SpanPredict: Extraction of Predictive Document Spans with Neural Attention

Vivek Subramanian, Matthew Engelhard, Samuel Berchuck,
Liqun Chen, Ricardo Henao, and Lawrence Carin
Duke University
{vivek.subramanian, matthew.engelhard, samuel.berchuck,
liqun.chen, ricardo.henao, lcarin}@duke.edu

Abstract

In many natural language processing applications, identifying predictive text can be as important as the predictions themselves. When predicting medical diagnoses, for example, identifying predictive content in clinical notes not only enhances interpretability, but also allows unknown, descriptive (*i.e.,* text-based) risk factors to be identified. We here formalize this problem as *predictive extraction* and address it using a simple mechanism based on linear attention. Our method preserves differentiability, allowing scalable inference via stochastic gradient descent. Further, the model decomposes predictions into a sum of contributions of distinct text spans. Importantly, we require only document labels, not ground-truth spans. Results show that our model identifies semantically-cohesive spans and assigns them scores that agree with human ratings, while preserving classification performance.

1 Introduction

Attention-based neural network architectures achieve human-level performance in many document classification tasks. However, understanding model predictions remains challenging. Common feature attribution methods are often inadequate, because the "features" of a document classification model – individual words or their embeddings – tend to have limited or ambiguous meaning in isolation, and must instead be interpreted in context. Rather than examining the importance of individual words and passing the contextualization task to the end-user, we may wish to extract distinct spans of text, such as sentences or paragraphs, and quantify the effect of each span on model predictions. However, the appropriate span boundaries depend on the document type, and processing all possible spans individually is computationally prohibitive.

In some settings, understanding model predictions can be as important as the predictions themselves. When predicting medical diagnoses from clinical notes, for example, attributing predictions to specific note content assures clinicians that the model is not relying on data artifacts that are not clinically meaningful or generalizable. Moreover, this process may illuminate previously unknown risk factors that are described in clinical notes but not captured in a structured manner. Our work is motivated by the problem of autism spectrum disorder (ASD) diagnosis, in which many early symptoms are behavioral rather than physiologic, and are documented in clinical notes using multiple-word descriptions, not individual terms. Morever, extended and nuanced descriptions are important in many common document classification tasks, for instance, the scoring of movie or food reviews.

Identifying important spans of text is a recurring theme in natural language processing. In *extractive summarization*, a document summary is created by selecting and concatenating important spans within a document (Narayan et al., 2018); and in many *question answering* tasks, including in the Stanford Question Answering Dataset (Rajpurkar et al., 2018), the goal is to identify a span within a paragraph of text that answers a given question. In both cases, training typically relies on ground truth spans, *i.e.*, correct start and end positions are available during training, which the model learns to predict.

In contrast, our goal is to identify distinct spans within a document that, taken together, are sufficient to predict its associated label. In this task, which we call *predictive extraction*, ground truth spans are not available; instead, training is based on document labels alone, and without predefined spans, *e.g.*, sentences or paragraphs. Moreover, similar to feature attribution methods, we wish to assign scores to each span such that predictions are effectively decomposed into the contributions of individual spans. In the current work, which for simplicity focuses on binary classification, we achieve this by summing individual span scores to

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 5234–5258
June 6–11, 2021.

obtain the log-odds of a positive label.

Since correct start and end positions are not known, they are represented as latent variables that must be learned to (a) optimize classification performance, and (b) satisfy additional span constraints; in particular, we wish to ensure that spans are concise, and do not significantly overlap. A brute-force approach – in which all sets of spans satisfying these constraints are evaluated – is computationally intractable, as the number of possibilities is $\mathcal{O}(n^k)$, where n is the length of the document and k is the number of spans. Alternatively, predicting discrete start and end positions would introduce categorical latent variables, necessitating the use of a continuous relaxation (Jang et al., 2016; Maddison et al., 2016) or gradient estimation alternatives (Tucker et al., 2017). Instead, we formulate a simple but effective approach in which span representations are derived directly from a continuous (probabilistic) representation of the start and end positions, avoiding more computationally expensive gradient estimation; and the positions themselves, are predicted using linear attention. Our contributions are as follows:

- We define *predictive extraction* and describe its importance particularly for prediction tasks in which model performance exceeds human performance.
- We formulate SpanPredict, a neural network model for predictive extraction in which predicted log-odds are formulated as the sum of contributions of distinct spans.
- We quantify prediction and span selection performance on five binary classification tasks, including three real-world medical diagnosis prediction tasks.
- In the context of these studies, we quantify the effect of span constraints on performance.

2 Related Work

Explaining neural network predictions is a well-known problem, one that is particularly challenging in natural language processing, due to the presence of complex semantic structure and interdependencies (Belinkov and Glass, 2019). The importance of individual words, or their embeddings, can be quantified using word-pooling strategies in which some words contribute to predictions, and others do not (Shen et al., 2018). In many settings, however, examining individual words in isolation provides limited insight. One solution is to ask the model to generate an explanation along with each prediction (Zhang et al., 2016); inconveniently, explanations must be available during training.

Alternatively, explanations may be selected from within the document itself. This strategy is closely related to question answering and extractive summarization, in which text spans are selected to answer a given question or summarize a document, respectively. If correct spans are known during training, representations of candidate spans can be generated and used to evaluate each span as the possible answer to a question, or for inclusion in a document summary. Representations for all short spans can be generated via bidirectional recurrent neural networks (Lee et al., 2016), for example, or candidate spans can be limited to individual words and sentences (Cheng and Lapata, 2016).

Clinical notes contain redundant information as well as medical jargon and abbreviations, making meaningful text extraction more useful but also more challenging. Concept recognition and relation detection have been used to identify salient note content, which is then used to create a summary (Liang et al., 2019). Alternatively, the importance of specific content can be evaluated based on its presence or absence in subsequent notes; this concept has been used to train extractive summarization models using discharge summaries, which distill information collected during a clinical encounter (Alsentzer and Kim, 2018), and using subsequent notes, which are more likely to repeat earlier information if it is important (Liu et al., 2018).

In contrast to these methods, our focus is on extracting *predictive* text in settings where span annotations are costly to obtain. (Lei et al., 2016) tackle this by introducing two networks, a generator and an encoder, which, respectively, filter for important words before making a prediction. However, theirs is a sampling-based method that must be trained via REINFORCE. Moreover, unlike our approach, they are unable to score individual phrases, limiting interpretability. Our work is perhaps most closely related to (Bastings et al., 2019), which defines candidate spans using a modified Kumaraswamy distribution and then selects spans that are predictive via fused LASSO. Instead, our approach uses an attention mechanism to identify promising start and end positions, which are then used to construct spans nonparametrically. Lastly, another approach is the prediction-constrained topic model, which provides interpretable topics that are useful for pre-

dicting labels of interest (Ren et al., 2019; Hughes et al., 2017).

3 Model

3.1 Predictive Extraction

We define predictive extraction as follows. Given a document $\mathbf{X}$ and its associated binary label y, the goal of predictive extraction is to select contiguous sequences of text called *spans* that, jointly, are sufficient to predict the label y effectively. One wishes to also assign each span a score reflecting its contribution to the prediction $\hat{y}$. In this work, span selection is regularized by quantifying span size and overlap among spans, and performance is evaluated via human rating of randomly selected spans.

3.2 Proposed Model: SpanPredict

The architecture for the proposed SpanPredict model is given in Figure 1. For a given passage of text, let $t = 1, \ldots, T$ index token s_t, and let $\mathbf{e}_t \in \mathbb{R}^D$ denote an embedding of token s_t. Note that the $\mathbf{e}_t$ may be linear token embeddings, but may also be contextualized embeddings generated by BERT (Devlin et al., 2018), for example. For each embedding $\mathbf{e}_t$, two probability vectors $\tilde{\mathbf{p}} = \text{softmax}\left(\mathbf{E}^\top \mathbf{w}_p\right)$ and $\tilde{\mathbf{q}} = \text{softmax}\left(\mathbf{E}^\top \mathbf{w}_q\right)$, where $\mathbf{E} = [\mathbf{e}_1, \ldots, \mathbf{e}_T]$, are computed using a pair of trainable, sentinel attention vectors $\mathbf{w}_p, \mathbf{w}_q \in \mathbb{R}^D$. Vectors $\tilde{\mathbf{p}} = [\tilde{p}_1, \ldots, \tilde{p}_T] \in \Delta^{T-1}$ and $\tilde{\mathbf{q}} = [\tilde{q}_1, \ldots, \tilde{q}_T] \in \Delta^{T-1}$, where Δ^{T-1} is the $T-1$ simplex, represent the set of probabilities of each token in the sequence being the start and end of a span of text, respectively. While it is tempting to create a span by choosing the start and end positions with highest probabilities, *i.e.*, $\arg\max_t \tilde{\mathbf{p}}$ and $\arg\max_t \tilde{\mathbf{q}}$, respectively, this is problematic since the $\arg\max$ function is not differentiable, precluding training by standard backpropagation.

To produce a span representation $\mathbf{r}$ that is amenable to backpropagation, we employ the cumulative sum function $\text{cumsum}(\mathbf{x}) : \mathbf{x} \in \mathbb{R}^T \mapsto \mathbf{c} \in \mathbb{R}^T$, where $c_t = \sum_{t' \leq t} x_{t'}$ is an element of $\mathbf{c}$. Using this function, we define $\mathbf{p} = \text{cumsum}(\tilde{\mathbf{p}})$ and $\mathbf{q} = \text{cumsum}(\tilde{\mathbf{q}}_{::-1})$, where $\mathbf{x}_{::-1}$ is the vector $\mathbf{x}$ with its elements reversed. Intuitively, p_t (element of $\mathbf{p}$) represents the probability that the start of a span has occurred by token t when coming from the left of the sequence and q_t (element of $\mathbf{q}$) represents the probability that the end has occurred by token t when coming from the right. We then calculate a set of weights $\tilde{\mathbf{r}} = \mathbf{p} \odot \mathbf{q}$, where $\odot$ denotes the element-wise product. The product $\tilde{\mathbf{r}}$ therefore assigns large weights to tokens which have high mass under both $\mathbf{p}$ and $\mathbf{q}$, *i.e.*, those that are identified as falling between the start and end points of a span.

Rather than directly using $\tilde{\mathbf{r}}$ to compute a span representation, we first normalize $\tilde{\mathbf{r}} = [\tilde{r}_1, \ldots, \tilde{r}_T]$ such that its elements sum to 1. We define the elements of $\mathbf{r}$ as $r_t = \tilde{r}_t / (\sum_t \tilde{r}_t + \epsilon)$ and $\epsilon \approx 10^{-8}$ is included for numeric stability, since $\tilde{\mathbf{r}}$ is zero everywhere if the support of $\mathbf{p}$ and $\mathbf{q}$ do not overlap, indicating a null span. Importantly, normalization allows us to compute a score that reflects each word's contribution to the span as a whole, regardless of the length of the overall sequence. We then construct a span representation $\mathbf{m} = \mathbf{E}\mathbf{r} \in \mathbb{R}^D$, by taking an average of the embeddings $\mathbf{E}$ weighted by $\mathbf{r}$. This method of constructing spans is a key feature of our model as it allows for span location and length to be dictated nonparametrically, driven only by the content within the identified spans and the quality of the predictions.

We repeat this procedure J times to identify J spans $\mathbf{m}_j$, $j = 1, \ldots, J$, using unique pairs of sentinel vectors $\{\mathbf{w}_{pj}, \mathbf{w}_{qj}\}$ for each span. Finally, we employ attention over the J span representations to generate span scores $z_j = \mathbf{w}_z^\top \mathbf{m}_j$. These scores are effectively *logits*, which can be interpreted as the *log-odds* of a positive label associated with the span. The output of the model, $\hat{y} = \sigma(\sum_j z_j)$, where $\sigma(\cdot)$ is the sigmoid function, is compared against the truth y, and the model is trained via backpropagation with binary cross-entropy loss.

In this work, we pad or truncate documents, as appropriate, to have fixed length $\tilde{T}$. Tokens are mapped to dense vectors using 100-dimensional GloVe embeddings, which are then contextualized with three parallel convolutional layers with filters of kernel sizes $K \in \{2, 3, 5\}$ prior to span selection (see Section 5.1 for details). We chose this simple approach over more complex embeddings, *e.g.*, BERT, to focus on the quality of span extraction and its effect on classification performance rather than on maximizing performance *per se*. However, our approach is agnostic to the choice of embedding, and alternative embeddings may be used if desired.

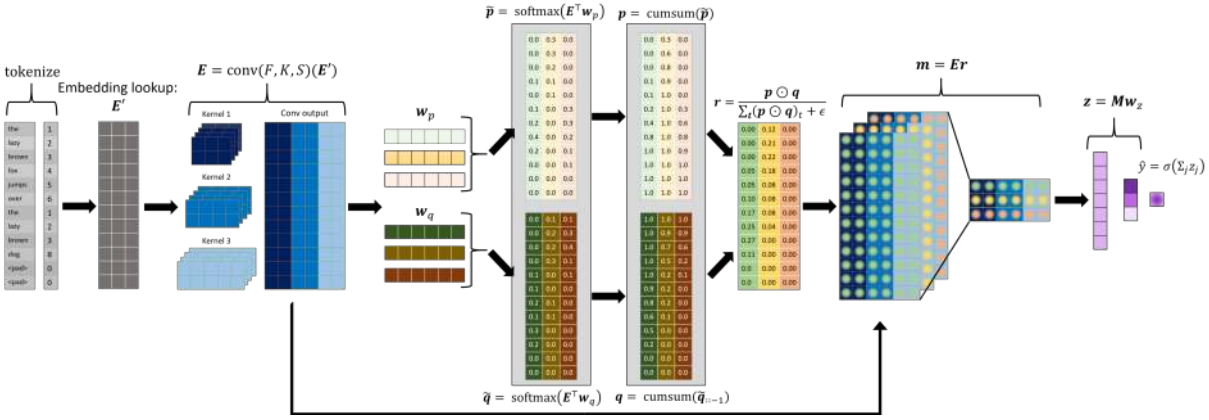


Figure 1: Model architecture. We begin with tokenization followed by an embedding lookup. Three convolutions with kernel sizes $K \in \{2, 3, 5\}$ (shades of blue) are performed in parallel, and outputs are concatenated to form contextual embeddings. The span detection module then identifies $J = 3$ (in this example) spans denoted by green, yellow, and red. Word scores from the span detectors are used to compute J weighted average span representations, each denoted by $\mathbf{m}$. These are stacked to form $\mathbf{M}$. Note that the red span weights are all 0, indicating a null span representation. Finally, we perform attention over the span representations to obtain scores z_j, which are added and passed through a sigmoid to predict $\hat{y} \in (0, 1)$.

3.3 Constraining span uniqueness and size

Our model already contains an *implicit* penalty for span size – specifically, the greater the number of tokens over which the model averages to compute a span representations, the smaller the contribution of influential words to the span logits. Hence, the model should *implicitly* prefer to have spans that are concise and not overwhelmed with "filler" words. Further, our model naturally encourages sparsity of number of spans. Spans that do not carry meaning are biased towards generating weights z_j of zero since, otherwise, they would inadvertently reduce the predictive performance. This also means that the model implicitly learns the number of spans required to make predictions on an individual document basis.

In practice, we observed that spans identified by our model tend to be rather long and suffer from significant overlap, which suggests the need for an additional *explicit* penalty to make the spans more concise and distinct. Methods involving L_2-regularization on the magnitudes of $\mathbf{r}_j$ or z_j may shrink the spans or encourage sparsity, but they do not directly address the overlap issue. Thus, we seek a regularization method that directly compares spans $\mathbf{r}_j$ with one another.

Since vectors $\{\mathbf{r}_j\}_{j=1}^J$ each constitute a discrete probability distribution, a natural choice is to consider divergences between them. Among these, the generalized Jensen-Shannon divergence (JSD) (Lin, 1991), a symmetric measure of similarity among a set of J probability distributions, is appealing for several reasons. The JSD is defined as

$$\text{JSD}_{\boldsymbol{\pi}}(\mathbf{r}_1, \ldots, \mathbf{r}_J) = \underbrace{H\left(\sum_{j=1}^J \pi_j \mathbf{r}_j\right)}_{\text{span overlap}} - \underbrace{\sum_{j=1}^J \pi_j H(\mathbf{r}_j)}_{\text{span conciseness}}, \quad (1)$$

where $H(\cdot)$ denotes the entropy and $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_J] \in \Delta^{J-1}$ is a distribution of mixing coefficients among the J distributions $\{\mathbf{r}_j\}_{j=1}^J$ (Lin, 1991). While the JSD is commonly expressed as a weighted average of Kullback-Leibler divergences (Manning et al., 1999), in this form, we emphasize that the JSD can be decomposed into two terms: the entropy of the (weighted) average of the $\mathbf{r}_j$s and the (weighted) average of the entropies of each $\mathbf{r}_j$. Thus, by maximizing the JSD, we simultaneously maximize the entropy of the average distribution (*i.e.*, minimize *overlap* between the $\mathbf{r}_j$s) while minimizing the entropy of each $\mathbf{r}_j$ (*i.e.*, maximize *conciseness* of each $\mathbf{r}_j$). In addition, the JSD is bounded below and above by 0 and $\log(J)$, respectively, allowing one to monitor convergence during training (see Appendix C) (Lin, 1991).

We can modify the JSD formulation by introducing a tunable parameter $\theta \in [0, 0.5]$ as follows:

Dataset	Num. Notes	Num. tokens	Age: patients (years)	Age: controls (years)
ASD	44458	560.1 ± 515.3	2.1 ± 2.4	2.2 ± 2.4
ADHD	45160	480.0 ± 437.3	5.6 ± 2.2	5.5 ± 1.6
Asthma	46588	505.7 ± 441.3	1.7 ± 2.0	1.7 ± 0.2

Table 1: Diagnosis prediction statistics. Each dataset was divided into training, validation, and testing subsets with a 45:10:45 split. Positive and negative examples are balanced in each dataset.

$$\begin{aligned}&\mathrm{JSD}_{\boldsymbol{\pi}}(\mathbf{r}_1, \ldots, \mathbf{r}_J; \theta) = \\ &2\left\{\theta\left[H\left(\sum_{j=1}^{J}\pi_j \mathbf{r}_j\right)\right] - \right. \\ &\left.(1-\theta)\left[\sum_{j=1}^{J}\pi_j H(\mathbf{r}_j)\right]\right\},\end{aligned} \tag{2}$$

where we recover (1) when $\theta = 0.5$. As we slide θ closer to 0, the contribution of the second term increases; hence, the smaller the value of θ, the smaller we can expect the entropies of the individual distributions to be. This implies that the span sizes can be made smaller by reducing θ.

Lemma 3.1. *The modified JSD is bounded above by a constant, independent of the entropies of the individual* $\{\mathbf{r}_j\}_{j=1}^{J}$.

Proof. Defining $H_1 = H\left(\sum_{j=1}^{J}\pi_j \mathbf{r}_j\right)$ and $H_2 = \sum_{j=1}^{J}\pi_j H(\mathbf{r}_j)$, we have:

$$\begin{aligned}&\mathrm{JSD}_{\boldsymbol{\pi}}(\mathbf{r}_1, \ldots, \mathbf{r}_J; \theta) = \\ &\quad = 2\left\{\theta H_1 - (1-\theta)H_2\right\} \\ &\quad = 2\left\{\theta H_1 - \theta H_2\right\} - 2(1-2\theta)H_2 \\ &\quad = 2\theta \mathrm{JSD}_{\boldsymbol{\pi}}(\mathbf{r}_1, \ldots, \mathbf{r}_J) - \\ &\qquad 2(1-2\theta)H_2 \\ &\quad \leq 2\theta \mathrm{JSD}_{\boldsymbol{\pi}}(\mathbf{r}_1, \ldots, \mathbf{r}_J),\end{aligned} \tag{3}$$

where the last line follows from the fact that $1 - 2\theta \geq 0 \,\forall\, \theta \in [0, 0.5]$ and $H_2 \geq 0$. ■

This result provides a lower bound on our JSD objective, useful for monitoring convergence during training, *i.e.*, $-\mathrm{JSD}_{\boldsymbol{\pi}}(\mathbf{r}_1, \ldots, \mathbf{r}_J; \theta) \geq -2\theta \log(J)$.

4 Learning

The complete objective function we aim to minimize is thus given by:

$$\begin{aligned}\mathcal{L} = -\mathbb{E}_{\mathcal{D}}\big[&(1-\alpha)\big(y \log \hat{y} + \\ &(1-y)\log(1-\hat{y})\big) + \\ &\alpha \mathrm{JSD}(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_J; \theta)\big]\end{aligned} \tag{4}$$

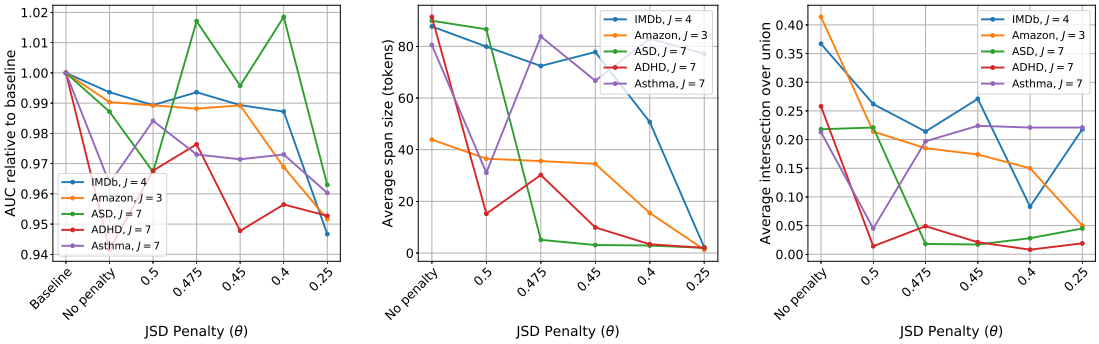
where $\mathcal{D}$ is our dataset, and $\alpha \in [0, 1)$ is a hyperparameter denoting the weight of the modified JSD penalty relative to the classification loss. For simplicity, we choose to take $\pi_j = 1/J$ in (2) and have therefore omitted $\boldsymbol{\pi}$ from the expression for $\mathrm{JSD}_{\boldsymbol{\pi}}(\mathbf{r}_1, \ldots, \mathbf{r}_J; \theta)$ in (4).

Aside from the learning rate, our model consists of only three hyperparameters J, θ, and α, making it highly attractive for experimentation. Predictive performance is not very sensitive to the choice of J; here we select J to be proportional to the average document length in each dataset, but we investigate the impact of a fixed larger value of J in Appendix B. To choose α, we employ a method similar to that used in (Smith, 2017) for choosing a learning rate. Specifically, we slowly ramp up α from a minimum value of 0 in increments of 10^{-5} batch by batch and monitor validation accuracy. When the accuracy starts to level off or drop, we mark the value of α; we found $\alpha = 0.1$ to be appropriate for our datasets. Parameter θ is selected via cross-validation (trading off performance for desired span length), and is a focus of our experiments, described below.

5 Experiments

Datasets We perform experiments on five datasets: two publicly available non-medical datasets, and three constructed from clinical notes from the Duke University Health System. We consider the IMDb movie reviews dataset[1] (Maas et al., 2011), which contains 25,000 training and testing

[1] https://www.tensorflow.org/datasets/catalog/imdb_reviews



(a) AUC relative to baseline (b) Average span size (c) Average intersection-over-union

Figure 2: Trends in performance. Baseline AUCs – IMDb: 0.938, Amazon: 0.931, ASD: 0.702, ADHD: 0.804, Asthma: 0.630. Performance tends to drop slightly as θ is decreased, but spans become more concise and distinct.

examples of movie reviews and a binary viewer rating; and the Amazon Fine Food Reviews dataset[2] (McAuley and Leskovec, 2013), which contains >500,000 reviews of food items, which we subsample to 25,000 training and testing examples and 5000 validation examples for consistency. Positive and negative examples are balanced in each subset. Reviews are on a 5-point scale, but we binarize by labeling ratings of 3 or higher as positive. Average document length for IMDb is 225.4 ± 166.1 tokens, and shorter for Amazon at 84.3 ± 86.1 tokens.

The three medical datasets were built by sampling the clinical progress notes of children visiting the Duke University Health System between October 1, 2013 and October 1, 2018. All analyses were approved by the Duke University Institutional Review Board. Diagnosis codes (ICD-9/10) were used to identify patients eventually diagnosed with autism spectrum disorder (ASD), attention deficit hyperactivity disorder (ADHD), or asthma. Notes from each patient group were then selected at random and labeled as positive for the condition corresponding to that group. While many of these notes are not directly related to the condition of interest, a large proportion contain related information or risk factors. Future work will focus on extracting predictive spans from *all* notes from a given patient; here we focus on individual notes to limit complexity and highlight span extraction performance. For each diagnosis prediction task, we then selected notes from age-matched controls not diagnosed with the condition as of October 1, 2018, and assigned them a negative label. Each dataset contains an even number of positive and negative examples. Descriptive statistics are shown in Table 1.

[2] https://www.kaggle.com/snap/amazon-fine-food-reviews

5.1 Methods

We first establish baseline performance for each dataset by training a CNN-based classifier that replaces span detection with max-pooling of all filter activations, but that is otherwise identical to SpanPredict. Pooled activations are fed into a linear layer that predicts the log-odds of a positive label. Our baseline model was motivated by our goal to understand how the SpanPredict module affects performance and highlight its flexibility with many baseline models, rather than to maximize performance, *per se*. A CNN-baseline was preferred over a BiLSTM, as the latter contains a context window of infinite length. Thus, a contiguous contiguous sequence of tokens can contain information from tokens outside the window, making span identification and interpretation difficult. Our baseline is closely related to hierarchical SWEM (Shen et al., 2018), and despite its simplicity, achieves an accuracy of 86.3% on IMDb, which is competitive against recent benchmarks (Papers with Code, 2020; Zhang et al., 2018). As shown in figure 2a, this same model achieves an AUC of 0.938.

To contextualize GloVe embeddings, we apply $C = 3$ parallel convolutional layers, each of filter size $F = 50$, stride $S = 1$, kernel sizes $K \in \{2, 3, 5\}$ and with ReLU activations. Tokens are padded such that the output of each convolution is of length $\tilde{T}$. We then concatenate the filters to obtain refined embeddings $\mathbf{e}_t \in \mathbb{R}^{CF}$, which are fed into the span detection module. Omitting the token embedding matrix, our model contains $100 \times (2 + 3 + 5) \times F + C \times F$ parameters in the

convolutional layers and $2J \times C \times F$ parameters in the span detection filters. Thus, SpanPredict contains $2J \times C \times F$ more parameters than our baseline model, and $\approx 50,000$ parameters in total.

We take a step-wise approach to assessing model hyperparameters by first training with only binary cross entropy loss ($\alpha = 0$). We then train three models with $\alpha = 0.1$ – chosen by comparing baseline performance on $\alpha \in \{0.01, 0.05, 0.1, 0.2\}$ – and a maximum of J spans, where J is proportional to the average document length in the dataset. For IMDb, we choose 4; for Amazon, 3; and for all diagnoses, 7. Within this set of three, we vary θ across the values $\{0.5, 0.475, 0.45, 0.4, 0.25\}$ to assess the impact of the JSD penalty on span size and prediction performance. In Appendix B, we show results when J is increased to 10.

For each experiment, we summarize classification performance using area under the ROC curve (AUC, for span size) and intersection over union (IoU, for span overlap). However, our goal is not to maximize classification performance, but rather to maintain good performance while also providing distinct, concise spans and scoring them accurately. To evaluate our span selection, we (a) quantify average span length and overlap for each model; (b) evaluate model-based span scoring, for which we have no ground truth, by having human raters score a random sample of spans; and (c) show a large number of spans selected by our models, which may be evaluated qualitatively (Appendix A).

For IMDb and Amazon, samples for human evaluation were selected by first filtering for correctly labeled spans ($z_{ij} < 0$ when $y_i = 0$, where i indexes documents in the testing set and j indexes spans; and vice versa). The remaining spans were divided by z_{ij} into quantiles, and 40 samples were drawn from each (to ensure a roughly uniform distribution of scores). We recruited 3 native English speakers to rate each span on a 5-point scale (very negative, negative, neutral, positive, very positive).

A similar procedure was used to select spans from each medical dataset. Here, we only considered correctly labeled, condition-positive notes ($y_i = 1$), since condition-negative notes ($y_i = 0$) are marked by the absence of information related to the diagnosis more than the presence of information denying it. To mitigate rater fatigue, we sampled 20 spans per quantile, per condition, rather than 40. Three neurology or psychiatry residents rated each span on a 5-point scale. Raters were asked to grade the conditional probability of seeing the span given that the patient has the condition.

5.2 Training

SpanPredict was built in Python using Tensorflow 2.1 and trained on a single NVIDIA Titan Xp GPU. We use the Adam optimizer with default values of $\eta = 0.001$, $\epsilon = 10^{-7}$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. Parameters are randomly initialized from $\mathcal{N}(0, 0.05)$ for the convolutional layers and $\mathcal{N}(0, 0.5)$ for the span detection layers. To regularize training, we employ Dropout (Srivastava et al., 2014); after selecting α, Dropout rates of $\{0.1, 0.25, 0.5, 0.7\}$ were tested and 0.5 was chosen. We train each of our models with a batch size of 8 for 300 epochs. Our model complexity is linear in space and time with respect to J. We report performance using the model stored at the epoch with the lowest overall validation loss. To allow the model to warm up to the JSD penalty, we linearly increase α from 0 to 0.1 over 150 epochs and then fix its value to 0.1 for the remainder of the experiment. We use the Keras tokenizer with a vocabulary size of 30,000 to tokenize our text and pad or truncate each sequence to a maximum length of 512 tokens.

6 Results and Discussion

In Figure 2, we describe trends in performance. Baseline AUCs are provided in the caption. Note that lower AUCs for diagnosis prediction reflect the comparative difficulty of these tasks. Figure 2a shows performance relative to the baseline model for varying JSD penalties. Performance decreases up to 6% as the penalty increases, with the exception of ASD, on which the model performs about as well as or better than baseline for $\theta \in [0.4, 0.475]$. Thus, while some information may be lost during summarization, depending on the dataset, summarization may also serve to denoise the text, improving predictive performance.

From Figure 2b, we find that as the penalty is increased, spans become considerably shorter. Inspecting the results when $\theta = 0.25$, we found that the model tends to focus in on key words rather than phrases. From Figure 2c, we see that overlap also shrinks with span size. The effect is more rapid for the medical datasets, likely because the non-medical passages contain text throughout that is relevant to the sentiment of the passage, whereas medical notes contain information not relevant to the prediction task. A notable exception is asthma,

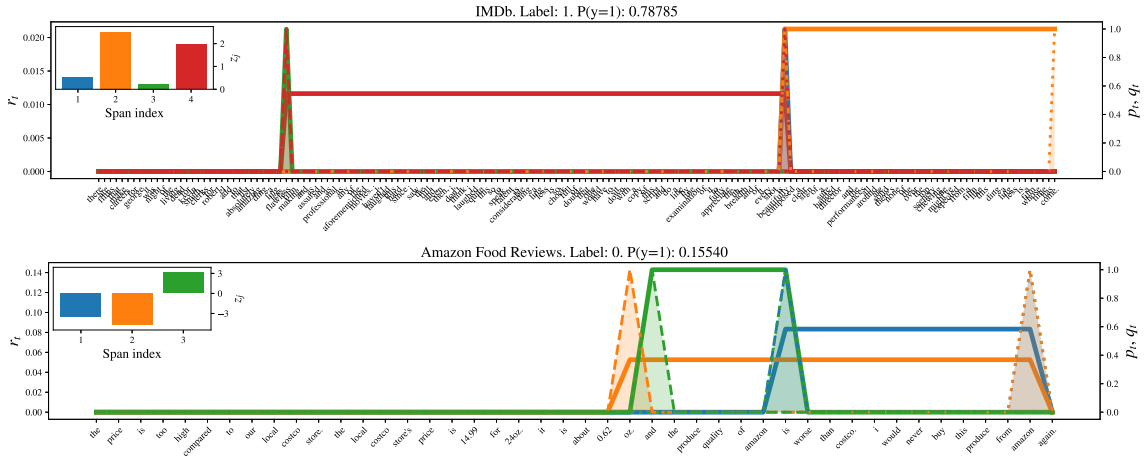


Figure 3: Example spans in the IMDb (top, positive sentiment) and Amazon (bottom, mixed sentiment) datasets. Colors represent the different spans ($J = 4$ for Amazon, $J = 3$ for IMDb). Solid lines denote $\mathbf{r}_j$ (heights of $\mathbf{r}_1$ and $\mathbf{r}_3$ rescaled for visualization purposes). Dashed lines denote $\tilde{\mathbf{p}}_j$ and dotted lines $\tilde{\mathbf{q}}_j$, with shading to resolve overlap. The inset plot shows the scores z_j, which are added to predict the log-odds of a positive label.

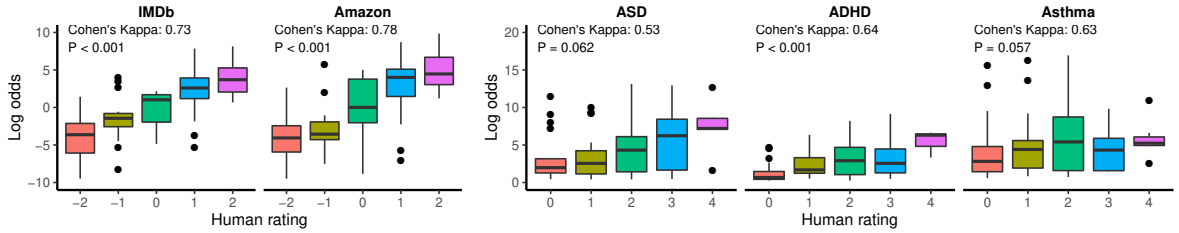


Figure 4: Predicted log-odds versus median sentiment (3 raters) for all five datasets.

which maintains a relatively constant span size and overlap, suggesting that diagnosing asthma requires identifying specific phrases (*e.g.*, "shortness of breath") that cannot be decomposed into individual words. Finally, we demonstrate in Appendix B that, for $J = 10$, AUC is, on average, greater but at the cost of greater sensitivity to θ.

Figure 3 provides an illustration of individual spans inferred by SpanPredict ($\theta = 0.5$). In the IMDb example (top), we see that the model captures two highly positive spans, each constituting 30-35% of the note, with words such as "professional," "laughed," and "appreciate" appearing in the red span. SpanPredict is also able to capture meanings of complex positive phrases, such as "chock full", "sure handed," "none of the over the top," and "time has come." The blue and green spans each cover only a single word; however, these words – "flawless" and "beautifully" – have significant positive connotation. This is a feature our model shares with (Shen et al., 2018), which also picks out individual tokens.

The Amazon review (Figure 3, bottom) contains mixed sentiment. The green span contains the word "quality," which, akin to words such as "care" or "workmanship," is slightly positive. However, the blue span is filled with negative phrases. This is reflected in the z_j scores in the inset plot, which are added to predict the log-odds of a positive label. We find that z_j is negative for the blue span while positive for the green span. The orange span is most negative, suggesting that the model is able to synthesize information from the blue and green spans it overlaps to extract an overall meaning.

Figure 4 shows the human evaluation results. For each span, we computed the median rating among the 3 reviewers and performed a non-parametric ANOVA (Kruskal-Wallis test) to assess agreement with model-predicted scores. Statistically significant differences in means ($p < 0.001$) were present in the IMDb, Amazon, and ADHD datasets, but not the ASD and Asthma datasets. Given our model's high agreement with human raters in the IMDb and Amazon tasks, the lower agreement observed on the medical diagnosis tasks may indicate that our model is identifying descriptive risk factors not familiar to our clinical raters. This hypothesis, which was suggested by our clinical collaborators, will be explored further in subsequent work. To measure inter-rater reliability, we computed Cohen's kappa

for each dataset – IMDb: 0.73, Amazon: 0.78, ASD: 0.53, ADHD: 0.64, Asthma: 0.63. These values illustrate the difficulty of evaluating the clinical notes compared to the review datasets.

7 Conclusions

We have introduced the task of *predictive extraction*, in which document labels are predicted from extracted contiguous segments of text called *spans*. We presented SpanPredict, which constructs span representations nonparametrically from contextualized embeddings by predicting start and end positions using linear attention. Our model is straightforward to tune, and assigns interpretable span scores that are added together to predict the log-odds of a positive label. Model performance and span quality are evaluated on two non-medical and three medical datasets. Notably, we observe high correlation between human span ratings and model-predicted span scores, particularly in the non-medical datasets, illustrating that our model selects meaningful spans and scores them accurately. Discrepancies between human ratings and model predictions in the medical datasets may suggest that our model is identifying condition-specific risk factors that are unfamiliar to trained clinicians. Future work will consider prediction and span extraction from a collection of documents rather than individual documents, allowing descriptive risk factors to be extracted from patient medical histories. Clinical findings consistently highlighted by SpanPredict will be analyzed as possible risk factors via standard statistical methods. Additionally, whereas SpanPredict identifies a set of spans sufficient to predict the label, future work will explore methods for ensuring that *all* predictive spans are identified.

Ethical considerations

This paper introduced the problem of predictive extraction, which attempts to identify distinct spans of text within a document that, taken together, are sufficient to predict its associated label. Its positive impact can best be described within the context of disease classification from narrative clinical text. For example, ASD is a classically difficult condition to diagnose, as its symptoms are often behavioral, rather than physiological, making clinical notes critical for classification. Focus on classification alone, however, is not sufficient, as a clinical decision support tool requires a level of interpretability to assure clinicians that the model is not relying on data artifacts that are not clinically meaningful or generalizable. This requirement is present in many document classification tasks, including the scoring of food or movie reviews. Our newly introduced algorithm, SpanPredict, addresses this need by identifying important and unlabeled predictive phrases without substantially worsening classification performance. As such, SpanPredict can be used as a real-time decision aid, providing narrative summaries optimized for disease classification, thus leading to faster diagnoses and long-term improvements in function, while minimizing healthcare cost and utilization.

While the positive impact of our contribution is clear, there are potential negative consequences related to biases in training. When algorithms are trained on patient datasets that are incomplete or under-/mis-representative of certain populations, they can develop discriminatory biases in their outcomes. When considering clinical notes, there is also potential for biased language in patient medical records related to race and ethnicity, including perpetuating of negative stereotypes, blaming a patient for their symptoms, or casting doubt on patient reports and experience. This biased language likely changes the context of words and may negatively impact classification performance. This is of particular importance in ASD, where white children with ASD receive their diagnoses substantially earlier Black children with ASD. Ignoring these biases might create self-fulfilling prophecies that confirm existing social biases or create new applications of bias altogether. In light of these negative impacts, it will become critical to evaluate the performance of SpanPredict in various populations prior to being put in production, so that all biases are well-characterized. Nonetheless, the overall impact of the paper is a net positive as it advances the field of interpretable document classification, using a novel methodology that only requires labels for the classification.

Acknowledgements

This work was funded by NIMH R01 MH121329 (Geraldine Dawson and Guillermo Sapiro, Co-PI). We gratefully acknowledge the conceptual input of Guillermo Sapiro, Geraldine Dawson, and Scott Kollins in this work.

References

Emily Alsentzer and Anne Kim. 2018. Extractive summarization of ehr discharge notes. *arXiv preprint*

arXiv:1810.12085.

Jasmijn Bastings, Wilker Aziz, and Ivan Titov. 2019. Interpretable neural predictions with differentiable binary variables. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2963–2977, Florence, Italy. Association for Computational Linguistics.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. *arXiv preprint arXiv:1603.07252*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Michael C Hughes, Leah Weiner, Gabriel Hope, Thomas H McCoy Jr, Roy H Perlis, Erik B Sudderth, and Finale Doshi-Velez. 2017. Prediction-constrained training for semi-supervised mixture and topic models. *arXiv preprint arXiv:1707.07341*.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Kenton Lee, Shimi Salant, Tom Kwiatkowski, Ankur Parikh, Dipanjan Das, and Jonathan Berant. 2016. Learning recurrent span representations for extractive question answering. *arXiv preprint arXiv:1611.01436*.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. *arXiv preprint arXiv:1606.04155*.

Jennifer Liang, Ching-Huei Tsou, and Ananya Poddar. 2019. A novel system for extractive clinical note summarization using ehr data. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 46–54.

Jianhua Lin. 1991. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151.

Xiangan Liu, Keyang Xu, Pengtao Xie, and Eric Xing. 2018. Unsupervised pseudo-labeling for extractive summarization on electronic health records. *arXiv preprint arXiv:1811.08040*.

Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pages 142–150. Association for Computational Linguistics.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. 2016. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*.

Christopher D Manning, Christopher D Manning, and Hinrich Schütze. 1999. *Foundations of statistical natural language processing*. MIT press.

Julian John McAuley and Jure Leskovec. 2013. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In *Proceedings of the 22nd international conference on World Wide Web*, pages 897–908.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Ranking sentences for extractive summarization with reinforcement learning. *arXiv preprint arXiv:1802.08636*.

Papers with Code. 2020. IMDb Leaderboard. Https://paperswithcode.com/sota/sentiment-analysis-on-imdb.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

Jason Ren, Russell Kunes, and Finale Doshi-Velez. 2019. Prediction focused topic models via vocab selection. *arXiv preprint arXiv:1910.05495*.

Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Chunyuan Li, Ricardo Henao, and Lawrence Carin. 2018. Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. *arXiv preprint arXiv:1805.09843*.

Leslie N Smith. 2017. Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 464–472. IEEE.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

George Tucker, Andriy Mnih, Chris J Maddison, John Lawson, and Jascha Sohl-Dickstein. 2017. Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. In *Advances in Neural Information Processing Systems*, pages 2627–2636.

Yong Zhang, Meng Joo Er, Rui Zhao, and Mahardhika Pratama. 2016. Multiview convolutional neural networks for multidocument extractive summarization. *IEEE transactions on cybernetics*, 47(10):3230–3242.

Yue Zhang, Qi Liu, and Linfeng Song. 2018. Sentence-state lstm for text representation. *arXiv preprint arXiv:1805.02474*.

Appendix A. Example spans

In Tables 2 through 11, we list example spans selected from each of the corpora whose log-odds scores were highly positive or highly negative.

Table 2: Selected spans among top 100 positive scores for IMDb

Span Text	Score
this wonderful film is a love story, and shows that not all relationships are destined to last. even so they can be great worth the pain suffering of	+6.13
was born to play this role, and her performance will most likely be remembered as she is supported by an ideal cast, and the direction and design are tops. it doesn't get any better than this.	+6.12
in love with the cats break into song. with the song everybody wants to be a cat. thomas gets to love music like the other cats. thomas and really like each other. i loved this movie and i like the cats to	+6.12
i have nothing but good things to say about this tasteful and heartwarming film. i think that the effort of	+6.12
this remarkable film just gets better every time you watch it. a true cinematic work of art from a visionary director.	+6.11
a wonderful film that everyone interested in should see. but it's not a perfect or definitive work on the subject.	+6.11

Table 3: Selected spans among top 100 negative scores for IMDb

Span Text	Score
poor ward, so lovely, but so surely she's been better in other movies.	-6.71
this turgid film that i can think of. any proper film lover will have an almost impossible time trying to find any redeeming value in this crap, definitely one to avoid.	-6.71
in another of the dreadful horror films i seem so attracted to, we have a bunch of	-6.71
of the most annoying characters ever captured on film. this crap is an insult to movies and i almost never rate a movie i don't see from start to finish, but in this case the former is impossible. 2 10	-6.70
poor souls from wasting their time and or money with this movie. i [unk] it and wish i never even wasted the hard drive space. if i spent 10 bucks to see this in theaters i would kill	-6.70
i would ward off any temptation to view this movie, it is quite simply dull. the characters are predictable and the assassin is quite [unk] there is no tension, fun, no style or even a glimmer of	-6.68

Table 4: Selected spans among top 100 positive scores for Amazon

Span Text	Score
this for the first time recently and found it awesome i made a perfect vegetable curry, very flavorful and spicy. i'm going to make up another batch over the weekend, and put this curry paste on my list to order again.	+8.40
brew it correctly and you really get a beautiful cup of tea. i highly	+8.40
getting harder and harder to find in the stores so i'm stocking up from amazon. my toddler loves it and we use it on all meats. i make rice using coconut milk to serve with this and it's yum	+8.39
a very well built mole trap that works great when set correctly. the safety latch is a nice	+8.39
bowl. wonderful product and so nice that i can buy in bulk since we go through it so fast.	+8.39
a decent price depending if you do subscribe and save . these bars are great for my little ones, they love them, and they are a good healthy alternative to candy or cookies.	+8.39

Table 5: Selected spans among top 100 negative scores for Amazon

Span Text	Score
was terrible it tasted like we were licking an ashtray. it has a burnt grounds flavor. i highly recommend not wasting your money on this product.	-7.30
the one can i tasted and threw out to the food pantry.	-7.30
really stale items from amazon.com and this was one. unedible. beware of the quality of food items on this website that are on special as they can be very close to due dates or in this case, not expired but stale and unedible just the same.	-7.30
this product claims and hours of entertainment. my dog had it completely destroyed on 20 minutes. i'm completely disappointed.	-7.30
life threatening . undigested pieces of these chews were in his waste. i do not reccomend	-7.30
this is disgusting, it doesn't taste like watermelon at all. it's actually a blend of several different juices, plus the ascorbic acid, and the blend does not meld at all. it's just bitter, overly sweet, and has a nasty aftertaste.	-7.29

Table 6: Selected spans among top 100 positive scores for ASD

Span Text	Score
subjective intake chief complaint problems with sleep, inattention, and behavioral concerns both in the home and school setting. DATE, recently more anger and recent tic like behavior	+6.95
psychologist presenting problem NAME is a 3 year, 4 month old female who was referred for a neurodevelopmental assessment due to concerns regarding her overall development, behavior, and social emotional functioning and to assess for autism spectrum disorder	+6.82
problem list diagnosis • disruptive behavior disorder • impaired speech articulation • daytime enuresis • other subjective visual disturbances • hypermetropia of both eyes • adhd attention deficit	+6.81
problem list diagnosis • anemia of prematurity • history of colitis • meconium tox for thc • extreme immaturity of newborn, 27 completed weeks • nasal congestion of newborn • presumed	+6.78
motor delay DATE • hypotonia DATE • clasped thumb DATE • polydactyly DATE • developmental	+6.74
therapy NAME was seen for developmental support during rop eye exam today. the	+6.65

Table 7: Selected spans among top 100 negative scores for ASD

Span Text	Score
subjective NAME is a 5 y.o. female who presents for her 5 year well child visit. history was obtained today by father. concerns had om a few weeks ago. check her throat.	-5.91
CLINIC sick visit patient active problem list diagnosis • routine child health maintenance chief complaint patient presents with • fussy x several days. dad ? possible ear infection hpi has never	-5.90
NAME is a male child here for his 15 month well child visit. concerns none diet varied voiding and stooling well. past medical history active ambulatory	-5.87
evaluation was performed today unless otherwise noted. assessment encounter diagnosis name primary? • regular astigmatism of both eyes yes plan 1. astigmatism	-5.83
breast bottle vitamins formula no 0 oz. per feeding of feedings in 24 hours 0 solids yes juice no elimination patterns loose sleep sleeps all night development	-5.81
subjective is a 17 m.o. male and is here for a well child visit. history was obtained today by is 17 months old and is here for a 15 month exam. he is doing well.	-5.76

Table 8: Selected spans among top 100 positive scores for ADHD

Span Text	Score
behavioral parent training patient and family response to interventions we discussed parenting stress and consistent plan to move to pdi 4 next week. objective mental status exam behavioral	+5.33
sensory disorder subjective pain assessment no pain. patient caregiver comments NAME is reportedly behind in reading is likely going to need summer school reports of difficulty keeping place when reading. family considering testing to rule out adhd. objective goals demonstrate improved	+5.23
outpatient prescriptions on file prior to visit medication sig dispense refill • clonidine hcl catapres 0.1 mg tablet take 0.1 mg by mouth nightly. 2 • melatonin 3 mg tablet take 3 mg by mouth nightly. • methylphenidate concerta 54 mg	+5.18
list diagnosis • dyslexia, developmental • intermittent asthma • right elbow pain • food allergy • fire ant sting past medical history active ambulatory problems diagnosis date noted • dyslexia, developmental DATE • intermittent asthma DATE	+5.18
diagnosis • gestational age, NUMBER weeks • apnea of prematurity • breech presentation • unconjugated hyperbilirubinemia	+5.15
5 y.o. male who presents with h o developmental delay, speech disorder, sensory and fine motor disorders, challenging behavior and who would likely continue to benefit from continued evaluation to address identified concerns. will obtain information to assist r o adhd	+4.92

Table 9: Selected spans among top 100 negative scores for ADHD

Span Text	Score
5 y.o. female presenting with 3 days of runny nose, congestion, sore throat, cough. today she woke up with a little bit of drainage from her right eye. as the day has gone	-3.08
presents for an established patient office visit here for wcc. is doing well hopes to go to early hs past medical history past medical	-3.06
y.o. female is here today for the influenza vaccine. vaccine administered today influenza quad patient guardian reviewed or provided the hard copy of the YEAR influenza vaccine information	-3.01
diagnosis • healthy infant or child • abdominal pain, periumbilical current outpatient	-2.99
3 y.o. female here for evaluation of woke up with abdominal pain temp to 100 at daycare mother says she vomited x 1 this morning her sister was seen last week for strep patient active	-2.95
subjective pain assessment no pain. patient caregiver comments NAME is doing very well in swim and basketball. he coped very well when parents were recently out of the in the care of extended family. objective goals NAME will... 1. open snack	-2.91

Table 10: Selected spans among top 100 positive scores for Asthma

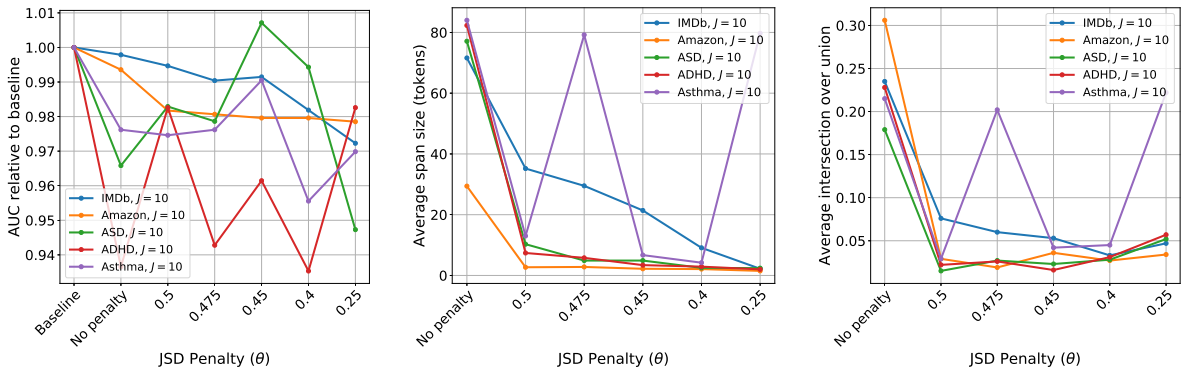
Span Text	Score
history diagnosis date • rad reactive airway disease , unspecified history reviewed. no pertinent surgical history. family history problem relation age of onset • asthma mother outpatient prescriptions marked as taking for the DATE encounter office visit with NAME medication sig • albuterol	+7.75
pt goal home airway clearance dates start DATE, description patient will increase airway clearance at home to three times a day when	+7.74
pediatric icu progress note DATE hospital day 1 icu admission indication 3 m.o. female with principal problem respiratory distress active problems hypotonia laryngomalacia chromosomal abnormality NAME is a 3	+7.68
patient active problem list diagnosis • gestation period, 28 weeks • respiratory insufficiency • breech birth overnight none medications caffeine citrated 5 mg kg	+7.63
CLINIC sick visit patient active problem list diagnosis • tof tetralogy of fallot • sacral dimple • chromosome abnormalities chief complaint patient presents with • nasal congestion • cough • sneezing • breathing problems hpi NAME is	+7.58
inhalation started as ordered and held near nose and mouth for ventilation administration.	+7.53

Table 11: Selected spans among top 100 negative scores for Asthma

Span Text	Score
vaccine less than 7yo im • hib prp omp conjugate vaccine 3 dose im pedvaxhib • pneumococcal conjugate vaccine 13 valent im prevnar 13 • hepatitis a vaccine pediatric	-5.67
19 m.o. here today with a red swollen slightly tender distal right 4th finger. patient injured that finger in a cabinet roughly 6 days ago. last 2 to 3 days it	-5.66
20 m.o. male who presents today for the evaluation of chief complaint patient presents with • cough • nasal congestion history was obtained today by father. uri symptoms for gt 1 week. no worse but	-5.65
motor runs and climbs well throws a ball stacks 3 or more blocks fine motor uses spoon and cup scribbles, tries to use	-5.61
21 m.o. with problems with gait and balance, some crying at night. grandmother thinks he is having difficulty with constant falling and running with some bruising of his head or face. no limp, no deformity or swelling. he has	-5.60
20 m.o. female brought in by mother. hpi NAME presents with a 2 days history of of the fever, with maximum temperature of 104. she was seen in er 2 days ago and diagnosed with viral illness. she is still running fever, has runny nose and is fussy.	-5.60

Appendix B. AUC, span size, and span overlap with $J = 10$

Figure 5 illustrates the performance of our model for a fixed value of $J = 10$, larger than that chosen for each dataset in the main paper (4, 3, and 7 for IMDb, Amazon, and the health datasets, respectively). While AUC is generally higher for each dataset compared to that obtained with a smaller value of J, we find that span length and overlap are now more sensitive to θ and drop more rapidly as θ is increased. In practice, we employ smaller values of J and adjust θ to achieve a desired level of span size and overlap to (1) allow for finer control of the tradeoff in performance, span size, and span overlap, and to (2) avoid overparameterizing our model.



(a) AUC relative to baseline (b) Average span size (c) Average intersection-over-union

Figure 5: Trends in performance. Baseline AUCs – IMDb: 0.938, Amazon: 0.931, ASD: 0.702, ADHD: 0.804, Asthma: 0.630. Performance tends to drop slightly as θ is decreased, but spans become more concise and distinct.

Appendix C. Training loss *vs.* epoch

In Figures 6 through 20, we show the traces of training loss as a function of epoch for each experiment. For all models except the baseline, we separate our loss into two components: one for the negative log likelihood (denoted "loglik") and another for the negative JSD (denoted "uniqueness"). In each experiment we find that the *lower* bound (LB) for *negative* JSD is not violated, providing experimental support for our proof of an upper bound on the modified JSD. Note that the bottom right subplot in each non-baseline model – titled "span_size" – can be ignored as this is related to a feature that was ultimately not incorporated into the model. There is no contribution to the total loss from this component; hence, the value is zero across all epochs.

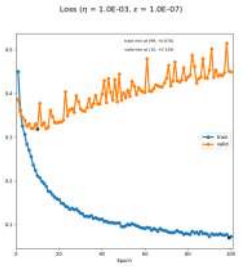
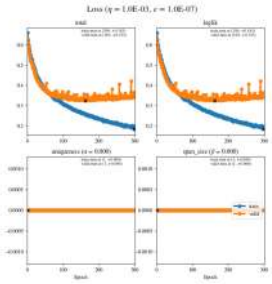
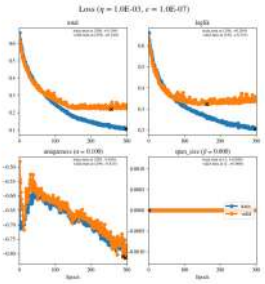


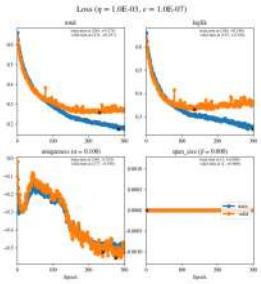
Figure 6: IMDb: Baseline



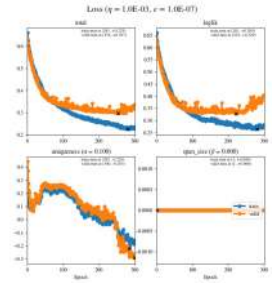
(a) No penalty



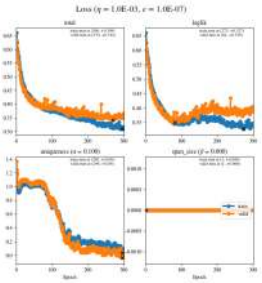
(b) $\theta = 0.5$, LB = -1.39



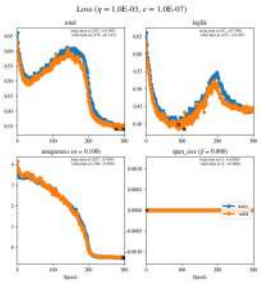
(c) $\theta = 0.475$, LB=-1.32



(d) $\theta = 0.45$, LB=-1.25

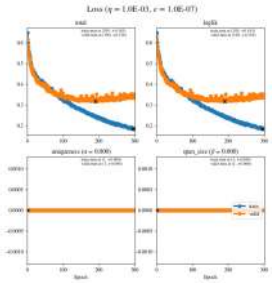


(e) $\theta = 0.4$, LB=-1.11

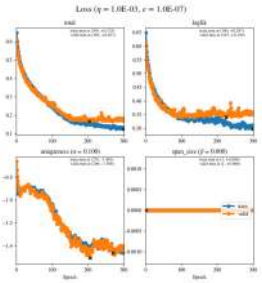


(f) $\theta = 0.25$, LB=-0.69

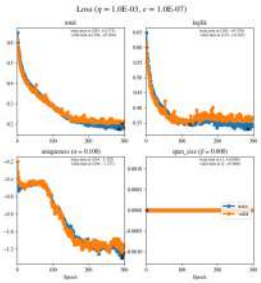
Figure 7: IMDb: $J = 4$



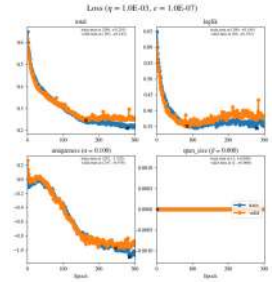
(a) No penalty



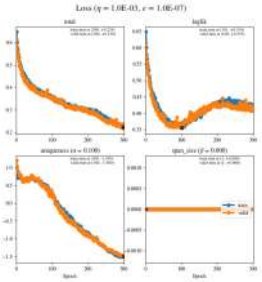
(b) $\theta = 0.5$, LB = -2.30



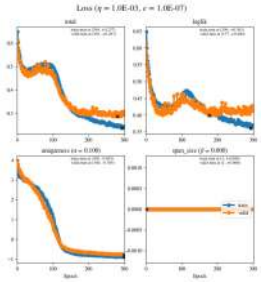
(c) $\theta = 0.475$, LB=-2.19



(d) $\theta = 0.45$, LB=-2.07



(e) $\theta = 0.4$, LB=-1.84



(f) $\theta = 0.25$, LB=-1.15

Figure 8: IMDb: $J = 10$

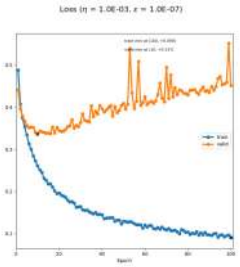
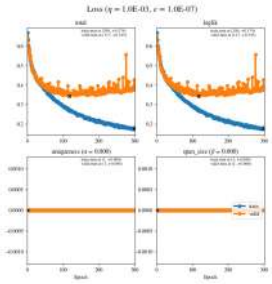
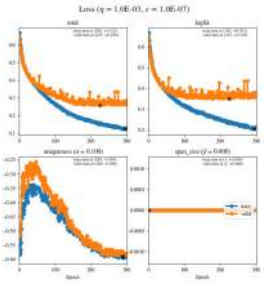


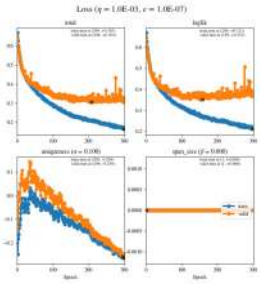
Figure 9: Amazon: Baseline



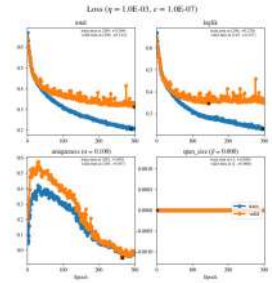
(a) No penalty



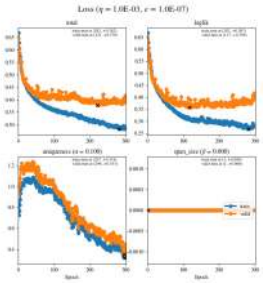
(b) $\theta = 0.5$, LB = -1.10



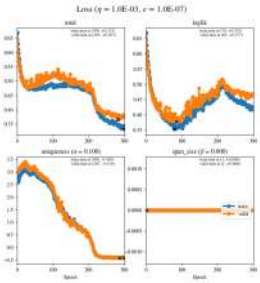
(c) $\theta = 0.475$, LB=-1.04



(d) $\theta = 0.45$, LB=-0.99

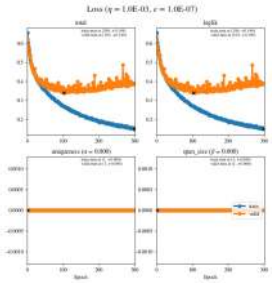


(e) $\theta = 0.4$, LB=-0.88

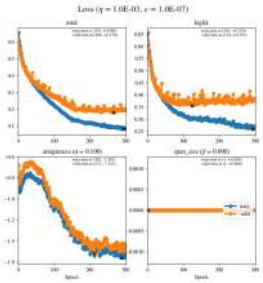


(f) $\theta = 0.25$, LB=-0.55

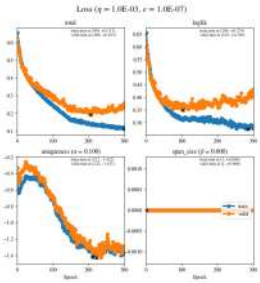
Figure 10: Amazon: $J = 3$



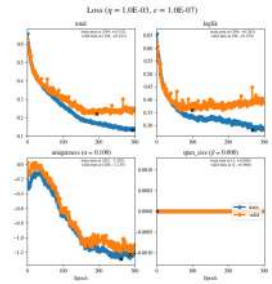
(a) No penalty



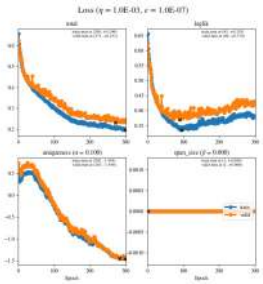
(b) $\theta = 0.5$, LB = -2.30



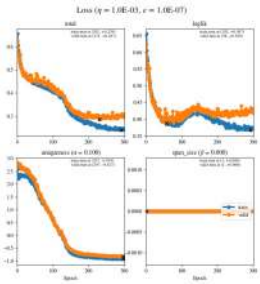
(c) $\theta = 0.475$, LB=-2.19



(d) $\theta = 0.45$, LB=-2.07



(e) $\theta = 0.4$, LB=-1.84



(f) $\theta = 0.25$, LB=-1.15

Figure 11: IMDb: $J = 10$

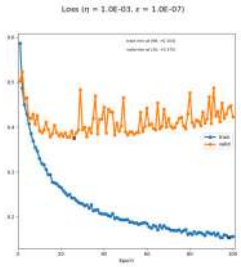
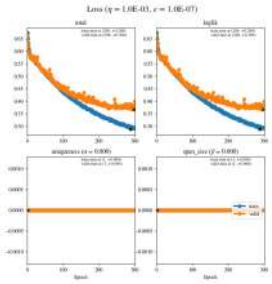
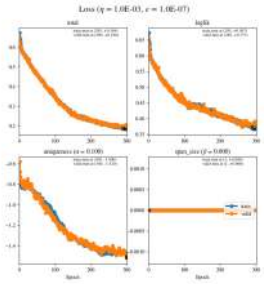


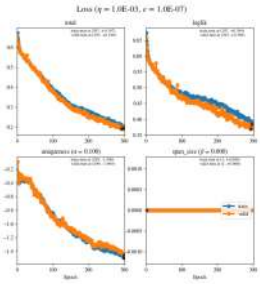
Figure 12: ASD: Baseline



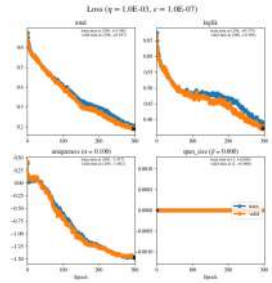
(a) No penalty



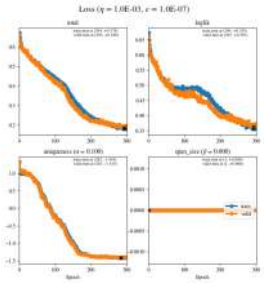
(b) $\theta = 0.5$, LB = -1.95



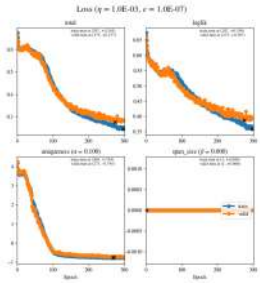
(c) $\theta = 0.475$, LB=-1.85



(d) $\theta = 0.45$, LB=-1.75

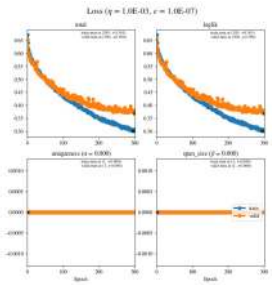


(e) $\theta = 0.4$, LB=-1.56

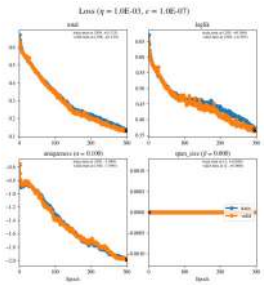


(f) $\theta = 0.25$, LB=-0.97

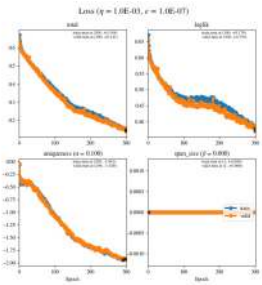
Figure 13: ASD: $J = 7$



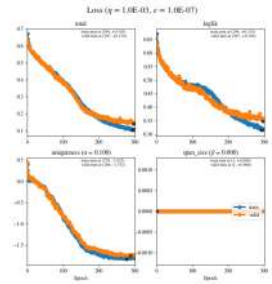
(a) No penalty



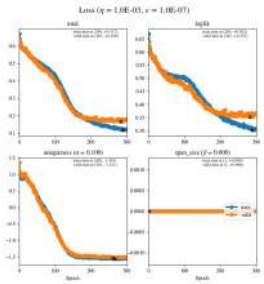
(b) $\theta = 0.5$, LB = -2.30



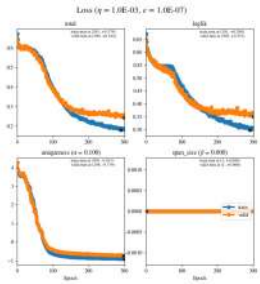
(c) $\theta = 0.475$, LB=-2.19



(d) $\theta = 0.45$, LB=-2.07



(e) $\theta = 0.4$, LB=-1.84



(f) $\theta = 0.25$, LB=-1.15

Figure 14: ASD: $J = 10$

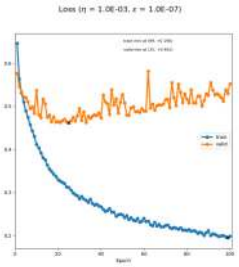
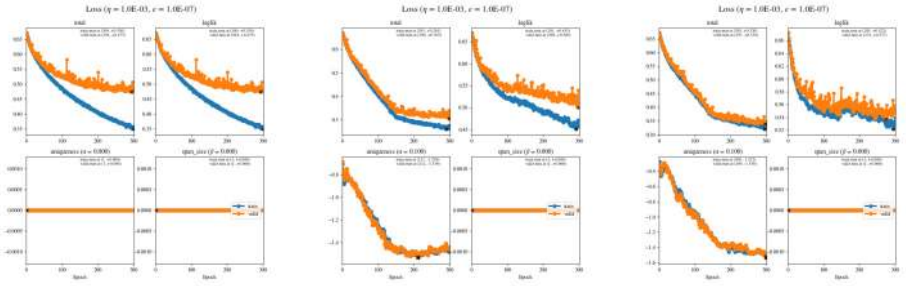
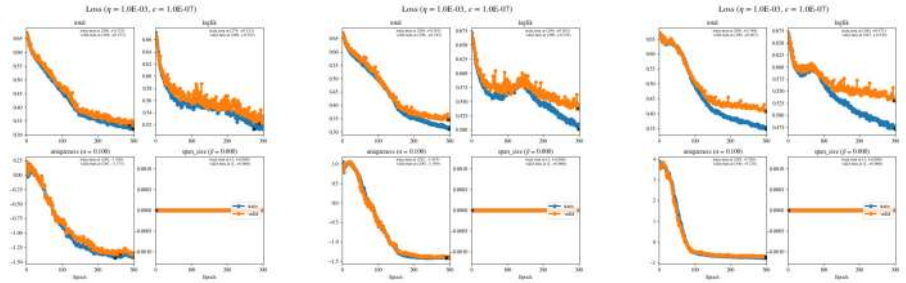


Figure 15: ADHD: Baseline

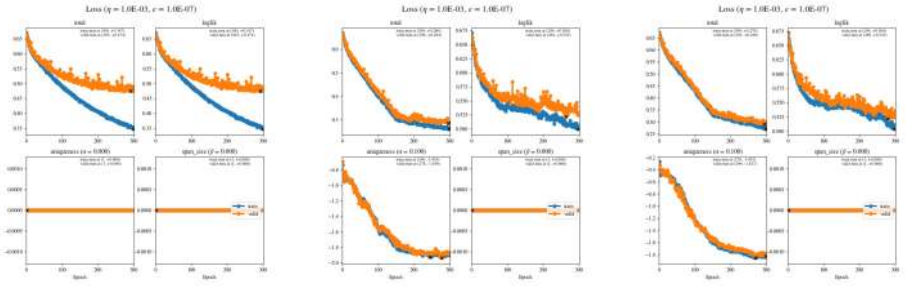


(a) No penalty (b) $\theta = 0.5$, LB = -1.95 (c) $\theta = 0.475$, LB=-1.85

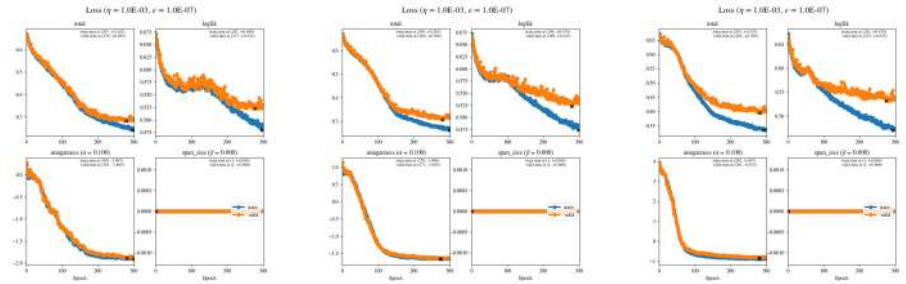


(d) $\theta = 0.45$, LB=-1.75 (e) $\theta = 0.4$, LB=-1.56 (f) $\theta = 0.25$, LB=-0.97

Figure 16: ADHD: $J = 7$



(a) No penalty (b) $\theta = 0.5$, LB = -2.30 (c) $\theta = 0.475$, LB=-2.19



(d) $\theta = 0.45$, LB=-2.07 (e) $\theta = 0.4$, LB=-1.84 (f) $\theta = 0.25$, LB=-1.15

Figure 17: ADHD: $J = 10$

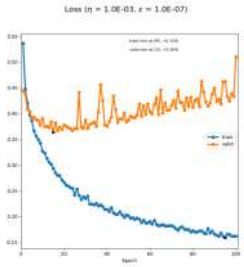
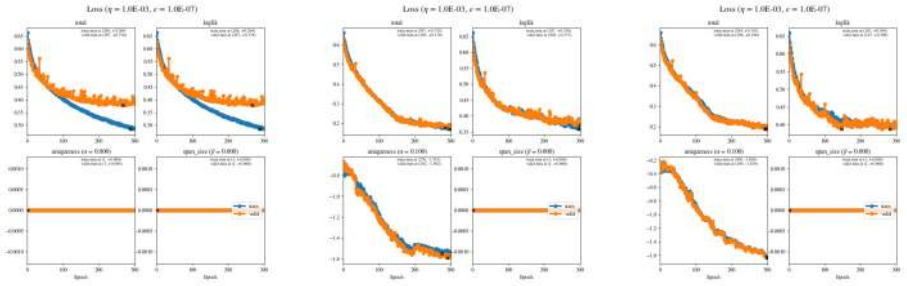
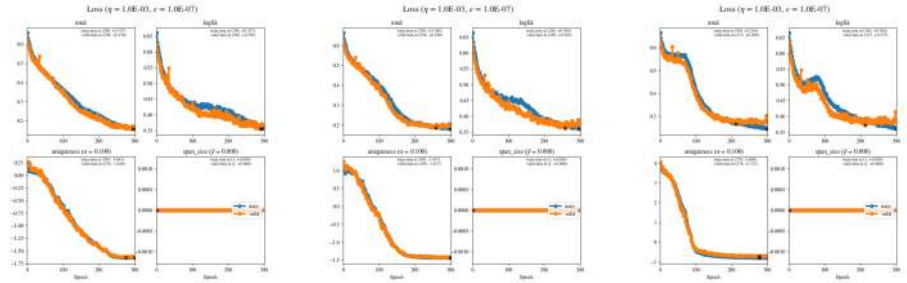


Figure 18: Asthma: Baseline

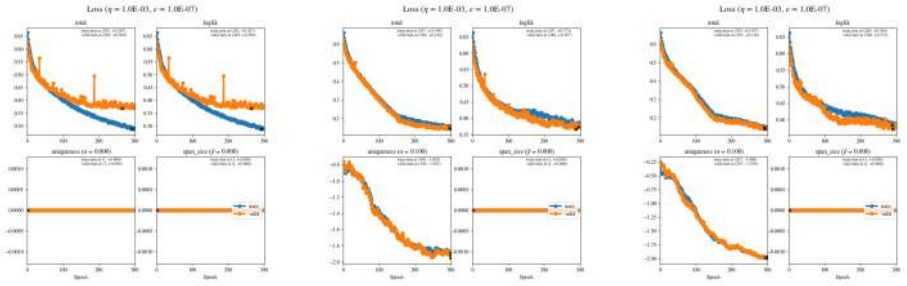


(a) No penalty (b) $\theta = 0.5$, LB = -1.95 (c) $\theta = 0.475$, LB=-1.85

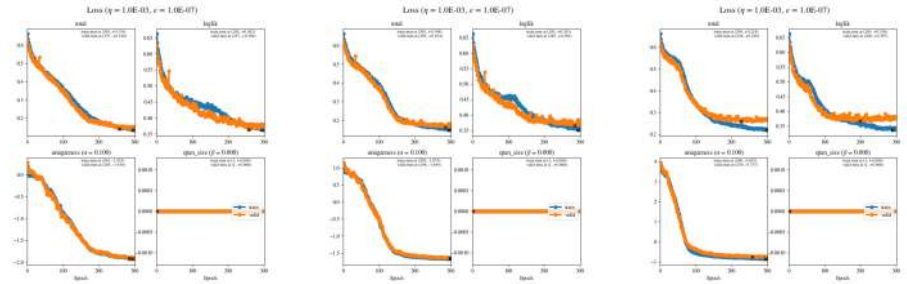


(d) $\theta = 0.45$, LB=-1.75 (e) $\theta = 0.4$, LB=-1.56 (f) $\theta = 0.25$, LB=-0.97

Figure 19: Asthma: $J = 7$



(a) No penalty (b) $\theta = 0.5$, LB = -2.30 (c) $\theta = 0.475$, LB=-2.19



(d) $\theta = 0.45$, LB=-2.07 (e) $\theta = 0.4$, LB=-1.84 (f) $\theta = 0.25$, LB=-1.15

Figure 20: Asthma: $J = 10$

Appendix D. Code

The IMDb and Amazon datasets were processed using the following scripts:

- `imdb_baseline.ipynb` – to train baseline model on IMDb and obtain AUC
- `imdb_main.ipynb` – to train SpanPredict model on IMDb and obtain (1) AUC and (2) average span length
- `imdb_spans.ipynb` – to obtain spans for human evaluation and supplementary material from IMDb
- `amazon_baseline.ipynb` – to train baseline model on Amazon Food Reviews and obtain AUC
- `amazon_main.ipynb` – to train SpanPredict model on Amazon Food Reviews and obtain (1) AUC and (2) average span length
- `amazon_spans.ipynb` – to obtain spans for human evaluation and supplementary material from Amazon Food Reviews
- `imdb_amazon_IoU.ipynb` – to obtain average intersection over union from both datasets on non-baseline models

The ASD, ADHD, and Asthma datasets were processed using the following scripts:

- `asd_adhd_asthma_baseline.ipynb` – to train baseline model on ASD / ADHD / Asthma datasets and obtain AUC
- `asd_adhd_asthma_main.ipynb` – to train SpanPredict model on ASD / ADHD / Asthma datasets and obtain (1) AUC and (2) average span length
- `asd_adhd_asthma_spans.ipynb` – to obtain spans for human evaluation and supplementary material from ASD / ADHD / Asthma datasets
- `asd_adhd_asthma_IoU.ipynb` – to obtain average intersection over union from ASD / ADHD / Asthma datasets on non-baseline models

To train a SpanPredict model from scratch, use the `{imdb, amazon}_main` script. J, α, and θ can be set using the `num_spans`, `uniqueness_weight_`, and `JSD_weight` fields of the `model_options` variable, respectively. Note that `JSD_weight` is defined as a tuple: $(\theta, 1-\theta)$. For instance, to set $J = 10$, $\alpha = 0.1$, and $\theta = 0.25$, use: `model_options = {'num_spans': 10, 'uniqueness_weight_': 0.1, 'JSD_weight': (0.25, 0.75), ...}`. All other parameters of `model_options` and all parameters of `training_options` can be left as is. This will create a database which stores every span extracted from every document in the testing corpus. The `{imdb, amazon}_spans` and `{imdb, amazon}_IoU` scripts can then be used to obtain (1) example spans (*e.g.*, those tabulated in Appendix A and those selected for human evaluation) and (2) intersection-over-union scores, respectively.

Note that due to privacy restrictions, we are unable to share models trained on the ASD, ADHD, and Asthma datasets. However, we provide baseline, no penalty, and $\theta = 0.5$ model checkpoints for the IMDb and Amazon datasets (with $J = 4$ and $J = 3$, respectively, for the non-baseline models). These were produced using the `{imdb, amazon}_main` script and can be loaded by making the appropriate modifications.

For all our models, we employ 100 dimensional GloVe embeddings, which can be found here: http://nlp.stanford.edu/data/glove.6B.zip.