF6	0.467	0.343	**0.396**	0.398	0.353	0.374	0.275	0.588	0.375
AF7	0.567	0.454	0.504	0.470	0.551	0.508	0.509	0.514	**0.512**
AF8a	0.188	0.286	0.226	*0.000*	*0.000*	*0.000*	0.200	0.286	**0.235**
AF8b	0.200	0.125	0.154	*0.000*	*0.000*	*0.000*	0.333	0.125	**0.182**
AF10	0.209	0.409	0.277	*0.000*	*0.000*	*0.000*	0.333	0.409	**0.367**
AF11a	0.013	0.250	0.025	*0.000*	*0.000*	*0.000*	0.500	0.125	**0.200**
AF11b	0.050	0.150	0.075	*0.000*	*0.000*	*0.000*	0.194	0.300	**0.235**
AF12	0.707	0.933	**0.805**	0.813	0.520	0.634	0.725	0.880	0.795
AF13a	0.491	0.274	0.351	0.416	0.416	**0.416**	0.315	0.568	0.405
AF13b	0.539	0.318	0.400	*0.000*	*0.000*	*0.000*	0.769	0.455	**0.571**
AF14	0.282	0.600	**0.384**	*0.000*	*0.000*	*0.000*	0.311	0.350	0.329
Micro-average	*0.481*	*0.453*	*0.467*	*0.526*	*0.407*	*0.459*	*0.426*	*0.531*	***0.472***

BI-LSTM-1 (without tuned weights) achieved overall micro-averaged F1 of 45.9%, slightly below the SVM. Compared to the SVM, BI-LSTM-1generally achieved higher precision and lower recall. Most notably, the results show clearly the low accuracy of the approach when there are few training examples. There were nine out of twelve classes for which the F1 measure was 0.

Augmented BI-LSTM achieved the highest micro-averaged F1 out of all three systems. After tuning, the weighted approach, BI-LSTM-W, achieved a micro-average of 47.2% in F1 score, just outperforming the highly tuned SVMs and the regular BI-LSTM. There were no classes for which performance was zero. The smallest increase in performance was for label AF8b (which had 19 training examples) and where the F1 value increased from 0 to 18.2%. The largest increase in performance was for label AF13b, (which had 11 training examples) and where the F1 value increased from 0 to 57.1%.

A comparison of the three approaches shows that all three systems obtained their best result with Associated Feature AF12 with the SVM, BI-LSTM, and BI-LSTM-W achieving F1 of 80.5%, 63.4%, and 79.5%, respectively. This class is defined fairly narrowly ("temper tantrums") so the expressions of the diagnostic criteria have been fairly consistent.

Looking at micro-averages across all classes, BI-LSTM-1 achieved the highest precision while BI-LSTM-W achieved the highest recall. After analyzing the optimized parameters of the tuned SVMs, we found that the best weight scheme is the "balanced" model in scikit-learn's implementation, which can be calculated with the formula below.

$$class\ weight = \frac{number\ of\ total\ samples}{number\ of\ classes \times number\ of\ positive\ samples\ for\ class}$$

Using this formula, we calculated class weights for each label based on the distributions observed in the training data and add these customized weights as training hyperparameters in Keras.

The effect of using customized class weights can be observed by comparing the classification results of BI-LSTM-W to BI-LSTM-1. While BI-LSTM-W did not outperform BI-LSTM-1 for every class, it was able to significantly increase the performance of the classes with very few examples that only saw null performance with BI-LSTM-1, such as Associated

Features AF8a - AF11b, AF13b, and AF14. (However, in the cases of AF11a and AF14, SVM achieved higher recall than BI-LSTM-W.) Even when BI-LSTM-W underperformed compared to the regular BI-LSTM-1, the margin is very small. The regular BI-LSTM-1 generally favors precision while BI-LSTM-W generally favors recall.

6. Discussion

In this paper, we demonstrated some of the advantages and challenges of using deep learning in a real-world setting where large training data sets are not available. Deep learning networks use word embeddings as input features, which can encode semantics more richly than BOW. Combined with the BI-LSTM's ability to track long term dependencies, the BI-LSTM was able to make significant contributions to the learning task. In classes A3 and B4, the deep models achieved more than 4% gain in F1 compared to the SVMs. For the very sparse criteria such as AF11a and AF11b, the weighted BI-LSTM-W was able to achieve F-values over 20% although the SVMs and the regular BI-LSTM virtually learned nothing. The advantages of these models are evident.

We also show that the performance of deep learning hinges on model hyperparameters. Our manual search for the deep learning model architecture is guided, as much as possible, by theory and experience. However, due to their complex internal structure, even small changes to the number of internal units or dropout ratio can lead to significant changes to the network. By incorporating class weights, a single number calculated from one formula, we were able to increase the F1-measure from 0 to up 51% for AF11b. In this study, we have also shown that adjusting some parts of the network will affect other aspects – by changing the class weights, we saw a change in performance of the BI-LSTM for all classes. Because we trained a single network for all 24 classes, adjusting the class weights or treatments for one class does affect the entire network. Yet, training a deep model for every single label would be very time-consuming.

Revisiting our first research question in this study, one of our interesting findings is the effectiveness of well-tuned SVMs on text data. Compared to deep learning models, it is much more feasible to carefully tune an algorithm like the SVM. In our study, a well-tuned SVM outperformed BI-LSTM-1 in all but three classes, and the optimized BI-LSTM-W nearly half the classes. With a few exception, such as when the data sparsity issue is extreme, the differences in F1 between SVM and deep learning are within 10%. The performance of the SVM can serve as a robust ML baseline, and even provide a rough estimate of the results to be expected from deep learning.

Our second research question focuses on whether insights from tuning traditional ML models can inform training and parameter tuning for deep learning. This study has shown that insights gained about our data through the SVM – such as the importance of class weights on this data – can be leveraged to improve the deep learning approach. Notably, optimized weights learned helped us avoid null performance for extremely rare classes. However, class weights are just one of many hyperparameters that may significantly affect the performance of deep learning. Hypermeters such as the number of layers, dropout ratio, or size of hidden layers do not have theoretical analogs in other ML models, so we still need to explore other methods for their optimization.

7. Lessons learned and future research

While deep learning methods have demonstrated the potential for a variety of machine learning applications, they are not the best approach for every scenario. In our real-world problem of ASD surveillance, with complex class definitions and small amounts of training data, SVMs can solve the problem nearly as well as a state-of-the-art BI-LSTM model. Since SVMs can be trained and tuned more quickly than deep learning models, they should be among the first options to be considered when experimenting with machine learning approaches for real-world problems. We are also able to glean useful information about our data and improve the training of deep learning models, such as correcting for class imbalance using optimal ratios found while tuning SVMs.

To continue our work with extracting ASD diagnostic criteria and associated features from EHR given our limited dataset, we will explore other non-deep learning-based ML approaches, such as ensemble methods and shallow neural networks. This study has also shown that there are several classes in our dataset for which we have extremely little training data. In future efforts to develop automated diagnostic criteria or feature selection, we will consider alternative approaches such as a rule-based system, bootstrapping, or data generation to generate training data.

8. Acknowledgement

The data presented in this paper were collected by the Centers for Disease Control (CDC) and

Prevention Autism and Developmental Disabilities Monitoring (ADDM) Network supported by CDC Cooperative Agreement Number 5UR3/DD000680. This project was supported by grant number R21HS024988 from the Agency for Healthcare Research and Quality. The content is solely the responsibility of the authors and does not necessarily represent the official views of the Agency for Healthcare Research and Quality.

9. References

[1] J. Baio, L. Wiggins, D. L. Christensen, M. J. Maenner, J. Daniels, Z. Warren, M. Kurzius-Spencer et al., "Prevalence of Autism Spectrum Disorder Among Children Aged 8 Years—Autism and Developmental Disabilities Monitoring Network, 11 Sites, United States, 2014," MMWR. Surveillance Summaries, vol. 67, 2018.

[2] A. V. S. Buescher, Z. Cidav, M. Knapp, and D. S. Mandell, "Costs of Autism Spectrum Disorders in the United Kingdom and the United States," JAMA Pediatrics, vol. 168, no. 8, 2014, pp. 721-728.

[3] American Psychiatric Association, Diagnostic and statistical manual of mental disorders (DSM-V). American Psychiatric Pub, 2013.

[4] A. Hevner, R, M. T. Salvatore, J. Park, and S. Ram, "Design science in information systems research," MIS quarterly, vol. 28, no. 1, 2004, pp. 75-105.

[5] T. Ching et al., "Opportunities and obstacles for deep learning in biology and medicine," Journal of The Royal Society Interface, vol. 15, no. 141, 2018, p. 20170387.

[6] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, "Deep learning for healthcare: review, opportunities and challenges," Briefings in Bioinformatics, vol. 19, no. 6, 2017, pp. 1236-1246.

[7] J. H. Martin and D. Jurafsky, Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition. Pearson/Prentice Hall Upper Saddle River, 2009.

[8] American Psychiatric Association, Diagnostic and statistical manual-text revision (DSM-IV-TR). American Psychiatric Association, 2000.

[9] J. Hirschberg and C. D. Manning, "Advances in natural language processing," Science, vol. 349, no. 6245, 2015, pp. 261-266.

[10] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," arXiv preprint arXiv:1301.3781, 2013.

[11] Z. Yin, M. Harrell, J. L. Warner, Q. Chen, D. Fabbri, and B. A. Malin, "The therapy is making me sick: how online portal communications between breast cancer patients and physicians indicate medication discontinuation," Journal of the American Medical Informatics Association, vol. 25, no. 11, pp. 1444-1451.

[12] C. Bejan et al., "Mining 100 million notes to find homelessness and adverse childhood experiences: 2 case studies of rare and severe social determinants of health in electronic health records," Journal of the American Medical Infomatics Association, 2017.

[13] S. Pradhan et al., "Evaluating the state of the art in disorder recognition and normalization of the clinical narrative," Journal of the American Medical Informatics Association, vol. 22, no. 1, 2015, pp. 143-154.

[14] R. Sullivan, R. Yao, R. Jarrar, J. Buchhalter, and G. Gonzalez, "Text Classification towards Detecting Misdiagnosis of an Epilepsy Syndrome in a Pediatric Population," AMIA Annual Symposium Proceedings, vol. 2014, 11/14 2014, pp. 1082-1087.

[15] K. Roberts et al., "A machine learning approach for identifying anatomical locations of actionable findings in radiology reports," AMIA Annual Symposium Proceedings, vol. 2012, 2012, pp. 779-88.

[16] K. Haerian, H. Salmasian, and C. Friedman, "Methods for identifying suicide or suicidal ideation in EHRs," AMIA Annual Symposium Proceedings, 2012, vol. 2012, 2012 , pp. 1244-53.

[17] G. Leroy, Y. Gu, S. Pettygrove, and M. Kurzius-Spencer, "Automated Lexicon and Feature Construction Using Word Embedding and Clustering for Classification of ASD Diagnoses Using EHR," in International Conference on Applications of Natural Language to Information Systems, 2017, pp. 34-37.

[18] S. Moon, S. Pakhomov, and G. B. Melton, "Automated disambiguation of acronyms and abbreviations in clinical texts: window and training size considerations," AMIA Annual Symposium Proceedings, vol. 2012, 2012, pp. 1310-9.

[19] N. V. Chawla, N. Japkowicz, and A. Kotcz, "Special issue on learning from imbalanced data sets," ACM Sigkdd Explorations Newsletter, vol. 6, no. 1, 2004, pp. 1-6.

[20] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," IEEE Computational intelligence magazine, vol. 13, no. 3, 2018, pp. 55-75.

[21] Y. Kim, "Convolutional Neural Networks for Sentence Classification," Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing EMNLP 14, Doha, Qatar, 2014.

[22] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, "Very Deep Convolutional Networks for Text

Classification," in European Chapter of the Association for Computational Linguistics EACL'17, 2017.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, 1997, pp. 1735-1780.

[24] S. Rosenthal, N. Farra, and P. Nakov, "SemEval-2017 task 4: Sentiment analysis in Twitter," in Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), 2017, pp. 502-518.

[25] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673-2681, 1997.

[26] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in 2013 IEEE international conference on acoustics, speech and signal processing, 2013, pp. 6645-6649

[27] X. Ma and E. Hovy, "End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF," in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol. 1, 2016, pp. 1064-1074.

[28] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural Architectures for Named Entity Recognition," in Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, pp. 260-270.

[29] J. Chiu and E. Nichols, "Named Entity Recognition with Bidirectional LSTM-CNNs," Transactions of the Association of Computational Linguistics, vol. 4, no. 1, 2016, pp. 357-370,.

[30] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in Advances in neural information processing systems, 2012, pp. 2951-2959.

[31] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in Advances in neural information processing systems, 2011, pp. 2546-2554.

[32] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, 2012.

[33] F. Chollet et al., "Keras," https://keras.io, 2015.

[34] Eclipse Deeplearning4j Development Team. "Deeplearning4j: Open-source distributed deep learning for the JVM, Apache Software Foundation License 2.0," http://deeplearning4j.org.