Detecting Local Insights from Global Labels: Supervised and Zero-Shot Sequence Labeling via a Convolutional Decomposition

Allen Schmaltz
Department of Epidemiology
Harvard University
aschmaltz@hsph.harvard.edu

We propose a new, more actionable view of neural network interpretability and data analysis by leveraging the remarkable matching effectiveness of representations derived from deep networks, guided by an approach for class-conditional feature detection. The decomposition of the filter-n-gram interactions of a convolutional neural network (CNN) and a linear layer over a pre-trained deep network yields a strong binary sequence labeler, with flexibility in producing predictions at—and defining loss functions for—varying label granularities, from the fully supervised sequence labeling setting to the challenging zero-shot sequence labeling setting, in which we seek token-level predictions but only have document-level labels for training. From this sequence-labeling layer we derive dense representations of the input that can then be matched to instances from training, or a support set with known labels. Such introspection with inference-time decision rules provides a means, in some settings, of making local updates to the model by altering the labels or instances in the support set without re-training the full model. Finally, we construct a particular K-nearest neighbors (K-NN) model from matched exemplar representations that approximates the original model's predictions and is at least as effective a predictor with respect to the ground-truth labels. This additionally yields interpretable heuristics at the token level for determining when predictions are less likely to be reliable, and for screening input dissimilar to the support set. In effect, we show that we can transform the deep network into a simple weighting over exemplars and associated labels, yielding an introspectable—and modestly updatable—version of the original model.

1. Introduction

The promise and peril of deep learning in computational linguistics, and AI in general, would seem, on the surface, to be that the strong effectiveness of the large neural networks is unavoidably accompanied by inscrutable model predictions. The models are often right, but when they are wrong, it is difficult to ascertain why, and furthermore, it is typically not obvious how to course-correct a model when errors are discovered, beyond altogether abandoning the model. The non-identifiable (cf., Hwang and Ding 1997; Jain and Wallace 2019) and extraordinarily large number of parameters suggest

a lost cause, in general, for tracing model predictions back to particular parameters, and it would seem then that deep networks are of limited use in settings where interpretability is paramount. However, interestingly, and surprisingly, we show that there is nonetheless a sense in which the deep networks can be leveraged to create a notion of actionable interpretability against the data that is not necessarily possible with simpler, less expressive models alone, and may yield precisely the characteristics desired in certain real-world applications. By leveraging the strong pattern matching behavior and the dense representations of the deep networks, we can form a mapping between test instances and training instances with known labels, which enables introspection of the model with respect to the data. In some settings, we can then update the model by updating the data and labels in these mappings. Interestingly, in this way, the application of deep neural networks begins to resemble some of the classic instance-based and metric learning methods from machine learning, as well as the exemplar systems (Clark 1990) from an earlier era of AI, but with less dependence on human-mediated feature engineering, which may prove critical for applications with high-dimensional input, at the very least as tools for data analysis.

A model for analyzing a natural language data set ideally needs some facility for class-conditional feature detection at the word level. However, the compositional, high-dimensional nature of language makes feature detection a challenging endeavor, with further empirical complications arising from the need to label at a granularity that is typically more fine-grained than many existing human-annotated data sets. We propose and demonstrate that a single-layer, one-dimensional, kernel-width-one max-pooled convolutional neural network (CNN) and a linear layer, as the final layer of a network, can be trained for document-level classification, and then decomposed in a straightforward way to produce token-level labels. This particular set of operations over a CNN and a linear layer yields flexibility in learning and predicting at disparate label resolutions and is efficient and simple to calculate and train. It can readily replace the standard final linear layer often used for classification in Transformer models (Vaswani et al. 2017), adding the properties described here. We empirically show across tasks, using data sets that have token-level labels for verification, that it yields surprisingly sharp token-level binary detections even when trained at the document level, when the input to the layer is a large, masked-language-model-trained BERT model (Devlin et al. 2019).

Feature detection in this way is a useful tool for analyzing data sets, detecting rather subtle distributional differences within documents that can be otherwise challenging to find at scale. Further, we show that the CNN filter applications corresponding to the token-level predictions are effective dense representations of the model predictions, with which we can form a mapping between test predictions and instances with known labels. We find qualitatively and quantitatively that the matches correspond to similar features in similar contexts, at least when the distances between representations are low. Finally, without loss of predictive effectiveness, we can altogether replace the model's output with a simple weighting over exemplar representations, converting the deep network into a K-nearest neighbor (K-NN) model, with concomitant benefits for interpretability, and straightforward heuristics for detecting domain-shifted and out-of-domain data.

In summary, this work contributes the following new approaches:

1. We present a new, effective model for supervised and zero-shot binary sequence labeling. We evaluate on token-level annotations for grammatical error detection and diff annotations on a sentiment data set,

          detecting both sentiment features and surprisingly, also subtle
          re-annotation artifacts.

2.   We propose a method for data and model analysis via dense
          representation matching, **exemplar auditing**, enabled by our binary
          sequence labeling method, creating inference-time decision rules linking
          feature-level exemplar representations and associated predictions from
          test with representations from a support set with known labels. We show
          that in some settings we can make local updates to the model by updating
          the data and labels in the support set without re-training the full model.

3.   We approximate the model's token-level output with a K-NN over the
          support set that is at least as effective as the original model, and can be
          used as an interpretable substitute for the original model. Incorrect model
          predictions tend to also be more difficult to approximate; our proposed
          approach yields simple, understandable heuristics at the token level for
          determining when predictions are less likely to be reliable, and for
          screening input unlike that seen in the support set.

We proceed by first introducing the notation for the tasks across label resolutions (Section 2) and the core methods (Section 3) used across all experiments, and then we apply these ideas to three tasks. First, we demonstrate effectiveness on the challenging, well-defined error detection task (Section 4), which enables careful examination of the behavior using available token-level labels. Next, we use sentiment data that has been usefully re-annotated via local changes (Section 5) to further examine updating the support set over domain-shifted data, and to motivate and analyze our approach for constraining out-of-domain data in the context of an existing approach for robust classification. Finally, we also use these sentiment data sets to examine the model's ability to detect subtle distributional changes across re-annotated and original data (Section 6), discovering features that are not readily detectable at scale without model-based assistance.

2. Tasks

Given a document, which may consist of a single sentence, we seek binary labels over the words in the document. For learning such a model, we may be given training examples with associated labels for each of the "words,"[1] which is the standard fully supervised binary sequence labeling setting, or we may only be given document-level labels, which is the zero-shot binary sequence labeling setting. This latter setting corresponds to notions of feature detection for document-level classification models, enabling quantitative evaluation when given token-level labeled held-out data.

Supervised Binary Sequence Labeling. Specifically, in the standard fully supervised sequence labeling setting, we are given a training data set $\mathbb{D}^* = \{(x_d, y_d) | 1 \le d \le |\mathbb{D}^*|\}$ of $|\mathbb{D}^*|$ documents paired with their corresponding token-level ground-truth labels. Each of N tokens in a document, $x = x_1, \ldots, x_n, \ldots, x_N$, has a known token-level label, $y_n \in \{-1, 1\}$. We seek a learned mapping, $x \mapsto \hat{y}$, for predicting the labels for a given

---

1 Hereafter, we will tend to use "token" instead of "word," as the lowest resolution of the input will be
  determined by the tokenization scheme of the particular data set.

document: At inference, we are given a new, previously unseen document instance, $x_{|\mathbb{D}^*|+1}$, over which we predict $\hat{y}_{|\mathbb{D}^*|+1} = \hat{y}_1, \ldots, \hat{y}_n, \ldots, \hat{y}_N$, the token-level labels for each token in the document. We will subsequently drop the subscript label, "$|\mathbb{D}^*| + 1$", on test-time instances when the distinction from training is otherwise unambiguous. We aim to minimize the distance between the predicted $\hat{y}$ and the ground-truth y.

Throughout we use * to indicate a data set includes, or a model otherwise has access to, token-level labels. Otherwise, the label signal is limited to the document level, with the exception of clearly indicated reference experiments simply tuning the decision boundary of document-level models with a limited number of token-level labels.

Document-level Binary Classification. In the standard document-level classification setting, we are given a training data set $\mathbb{D} = \{(x_d, Y_d) | 1 \leq d \leq |\mathbb{D}|\}$ of $|\mathbb{D}|$ documents paired with their corresponding document-level ground-truth labels. Token-level labels are not present in $\mathbb{D}$. At inference, we seek to predict $\hat{Y}$ given a new, unseen document x, via the learned mapping $F : x \mapsto \hat{Y}$. We aim for $\hat{Y}$ to be close to the true document-level label, $Y \in \{-1, 1\}$.

Zero-shot Binary Sequence Labeling. The zero-shot binary sequence labeling models have access to the same training data set $\mathbb{D}$ as in the standard document-level classification task. However, at inference, we then seek to predict the *token-level* labels, $\hat{y}$, for each token in the new document instance x, via a mapping $x \mapsto \hat{y}$, even though we can only query the document-level labels of $\mathbb{D}$ during training. In other words, the learning signal is the same for document-level classification and zero-shot sequence labeling, but the inference-time task is the same in the zero-shot sequence labeling and fully supervised sequence labeling settings.

We will be primarily concerned with analyzing the sequence labeling settings. We also report document-level classification results for a subset of the zero-shot sequence labeling models, illustrating how the proposed token-level predictions can be used to analyze and constrain typical text data sets that only have labels at the document level, rather than at finer-grained resolutions, at least at scale.

3. Methods

We propose a new method for class-conditional feature detection from a large, expressive deep network that enables the interlinked view of interpretability, constrained inference, and updatability via an external database introduced in this work. We demonstrate that a particular max-pool attention-style mechanism from a CNN and a linear layer over a deep network enables the following:

1. We show that we can derive token-level predictions across the full document, $f(x_1), \ldots, f(x_n), \ldots, f(x_N)$, from the document-level prediction, $F(x)$. This decomposition provides flexibility in learning and analyzing at varying label resolutions.

2. We further show that the token-level predictions can themselves be approximately decomposed via $f(x_n) \approx f(x_n)^{KNN}$, where $f(x_n)^{KNN}$ is an explicit weighting over a set of nearest exemplar representations and their associated labels and predictions.

We proceed by first introducing the base document-level classifier (Section 3.1). We then introduce the approach for deriving token-level predictions from the document-level classifier (Section 3.2). We show how this can be used for supervised labeling (Section 3.3); yields flexibility in adding task-specific priors (Section 3.4); and provides a means of aggregate feature extraction for analyzing data sets (Section 3.5). Next, we introduce the approach for mapping a test-time prediction to a database of exemplars by leveraging dense representations coupled with the class-conditional feature detection (Section 3.6), before introducing the K-NN approximations (Section 3.7).[2] Figure 1 provides a high-level overview of the approaches further detailed below.

3.1 CNN Binary Classifier Over a Deep Network: Document-Level Predictions

We use a CNN architecture similar to that of Kim (2014) over a pre-trained Transformer model (Devlin et al. 2019) and fine-tuned word embeddings as our document-level classifier, F. Each token $x_n \in x$ in the document, including padding symbols as necessary, is represented by a D-dimensional vector, $t_n = (e^{BERT}, e^{word})$, the concatenation of the top hidden layer(s) of a Transformer and a vector of word embeddings, $D = |e^{BERT}| + |e^{word}|$. The convolutional layer is then applied to this $\mathbb{R}^{D \times N}$ matrix, using a filter of width Q, sliding across the dense vectors corresponding to the Q-sized n-grams of the input. The convolution results in a feature map $h_m \in \mathbb{R}^{N-Q+1}$ for each of M total filters.

We then compute

$$g_m = \max \mathrm{ReLU}(h_m) \tag{1}$$

a ReLU non-linearity followed by a max-pool over the n-gram dimension resulting in $g \in \mathbb{R}^M$. A final linear fully connected layer, $W \in \mathbb{R}^{C \times M}$, with a bias, $b \in \mathbb{R}^C$, followed by a softmax, produces the output distribution over C class labels, $o \in \mathbb{R}^C$:

$$o = \mathrm{softmax}(Wg + b) \tag{2}$$

The base model is trained for document classification with a standard cross-entropy loss. We primarily use a filter width of 1, $Q = 1$. In experiments with multiple filter widths, we concatenate the output of the max-pooling prior to the fully connected layer.

3.2 Zero-Shot Sequence Labeling with a CNN Binary Classifier: From Document-Level Labels to Token-Level Labels

The matrix multiplication of the output of the max-pooling operation with the fully connected layer can be viewed as a weighted sum of the most relevant filter-n-gram interactions for each prediction class. This can be deterministically decomposed to produce predictions at the resolution of the CNN's input for each class. Specifically, we use the notation

$$n_m = \arg \max \mathrm{ReLU}(h_m) \tag{3}$$

---

2 Our replication code is publicly available at `https://github.com/allenschmaltz/exa`.
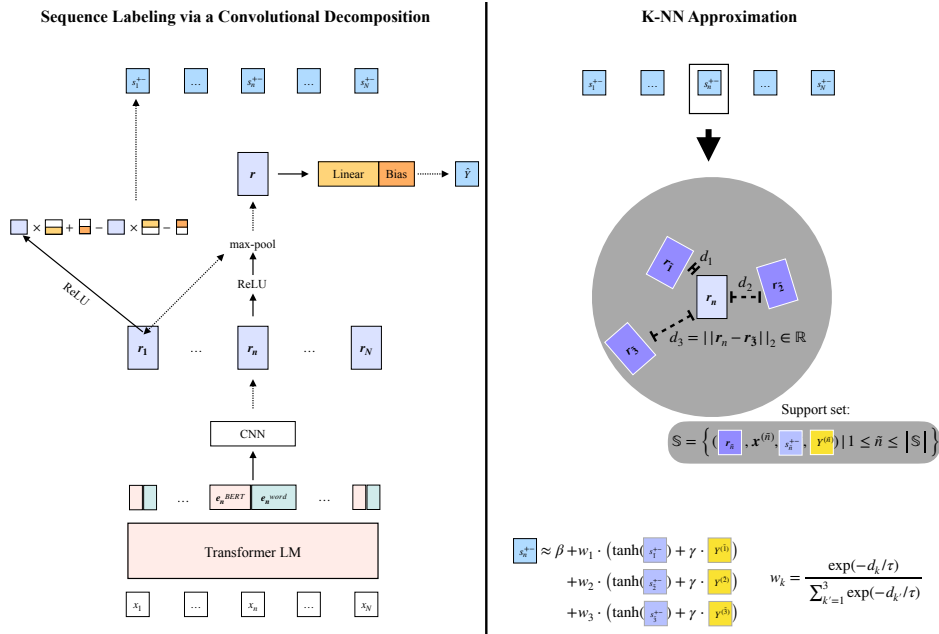
Figure 1
High-level overview of the proposed methods. We derive token-level predictions from a model trained with document-level labels via the decomposition of a max-pooled, kernel-width-one CNN and a linear layer over a large Transformer language model (*left*). These token-level predictions can themselves be approximated as an interpretable weighting over a support set with known labels (*right*, where $K = 3$ in the illustration) by leveraging the CNN's feature-specific, summarized representations of the deep network to measure distances to the support set.

to identify the index into the feature map h_m that survived the max-pooling operation, which corresponds to the application of filter m starting at index n_m of the input (i.e., the set $\{n_m, \ldots, n_m + (Q - 1)\}$ contains all of the indices of the input covered by this particular application of the filter of width Q). We then have a corresponding negative contribution score $s_n^- \in \mathbb{R}$ for each input token:

$$s_n^- = \left(\sum_{m=1}^{M} W_{1,m} \cdot g_m \cdot \sum_{q=1}^{Q} [n = n_m + (q - 1)] \right) + b_1 \qquad (4)$$

where we have used an Iverson bracket for the indicator function. The corresponding positive contribution score s_n^+ is analogous:

$$s_n^+ = \left(\sum_{m=1}^{M} W_{2,m} \cdot g_m \cdot \sum_{q=1}^{Q} [n = n_m + (q - 1)] \right) + b_2 \qquad (5)$$

This decomposition then affords considerable flexibility in defining loss constraints to bias the filter weights according to the granularity of the available labels, and/or according to other priors we may have regarding our data.

3.3 Supervised Sequence Labeling

We can use the aforementioned decomposition to fine-tune against token-level labels, when available. We subtract the *negative class contribution* scores from the *positive class contribution* scores, passing the result through a sigmoid transformation for each token. We minimize a binary cross-entropy loss, averaged over the non-padding tokens in the mini-batch:

$$\mathcal{L}_n = -y'_n \cdot \log \sigma(s_n^{+-}) - (1 - y'_n) \cdot \log(1 - \sigma(s_n^{+-})) \tag{6}$$

where $s_n^{+-} = s_n^+ - s_n^-$ and $y'_n \in \{0, 1\}$ is the corresponding true token label, transformed via:

$$y'_n = \begin{cases} 1 & \text{if } y_n = 1 \\ 0 & \text{if } y_n = -1 \end{cases} \tag{7}$$

For inference, token-level detection labels are determined in the same manner as in the zero-shot setting.

3.4 Task-Specific Zero-Shot Loss Constraints: Min-Max

The base zero-shot formulation is appealing because it only requires labels at the document level, and does not entail additional losses nor other constraints beyond the standard classifier. This mechanism also enables adding task-specific constraints, where applicable, to bias the token contributions based on priors we may have about our data. For example, Rei and Søgaard (2018) propose a min-max squared loss constraint for grammatical error detection. We can capture this idea in our setting in the following manner by fine-tuning the CNN parameters with the following binary cross-entropy losses:

$$\mathcal{L}_{min} = -\log(1 - \sigma(s_{min}^{+-})) \tag{8}$$

where $s_{min}^{+-} = \min(s_1^{+-}, \ldots, s_n^{+-}, \ldots, s_N^{+-})$ is the smallest combined token contribution in the sentence; and

$$\mathcal{L}_{max} = -Y' \cdot \log \sigma(s_{max}^{+-}) - (1 - Y') \cdot \log(1 - \sigma(s_{max}^{+-})) \tag{9}$$

where $s_{max}^{+-} = \max(s_1^{+-}, \ldots, s_n^{+-}, \ldots, s_N^{+-})$ is the largest combined token contribution in the sentence and Y' is the true document-level label, Y, transformed to be in $\{0, 1\}$. These two losses are then averaged together over the mini-batch.

The intuition is to encourage correct sentences to have aggregated token contributions less than zero (i.e., no detected errors), and to encourage sentences with errors to have at least one token contribution less than zero and at least one greater than zero (i.e., to encourage even incorrect sentences to have one or more correct tokens, since errors are, in general, relatively rare).

3.5 Aggregate, Comparative Feature Extraction

From the token-level contributions, we can then score spans of text, from n-grams to full sentences and documents, serving as a type of feature extractor for each class. We can aggregate token contributions across spans of text, which can have the effect of comparative, extractive summarization, an additional useful view of a data set under a model. Here we assign scores to the negative class n-grams of size z as follows:[3]

$$n\text{-gram}_{n:n+(z-1)}^{-} = \sum_{i=n}^{n+(z-1)} (s_i^{-} - b_1) \tag{10}$$

The score for the full document is then $n\text{-gram}_{1:N}^{-}$. The negative class n-grams are only calculated from documents for which the document-level model predicts the document as being negative. In our analysis below, we consider unigram to 5-gram scores that are summed, $^{\text{total}}n\text{-gram}_{n:n+(z-1)}^{-}$, or averaged, $^{\text{mean}}n\text{-gram}_{n:n+(z-1)}^{-}$, over the number of occurrences. Similarly, each document is scored by calculating $n\text{-gram}_{1:N}^{-}$, and then optionally, normalizing by the document length. The corresponding scores for the positive class, $n\text{-gram}_{n:n+(z-1)}^{+}$, are calculated in an analogous manner.

With the true document-level labels, we can then identify the n-grams and documents most salient for each class under this metric, and just as importantly for many applications, the n-grams and documents that the model misclassifies.

3.6 Exemplar Auditing: Inference-Time Decision Rules and Data/Model Introspection via Dense Representation Matching

We can view each token-level prediction, $f(x_n) = s_n^{+-}$, as the composition $f = u \circ v$, where $v : e_n \in \mathbb{R}^D \mapsto r_n \in \mathbb{R}^M$ and $u : r_n \in \mathbb{R}^M \mapsto s_n^{+-} \in \mathbb{R}$. The mapping v takes as input the word embeddings and hidden layers of the deep network corresponding to the particular token and produces a dense representation, a distilled summarization of the expressive deep network at the local level which we refer to as an **exemplar** representation, derived from the CNN filter applications corresponding to the token.[4]

More specifically, with $Q = 1$, for each token we have a vector

$$r_n = h_{1,n}, \ldots, h_{m,n}, \ldots, h_{M,n} \tag{11}$$

consisting of the components from each of the M feature maps corresponding to the token at index n. With our model, the mapping u is then the max-pool, ReLU, and the corresponding weights of the final fully connected layer that produce s_n^{+-}.

Over a set of instances with known labels containing $|\mathbb{S}|$ tokens, we can then form what we term a *support set*:

$$\mathbb{S} = \left\{ (r_{\tilde{n}}, x^{(\tilde{n})}, s_{\tilde{n}}^{+-}, Y^{(\tilde{n})}) | 1 \leq \tilde{n} \leq |\mathbb{S}| \right\} \tag{12}$$

---

3 We drop the constant bias term because we are ranking negative and positive class n-grams separately.
4 We use the term "exemplar" rather than "prototype," as we use these representations directly, unique to each feature, rather than as class-based centroids.

a database of meta-data associated with the model's predictions over the document instances for each token index $\tilde{n}$: The token-level representation $\boldsymbol{r}_{\tilde{n}}$, the associated document $\boldsymbol{x}^{(\tilde{n})}$, the prediction $s_{\tilde{n}}^{+-}$, and the ground-truth document-level label $Y^{(\tilde{n})}$. When token-level labels are available, we additionally add $y_{\tilde{n}}$. We treat each $\tilde{n}$ as uniquely describing a single token in the database. The set of documents in the support set and that of the model's training set can be identical, partially overlapping, or even disjoint.

To aid in analyzing the decision-making process of the model, as well as to explore the characteristics of the data, we can then relate a new test instance to this support set by matching against representations, searching[5] for the index $\tilde{n}$ that minimizes the Euclidean distance between $\boldsymbol{r}_{\tilde{n}}$ and that of the test token's vector $\boldsymbol{r}_n$:

$$\arg \min_{\tilde{n}} \| \boldsymbol{r}_n - \boldsymbol{r}_{\tilde{n}} \|_2 \tag{13}$$

This connection enables inference-time decision rules with which we can inspect and constrain predictions, which we refer to as exemplar auditing. We will use the label EXAG for the rule in which positive token-level predictions are only admitted when the token-level prediction of the corresponding exemplar token from the support set matches that of the test token, and the exemplar's document has a positive ground-truth label: $s_n^{+-} > 0 \wedge s_{\tilde{n}}^{+-} > 0 \wedge Y^{(\tilde{n})} = 1$. Similarly, we use the label EXAT when token-level ground-truth labels are available in the support set: $s_n^{+-} > 0 \wedge s_{\tilde{n}}^{+-} > 0 \wedge y_{\tilde{n}} = 1$. In this way, updates to the support set can be a means of making local updates to the model without modifying the parameters of the original model, including in some cases for domain-shifted data over which the original model is otherwise a weak predictor, provided the dense representations yield adequate matching effectiveness across the new domain. The distances to the matches can also be used for constraining predictions, which we consider in the context of the K-NN approximations described next.

3.7 K-NN Model Over Exemplar Representations

The inference-time decision rules are appealing, as once a dense search infrastructure is in place, they are easy to implement and for end-users and auditors to understand: *If a prediction does not resemble that of its nearest matched exemplar, as via a large distance and/or label and prediction discrepancies, reject the prediction and send the decision to a human for adjudication.* Additionally, because the original model's output is used for non-rejected predictions, the prediction effectiveness is guaranteed to be the same as that of the original model for the non-rejected predictions. However, in some settings where explainability is paramount, we may require the stronger sense of fully describing a prediction as a weighting over exemplars from the support set. Interestingly, we show that we can construct a K-NN from a simple transformation of the predictions and class labels of the nearest K exemplars that closely matches the sign directions of the original prediction and is at least as strong a predictor on the metrics over the ground-truth.

We consider one primary formulation and two additional variations for further analysis. We aim to keep the number of parameters to a minimum to avoid over-fitting; since our goal is to simply reproduce the sign of the original prediction, rather than to

---

5 We restrict our experiments to exact search, which is nonetheless reasonably fast using GPUs at this scale, to avoid introducing another source of variation, but approximate search could be used in practice for larger support sets.

construct a significantly larger or more expressive model; and since we seek a weighting that is easily inspectable by an end-user.

We seek a simple function that approximates the original model's prediction for a token x_n as a weighting over the support set:

$$\hat{y}_n = \text{sgn}\left(f(x_n)\right) = \text{sgn}\left(s_n^{+-}\right) \approx$$

$$\hat{y}_n^{KNN} = \text{sgn}\left(f(x_n)^{KNN}\right) = \text{sgn}\left(\beta + \sum_{k \in \arg K \min_{\tilde{n}} ||r_n - r_{\tilde{n}}||_2} w_k \cdot \left(\tanh(s_k^{+-}) + \gamma \cdot Y^{(k)}\right)\right) \tag{14}$$

where $\gamma \in \mathbb{R}$ and $\beta \in \mathbb{R}$ are parameters learned via gradient descent; with K treated as a hyper-parameter; and sgn is the binary threshold function

$$\text{sgn}(x) = \begin{cases} 1 & \text{if } x > 0 \\ -1 & \text{if } x \le 0 \end{cases} \tag{15}$$

The three considered variations differ in their particular formulation of w_k, detailed below, but in all cases $\sum w_k = 1, w_k \in [0, 1]$. We take s_k^{+-} to mean the token-level prediction of the kth nearest exemplar in the support set, and $Y^{(k)} \in \{-1, 1\}$ as the document-level label associated with the *document* to which the kth exemplar belongs in the support set. When token-level labels are available, as with the fully supervised setting, we replace $Y^{(k)}$ with $y_k \in \{-1, 1\}$, the ground-truth token-level label associated with the kth exemplar. The $\gamma \cdot Y^{(k)}$ term is in effect a class-specific bias offset given the matched document, and the $\gamma \cdot y_k$ variation directly balances the signal from the true token-level label and the prediction. The predictions and exemplar matchings are at the *token level*, but importantly r is a representation of the token that encodes contextual dependencies over the full input, as a result of the deep network.

3.7.1 Distance-Weighted K-NN (KNN$_{DIST.}$). Our main form for w_k accounts for the relative distribution of distances in the top-K:

$$w_k = \frac{\exp\left(-||r_n - r_k||_2/\tau\right)}{\sum_{k' \in \arg K \min_{\tilde{n}} ||r_n - r_{\tilde{n}}||_2} \exp\left(-||r_n - r_{k'}||_2/\tau\right)} \tag{16}$$

where $\tau \in \mathbb{R}$ is the single additional learnable parameter. *We separately use the raw, unnormalized distance to the nearest match as an exogenous factor to consider when assessing the reliability of the predictions.*

We train the K-NN's parameters with a binary cross-entropy loss, after having trained the original model, the parameters of which remained fixed, by minimizing the difference between the original model's output and the K-NN's output:

$$\mathcal{L}_n^{KNN} = -\sigma(s_n^{+-}) \cdot \log \sigma\left(f(x_n)^{KNN}\right) - (1 - \sigma(s_n^{+-})) \cdot \log\left(1 - \sigma\left(f(x_n)^{KNN}\right)\right) \tag{17}$$

$\mathcal{L}_n^{KNN}$ is averaged over mini-batches constructed from the tokens of shuffled documents. Across data sets, we treat the original training set, or a subset thereof, as the

support set during training, and we randomly split the held-out dev set into two sets: We use half of the data for learning via $\mathcal{L}^{KNN}$, and the other half serves as the held-out KNN DEV set. We choose the epoch that minimizes

$$\delta^{KNN} = \sum_{n \in \text{dev}} [\text{sgn}\left(s_n^{+-}\right) \neq \text{sgn}\left(f(x_n)^{KNN}\right)] \tag{18}$$

the total number of prediction discrepancies between the original model and the K-NN approximation over the KNN DEV set. During training, if the immediately preceding epoch does not yield the minimal δ^{KNN} among all epochs, we subsequently only calculate $\mathcal{L}^{KNN}$ for the tokens with prediction discrepancies until a new minimum δ^{KNN} is found, or the maximum number of epochs is reached.

3.7.2 Constraint-Weighted K-NN (KNN$_{CONST.}$). We additionally consider a variation to assess the significance of the relative distances by dropping the dependence of w_k on the distances, at the expense of adding K additional learned parameters:

$$w_k = \frac{\exp\left(\bar{w}_k/\tau\right)}{\sum\limits_{k'=1}^{K} \exp\left(\bar{w}_{k'}/\tau\right)} \tag{19}$$

with $\tau \in \mathbb{R}$ and $\bar{w} \in \mathbb{R}^K$.

To avoid overfitting and to encourage the normalized weights to be of decreasing magnitude, $w_k \geq w_{k+1}$, a prior that the closer exemplars should be more prominent in the prediction as with the distance-weighted version above, we add additional loss constraints when training this version:

$$\mathcal{L}_{mm}^{\text{KNNconst.}} = \frac{1}{K+1}\left(-\log(1 - \sigma(\bar{w}_{min})) - \log\sigma(\bar{w}_{max}) - \sum_{k=1}^{K-1} \log\sigma(\bar{w}_k - \bar{w}_{k+1})\right) \tag{20}$$

where $\bar{w}_{min} = \min(\bar{w}_1,\ldots,\bar{w}_k,\ldots,\bar{w}_K)$ is the smallest element of $\bar{w}$, the unnormalized weights; $\bar{w}_{max} = \max(\bar{w}_1,\ldots,\bar{w}_k,\ldots,\bar{w}_K)$ is the largest element of $\bar{w}$; and the final term encourages decreasing weights. The unnormalized weights, $\bar{w}$, are initialized to be decreasing. The final combined loss in a mini-batch for this model is then

$$\mathcal{L}^{\text{KNNconst.}} = \frac{1}{2}\left(\mathcal{L}_{mm}^{\text{KNNconst.}} + \frac{1}{|\text{batch}|}\sum_{n \in \text{batch}}^{|\text{batch}|} \mathcal{L}_n^{KNN}\right) \tag{21}$$

3.7.3 Equally Weighted K-NN (KNN$_{EQUAL}$). Finally, we consider $w_k = \frac{1}{K}$. An advantage of this approach is that it requires learning and interpreting only two parameters, γ and β; it is just a simple transformation of the nearest exemplar predictions and associated labels. A disadvantage is that even relatively far exemplars will play an equal role in the final K-NN prediction. In this way, an interpretation of the model is obligated to equally consider even the farthest exemplars, which requires an end-user to examine the full set of size K, some members of which may have near-zero weights in the above alternatives that explicitly enforce a ranking. For comparison purposes, we train this version via gradient descent with $\mathcal{L}^{KNN}$, as with KNN$_{DIST.}$ above.

4. Grammatical Error Detection

The task of grammatical error detection is to detect the presence or absence of grammatical errors in a sentence[6] at the token level.

4.1 Grammatical Error Detection: Experiments

We evaluate detection in both the zero-shot and fully supervised sequence labeling settings, comparing the behavior of the proposed sequence labeling layer to previous models, as well as investigating the behavior of the inference-time decision rules and the K-NN approximations.

4.1.1 Data: FCE. We follow past work on error detection and use the standard training, dev, and test splits of the publicly released subset of the First Certificate in English (FCE) data set (Yannakoudakis, Briscoe, and Medlock 2011; Rei and Yannakoudakis 2016),[7] consisting of 28.7k, 2.2k, and 2.7k labeled sentences, respectively.

4.1.2 Data: Domain-Shifted News Data. In a real deployment, we might reasonably expect an error detection model to encounter well-formed, correct documents from another domain, over which we would want the model to be robust to false positives. To emulate this scenario, we also consider a series of experiments in which we augment the FCE data set with sentences from the news-oriented One Billion Word Benchmark data set (Chelba et al. 2014), which are assigned negative class ($Y = -1$) sentence-level labels. We augment the FCE training set with a sample of 50,000 sentences (FCE+NEWS50K) and add a disjoint sample of 2,000 sentences to the FCE test set for evaluation (FCE+NEWS2K).

4.1.3 Models

uniCNN+BERT *Model.* Our primary model uses a filter width of 1 with 1,000 filter maps, $Q = 1, M = 1,000$. The CNN layer takes as input, for each token, the top four hidden layers of the large, pre-trained Bidirectional Encoder Representations from Transformers (BERT$_{\text{LARGE}}$) model of Devlin et al. (2019), a multilayer bidirectional Transformer (Vaswani et al. 2017), concatenated with the pre-trained Word2Vec word embeddings of Mikolov et al. (2013), $D = 4{,}396$. The BERT model is pre-trained with masked-language modeling and next-sentence prediction objectives with large amounts of unlabeled data from 3.3 billion words. BERT's contextualized embeddings are capable of modeling dependencies between words and position information. The CNN can be viewed as summarizing the signal from this deep network for the fine-tuned task. We use the pre-trained, 340-million-parameter BERT$_{\text{LARGE}}$ model with case-preserving WordPiece (Wu et al. 2016) tokenization.[8] In our experiments, we fine-tune the 300-dimensional word-embeddings with the CNN parameters, while the parameters of the BERT$_{\text{LARGE}}$ model remain fixed. The BERT model takes as input WordPiece tokens, using its full vocabulary, and we limit the vocabulary size to 7,500 only for the fine-tuned word embeddings. Prior to evaluation, to maintain alignment with the original tokenization and labels, the

---

6 For the FCE data set, each "document" consists of a single sentence.

7 https://ilexir.co.uk/datasets/index.html.

8 We use the PyTorch (https://pytorch.org/) reimplementation of the original code base available at https://github.com/huggingface/pytorch-pretrained-BERT (Wolf et al. 2020).

WordPiece tokenization is reversed (i.e., de-tokenized), with positive/negative token contribution scores averaged over fragments for original tokens split into separate WordPieces. We also consider fine-tuning the trained UNICNN+BERT model with the min-max loss, which we label UNICNN+BERT+MM.

Our model only adds approximately 2% more parameters than BERT$_{\text{LARGE}}$ alone. With $Q = 1$, the CNN consists of the kernel-width and bias, $M \cdot D + M$, and the linear layer consists of $2 \cdot M + 2$ parameters, which includes the 2 bias terms. The word-embeddings contribute $300 \cdot (7{,}500 + 2)$ parameters, which includes 2 additional holder symbols we use in practice for padding and out-of-vocabulary input tokens. For UNICNN+BERT, this results in around 6.6 million parameters added to the 340 million parameters of BERT$_{\text{LARGE}}$.

Reference Models. We also include a reference base model, CNN, with filter widths of 3, 4, and 5, with 100 filter maps each, fine-tuning 300 dimensional GloVe embeddings (Pennington, Socher, and Manning 2014), with a vocabulary of size 7,500, comparable to early work on zero-shot detection with lower parameter models. We additionally consider a model, CNN+BERT, similar to the primary UNICNN+BERT model, which uses Word2Vec word embeddings for consistency with the past supervised detection work of Rei and Yannakoudakis (2016), but with Q and M identical to CNN.

Optimization and Tuning. For our zero-shot detection models, CNN, CNN+BERT, and UNICNN+BERT, we optimize for sentence-level classification, choosing the training epoch with the highest sentence-level F_1 score on the dev set, without regard to token-level labels. These models do not have access to token-level labels for training or tuning.

We set aside 1k token-labeled sentences from the dev set to tune the token-level $F_{0.5}$ score for comparison purposes for the experiments labeled CNN+BERT+1K and UNICNN+BERT+1K.

uniCNN+BERT+S* *Model.* We also fine-tune a model with token-level labels, UNICNN+BERT+S*, with weights initialized with those of the UNICNN+BERT model trained for binary sentence-level classification. For calculating the loss at training, we assign each WordPiece to have the detection label of its original corresponding token, with the loss of a mini-batch averaged across all of the WordPieces. Inference is performed as in the zero-shot setting.

All models use dropout, with a probability of 0.5, applied on the output of the max-pooling operation, and we train with Adadelta (Zeiler 2012) with a batch size of 50.

4.1.4 Exemplar Auditing Decision Rules

In-Domain Data. For each of the UNICNN models, we also evaluate using the inference-time decision rules of Section 3.6, which we indicate with +EXAG and +EXAT appended to the model labels. The Euclidean distances are calculated at the word level of the original sentences, where we average the exemplar vectors when a word is split across multiple WordPiece tokens.

Expanded Database with Domain-shifted Data. We also consider adding the FCE+NEWS50K data to the support set, and evaluating on the augmented FCE+NEWS2K test set. For reference, we also *train* the primary zero-shot models using the FCE+NEWS50K data, for which we use the labels UNICNN+BERT+NEWS50K and UNICNN+BERT+MM+ NEWS50K.

4.1.5 K-NN Approximations. We train each of the 3 proposed K-NN approximations on the held-out KNN DEV set to minimize δ^{KNN}, for up to 40 epochs, only using the predictions from the original models, rather than ground-truth labels. Only for the fully supervised model, UNICNN+BERT+S* , do we then subsequently use token-level labels to tune the decision boundary, as with that original model. We add the labels of Section 3.7 as suffixes to the original models to indicate the type of K-NN used, $+K_8NN_{DIST.}$, $+K_8NN_{CONST.}$, $+K_8NN_{EQUAL}$, with the subscript indicating $K = 8$. We chose $K = 8$ on the held-out dev set based on minimizing δ^{KNN} with the UNICNN+BERT+MM model with $K \in \{1, 3, 5, 8, 25\}$. The approximations are only marginally better with $K = 25$ for some of the models, so we hold $K = 8$ constant for comparison purposes, and since smaller values of K are preferable for interpretability, ceteris paribus. For reference, we also include results with K_1NN_{equal}, which only considers the nearest match.

Constraints for Domain-shifted Data. We also demonstrate constraining the output based on the maximum allowed distance to the nearest match in the support set, among matches for which the K-NN prediction equals that of the sentence-level label of the nearest match, and/or limited to minimum output magnitudes of the K-NN. We determine these constraints on the KNN DEV set, based on δ^{KNN}, determined without access to token-level labels; for simplicity, we use the mean values among correct *approximations*. We examine this with weak models over the FCE+NEWS2K domain-shifted test set that only have the FCE training set in the support set, investigating whether we can nonetheless identify subsets with strong effectiveness. This is a challenging but very practical setting, as in real deployments, the input data will often diverge from what we have seen in training. Such constraints serve as heuristics, tied to the model itself, for determining when to refrain from predicting, as is critical in higher-risk settings.[9]

4.1.6 Previous Approaches and Baselines

Previous Zero-shot Sequence Models. Recent work has approached zero-shot error detection by modifying and analyzing bidirectional LSTM taggers, which have been shown to work comparatively well on the task in the supervised setting. Rei and Søgaard (2018) adds a soft-attention mechanism to a bidirectional LSTM tagger, training with additional loss functions to encourage the attention weights to yield more accurate token-level labels (LSTM-ATTN-SW). Previous work also considered a gradient-based approach to analyze this same model (LSTM-ATTN-BP) and the model without the attention mechanism (LSTM-LAST-BP), by fitting a parametric Gaussian model to the distribution of magnitudes of the gradients of the word representations.

Previous Supervised Sequence Models. For comparison, we include recent fully supervised sequence models. Rei and Yannakoudakis (2016) compares various word-based neural sequence models, finding that a word-based bidirectional LSTM model was the most effective (LSTM-BASE+S*). Rei and Søgaard (2018) compares against a bidirectional LSTM tagger with character representations concatenated with word embeddings (LSTM+S*). The model of Rei (2017) extends this with an auxiliary language

---

9 Within the set of admitted predictions, we might then consider approaches for quantifying uncertainty, which we leave for future work. Here we focus on examining and establishing the K-NN behavior relative to the original model to justify its use as an interpretable substitute, as well as the types of interpretable heuristics useful for avoiding domain-shifted and out-of-domain data this enables.

modeling objective (LSTM+LM+S*). This model is further enhanced with a character-level language modeling objective and supervised attention mechanisms in Rei and Søgaard (2019) (LSTM+JOINT+S*). Bell, Yannakoudakis, and Rei (2019) consider BERT embeddings with the LSTM+LM+S* model, establishing a new state-of-the-art for the supervised setting, using a frozen $BERT_{BASE}$ model (LSTM+LM+$BERT_{BASE}$+S*), and also providing results with a $BERT_{LARGE}$ model (LSTM+LM+BERT+S*).

Additional Baselines. For reference, we also provide a RANDOM baseline, which classifies based on a fair coin flip, and a MAJORITYCLASS baseline, which in this case always chooses the positive ("error detected") class.

4.2 Grammatical Error Detection: Results

4.2.1 Zero-shot Results. Table 1 contains the main results with the models only given access to sentence-level labels, as well as LSTM+S* for reference, using F_1, as in previous zero-shot work. The task is very challenging, in general, with some baselines falling below random at the token level. The CNN model has a similar F_1 score as LSTM-ATTN-SW, and is stronger than the back-propagation-based approaches of LSTM-ATTN-BP and LSTM-LAST-BP. This is important, as it suggests the decomposition used with the basic CNN model, which amounts to a very lightweight attention mechanism, has the inductive bias suitable for such local detections, while being trivial to break apart into representative dense vectors of the input, enabling our analysis and interpretability methods. This is further confirmed when adding the pre-trained contextualized embeddings from BERT; remarkably, as a point of reference, these models exceed basic *supervised* LSTM models that use pre-trained word embeddings. In Table 2 against $F_{0.5}$, which is the typical metric for evaluating supervised grammatical error detection, used under the assumption that end users prefer higher precision systems, the UNICNN+BERT model exceeds the fully supervised LSTM-BASE+S* model, which was the state-of-the-art model on the task as recently as 2016.

Table 1
FCE test set results. The LSTM model results are as reported in Rei and Søgaard (2018). With the exception of LSTM+S*, all models only have access to sentence-level labels while training. The sentence-level F_1 scores for the CNN models are from the fully connected layer and are provided for reference.

Model	Sent F_1	Token-level P	R	F_1
LSTM+S*	−	49.15	26.96	34.76
RANDOM	58.30	15.30	50.07	23.44
MAJORITYCLASS	80.88	15.20	100.00	26.39
LSTM-LAST-BP	85.10	29.49	16.07	20.80
LSTM-ATTN-BP	85.14	27.62	17.81	21.65
LSTM-ATTN-SW	85.14	28.04	29.91	28.27
CNN	84.24	20.43	50.75	29.13
CNN+BERT	86.35	26.76	61.82	37.36
UNICNN+BERT	86.28	47.67	36.70	41.47

Table 2
Comparisons with recent state-of-the-art supervised detection models on the FCE test set. Models marked with +S* have access to approximately 28.7k token-level labeled sentences for training and 2.2k for tuning. Models marked with +1K have access to 28.7k sentence-level labeled sentences for training and 1k token-level labeled sentences for tuning. The UNICNN+BERT and UNICNN+BERT+MM models only have access to sentence-level labeled sentences. The results of the LSTM models are as previously reported in the literature.

Model	Token-level		
	P	R	$F_{0.5}$
LSTM+JOINT+S*	65.53	28.61	52.07
LSTM+LM+S*	58.88	28.92	48.48
LSTM-BASE+S*	46.1	28.5	41.1
LSTM+LM+BERT$_{BASE}$+S*	64.96	38.89	57.28
LSTM+LM+BERT+S*	64.51	38.79	56.96
UNICNN+BERT+S*	75.00	31.40	58.70
CNN+BERT+1K	47.11	28.83	41.81
UNICNN+BERT+1K	63.89	23.27	47.36
UNICNN+BERT	47.67	36.70	44.98
UNICNN+BERT+MM	54.87	29.10	46.62

Fine-tuning the zero-shot model UNICNN+BERT with the min-max loss constraint (UNICNN+BERT+MM) has the effect of increasing precision and decreasing recall, as seen in Table 2. This results in a modest increase in $F_{0.5}$, but also a decrease in F_1 to 38.04. Whether or not this is a desirable tradeoff depends on the particular use case, but illustrates biasing the detections via task-specific constraints in the absence of token-level labels.

The inductive bias of the architecture is important for token-level detections: Models with similar sentence-level classification results can have significantly different token-level results. For example, CNN+BERT and UNICNN+BERT have similar sentence-level F_1 scores of around 86, despite differing token-level effectiveness, and the LSTM baselines all exhibit similar sentence-level F_1 scores yet have significantly different token-level scores. As such, attention-style approaches are useful, but not sufficient, for analyzing model predictions over the non-identifiable parameters of deep models, further justifying the need for the proposed methods establishing auditable mappings to the support set.

4.2.2 Supervised and Dev-set-tuned Results. Table 2 also compares dev-set-tuned and fully supervised models. For illustrative purposes, CNN+BERT+1K and UNICNN+BERT +1K are given access to 1,000 token-labeled sentences to tune a single parameter, an offset on the decision boundary, for each model. This yields modest gains for both models, but interestingly, the UNICNN+BERT, in particular, already has a strong $F_{0.5}$ score without modification of the decision boundary in the true zero-shot setting.

The UNICNN+BERT+S* model is a strong supervised sequence labeler. As seen in Table 2, it is nominally stronger than the current state-of-the-art models recently presented in Bell, Yannakoudakis, and Rei (2019). This is critical, as it suggests we can forgo more complicated, expressive final layers, and instead use our proposed CNN and linear decomposition to, in effect, summarize the signal from the deep network, from which it is then straightforward to yield representations for matching, as analyzed next.

4.2.3 Inference-time Decision Rules and K-NN Approximations.

In-domain Data. Table 3 shows the proposed exemplar auditing decision rules and the K-NN approximations on in-domain data across models. Compared with the results in Table 2, the EXAG rule increases precision. In practice, matches tend to correspond to similar contexts, at least when the distance to the nearest exemplar in the support set is low, as shown in the examples in Appendix B. Further, the $F_{0.5}$ scores suggest that with $K = 8$, the distance-weighted K-NNs (KNN$_{\text{DIST.}}$) are *sufficient for replacing the original models' predictions*: The zero-shot K-NNs are nominally stronger than the corresponding original models, and the supervised version has the same effectiveness as the original for all practical purposes (± 1 point). Note, too, that the precision vs. recall patterns for UNICNN+BERT+MM+K$_8$NN$_{\text{DIST.}}$ vs. UNICNN+BERT+K$_8$NN$_{\text{DIST.}}$ parallel those of UNICNN+BERT+MM vs. UNICNN+BERT, reflecting that the approximations are reasonably similar to the original models' predictions, especially over the subset of data for which the original models' predictions are correct, as discussed below.

We further examine the K-NN behavior on the held-out dev set in Table 4. We find that with $K = 8$, across models, each of the proposed K-NN formulations can be trained to be roughly similar in approximation effectiveness, and when we reveal the true labels, there is not a clear winner. In this way, the modeling choice shifts to other aspects of the model: The relative distances within the top-K appear not to be critical on this data set and can be replaced with constant learned weights with KNN$_{\text{CONST.}}$; however, that comes at the expense of additional parameters and is harder to train due to the sensitivity of parameter initialization. The simplicity of KNN$_{\text{EQUAL}}$ is appealing, but KNN$_{\text{DIST.}}$ provides an explicit ranking over the exemplars with the addition of just a single learned parameter, so we take it as our primary model.

As shown in Figure 2, across both classes and all models, the approximation effectiveness *and* the K-NN's prediction effectiveness increase as the magnitude of the K-NN's output increases. This reflects a more general pattern: When the original model and/or K-NN produce incorrect predictions, the original model and the K-NN are more likely to produce different predictions. Put another way, *difficult instances to predict also tend to be difficult instances over which to approximate the model*, which we can exploit as a heuristic to abstain from predicting, discussed below.

Table 3
FCE test set results with the inference-time decision rules and replacing the original model with a K-NN approximation.

| Model	Token-level		
	P	R	$F_{0.5}$
UNICNN+BERT+S*+EXAG	85.17	21.86	53.93
UNICNN+BERT+S*+K$_1$NN$_{\text{EQUAL}}$	72.64	25.52	53.05
UNICNN+BERT+S*+K$_8$NN$_{\text{DIST.}}$	71.91	32.24	57.71
UNICNN+BERT+EXAG	56.79	26.74	46.37
UNICNN+BERT+K$_1$NN$_{\text{EQUAL}}$	47.23	32.01	43.13
UNICNN+BERT+K$_8$NN$_{\text{DIST.}}$	51.19	35.53	47.04
UNICNN+BERT+MM+EXAG	63.88	20.03	44.43
UNICNN+BERT+MM+K$_1$NN$_{\text{EQUAL}}$	60.76	21.17	44.23
UNICNN+BERT+MM+K$_8$NN$_{\text{DIST.}}$	62.06	25.38	48.14

Table 4
Additional results on the K-NN held-out dev set. $F_{0.5}$ and accuracy of the approximation ($\hat{y}^{KNN} = \hat{y}$), and $F_{0.5}$ of the K-NN against ground-truth ($\hat{y}^{KNN} = y$). The effectiveness of the original models ($\hat{y} = y$) on this subset of 14,867 tokens from 972 sentences is included for reference.

Model	True Labels $\hat{y}^{KNN} = y$	Model Approx. $\hat{y}^{KNN} = \hat{y}$	
	$F_{0.5}$	Accuracy	$F_{0.5}$
UNICNN+BERT+S*+K_1NN_{EQUAL}	56.5	96.5	72.5
UNICNN+BERT+S*+K_8NN_{EQUAL}	58.1	96.9	75.9
UNICNN+BERT+S*+$K_8NN_{CONST.}$	60.0	97.0	75.8
UNICNN+BERT+S*+$K_8NN_{DIST.}$	59.4	97.0	75.9
UNICNN+BERT+K_1NN_{EQUAL}	45.1	92.8	69.1
UNICNN+BERT+K_8NN_{EQUAL}	50.5	94.2	78.0
UNICNN+BERT+$K_8NN_{CONST.}$	47.5	94.2	75.5
UNICNN+BERT+$K_8NN_{DIST.}$	48.1	94.3	76.4
UNICNN+BERT+MM+K_1NN_{EQUAL}	47.7	95.8	72.4
UNICNN+BERT+MM+K_8NN_{EQUAL}	52.3	96.4	76.9
UNICNN+BERT+MM+$K_8NN_{CONST.}$	53.1	96.4	75.9
UNICNN+BERT+MM+$K_8NN_{DIST.}$	52.9	96.5	76.9

Model	True Labels $\hat{y} = y$		
	$F_{0.5}$		
UNICNN+BERT+S*	59.5	–	–
UNICNN+BERT	44.9	–	–
UNICNN+BERT+MM	49.6	–	–

Domain-shifted Data. Table 5 considers the more challenging setting in which the FCE test set has been augmented with 2,000 already correct sentences in the news domain. Just applying the UNICNN+BERT+MM model to this modified test set yields a large number of false positives on the already correct data, yielding a $F_{0.5}$ of 25.76 (c.f., the $F_{0.5}$ score of 46.62 on the original test set, as shown in Table 2), and similarly for the other models, including that with full supervision. Simply training with the domain-shifted data, as with UNICNN+BERT+NEWS50K, still results in low effectiveness for the zero-shot models, presumably owing to the class imbalance. Furthermore, the $F_{0.5}$ score of the UNICNN+BERT+news50k model on the original FCE test set (a result not shown in the tables) is 39.57, which is lower than the result of 44.98 of UNICNN+BERT, the equivalent model trained only with the original FCE set (Table 2).

However, when we update the support set with the domain-shifted data, in conjunction with the decision rules or the K-NN approximations, the $F_{0.5}$ scores jump significantly across models. The models are generally weak predictors over the domain-shifted data, but the improved scores reflect the capacity of the representations to match to the new data, and by extension, the associated labels. This mechanism opens the potential to update the model locally without a full re-training.
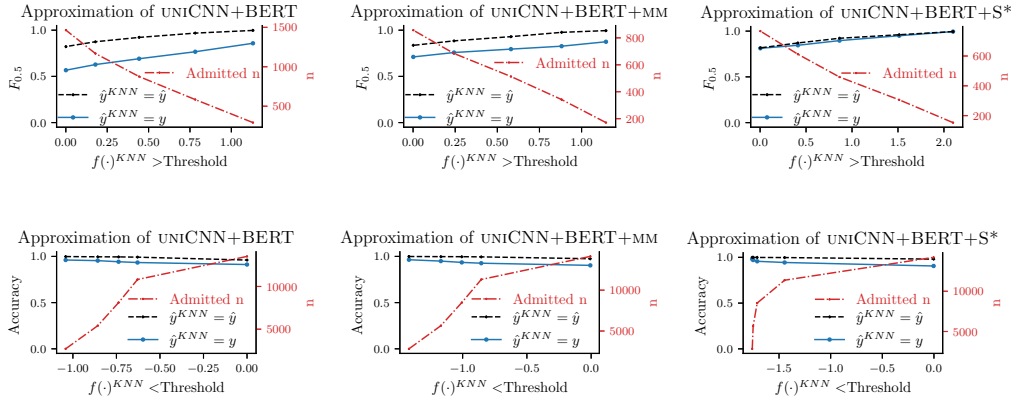
Figure 2
On the in-domain K-NN dev split, across models using $K_8NN_{dist.}$, for both $\hat{y}_n^{KNN} = 1$ (top row) and $\hat{y}_n^{KNN} = -1$ (bottom row), the $F_{0.5}$ and accuracy scores of the approximation (black dotted lines) generally track those of the K-NN against the ground-truth (blue lines) as the magnitude of the K-NN output varies. That is, both the approximation and the prediction effectiveness increase with greater output magnitudes.

Table 5
Domain-shifted FCE+NEWS2K test set. The training set and the support set, $\mathbb{S}$, differ in whether they include the FCE training set (F) or the FCE+NEWS50K set (F+50k), or those sets with token-level labels (F* and F*+50k*, respectively).

Model	Training		Token-level		
		$\mathbb{S}$	P	R	$F_{0.5}$
UNICNN+BERT+S*	F*	–	43.44	31.42	40.35
UNICNN+BERT+S*+ExAT	F*	F*	59.23	21.02	43.43
UNICNN+BERT+S*+ExAT	F*	F*+50k*	83.31	18.92	49.57
UNICNN+BERT+S*+$K_8NN_{DIST.}$	F*	F*	43.98	32.23	40.99
UNICNN+BERT+S*+$K_8NN_{DIST.}$	F*	F*+50k*	65.39	29.58	52.64
UNICNN+BERT+NEWS50K	F+50k	–	26.64	40.13	28.56
UNICNN+BERT+NEWS50K+ExAG	F+50k	F+50k	47.10	26.55	40.79
UNICNN+BERT+MM+NEWS50K	F+50k	–	61.80	11.67	33.25
UNICNN+BERT+MM+NEWS50K+ExAG	F+50k	F+50k	68.89	06.39	23.31
UNICNN+BERT	F	–	21.84	36.65	23.76
UNICNN+BERT+ExAG	F	F	29.19	26.74	28.66
UNICNN+BERT+ExAG	F	F+50k	56.39	23.52	44.07
UNICNN+BERT+ExAT	F	F*+50k*	75.98	18.51	46.87
UNICNN+BERT+$K_8NN_{DIST.}$	F	F	24.65	35.54	26.26
UNICNN+BERT+$K_8NN_{DIST.}$	F	F+50k	43.64	30.91	40.32
UNICNN+BERT+MM	F	–	25.04	29.10	25.76
UNICNN+BERT+MM+ExAG	F	F	31.29	20.03	28.13
UNICNN+BERT+MM+ExAG	F	F+50k	65.08	17.62	42.30
UNICNN+BERT+MM+ExAT	F	F*+50k*	78.16	14.53	41.66
UNICNN+BERT+MM+$K_8NN_{DIST.}$	F	F	27.41	25.38	26.98
UNICNN+BERT+MM+$K_8NN_{DIST.}$	F	F+50k	64.48	21.71	46.26

Table 6
FCE+NEWS2K test set. The output is constrained by a maximum allowed distance to the nearest match in the support set, among matches for which the K-NN prediction equals that of the sentence-level label of the nearest match, and/or limited to minimum output magnitudes of the K-NN. Constraints and thresholds are the mean values among correct *approximations* on the K-NN dev set, determined without access to token-level labels. These limits identify subsets with significantly increased $F_{0.5}$ (cf., Table 5), at the expense of not producing predictions for tokens over which the model is less reliable. Only 41,477 (out of $N = 92,597$) of the tokens in this set are from the original FCE in-domain sentences. Without thresholds, the decision boundary is 0.

Model	$F_{0.5}$	L^2 distance max constraint (Class -1, Class 1)	Output min threshold (Class -1, Class 1)	Admitted n	n/N
UNICNN+BERT+S*+K$_8$NN$_{DIST.}$	42.5			92,597	1.0
UNICNN+BERT+S*+K$_8$NN$_{DIST.}$	62.5		$(-1.6, 1.3)$	53,396	0.58
UNICNN+BERT+S*+K$_8$NN$_{DIST.}$	67.5	$(25.3, 38.9)$		7,896	0.09
UNICNN+BERT+S*+K$_8$NN$_{DIST.}$	86.9	$(25.3, 38.9)$	$(-1.6, 1.3)$	4,219	0.05
UNICNN+BERT+K$_8$NN$_{DIST.}$	26.3			92,597	1.0
UNICNN+BERT+K$_8$NN$_{DIST.}$	46.5		$(-0.8, 0.7)$	40,691	0.44
UNICNN+BERT+K$_8$NN$_{DIST.}$	42.6	$(31.0, 47.6)$		8,779	0.09
UNICNN+BERT+K$_8$NN$_{DIST.}$	67.4	$(31.0, 47.6)$	$(-0.8, 0.7)$	4,388	0.05
UNICNN+BERT+MM+K$_8$NN$_{DIST.}$	27.0			92,597	1.0
UNICNN+BERT+MM+K$_8$NN$_{DIST.}$	45.9		$(-1.2, 0.8)$	38,110	0.41
UNICNN+BERT+MM+K$_8$NN$_{DIST.}$	53.5	$(34.2, 53.3)$		7,879	0.09
UNICNN+BERT+MM+K$_8$NN$_{DIST.}$	75.8	$(34.2, 53.3)$	$(-1.2, 0.8)$	4,180	0.05

Matching to the support set in this way can improve effectiveness over domain-shifted data, but of course, it also requires such data to be in the support set prior to inference. In practice, it may be advisable to include as much data in the support set as computationally feasible, refraining from predicting for matches to unlabeled data, as applicable. In higher-risk settings, we can also constrain predictions based on the L^2 distance to the nearest match and the magnitude of the K-NN output, as demonstrated in Table 6 on the FCE+NEWS2K test set. These constraints limit predictions to reliable subsets, even for these models that are weak predictors over the full set. These heuristics are interpretable in that the matched distance can be compared to that of other instances, and the K-NN output is a bounded value that is an explicit weighting over instances with known labels, tracking prediction reliability at least as well as the magnitude of the token-level output of the original model (Figure 3).

4.3 Grammatical Error Detection: Discussion

The baseline expectations for zero-shot grammatical error detection models are low given the difficulty of the supervised case. It is therefore relatively surprising that a model such as UNICNN+BERT, when given only sentence-level labels, can yield a reasonably decent sequence model that is in the ballpark of some recent—even if lower parameter—fully supervised models. The inductive bias of the proposed method over a strong deep network is effective for such class-conditional detection, as well as supervised labeling. The approach additionally enables dense representation matching against a support set with known labels, with both inference-time decision rules and particular K-NN approximations. In this way, we gain the ability to make updates to a model without re-training; to constrain predictions based on interpretable heuristics;
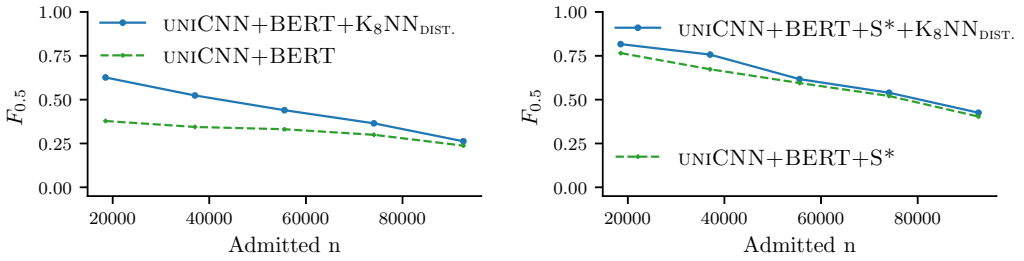
Figure 3
The original model output and the K-NN approximation output as comparative measures of prediction reliability on the domain-shifted FCE+NEWS2K test set. The predictions are sorted by the magnitude of the output and scored in 5 bins. We consider both classes together, holding n constant within bins. The magnitude of the K-NN output tracks prediction reliability at least as well as that of the original model, with the advantage that the K-NN has an explicit, interpretable connection to the support set and available labels. Appendix Table B5 similarly examines UNICNN+BERT+MM+$K_8NN_{DIST.}$ looking at each class separately.

and more generally, to recast the otherwise black-box predictions of the network as an explicit weighting over instances with known labels.

5. Sentiment Data: Binary Prediction of Polarity

We further analyze the behavior of updating the support set over domain-shifted data for the task of predicting sentiment features in IMDb movie reviews. We consider recent work that re-annotates document-level classification data with minimal, local revisions that change the class labels (Kaushik, Hovy, and Lipton 2020; Gardner et al. 2020), from which we back-out token-level labels for evaluation. We use this existing data-oriented approach for robust classification for controlled tests of the internal validity of our approach. We observe an ability to adapt the models via matching as with the grammar experiments. Additionally, in this context, we find that robust prediction over new, unseen domains remains challenging, but simple token-level heuristics tied to the K-NN approximation are nonetheless at least reasonably effective at constraining predictions to reliable subsets, and for screening data unlike that seen in training. This provides further justification for methods, such as proposed here, with which we can analyze and curate the data under the current generation of deep networks.

5.1 Sentiment Data: Experiments

We consider the task of predicting binary document-level sentiment in IMDb movie reviews. We analyze detection of sentiment features at the token level, treating it as a zero-shot sequence labeling task, and additionally provide document-level classification results when constraining the predictions based on the token-level heuristics.

5.1.1 Data: IMDb Sentiment (Negative vs. Positive) with Local Re-edits. We use the IMDb data of Kaushik, Hovy, and Lipton (2020).[10] This consists of movie reviews with negative sentiment ($Y = -1$) and positive sentiment ($Y = 1$), including reviews from the

---

10 Available at `https://github.com/acmi-lab/counterfactually-augmented-data`.

original review site (original, or ORIG.) and "counterfactually augmented" revisions (REV.), the latter of which were created by crowd-workers who annotated the original reviews with local, minimal changes that change the document-level label. For document/review-level sentiment, we follow the main splits of the original work and train on a sample of 3.4k original reviews, ORIG. (3.4k), and the original reviews combined with their corresponding revisions, ORIG.+REV. (1.7k+1.7k). For experiments modifying the support set, we will also consider each of these halves separately, ORIG. (1.7k) and REV. (1.7k). For reference, we additionally train with the full set of original reviews, ORIG. (19k), and the full set combined with the revisions, ORIG.+REV. (19k+1.7k). For evaluation, we consider the ORIG. and REV. test sets from previous work.

To control for the language distribution of the revisions, we also create a new set of disjoint source-target pairs for training by removing the corresponding original reviews and leaving the revisions. We then add in disjoint samples from the remaining full set of original reviews to fill out the remaining sample size. For the smaller set this results in a set of 3.4k reviews, ORIG.$_{\text{DISJOINT}}$+REV. (1.7k+1.7k), the same size as the comparable parallel set. For the larger set, we simply remove any original reviews that match the original reviews paired with revised reviews, creating ORIG.$_{\text{DISJOINT}}$+REV. (19k-1.7k+1.7k).

Sentiment Diffs for Token-Level Detection. We use the parallel original and revision data to create token-level feature labels. Treating positive reviews as the source, we deterministically generate source-target transduction diffs in the same manner as Schmaltz et al. (2017). We then assign the positive class ($y_n = 1$) to tokens associated with diffs that transduce to documents for which $Y = 1$, assigning all other tokens to the negative class ($y_n = -1$). We use a similar convention as the FCE data set in Section 4 with respect to insertions, deletions, and replacements. Table C1 provides an example.

5.1.2 Data: IMDb Sentiment (Negative vs. Positive) with Contrast Sets. We additionally evaluate on the IMDb reviews of Gardner et al. (2020),[11] which are revised with local re-edits by professional researchers familiar with the task instead of by crowd-sourced workers. This test set (CONTRAST) corresponds to the same set of reviews in the test set of Kaushik, Hovy, and Lipton (2020). We do not have a corresponding training set, nor do we use the corresponding dev set for tuning, so we consider all evaluation on this set to be a domain-shifted setting.

5.1.3 Data: Out-of-domain Twitter Document-Level Sentiment (Negative vs. Positive). Finally, we also evaluate on the test set of SemEval-2017 Task 4a (Rosenthal, Farra, and Nakov 2017).[12] This consists of Twitter messages, which are significantly different from the IMDb movie reviews in terms of the topics covered, the language distribution, and the length of the documents, so we consider this to be an out-of-domain setting. We follow the previous work of Kaushik, Hovy, and Lipton (2020) in evaluating the binary classification results with accuracy. We balance the test set, using equal numbers of negative and positive Tweets, and drop the third class (neutral) for consistency with the earlier work, resulting in 4,750 Twitter messages for evaluation.

---

11 Available at `https://github.com/allenai/contrast-sets/tree/master/IMDb`.
12 Available at `http://alt.qcri.org/semeval2017/task4/data/uploads/semeval2017-task4-test.zip`.

5.1.4 Models. Our core model is the UNICNN+BERT model from Section 4, with which we vary the training set and the data in the support set. The only differences from UNICNN+BERT in the grammar detection experiments is that we set the maximum length, by WordPiece, to 350 as in previous works, and we choose the training epoch (up to a max of 60 epochs) by the highest accuracy on the dev set.

We evaluate token-level predictions of sentiment diffs using the $F_{0.5}$ metric, as with grammatical error detection above. We vary whether the support set includes data from the ORIG. and/or REV. training sets, using the labels +EXAG and +EXAT from Section 3.6 to identify the particular rules used. We also present results where we allow the models a small amount of data to tune the decision boundary for the token-level predictions. For consistency, we always use the dev set of the ORIG. reviews subset, using the subscript $_{+ORIG\_DEV}$ to indicate that the models have access to 245 sentences with token-level labels. This provides a point of comparison to the exemplar auditing decision rules.

K-NN. We train the distance-weighted K-NN approximation on the held-out KNN DEV set to minimize δ^{KNN} as in Section 4, but for up to 60 epochs, UNICNN+BERT+ $K_8NN_{DIST.}$. The original model is trained on the ORIG. (3.4k) data. For comparisons with experiments with the inference-time decision rules, the K-NN is trained with ORIG. (1.7k) as the support set, using half of the $_{+ORIG\_DEV}$ for setting the K-NN parameters and the other half as the held-out KNN DEV. This is a relatively limited amount of data, but it is sufficient for training the 3 parameters of the K-NN to at least match the accuracy of the original model.

K-NN Token-Level Constraints for Document-Level Classification. The K-NN enables interpretable heuristics for constraining predictions to the most reliable subsets of the data. In Section 4, we demonstrated this for token-level detection; here, we show how this idea can be applied toward document-level classification, as well. As with detection in Table 6, token-level predictions are constrained by a maximum allowed distance to the nearest match in the support set and K-NN output magnitude limits derived from correct *approximations* on the KNN DEV set, determined without access to token-level labels. For both distances and magnitudes, we use the mean for each class among correct approximations. Using the full $_{+ORIG\_DEV}$ set we then set limits on the proportion and/or range of admitted tokens per document required to admit the overall document-level classification from the original UNICNN+BERT model. To emulate a high-risk setting, we set the minimum threshold such that all admitted document-level predictions are correct on the dev set. We also optionally further require the total number of tokens admitted to be within $\pm$ 1 standard deviation from the mean of correct predictions to control for unexpected lengths.[13]

Previous Approaches. Our primary focus in this section is holding the model architecture from Section 4 constant while changing the data subsets. For reference, we include the results of Kaushik, Hovy, and Lipton (2020), which fine-tunes the $BERT_{BASE}$ uncased model with the standard final linear layer for classification, $BERT_{BASE_{UNCASED}}$+FT. For comparison, we then also train a model using this same Transformer as frozen input with uncased GloVe embeddings, UNICNN+BERT$_{BASE_{UNCASED}}$, and also an analogous cased model with Word2Vec embeddings, UNICNN+BERT$_{BASE}$.

---

13 This differs from a simple hard constraint on token input lengths. In principle, most Twitter messages could still be admitted by this model-dependent constraint, as the lower bound is around 5 tokens.

5.2 Sentiment Data: Results

Document-Level Classification. For context, Table 7 shows the document-level accuracy of UNICNN+BERT when varying the training data, tested on the original (ORIG.) and revised (REV.) test sets. Training with ORIG. vs. ORIG.+REV. reflects the same patterns seen in the experiments of Kaushik, Hovy, and Lipton (2020); however, if we control for the language of the revised reviews by training with disjoint source-target pairs (ORIG.$_{\text{DISJOINT}}$+REV.), the difference across test sets is more modest. For reference, we find that UNICNN+BERT is at least as effective as fine-tuning all parameters of the BERT$_{\text{BASE}}$ model, with the UNICNN+BERT$_{\text{BASE}_{\text{UNCASED}}}$ variant within 2–3 points (Table C2).

Document-Level Classification with Token-Level Constraints. Table 8 shows review-level test accuracy with UNICNN+BERT trained on the ORIG. (3.4k) data using UNICNN+BERT+K$_8$NN$_{\text{DIST.}}$ to determine constraints. Token-level predictions are constrained by a maximum allowed distance to the nearest match in the support set and K-NN output magnitude limits derived from correct *approximations* on the K-NN dev set, determined without access to token-level labels (as in Table 6). The document-level predictions are then constrained by a minimum threshold ($\approx 10\%$) on the proportion of admitted tokens among all tokens in the document and optionally, an additional constraint on the allowed range of admitted tokens (between 5 and 15, which is ± 1 standard deviation from the mean), both determined from sentence-level labels on the dev set.

These simple, understandable constraints derived from the token-level predictions are effective at restricting the model to the most reliable document-level predictions, including on dramatically different out-of-domain input (SEMEVAL-2017). For the constraints with the original (ORIG.) and revised (REV.) test sets, the same 3 and 1 reviews, respectively, are missed with both constraint variants, which accounts for the nominally lower accuracy as a result of a smaller denominator, and notably, 1 review in each of these sets is incorrectly or ambiguously annotated in the ground-truth data. On average, only around 1 token is admitted per Tweet in the SEMEVAL-2017 data with the distance and magnitude constraints, so the hard token count constraints readily filter most such data for document-level predictions, which is desirable given the mis-match with the

Table 7
Predicting *sentiment* on the original (ORIG.) and revised (REV.) test sets at the review level, using UNICNN+BERT, varying the training data (rows).

Model Train. Data (Num. Reviews)	Review-level Sentiment (Accuracy)	
	ORIG.	REV.
RANDOM	50.2	49.8
ORIG. (3.4k)	92.8	88.7
ORIG.+REV. (1.7k+1.7k)	90.6	96.5
ORIG.$_{\text{DISJOINT}}$+REV. (1.7k+1.7k)	89.5	95.7
ORIG. (19k)	93.0	87.9
ORIG.+REV. (19k+1.7k)	93.0	94.3
ORIG.$_{\text{DISJOINT}}$+REV. (19k-1.7k+1.7k)	93.0	90.2

Table 8
Review-level test accuracy with UNICNN+BERT trained on the ORIG. (3.4k) data using UNICNN+BERT+$K_8NN_{DIST.}$ to constrain predictions. Token-level predictions are constrained by a maximum allowed distance to the nearest match in the support set and K-NN output magnitude limits derived from correct approximations on the K-NN dev set. Document-level predictions are admitted based on a minimum threshold on the proportion of admitted tokens among all tokens in the document ("Admitted token % min") and, optionally, an additional constraint on the allowed range of admitted tokens ("Admitted token min, max").

Test Data	Review-level Sentiment (Accuracy)	Admitted token % min	Admitted token min, max	Admitted n	n/N
ORIG.	92.8			488	1.0
ORIG.	96.2	●		78	0.16
ORIG.	93.0	●	●	43	0.09
REV.	88.7			488	1.0
REV.	98.1	●		52	0.11
REV.	97.0	●	●	33	0.07
SEMEVAL-2017	77.8			4,750	1.0
SEMEVAL-2017	81.4	●		576	0.12
SEMEVAL-2017	100.0	●	●	1	0.0002

training data. In contrast, the orthogonal approach of seeking more robust predictions by including source-target pairs was not consistently beneficial, as shown in Table C3.

Token-Level Feature Detection. The token-level feature detections follow a similar pattern with regard to the training data sets as the document-level predictions, with gains observed with the locally re-edited data, and to a lesser extent, the disjoint sets, as shown in Table 9 and the true zero-shot setting shown in the `red` and black rows of Table 10. The predictions from the K-NN are at least as effective as the original model. As with the error detection experiments, the inference-time decision rules can be used to make updates to the model without retraining (Table 10), which in some cases, results in $F_{0.5}$ scores approaching that of training on that same data.

The observed patterns are analogous on the professionally annotated CONTRAST test set, as shown in Tables 11 and 12. A relatively modest amount of labeled data in the support set is sufficient for improving effectiveness in detecting the token-level sentiment features as seen in the rightmost column of Table 12.

5.3 Sentiment Classification and Feature Detection: Discussion

As with error detection, on the sentiment data sets we demonstrate that we can leverage dense representation matching to update a model and to improve token-level feature detection. Remarkably, with a strong neural model and an inductive bias conducive to matching, we can start to close the distance with models trained with domain-shifted data by just updating the support set, which points to new flexibility in adapting models. However, this still requires a least some data from the distribution of the new domain to be available. When we carefully control for data distributions, robust prediction over data from unseen domain-shifted and out-of-domain distributions remains challenging, ceteris paribus, even with recently proposed data perturbation approaches,

Table 9
Predicting *sentiment diffs* at the token level ($F_{0.5}$). All results are with the UNICNN+BERT model, varying the training data, except for the second row with UNICNN+BERT+K_8NN$_{\text{DIST.}}$. The decision boundary is tuned with token-level diffs from 245 ORIG. dev set reviews (cf., the true zero-shot setting in Table 10).

Model Train. Data (Num. Reviews)	Token-level Sentiment Diffs ($F_{0.5}$)	
	ORIG.	REV.
RANDOM	6.0	7.6
ORIG. (3.4k)$_{+\text{ORIG\_DEV}}$ (K_8NN$_{\text{dist.}}$)	29.5	23.5
ORIG. (3.4k)$_{+\text{ORIG\_DEV}}$	26.2	22.5
ORIG.+REV. (1.7k+1.7k)$_{+\text{ORIG\_DEV}}$	32.4	33.1
ORIG.$_{\text{DISJOINT}}$+REV. (1.7k+1.7k)$_{+\text{ORIG\_DEV}}$	32.4	31.5
ORIG. (19k)$_{+\text{ORIG\_DEV}}$	24.8	21.7
ORIG.+REV. (19k+1.7k)$_{+\text{ORIG\_DEV}}$	28.8	27.9
ORIG.$_{\text{DISJOINT}}$+REV. (19k-1.7k+1.7k)$_{+\text{ORIG\_DEV}}$	28.2	26.8

Table 10
Predicting *sentiment diffs* at the token level ($F_{0.5}$) with UNICNN+BERT, applying the exemplar auditing decision rules. Predictions without accessing the support set (𝕊) are displayed in red. Underlined results indicate 𝕊 contains additional reviews or labels not seen by the model during training. Results with access to token-level labels in 𝕊 are further highlighted in blue.

	Token-level Sentiment Diffs ($F_{0.5}$)					
	𝕊 = ORIG. (1.7k)		𝕊 = REV. (1.7k)		𝕊 = ORIG.+REV. (1.7k+1.7k)	
Model Train. Data (Num. Reviews)	Test: ORIG.	Test: REV.	Test: ORIG.	Test: REV.	Test: ORIG.	Test: REV.
RANDOM	6.0	7.6	6.0	7.6	6.0	7.6
ORIG. (3.4k)	10.5	11.2	10.5	11.2	10.5	11.2
ORIG.+EXAG (3.4k)	16.9	15.8	18.2	18.3	18.0	17.8
ORIG.+EXAT (3.4k)	34.7	28.2	30.1	29.1	32.8	28.8
ORIG.+REV. (1.7k+1.7k)	20.1	22.0	20.1	22.0	20.1	22.0
ORIG.+REV.+EXAG (1.7k+1.7k)	32.0	31.4	30.4	33.2	31.5	33.7
ORIG.+REV.+EXAT (1.7k+1.7k)	39.2	33.3	31.6	36.1	37.4	37.1
ORIG.$_{\text{DISJOINT}}$+REV. (1.7k+1.7k)	15.8	17.4	15.8	17.4	15.8	17.4
ORIG.$_{\text{DISJOINT}}$+REV.+EXAG (1.7k+1.7k)	28.1	26.8	26.0	28.3	27.3	28.1
ORIG.$_{\text{DISJOINT}}$+REV.+EXAT (1.7k+1.7k)	38.8	32.8	33.3	34.9	37.0	34.5

which is consistent with the broad patterns observed in the contemporaneous works of Taori et al. (2020) and Gulrajani and Lopez-Paz (2021) for image data. This is a point of concern for higher-risk settings, as some amount of domain shift or subpopulation shift will invariably occur in many real-world settings.

Faced with these challenges, we can instead constrain document-level predictions based on an interpretable token-level K-NN derived from the deep model. This combination of feature-level detection derived from document-level labels, dense matching, and heuristics that can be traced back to individual token-level predictions across the support set offers an alternative, practical approach for deploying deep models in higher-risk settings, in which we refrain from predicting over domain-shifted data and out-of-domain data over which reliable predictions and bounds remain elusive. In this way, we can refrain from predicting when necessary and then re-label, update, and as

Table 11
Predicting review-level sentiment (accuracy) and token-level *sentiment diffs* ($F_{0.5}$) on the professionally annotated CONTRAST test set. In the second column, the decision boundary is the same as that tuned for Table 9 using 245 ORIG. dev set reviews, as indicated by the ($_{+\text{ORIG\_DEV}}$) label (cf., the true zero-shot setting in Table 12).

	Contrast Sets	
Model Train. Data (Num. Reviews)	Review-level Sentiment (Accuracy) CONTRAST	Token-level Sentiment Diffs ($F_{0.5}$) CONTRAST
RANDOM	49.8	8.4
ORIG. (3.4k)	82.4	17.1 ($_{+\text{ORIG\_DEV}}$)
ORIG.+REV. (1.7k+1.7k)	93.0	28.4 ($_{+\text{ORIG\_DEV}}$)
ORIG.$_{\text{DISJOINT}}$+REV. (1.7k+1.7k)	91.2	26.9 ($_{+\text{ORIG\_DEV}}$)
ORIG. (19k)	81.4	18.0 ($_{+\text{ORIG\_DEV}}$)
ORIG.+REV. (19k+1.7k)	90.0	23.5 ($_{+\text{ORIG\_DEV}}$)
ORIG.$_{\text{DISJOINT}}$+REV. (19k-1.7k+1.7k)	88.1	23.4 ($_{+\text{ORIG\_DEV}}$)

Table 12
Predicting *sentiment diffs* at the token level ($F_{0.5}$) with UNICNN+BERT on the CONTRAST test set, applying the exemplar auditing decision rules. Predictions without accessing the support set ($\mathbb{S}$) are displayed in red. Underlined results indicate $\mathbb{S}$ contains additional reviews or labels not seen by the model during training. Results with access to token-level labels in $\mathbb{S}$ are further highlighted in blue. $|\mathbb{S}|$ is relatively small in the rightmost column. None of the models see data from the CONTRAST set dev set, either in training or in $\mathbb{S}$.

	Contrast Sets			
	Token-level Sentiment Diffs ($F_{0.5}$)			
	$\mathbb{S} =$ ORIG. (1.7k)	$\mathbb{S} =$ REV. (1.7k)	$\mathbb{S} =$ ORIG.+REV. (1.7k+1.7k)	$\mathbb{S} =$ ORIG._DEV+REV._DEV (245+245)
Model Train. Data (Num. Reviews)	Test: CONTRAST	Test: CONTRAST	Test: CONTRAST	Test: CONTRAST
RANDOM	8.4	8.4	8.4	8.4
ORIG. (3.4k)	12.3	12.3	12.3	12.3
ORIG.+EXAG (3.4k)	15.1	18.2	17.6	16.4
ORIG.+EXAT (3.4k)	25.7	27.0	27.5	24.0
ORIG.+REV. (1.7k+1.7k)	21.2	21.2	21.2	21.2
ORIG.+REV.+EXAG (1.7k+1.7k)	27.8	29.0	29.8	29.1
ORIG.+REV.+EXAT (1.7k+1.7k)	28.5	29.6	31.6	27.9
ORIG.$_{\text{DISJOINT}}$+REV. (1.7k+1.7k)	17.6	17.6	17.6	17.6
ORIG.$_{\text{DISJOINT}}$+REV.+EXAG (1.7k+1.7k)	24.7	25.7	25.8	26.0
ORIG.$_{\text{DISJOINT}}$+REV.+EXAT (1.7k+1.7k)	27.2	30.2	29.6	28.7

needed, re-train models in a continual loop based on these methods. For instructive purposes, we contrast such a framework with local re-edits in Figure 4.

6. Sentiment Data: Binary Prediction of Local Annotation Edits

In Section 5.1, we found locally re-edited data to be useful in analyzing and evaluating feature detection for a classification task typically only labeled at the document-level. In
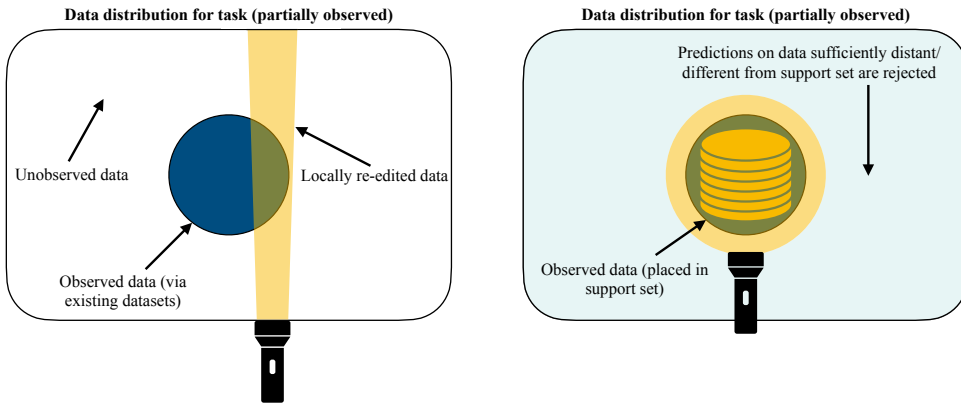
Figure 4
Local re-edits and the proposed approach for dense representation matching can be used in conjunction, but here we emphasize the contrasts for instructional purposes. Manually perturbing data around identified features, creating source-target pairs (over this small slice, illuminated by the flashlight at *left*), can expand a training set and be useful for analysis; however, re-annotating in this manner can be a non-trivial task to avoid inadvertently creating annotation artifacts. As an alternative outlook for higher-risk settings (*right*), we can place as much data as possible into the support set—including data not seen in training—and then conservatively only admit predictions matched closely to the support set, with flexibility over the unit of analysis using our proposed methods, sending rejected predictions to a human for further adjudication, and/or labeling.

this section, we use the same data sets to demonstrate that our proposed methods can be used to uncover subtle distributional differences across annotations, which can be used, for example, for filtering and performing quality control on data sets for training and evaluation.

6.1 Binary Prediction of Local Annotation Edits: Experiments

Kaushik, Hovy, and Lipton (2020) report that the $BERT_{BASE_{UNCASED}}$+FT model is able to distinguish original vs. revised reviews (hereafter, "annotator domain") with an accuracy of about 77 percent. We investigate this further, illustrating how the proposed approach for token-level detections can be used for fine-grained text analysis.

6.1.1 Data: Predicting Annotator Domain (Original vs. Revised). We assign $Y = -1$ to the original reviews and $Y = 1$ to the revised reviews. We report results at the review level on varying subsets of the data, including splits by sentiment. We refer to the subset of original *and* revised reviews restricted to reviews with negative sentiment with the label (ORIG.+REV.)∧NEG., and similarly for other subsets. We derive token-level labels analogously to those created for sentiment in Section 5.1, except the diffs here represent the transduction from revised reviews (source) to original reviews (target). Applicable tokens in revised reviews receive a class 1 ($y_n = 1$) label, whereas tokens in original reviews are all assigned a $y_n = -1$ label. We similarly analyze the professionally annotated CONTRAST test set of Gardner et al. (2020), predicting the original reviews vs. the professionally annotated alternatives.

Table 13
Predicting original (ORIG.) vs. revised (REV.) data using the UNICNN+BERT model on the test set, with additional results subdivided by sentiment and the annotator domain classes. RANDOM has an accuracy of ≈ 50 for each subset.

Test (Sub-)Set	Review-level Annotator Domain (*Not Sentiment*)	
	Accuracy	Num. Reviews
ORIG.+REV.	79.6	976
ORIG.	78.7	488
REV.	80.5	488
(ORIG.+REV.)∧NEG.	84.0	488
(ORIG.+REV.)∧POS.	75.2	488
ORIG.∧NEG.	84.0	243
ORIG.∧POS.	73.5	245
REV.∧NEG.	84.1	245
REV.∧POS.	77.0	243

6.1.2 Models. We train the UNICNN+BERT model on the 3414 parallel original and counterfactually augmented revised reviews, using the 490 paired reviews of the dev set to choose the epoch with highest accuracy.

6.2 Binary Prediction of Local Annotation Edits: Results

Predicting Annotator Domain (Original vs. Revised). With the UNICNN+BERT model, original reviews are distinguishable from counterfactually revised reviews with an accuracy of around 80%, as shown in Table 13. The revised reviews are slightly easier to distinguish in general (accuracy of 80.5 vs. 78.7). The negative reviews are particularly distinct in relative terms, with the accuracy nearly 9 points higher on the negative reviews in the combined set, with an accuracy of 84.0 vs. 75.2 for the positive reviews.

We further examine the particularly distinctive language used in the negative reviews using the aggregate feature extraction of Section 3.5. We split the dev set[14] according to the true document-level labels. Table 14 presents the top and lowest scoring negative class (i.e., original reviews) unigrams and positive class (i.e., revised reviews) unigrams, by total score ($^{\text{total}} n\text{-gram}^{-}$ and $^{\text{total}} n\text{-gram}^{+}$) for the dev set reviews for each class,[15] as well as the corresponding unigram frequency. We see a sharp distinction between the words most discriminative for each class. Certain unigrams, such as `not` and `bad`, occur with similar frequency in the original and revised reviews, but have diametrically opposed weightings for the respective classes. It seems that words that tend to be sentiment-laden, especially those that are of negative sentiment, are particularly discriminative features for distinguishing revised reviews. In Table 15, we show the

---

14 The overall accuracy for annotator domain prediction is 79.4 on the dev set, which is similar to that of the test set (79.6).

15 The analogous $^{\text{total}} n\text{-gram}^{-}$ and $^{\text{total}} n\text{-gram}^{+}$ scores for REV. and ORIG., respectively, which are not shown, exhibit patterns in the expected, corresponding directions.

Table 14
The top and lowest scoring negative class (i.e., original reviews) unigrams and positive class (i.e., revised reviews) unigrams, by total score ($^{\text{total}}n\text{-gram}^-$ and $^{\text{total}}n\text{-gram}^+$) for the dev set reviews for the respective class. We display the total score to highlight that certain unigrams, such as not and bad, occur with similar frequency in the original and revised reviews, but have diametrically opposed weightings for the respective classes.

Review-level	Annotator Domain	(*Not Sentiment*)			
Orig.			Rev.		
unigram	$^{\text{total}}n\text{-gram}^-$ score	Total Frequency	unigram	$^{\text{total}}n\text{-gram}^+$ score	Total Frequency
but	41.5	249	not	61.4	229
waste	18.5	22	terrible	54.3	20
any	11.9	56	least	44.1	26
just	11.0	112	bad	43.1	61
still	8.6	40	worst	32.6	22
that	7.7	394	poor	31.9	21
only	7.6	70	awful	22.4	13
But	7.6	42	dislike	20.2	9
moving	5.8	7	great	18.5	69
completely	5.3	18	boring	18.1	25
		. . . SKIPPED . . .			
hated	−1.2	3	missed	−1.8	4
excited	−1.3	3	without	−1.8	21
horrible	−1.4	5	just	−2.1	97
worst	−1.4	19	lacks	−2.2	3
usual	−1.6	4	lost	−2.3	9
disliked	−1.6	1	I	−2.6	561
worse	−1.9	8	that	−3.1	395
hate	−1.9	3	any	−4.3	40
bad	−5.5	64	waste	−4.5	9
not	−7.6	217	but	−10.0	203

5-grams normalized by occurrence.[16] The most discriminating phrases across classes are distinct, with the contextual use of words such as "bad," "not," and "waste" recognized by the model as being distinctive of original vs. revised reviews.

In Table 16 we display the top two revised reviews, ranked by $n\text{-gram}^+_{1:N}$, normalized by length. We have further highlighted both the ground-truth token-level domain diffs and the zero-shot sequence labeling predictions by the model (i.e., $s_n^{+-} > 0$). The token-level domain diff predictions typically are subsets of the true diffs, with a focus on particularly sentiment-laden words, along the lines of what was shown in tables 14 and 15. More generally, rather remarkably, the zero-shot sequence labeling is sufficiently effective that the approach can be used as a tool for quickly scanning through a data set for distinctive words and phrases conditional on the document-level label, as demonstrated with additional examples in Table D1. Interestingly, just reading the documents in isolation, it is not always obvious that many of the detected diffs are from revisions, yet the model is nonetheless often able to detect such subtle distributional differences.

Predicting Annotator Domain (Original vs. Professional Revisions). The model is nearly as effective at distinguishing the professionally annotated reviews as the crowd-sourced

---

16 For display purposes, we have dropped subsequent *n*-grams with the same score, which typically just differ by a single non-discriminating word as the prefix or suffix token.

Table 15
The top and lowest scoring negative class (i.e., original reviews) 5-grams and positive class (i.e., revised reviews) 5-grams, normalized by occurrence ($^{\mathrm{mean}}n\text{-gram}^-$ and $^{\mathrm{mean}}n\text{-gram}^+$) for the dev set reviews for the respective class.

	Review-level **Annotator Domain** (*Not Sentiment*)		
Orig.		Rev.	
5-gram	$^{\mathrm{mean}}n\text{-gram}^-$ score	5-gram	$^{\mathrm{mean}}n\text{-gram}^+$ score
little bit, but it still	3.9	his awful performance did not	11.3
bit, but it still managed	3.7	dominated this film, his awful	10.4
movie, but many elements ruined	3.4	Come is indeed a terrible	10.3
killer down. A serious waste	3.4	a terrible work of speculative	10.3
this slow paced, boring waste	3.4	This was a very bad	10.0
movie is just a waste	3.1	/> A terrible look at	8.8
waste of time. The most	2.9	dream home. A terrible	8.8
to be nice people, but	2.8	movie is not a lot	8.2
nice people, but can't carry	2.8	This movie is not a	8.2
people, but can't carry a	2.8	remains one of my least	7.9
	…SKIPPED…		
film. The usual superb acting	−1.6	either been reduced to stereo	−1.7
disliked it and looking at	−1.6	around have either been reduced	−1.7
the reasons that I disliked	−1.6	would simply be a waste	−1.7
film or an even worse	−2.0	don't waste your time and	−1.7
this is such a bad	−2.5	about lovey-dovey romance, don't waste	−1.9

Table 16
Top two revised reviews in the counterfactually augmented dev set, ranked by $n\text{-gram}^+_{1:N}$, normalized by length. We have included the original review, ORIGINAL, and the revised review, TRUE (REV.), where underlines indicate ground-truth token-level annotator domain diffs (i.e., that the token participated in a transduction between an original and revised review). We show the prediction by the UNICNN+BERT model to predict original vs. revised reviews, with token-level predictions underlined, and correct predictions further highlighted in blue.

	Review-level **Annotator Domain** (*Not Sentiment*)
	Dev. Set Document 244/245
ORIGINAL	This is actually one of my favorite films, I would recommend that EVERYONE watches it. There is some great acting in it and it shows that not all ''good'' films are American....
TRUE (REV.)	This is actually one of my <u>least</u> favorite films, I would <u>not</u> recommend that <u>ANYONE</u> watches it. There is some <u>bad</u> acting in it and it shows that <u>all ''bad''</u> films are American....
UNICNN+BERT (REV.) LEN. NORM. SCORE: 0.164	This is actually one of my <u>least</u> favorite films, I would <u>not</u> recommend that ANYONE watches it. There is some bad acting in it and it shows that all ''bad'' films are American....
	Dev. Set Document 266/267
ORIGINAL	One of the great classic comedies. Not a slapstick comedy, not a heavy drama. A fun, satirical film, a buyers beware guide to a new home. /> />Filled with great characters all of whom, Cary Grant is convinced, are out to fleece him in the building of a dream home. /> />A great look at life in the late 40's. /> />
TRUE (REV.)	One of the <u>bad</u> classic comedies. Not a slapstick comedy, not a heavy drama. A <u>boring, unfunny</u> film, a buyers beware guide to a new home. /> />Filled with <u>terrible</u> characters all of whom, Cary Grant is <u>falsely</u> convinced, are out to fleece him in the building of a dream home. /> <u>/>A terrible</u> look at life in the late 40's. /> />
UNICNN+BERT (REV.) LEN. NORM. SCORE: 0.133	One of the bad classic comedies. Not a slapstick comedy, not a heavy drama. A <u>boring,</u> unfunny film, a buyers beware guide to a new home. /> />Filled with terrible characters all of whom, Cary Grant is falsely convinced, are out to fleece him in the building of a dream home. /> />A <u>terrible</u> look at life in the late 40's. /> />

Table 17
Predicting original (ORIG.) vs. revised contrast set (CONTRAST) data using the UNICNN+BERT model on the test set, with additional results subdivided by sentiment and the annotator domain classes. RANDOM has an accuracy of ≈ 50 for each subset.

	Contrast Sets	
Test (Sub-)Set	Review-level Annotator Domain (*Not Sentiment*)	
	Accuracy	Num. Reviews
ORIG.+CONTRAST	77.8	976
ORIG.	78.7	488
CONTRAST	76.8	488
(ORIG.+CONTRAST)∧NEG.	78.7	488
(ORIG.+CONTRAST)∧POS.	76.8	488
ORIG.∧NEG.	84.0	243
ORIG.∧POS.	73.5	245
CONTRAST∧NEG.	73.5	245
CONTRAST∧POS.	80.2	243

revised reviews, with the overall accuracy only a couple of points lower, as shown in Table 17, even though the model only sees crowd-sourced revisions in training and development. The negative reviews are again easier to distinguish overall, but in this case we see that this is driven by accuracy on the original reviews, which are more readily distinguished. This might be attributable to the effects of the domain shift, with the original reviews being seen in training, while the professionally annotated counterparts are not. As with the counterfactually augmented edits, without such model assistance, it is often not obvious that a review has been revised, especially given the otherwise informal language of movie reviews. However, the class-conditional feature detection is strong enough that the token-level predictions can be visualized and some of the discriminative words and phrases participating in the diffs identified, as shown in Table D2.

6.3 Prediction of Local Annotation Edits: Discussion

With effective zero-shot sequence labeling, we gain a straightforward means of aggregating features from a deep network when only given document-level labels. As we have shown, this can be used to analyze text data sets, detecting rather subtle distributional differences that are not readily perceptible without such model assistance, at least at scale. Deep networks are typically viewed as strong predictors at the unit of analysis of the training set's labels; with the mechanism proposed here, we gain a means of leveraging that discriminative ability at lower resolutions to analyze the input data.

7. Discussion

This new facility for dense representation matching at resolutions of the input more fine-grained than available labels is a substantive departure from existing approaches in computational linguistics, providing new flexibility for locally updating a model

and analyzing data sets under the model. It draws a connection between attention-style mechanisms and the older distance metric learning literature (Weinberger and Saul 2009, inter alia), relying on the inductive bias of the CNN to learn summarized representations of the expressive deep network for subsequent matching via simple Euclidean distances. Fortunately, from an efficient compute perspective, this works well when training with standard cross-entropy losses against the available labels without resorting to expensive supervised contrastive losses searching through representations during initial training. When a stronger sense of interpretability is needed, we can then subsequently train an effective K-NN approximation with just 3 learnable parameters from the frozen representations.

Prototypical networks (Snell, Swersky, and Zemel 2017) and matching networks (Vinyals et al. 2016) can also be updated by modifying a support set, but the means of doing so are markedly different from what we have proposed, motivated by different intended use cases. Critical for NLP settings, we are concerned with fine-grained feature detection, which necessitates the proposed indirect approach for deriving predictions and representations from an imputation-trained deep network, and a different approach for training. Additionally, unlike prototypical networks, we perform matching against every instance (in fact, every *token*) in the support set, rather than class means, which is a strength rather than a weakness for the intended interpretability and data set analysis applications. Finally, matching networks can also be viewed as a particular weighted K-NN. In contrast, our K-NN approximation of an already trained model is proposed as a parsimonious, interpretable replacement of the original model, and is trained accordingly.[17]

8. Conclusion

Deep networks are typically viewed as strong predictors that are otherwise immutable and inscrutable black boxes, with the non-identifiable parameters running into the millions and higher. In this context, we have demonstrated a series of approaches toward a more actionable understanding of a deep network over its input data. We have shown that a kernel-width-one CNN and a linear layer over a deep network is effective for deriving token-level predictions when only given document-level labels for training. This approach for class-conditional feature detection enables dense representation matching against a support set with known labels, which can be used with inference-time decision rules to constrain predictions. Additionally, we have shown that we can altogether replace a model's output with an interpretable weighting over instances with known labels without loss of predictive effectiveness. In this way, we gain sequence labeling at varying label resolutions; local updatability of a model without re-training; interpretable token-level constraints over domain-shifted and out-of-domain data; and more generally, a model-assisted means for uncovering patterns in large data sets that may not be readily detectable at scale without the expressive, deep networks.

---

17 With regard to model approximations, there is also an indirect connection to work relating kernel machines to neural architectures and vice versa (Cho and Saul 2009; Alber et al. 2017, inter alia).

Appendix A. Contents

In Appendices B, C, and D, we provide additional results and output for the experiments on the grammatical error detection task, the sentiment data sets, and for the experiments predicting annotator re-edits, respectively.

Appendix B. Grammatical Error Detection Analysis and Examples

Table B1 shows five random examples of original sentences from the FCE test set and the corresponding labeled outputs from the CNN, UNICNN+BERT, UNICNN+BERT+MM, and UNICNN+BERT+S* models.

Tables B2 and B3 show the nearest matches used for the proposed inference-time decision rules for the first three sentences with ground-truth grammatical errors from Table B1 for the UNICNN+BERT+MM and UNICNN+BERT+S* models, respectively. We have provided the exemplar tokens and associated sentences from the support set (here, consisting of the FCE training set) wherever the model makes a positive prediction. For reference, we have also provided the sentence corresponding to the exemplar representation for any tokens marked in the ground-truth labels but missed by the model. The qualitative analysis is consistent with the quantitative results in the main text: When the test prediction is in the same direction as the prediction of the exemplar from the support set, the corresponding contexts, and the exemplar word itself—which is not always a verbatim lexical match—are often similar, particularly when the L^2 distances are low.

Table B4 contains the unigram positive class n-grams normalized by occurrence ($^{\text{mean}} n$-gram$^+$) for the training sentences for which $Y = 1$. The top scoring such unigrams constitute a relatively sharp list of misspellings. We also include the lowest scoring such unigrams at the bottom of the table, as a check on our featuring scoring method. The ranked features are as we would expect, with the lowest scoring unigrams being names and other words that are otherwise correctly spelled.

Table B5 compares the K-NN output with that of the original model, UNICNN+BERT+MM, on the domain-shifted test set, as with Figure 3 in the main text.

Table B1
Five random sentences from the FCE test set. The ground-truth labeled sentences are marked
TRUE, with ground-truth token-level labels underlined. In the case of model output, underlines
indicate predicted error labels. Note that sentence 1551, as with the other sentences, is verbatim
from the gold test set.

	Sentence 174
TRUE	There <u>are</u> some <u>informations</u> you have asked me about .
CNN	There <u>are some informations you have</u> asked me about .
UNICNN+BERT	There are some <u>informations</u> you have asked me about .
UNICNN+BERT+MM	There are some <u>informations</u> you have asked me about .
UNICNN+BERT+S*	There are some <u>informations</u> you have asked me about .

	Sentence 223
TRUE	There is space for about five hundred people .
CNN	There is space for <u>about five hundred people .</u>
UNICNN+BERT	There is space for about five hundred people .
UNICNN+BERT+MM	There is space for about five hundred people .
UNICNN+BERT+S*	There is space for about five hundred people .

	Sentence 250
TRUE	It is n't easy <u>giving</u> an answer <u>at</u> this question .
CNN	It <u>is</u> n't easy <u>giving an answer</u> at this question .
UNICNN+BERT	It is n't easy giving an <u>answer at</u> this <u>question</u> .
UNICNN+BERT+MM	It is n't easy giving an answer <u>at</u> this question .
UNICNN+BERT+S*	It is n't easy giving an answer <u>at</u> this question .

	Sentence 1302
TRUE	Your group has been booked <u>in Palace</u> Hotel which is one of the most comfortable hotels in London .
CNN	<u>Your group has been booked in Palace Hotel</u> which is one of <u>the most comfortable</u> hotels in London .
UNICNN+BERT	Your group has been booked <u>in</u> Palace Hotel <u>which</u> is one of the most <u>comfortable</u> hotels in London .
UNICNN+BERT+MM	Your group has been booked <u>in</u> Palace Hotel which is one of the most comfortable hotels in London .
UNICNN+BERT+S*	Your group has been booked in <u>Palace</u> Hotel which is one of the most comfortable hotels in London .

	Sentence 1551
TRUE	By the way <u>you</u> can visit the
CNN	By the way <u>you can visit the</u>
UNICNN+BERT	By the way you can visit the
UNICNN+BERT+MM	By the way you can visit the
UNICNN+BERT+S*	By the way you can visit the

Table B2
Exemplar auditing output for three sentences from Table B1 for the UNICNN+BERT+MM model. Ground-truth labeled sentences are marked TRUE with ground-truth token-level labels underlined. Underlines in the UNICNN+BERT+MM rows indicate predictions. We show the exemplars for the predicted tokens and for reference, any true token labels missed by the model. In both cases, the exemplar tokens from training are labeled by the index into the test sentence, as indicated in brackets. The Euclidean distance between the test token and the exemplar token is labeled with EXEMPLAR DIST. The full training sentence for the exemplar is provided, with underlines indicating ground-truth labels in the case of EXEMPLAR TRUE and training predications from UNICNN+BERT+MM in the case of EXEMPLAR PRED.

	Sentence 174
TRUE	There are[1] some informations[3] you have asked me about .
UNICNN+BERT+MM	There are some informations[3] you have asked me about .
EXEMPLAR [1] DIST.	41.1
EXEMPLAR [1] TRUE	But, there are[1] three things which I would like to tell you .
EXEMPLAR [1] PRED.	But, there are[1] three things which I would like to tell you .
EXEMPLAR [3] DIST.	34.0
EXEMPLAR [3] TRUE	I am very glad to hear that and would like to tell you all the informations[3] you need to know from me .
EXEMPLAR [3] PRED.	I am very glad to hear that and would like to tell you all the informations[3] you need to know from me .
	Sentence 250
TRUE	It is n't easy giving[4] an answer at[7] this question .
UNICNN+BERT+MM	It is n't easy giving an answer at[7] this question .
EXEMPLAR [4] DIST.	71.8
EXEMPLAR [4] TRUE	It 's very difficult describing[4] all emotion I felt .
EXEMPLAR [4] PRED.	It 's very difficult describing[4] all emotion I felt .
EXEMPLAR [7] DIST.	63.3
EXEMPLAR [7] TRUE	I 'm going to reply at[7] your question .
EXEMPLAR [7] PRED.	I 'm going to reply at[7] your question .
	Sentence 1302
TRUE	Your group has been booked in[5] Palace[6] Hotel which is one of the most comfortable hotels in London .
UNICNN+BERT+MM	Your group has been booked in[5] Palace Hotel which is one of the most comfortable hotels in London .
EXEMPLAR [5] DIST.	57.0
EXEMPLAR [5] TRUE	Secondly I would prefer to be accommodate in[5] log cabins .
EXEMPLAR [5] PRED.	Secondly I would prefer to be accommodate in[5] log cabins .
EXEMPLAR [6] DIST.	59.6
EXEMPLAR [6] TRUE	I insisted on going to your theatre, to the Circle[6] Theatre, because I have heard that it is one of the best theatres in London .
EXEMPLAR [6] PRED.	I insisted on going to your theatre, to the Circle[6] Theatre, because I have heard that it is one of the best theatres in London .

Table B3
Exemplar auditing output for three sentences from Table B1 for the UNICNN+BERT+S* model.
Ground-truth labeled sentences are marked TRUE with ground-truth token-level labels
underlined. Underlines in the UNICNN+BERT+S* rows indicate predictions. We show the
exemplars for the predicted tokens and for reference, any true token labels missed by the model.
In both cases, the exemplar tokens from training are labeled by the index into the test sentence,
as indicated in brackets. The Euclidean distance between the test token and the exemplar token
is labeled with EXEMPLAR DIST. The full training sentence for the exemplar is provided, with
underlines indicating ground truth labels in the case of EXEMPLAR TRUE and training
predications from UNICNN+BERT+S* in the case of EXEMPLAR PRED.

	Sentence 174
TRUE	There are[1] some informations[3] you have asked me about .
UNICNN+BERT+S*	There are some informations[3] you have asked me about .
EXEMPLAR [1] DIST.	32.7
EXEMPLAR [1] TRUE	But, there are[1] three things which I would like to tell you .
EXEMPLAR [1] PRED.	But, there are[1] three things which I would like to tell you .
EXEMPLAR [3] DIST.	24.0
EXEMPLAR [3] TRUE	I am very glad to hear that and would like to tell you all the informations[3] you need to know from me .
EXEMPLAR [3] PRED.	I am very glad to hear that and would like to tell you all the informations[3] you need to know from me .
	Sentence 250
TRUE	It is n't easy giving[4] an answer at[7] this question .
UNICNN+BERT+S*	It is n't easy giving an answer at[7] this question .
EXEMPLAR [4] DIST.	54.6
EXEMPLAR [4] TRUE	To say nothing about his or her giving[4] advice !
EXEMPLAR [4] PRED.	To say nothing about his or her giving[4] advice !
EXEMPLAR [7] DIST.	44.0
EXEMPLAR [7] TRUE	I 'm going to reply at[7] your question .
EXEMPLAR [7] PRED.	I 'm going to reply at[7] your question .
	Sentence 1302
TRUE	Your group has been booked in[5] Palace[6] Hotel which is one of the most comfortable hotels in London .
UNICNN+BERT+S*	Your group has been booked in Palace[6] Hotel which is one of the most comfortable hotels in London .
EXEMPLAR [5] DIST.	43.6
EXEMPLAR [5] TRUE	Accommodation in[5] log cabins would be better for me, because they are more comfortable .
EXEMPLAR [5] PRED.	Accommodation in[5] log cabins would be better for me, because they are more comfortable .
EXEMPLAR [6] DIST.	46.8
EXEMPLAR [6] TRUE	There will be The London Fashion and Leisure Show in Central[6] Exhibition Hall on the 14th of March .
EXEMPLAR [6] PRED.	There will be The London Fashion and Leisure Show in Central[6] Exhibition Hall on the 14th of March .

Table B4
The top and lowest scoring unigram positive class n-grams normalized by occurrence
($^{\text{mean}} n$-gram$^+$) for the training sentences that are marked as incorrect (i.e., belonging to the
positive class) for the UNICNN+BERT model.

unigram	$^{\text{mean}} n$-gram$^+$ score	Total Frequency
wating	22.5	1
noize	21.9	1
exitation	21.5	1
exitement	21.2	1
toe	20.1	1
fite	20.0	1
ofer	20.0	2
n	19.7	5
intents	18.6	1
wit	17.7	2
defences	17.5	1
meannes	17.5	1
baying	17.3	1
saing	17.1	2
dipends	17.0	1
lair	16.7	2
torne	16.7	1
farther	16.2	1
andy	16.0	1
seasonaly	15.9	1
remainds	15.6	1
sould	15.5	4
availble	15.5	3
...SKIPPED...		
sixteen	−1.7	3
Uruguay	−1.7	1
Jose	−1.7	1
leg	−1.7	3
Joseph	−2.0	1
deny	−2.1	1
Sandre	−2.2	1
leather	−2.4	2
shoulder	−2.6	1
apartheid	−2.8	1
tablets	−2.8	1
Martial	−3.0	1
Lorca	−3.1	1

Table B5

The original model (UNICNN+BERT+MM) output, $f(\cdot)$, and the K-NN approximation output, $f(\cdot)^{KNN}$, as comparative measures of prediction reliability on the domain-shifted FCE+NEWS2K test set. The K-NN only has access to the original FCE training set. Quantiles are constructed by equally dividing the data after sorting based on the magnitude of the output, separated by class. When considering all of the data (4th quartile), the K-NN is already a modestly stronger predictor, but the difference amplifies with the smaller subsets because the K-NN output is a slightly stronger measure of prediction uncertainty and/or a stronger predictor conditioned on output magnitude, with relatively more of the correct predictions clustered at higher magnitudes. The K-NNs of the remaining models also track prediction reliability at least as closely as that of the original models in similar oracle sorting, as shown in Figure 3, with the advantage that the K-NNs' model terms are readily inspectable and interpretable, as described in the main text.

Model	Quantiles				Metric	Threshold	Prediction
	1st	2nd	3rd	4th			
		$\rightarrow$ cumulative $\rightarrow$					
UNICNN+BERT+MM+K$_8$NN$_{DIST.}$							
							$\hat{y}^{KNN} = 1$
	0.91	0.52	0.24	0.00		$> |f(\cdot)^{KNN}|$	
	56.3	44.2	36.7	32.1	$F_{0.5}$		
	50.7	38.8	31.7	27.4	Acc.		
	1,459	2,918	4,377	5,837	N		
							$\hat{y}^{KNN} = -1$
	1.32	1.11	0.86	0.00		$> |f(\cdot)^{KNN}|$	
	97.8	97.1	96.0	94.6	Acc.		
	21,690	43,380	65,070	86,760	N		
UNICNN+BERT+MM							
							$\hat{y} = 1$
	16.55	7.34	2.80	0.00		$> |f(\cdot)|$	
	48.7	38.7	32.8	29.5	$F_{0.5}$		
	43.1	33.5	28.1	25.0	Acc.		
	1,831	3,662	5,493	7,327	N		
							$\hat{y} = -1$
	12.13	12.10	9.25	0.00		$> |f(\cdot)|$	
	96.4	96.2	95.7	94.8	Acc.		
	21,317	42,634	63,951	85,270	N		

Appendix C. Sentiment Analysis

Sentiment Diffs for Token-Level Detection. An example of the process to create the token-level detection labels for the sentiment data sets is shown in Table C1. Note that the in-line diffs of the first row are used for data creation, but are not subsequently directly used in training or inference. The diffs are guaranteed to transduce to the source and target and the resulting positive class labels often correspond to positive sentiment. Occasionally there are edge cases created by the diff process and/or the underlying data for which an independent annotator tasked with labeling positive words might conceivably label differently. For example, in this review, "not" is assigned to the positive class, which is consistent with the original and revised diff of the reviews.

Table C1
Example of creating the ground-truth token-level sentiment features diffs data from parallel source (positive sentiment, $Y = 1$) and target (negative sentiment, $Y = -1$) data. Source-target diffs that transduce to $Y = 1$ are colored blue, and those that transduce to $Y = -1$ are colored red. Tokens with positive class token-level feature labels ($y_n = 1$) are underlined in the second row. Under this convention, the corresponding negative review (the final row) is never assigned positive token labels (i.e., the colored red tokens and all other non-blue tokens are assigned $y_n = -1$).

Sentiment Features Diffs Data Creation (Ground-Truth Labels)	
source (positive sentiment) to target (negative sentiment)	I saw this in the summer of 1990. I'm still `<del> amazed </del> <ins> annoyed </ins>` by how `<del> good </del> <ins> bad </ins>` this movie is in 2001.` <br <del> />Incredible </del> <ins> />Implausible </ins>` plot. You'd have to be a child to think this could `<del> not </del>` happen. `  `I'm just really `<del> amazed </del> <ins> annoyed </ins>` by it. `<del> Definitely </del> <ins> Don't </ins>` see this.
Resulting ground-truth labels (positive review: $Y = 1$)	I saw this in the summer of 1990. I'm still <u>amazed</u> by how <u>good</u> this movie is in 2001.` <br` <u>/>Incredible</u> plot. You'd have to be a child to think this could <u>not</u> happen.`  `I'm just really <u>amazed</u> by it. <u>Definitely</u> see this.
Resulting ground-truth labels (negative review: $Y = -1$)	I saw this in the summer of 1990. I'm still annoyed by how bad this movie is in 2001.` <br` />Implausible plot. You'd have to be a child to think this could happen.`  `I'm just really annoyed by it. Don't see this.

Table C2
Accuracy results for predicting *sentiment* on the original (ORIG.) and revised (REV.) test sets. These are reference results placing the proposed models in the context of fine-tuning the Transformer parameters. These models are all trained on the full original training set (19k) *and* the revised training set (1.7k). The results for $BERT_{BASE_{UNCASED}}$+FT, which fine-tunes the $BERT_{BASE_{UNCASED}}$ parameters, are those of Kaushik, Hovy, and Lipton (2020).

Model	Review-level Sentiment (Accuracy)	
	ORIG.	REV.
$BERT_{BASE_{UNCASED}}$+FT	93.2	93.9
UNICNN+$BERT_{BASE_{UNCASED}}$	91.8	91.4
UNICNN+$BERT_{BASE}$	92.2	93.4
UNICNN+BERT	93.0	94.3

Table C3
Predicting *sentiment* on out-of-domain data, the SemEval-2017 Task 4a test set, with
UNICNN+BERT and UNICNN+BERT$_{\text{BASE}_{\text{UNCASED}}}$ ($_{\text{BASE}_{\text{uncased}}}$).

Model Train. Data (Num. Reviews)	Review-level Sentiment (Accuracy) SemEval-2017
RANDOM	50.
ORIG. (3.4k)	77.8
ORIG.+REV. (1.7k+1.7k)	64.2
ORIG.$_{\text{DISJOINT}}$+REV. (1.7k+1.7k)	75.1
ORIG. (19k)	72.0
ORIG.+REV. (19k+1.7k)	66.9
ORIG.$_{\text{DISJOINT}}$+REV. (19k-1.7k+1.7k)	76.5
ORIG. (3.4k) $_{\text{BASE}_{\text{uncased}}}$	75.7
ORIG.+REV. (1.7k+1.7k) $_{\text{BASE}_{\text{uncased}}}$	73.5
ORIG.$_{\text{DISJOINT}}$+REV. (1.7k+1.7k) $_{\text{BASE}_{\text{uncased}}}$	75.2
ORIG. (19k) $_{\text{BASE}_{\text{uncased}}}$	68.5
ORIG.+REV. (19k+1.7k) $_{\text{BASE}_{\text{uncased}}}$	72.6
ORIG.$_{\text{DISJOINT}}$+REV. (19k-1.7k+1.7k) $_{\text{BASE}_{\text{uncased}}}$	76.9

Appendix D. Sentiment Data: Binary Prediction of Local Annotation Edits

Tables D1 and D2 illustrate how the zero-shot sequence labeling predictions from
the UNICNN+BERT model can be used as an assistant for analyzing text data sets,
uncovering subtle patterns that are not easily discoverable in large data sets.

Table D1
Selected sentences pulled from the counterfactually augmented dev set. Underlined words are zero-shot sequence label predictions from the UNICNN+BERT model for predicting annotator domain, with correct predictions further highlighted in blue and incorrect predictions (i.e., in which the token did not participate in a ground-truth token-level diff) in red. For reference, we also provide the original document of the parallel source-target pair.

	Counterfactually Augmented Data
	Review-level Annotator Domain (*Not Sentiment*)
ORIGINAL	Dev. Set Document 40/41 [...] It shocks me that something exceptional like Firefly lasts one season, while garbage like the Battlestar Galactica remake spawns a spin off. [...]
UNICNN+BERT (REV.)	[...] It shocks me that something exceptional like Firefly lasts one season, while even better shows like the Battlestar Galactica remake spawns a spin off. [...]
ORIGINAL	Dev. Set Document 254/255 [...] A well made movie, one which I will always remember, and watch again.
UNICNN+BERT (REV.)	[...] A feeble movie, one which I will always remember poorly, and never watch again.
ORIGINAL	Dev. Set Document 258/259 [...] We need that time again, now more than ever. [...]
UNICNN+BERT (REV.)	[...] We do need that time again, now less than ever. [...]
ORIGINAL	Dev. Set Document 276/277 [...] Highly, hugely recommended!
UNICNN+BERT (REV.)	[...] Highly, hugely not recommended!
ORIGINAL	Dev. Set Document 278/279 almost every review of this movie I'd seen was pretty bad. It's not pretty bad, it's actually pretty good, though not great. [...]
UNICNN+BERT (REV.)	almost every review of this movie I'd seen was pretty bad. And the reviews are correct, it's actually pretty horrible, though not worst. [...]

Table D2
Selected sentences pulled from the contrast sets dev set. <u>Underlined</u> words are zero-shot sequence label predictions from the UNICNN+BERT model for predicting annotator domain, with correct predictions further highlighted in blue and incorrect predictions (i.e., in which the token did not participate in a ground-truth token-level diff) in red. For reference, we also provide the original document of the parallel source-target pair.

Contrast Sets
Review-level **Annotator Domain** (*Not Sentiment*)

	Dev. Set Document 38/39
ORIGINAL	[...] The content of the film was very very moving. [...]
UNICNN+BERT (CONTRAST)	[...] The content of the film was very very <u>missing.</u> [...]

	Dev. Set Document 58/59
ORIGINAL	[...] Anyone who has the slightest interest in Gaelic, folk history, folk music, oral culture, Scotland, British history, multi-culturalism or social justice should go and see this film.
UNICNN+BERT (CONTRAST)	[...] Anyone who has the slightest interest in Gaelic, folk history, folk music, oral culture, Scotland, British history, multi-culturalism or social justice should go and <u>avoid</u> this film.

	Dev. Set Document 146/147
ORIGINAL	[...] It is hard to describe the incredible subject matter the Maysles discovered but everything in it works wonderfully. [...]
UNICNN+BERT (CONTRAST)	[...] It is hard to describe the flawed subject matter the Maysles discovered but everything in it <u>works</u> hopelessly. [...]

	Dev. Set Document 164/165
ORIGINAL	[...] The characters are cardboard clichs of everything that has ever been in a bad Sci-Fi series. [...]
UNICNN+BERT (CONTRAST)	[...] The characters are imaginations <u>of</u> everything that has ever been in a good Sci-Fi series. [...]

	Dev. Set Document 176/177
ORIGINAL	[...] There was also a forgettable sequel several years later, but this instant classic is not to be missed.
UNICNN+BERT (CONTRAST)	[...] There was also a <u>forgettable</u> sequel several years later, which made this <u>instant</u> film even more missable.

	Dev. Set Document 182/183
ORIGINAL	[...] It has very little plot,mostly partying,beer drinking and fighting. [...]
UNICNN+BERT (CONTRAST)	[...] It has very <u>dense</u> plot,mostly partying,beer drinking and fighting. [...]

	Dev. Set Document 184/185
ORIGINAL	[...] Whatever originality exists in this film - unusual domestic setting for a musical, lots of fantasy, some animation - is more than offset by a script that has not an ounce of wit or thought-provoking plot development. [...]
UNICNN+BERT (CONTRAST)	[...] Whatever originality exists in this film - unusual domestic setting for a musical, lots of fantasy, some animation - is more than offset by a script that has <u>so much</u> wit <u>or</u> thought-provoking plot development. [...]

References

Alber, Maximilian, Pieter-Jan Kindermans,
Kristof Schütt, Klaus-Robert Müller, and
Fei Sha. 2017. An empirical study on the
properties of random bases for kernel
methods. In *Advances in Neural Information
Processing Systems*, volume 30,
pages 2760–2771.

Bell, Samuel, Helen Yannakoudakis, and
Marek Rei. 2019. Context is key:
Grammatical error detection with
contextual word representations. In
*Proceedings of the Fourteenth Workshop on
Innovative Use of NLP for Building
Educational Applications*, pages 103–115,
Florence. `https://doi.org/10
.18653/v1/W19-4410`

Chelba, Ciprian, Tomás Mikolov, Mike
Schuster, Qi Ge, Thorsten Brants, Phillipp
Koehn, and Tony Robinson. 2014. One
billion word benchmark for measuring
progress in statistical language modeling.
In *INTERSPEECH 2014, 15th Annual
Conference of the International Speech
Communication Association, Singapore,
September 14-18, 2014*, pages 2635–2639.
`https://doi.org/10.21437
/Interspeech.2014-564`

Cho, Youngmin and Lawrence Saul. 2009.
Kernel methods for deep learning. In
*Advances in Neural Information Processing
Systems*, volume 22, pages 342–350.

Clark, Peter. 1990. A comparison of rule and
exemplar-based learning systems. In
*Machine Learning, Meta-Reasoning and
Logics*, Springer, 159–186. `https://doi
.org/10.1007/978-1-4613-1641-1_8`

Devlin, Jacob, Ming-Wei Chang, Kenton Lee,
and Kristina Toutanova. 2019. BERT:
Pre-training of deep bidirectional
transformers for language understanding.
In *Proceedings of the 2019 Conference of the
North American Chapter of the Association
for Computational Linguistics: Human
Language Technologies, Volume 1 (Long and
Short Papers)*, pages 4171–4186,
Minneapolis, MN.

Gardner, Matt, Yoav Artzi, Victoria Basmov,
Jonathan Berant, Ben Bogin, Sihao Chen,
Pradeep Dasigi, Dheeru Dua, Yanai Elazar,
Ananth Gottumukkala, Nitish Gupta,
Hannaneh Hajishirzi, Gabriel Ilharco,
Daniel Khashabi, Kevin Lin, Jiangming
Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang

Ning, Sameer Singh, Noah A. Smith,
Sanjay Subramanian, Reut Tsarfaty, Eric
Wallace, Ally Zhang, and Ben Zhou. 2020.
Evaluating models' local decision
boundaries via contrast sets. In *Findings of
the Association for Computational Linguistics:
EMNLP 2020*, pages 1307–1323. Online.
`https://doi.org/10.18653/v1/2020
.findings-emnlp.117`

Gulrajani, Ishaan and David Lopez-Paz.
2021. In search of lost domain
generalization. In *International Conference
on Learning Representations*. `https://doi
.org/10.1080/01621459.1997.10474027`

Hwang, J. T. Gene and A. Adam Ding. 1997.
Prediction intervals for artificial neural
networks. *Journal of the American Statistical
Association*, 92(438):748–757.

Jain, Sarthak and Byron C. Wallace. 2019.
Attention is not explanation. In *Proceedings
of the 2019 Conference of the North American
Chapter of the Association for Computational
Linguistics: Human Language Technologies,
Volume 1 (Long and Short Papers)*,
pages 3543–3556, Minneapolis, MN.

Kaushik, Divyansh, Eduard Hovy, and
Zachary Lipton. 2020. Learning the
difference that makes a difference with
counterfactually-augmented data. In
*International Conference on Learning
Representations*.

Kim, Yoon. 2014. Convolutional neural
networks for sentence classification. In
*Proceedings of the 2014 Conference on
Empirical Methods in Natural Language
Processing (EMNLP)*, pages 1746–1751,
Doha, Qatar. `https://doi.org/10
.3115/v1/D14-1181`

Mikolov, Tomas, Ilya Sutskever, Kai Chen,
Greg S. Corrado, and Jeff Dean. 2013.
Distributed representations of words and
phrases and their compositionality. In
C. J. C. Burges, L. Bottou, M. Welling,
Z. Ghahramani, and K. Q. Weinberger,
editors, *Advances in Neural Information
Processing Systems 26*, pages 3111–3119.

Pennington, Jeffrey, Richard Socher, and
Christopher Manning. 2014. GloVe: Global
vectors for word representation. In
*Proceedings of the 2014 Conference on
Empirical Methods in Natural Language
Processing (EMNLP)*, pages 1532–1543,
Doha. `https://doi.org/10.3115/v1
/D14-1162`

Rei, Marek. 2017. Semi-supervised multitask
learning for sequence labeling. In
*Proceedings of the 55th Annual Meeting of the
Association for Computational Linguistics
(Volume 1: Long Papers)*, pages 2121–2130,

Vancouver. `https://doi.org/10.18653/v1/P17-1194`

Rei, Marek and Anders Søgaard. 2018. Zero-shot sequence labeling: Transferring knowledge from sentences to tokens. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 293–302, New Orleans, LA. `https://doi.org/10.18653/v1/N18-1027`

Rei, Marek and Anders Søgaard. 2019. Jointly learning to label sentences and tokens. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6916–6923. `https://doi.org/10.1609/aaai.v33i01.33016916`

Rei, Marek and Helen Yannakoudakis. 2016. Compositional sequence labeling models for error detection in learner writing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1181–1191, Berlin. `https://doi.org/10.18653/v1/P16-1112`

Rosenthal, Sara, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 Task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518, Vancouver. `https://doi.org/10.18653/v1/S17-2088`

Schmaltz, Allen, Yoon Kim, Alexander Rush, and Stuart Shieber. 2017. Adapting sequence models for sentence correction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2807–2813, Copenhagen. `https://doi.org/10.18653/v1/D17-1298`

Snell, Jake, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4077–4087.

Taori, Rohan, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. 2020. Measuring robustness to natural distribution shifts in image classification. In *Advances in Neural Information Processing Systems*, volume 33, pages 18583–18599.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 6000–6010.

Vinyals, Oriol, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. 2016. Matching networks for one shot learning. In D. D. Le, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3630–3638.

Weinberger, Kilian Q. and Lawrence K. Saul. 2009. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10(9):207–244.

Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. `https://doi.org/10.18653/v1/2020.emnlp-demos.6`

Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Yannakoudakis, Helen, Ted Briscoe, and Ben Medlock. 2011. A new data set and method for automatically grading ESOL texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189, Portland, OR.

Zeiler, Matthew D. 2012. ADADELTA: An adaptive learning rate method. *CoRR*, abs/1212.5701.