Making the Most of Your Data: Few Shot Learning for Automated Essay Scoring

Stanford CS224N Custom Project MIGHT RESUBMIT BEFORE THE LATE DEADLINE

Samarth Kadaba
Department of Computer Science
Stanford University
skadaba@stanford.edu

Abel John
Department of Computer Science
Stanford University
abeljohn@stanford.edu

Abstract

Developing content-based, classroom-oriented automated essay scoring systems gives teachers the ability to bias large scale score estimation models with their own preferences for essay writing. This preserves the personal nature of student-teacher relationships however poses an algorithmic challenge due to the inherent scarcity of teacher-provided reference essays. Here, we aim to tackle this problem through learning context-dense embeddings which more closely reflect teacher-provided scores from limited training samples. To this end, we 1) demonstrate novel methods of augmenting reference samples using semantic substitution, 2) analyze performance trade-offs using different pairwise loss functions and 3) investigate recurrent architectures for constructing second-order document embeddings. We show that in classifying essay samples according to a non-binary rubric, our method outperforms baseline models evaluated with the same data scarcity constraints. Furthermore, our learned embeddings perform well in clustering reflecting their applicability towards Semantic Textual Similarity tasks and giving instructors the ability to quickly identify groups of students in need of greater support. We simultaneously investigate ensemble methods for dealing with data scarcity in automated essay scoring, providing a combinatorial analysis of the above-mentioned approaches.

1 Key Information to include

- Mentor: Abhinav Garg
- External Collaborators (if you have any): N/A
- Sharing project: N/A

2 Introduction

Motivation In domain-specific, data-sparse contexts it is more relevant for students to be evaluated against peer examples and instructor-provided references than it is to be measured against large corpuses' of score-annotated essays from potentially different domains. However, multi-class classification is notoriously difficult with sparse data from out-of-domain sources (Roa, 2018). Limited instructor examples make traditional methods for automated essay scoring (AES) via classification infeasible due to inherent complexity of document embeddings and poor reflection of semantic textual similarity (STS) in the latent space. To develop small-scale predictive models for use within classroom, we require the ability to generalize from few training examples.

Primer on Approach Therefore we look to few-shot learning as a method for classifying student essays within specific domains. To achieve this, we employ Siamese networks - a contrastive architecture in which loss is computed pair-wise relative to both the distance between two samples in the embedding space and their respective differences in class labels (Bertinetto, 2016). Using Siamese

Networks, we explore methods for learning with limited data to predict multi-class rubric scores for a range of student essays across different domains. Our ensemble approach is threefold. First, we introduce a multicriterion objective to simultaneously learn useful embeddings (for downstream tasks) and classification scores of student essays. Second, we experiment with data augmentation methods, using semantic-substitution weighted based on differential gradients. Third, we increase model complexity by introducing recurrent and convolutional architectures for constructing higher-order embeddings.

Rational Design An intuitive justification of the above methods follows. We implicitly infer that embeddings of documents with the same score have textual similarity that can be represented in their latent space. By optimizing for the simultaneous representation of this similarity and the prediction of rubric scores via a classification head, we introduce two complementary loss terms that help our optimizer move off of plateaus in the loss landscape (similar to momentum). Our data augmentation process computes gradients of the model's embedding input layer with respect to class predictions. Finally, in many cases, contextual sentence embeddings are insufficient to capture underlying meaning. Second-order embeddings more accurately reflect properties of text and thus are relevant in assessing essay content and characteristics including argumentation, persuasiveness, and cohesion.

3 Related Work

Sentence-level Encoding with Transformers Bertinetto (2016) shows that BERT contexts can be made more meaningful via fine-tuning with Siamese Networks (S-BERT). The authors train BERT architectures using Triplet loss, exploring pooling layers for the contextual output and show that semantic textual similarity is better reflected in S-BERT embeddings. We specifically focus on utilizing document embeddings for the task of essay scoring in which semantic textual similarity is not necessarily reflective of identical rubric scores. While we follow a similar training procedure to that implemented by Reimers and Gurevych (2019), for S-BERT, empirically we observe better results fine-tuning from BERT context embeddings directly rather than those from S-BERT. For that reason, below we consider an analysis of finetuning only the original BERT architecture.

Data Augmentation with Semantic Substitution Perturbations either in the latent space of document embeddings (via sampling of a convex hull) or at the word-level of the input text can help models train robustly (Jin et al., 2020). Adversarial training in few-shot settings are known to prevent over-fitting Mondal et al. (2018) and boost generalizability. Whereas previous work focuses on arbitrary perturbation of words/phrases for replacement, deletion, and/or modification, here we look to gradient-based methods for explainability of predictions with respect to individual words from the input. We use these as heuristics for identifying candidates for synonym substitution.

Higher-order Embeddings with Recurrent Architectures and Transfer Learning Transformer-derived embeddings for long-sentence documents often lack critical context. In the case of BERT, bidirectional context stored at the hidden-state of the start token is used as heuristic for the embedding of the whole document. However, in lengthy documents, context from the periphery are not included in this vector representation (Oniani et al., 2022) Hence, we explore the use of recurrent layers on the hidden state outputs of transformers to investigate the utility of higher-order embeddings for downstream essay scoring classification tasks.

4 Approach

We implement an ensemble approach, evaluating methods independently in terms of their contribution towards increasing test accuracy and average f1 score for class predictions under various data sparsity constraints. The task is given by estimating the average rater score for student essays provided by the Hewlett Foundation (2012) dataset. We first describe stand alone modules then explain their integration in a few-shot method for essay score prediction.

4.1 Siamese Networks

4.1.1 Multicriterion optimization

Caruana (1997) first demonstrated the success of multi-task learning (MLT) in broad domains, showing the ability of MLT system to generalize better across a range of modalities including text and image classification. Towards the development of embeddings that more accurately reflect semantic similarity between similarly-scoring essays we implement a variant of multi-task learning here. Coupling cross-entropy loss with pair-wise distance metrics between samples helps the optimizer move off of plateaus in the loss landscape and promotes faster and more reliable convergence. Our loss (1, 2) is a combination of three terms: the classification cross-entropy loss (3) and the contrastive (4) or triplet loss (5) between two samples scalarized to reflect optimization priority of classification versus embedding accuracy. Drawn from Mnih et al. (2016), we introduce an entropy-regularization term which is essentially Cross-Entropy loss (3) over post-softmax logits, with constant parameter weighting 0.3. This regularizer helps widen the distribution over predicted class scores to prevent the model from fitting to a single essay score.

$$\mathcal{L} = \mathcal{L}_{CE}(\hat{y^a}, y^a) + \mathcal{L}_{triplet}(x^a, x^p, x^n) \tag{1}$$

$$\mathcal{L} = \mathcal{L}_{contrastive}(x_1, x_2) + \mathcal{L}_{CE}(\hat{y_1}, y_1) + \mathcal{L}_{CE}(\hat{y_2}, y_2) \tag{2}$$

$$\mathcal{L}_{CE}(\hat{y}, y) = -\sum_{i=1}^{n} y_i \log \hat{y}_i \tag{3}$$

The former (5) uses a anchor, positive, and negative sample to simultaneously drive euclidean distance of similar samples closer and dissimilar samples further. The latter (4) minimizes pairwise euclidean distance between samples with with the same label and penalizes euclidean closeness of samples with unequal labels. Our training procedure employees a shared-weight architecture to update weights according to the cumulative gradient of these loss terms (Figure 3).

$$\mathcal{L}_{contrastive}(x_1, x_2) = \mathbb{1}_{x_1=x_2} \cdot \|\mathbf{x_1} - \mathbf{x_2}\|_2^2 + (1 - \mathbb{1}_{x_1=x_2}) \cdot \max(0, \alpha - \|\mathbf{x_1} - \mathbf{x_2}\|)^2 \tag{4}$$

$$\mathcal{L}_{triplet}(x^a, x^p, x^n) = \Big[\|x^a - x^p\|^2 - \|x^a - x^n\|^2 + \alpha\Big]_+ \tag{5}$$

4.1.2 Semantic Substitution

Training with sparse datasets often results in overfitting Luo et al. (2014). In the context of few-shot learning, this means learned weights will be unable to generalize to potentially out-of-domain examples. To mitigate for this, we introduce a form of adversarial training in which robustness is introduced by adding noise to our training samples. Here we present a rationale design of such noise based on gradient explanations of class predictions with respect to individual tokens of the inputted document. We compute the differential sensitivity of class predictions to each token, using these as a heuristic for probabilistically weighting semantic substitution of the given word (Figure 1). While directly observing the Attenion layers of a finetuned BERT model may also yield information about token importance to model predictions, this information is often far removed from the token itself. To preserve class labels under random word substitution, we use part-of-speech tagging of the inputted essay and match lemmatization of candidates for substitution to ensure that inserted words maintain the expected participle form and part of speech.

4.2 Recurrent Architectures

4.2.1 Higher-order embeddings

Higher-order embeddings introduce a number of benefits for text classification, including the capture of contextual meaning, better performance on downstream tasks, and a robustness to noise - especially important in few shot learning and preventing the overfitting on irrelevant features. We produced second-order embeddings to BERT's output by incorporating a bi-directional LSTM layer (Figure 2). LSTMs are especially well suited for the dataset considered here due to the shorter nature of essay

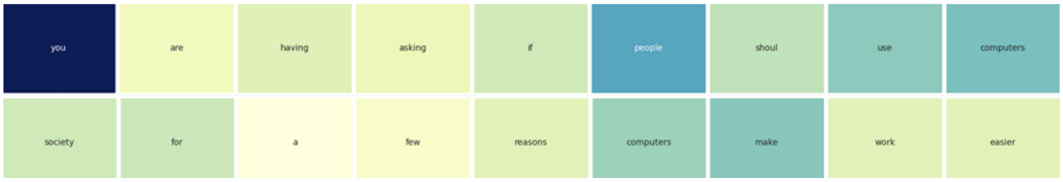

Figure 1: Two excerpts from sample essays showing the weights of each token with respect to model outputs (class predictions). The magnitude of the gradient sheds light on how perturbation of high-weight words may affect training. Namely, we hope to maximize variance between samples and their augments (counterparts) such that the model derives some benefit from robust exposure to adversarial data.The weights also shed light on where the model gives attention. We observe words intrinsic to the prompt such as "computer" or "society" to be weighted highly as expected.

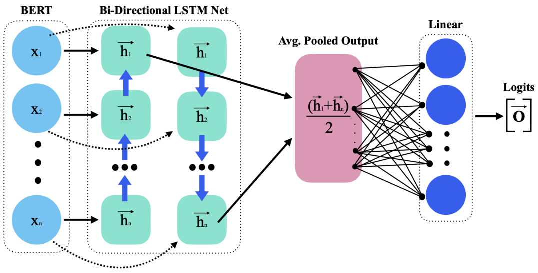

Figure 2: Diagram representing our BERT-LSTM architecture. Note that our siamese network implementation works by feeding one to three distinct embeddings into the model above (depending on selection of contrastive, triplet, or just CE loss), and then classifying each and applying the corresponding loss function to the output

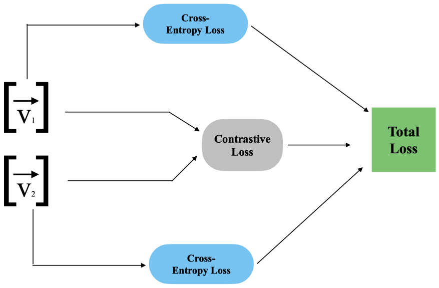

Figure 3: Shared-weight architecture for pairwise loss. Pairwise architectures feed tuples of length 2 (Contrastive) or 3 (Triplet). Loss is computed with respect to a flag that signifies whether a pair of samples originated from the same class or not. Our modified architecture update weights with respect to both pairwise loss and classification (cross entropy) loss.

responses. Additional layers (Dependency Sensitive Convolutional Neural Networks, C-LSTMs, etc.) were intentionally excluded due to literature (Lu, 2022) indicating severely diminishing returns in testing improvements and the exaggerated overhead when training a much more complex model. Thus, the final hidden states can adequately capture the context and dependencies present in the sample. We also attempt to show that higher-order embeddings improve the performance of semantic substitution, as the embeddings are more finely attuned to sequential data across the sample.

4.2.2 Transfer Learning

Few shot learning across varying domains makes model training a difficult task. With limited samples, it can be difficult for a model to identify the general characterizations of what makes a good essay. This goes beyond randomly initialized layers and includes the pretrained BERT architecture, which isn't optimized for essay evaluation. To augment our limited dataset we used the LDC dataset of TOEFL scores, which employed a similar classification standard to our own (see 5.1). We trained DistilBERT on this dataset with a bi-directional LSTM layer. Training occurred in a prompt-agnostic fashion, with the intent to capture the abstract representations of essay mechanics as opposed to any one factoid. To ensure learning on TOEFL scores resulted in less-context dependent weights, we saved only the weights from BERT's fourth and fifth layers and transferred them for training-time to our Siamese network. Ethayarajh (2019) showed that the upper layers of models produce more context-specific representations. For this reason we decided against using the weights from the uppermost LSTM layer as this would likely lead to overfitting on characteristics specific to TOEFL data.

5 Experiments

5.1 Data

5.1.1 Hewlett Foundation: Automated Scoring Essay Competition

We utilized data from the “Hewlett Foundation: Automated Scoring Essay Competition” released by Kaggle (Foundation (2012)). The data is divided into eight essay prompts, each of which is answered in a 150-450 word response by students from grades 7 through 10. Across the 7 essay topics, there are a total of 13,000 training, 4,000 validation, and 4,000 test samples.Most essays are evaluated by two annotators across 1 domain, with the exception of prompt 2 having two evaluation domains. Because different prompts were scored using varying rubrics, for simplicity, we decided to normalize scores according to percentile distribution of labels across essay prompts. In this report we consider a range of scores for a single prompt (prompt "1") to evaluate model performance. The average length of the essay was 350 words. Additional data, for validation, is provided for prompt "7" (A.3). We also modify the training data such that every score class has a set number of examples to avoid overfitting on an imbalanced dataset. When validating and testing, we also randomly select an equal number of samples from each class, ensuring that our evaluation metrics are not biased by an unabalanced validation/test set.

5.1.2 ETS Corpus of Non-Native Written English

To finetune our recurrent neural network, prior to training it on the intentionally limited number of responses from the Hewlett Foundation dataset, we utilize data comprised of essay responses from the TOEFL exam (Shparberg, 2023). We chose to use this dataset for transfer learning for the larger number of samples and its evaluation standards. With 9,899 samples classified in three tiers (low, medium, and high), this format of evaluation closely mirrors our own objective of less-granular, more holistic essay scoring. Furthermore, the short essay samples mirror the Hewlett Foundation dataset we train on after finetuning, with response length and prompts of similar nature.

5.2 Evaluation method

Our model was evaluated using F1, accuracy, and semantic clustering. F1 is a measure of the model’s performance in terms of confusion that takes into account class imbalance, calculated using the harmonic mean of the model’s precision and recall. We compute class-dependent F1 and average over all labels to produce a class-agnostic score. Because our validation and testing sets are rigged to contain equal amounts of data from each class label, we also consider accuracy as a high-fidelity metric below. We further plot confusion on a per-class basis to help identify generalization or overfittting to certain scores. We project 768-dimensional embeddings to 2-D using Principal Component Analysis. Our derived clusters help qualitatively confirm euclidean separation between samples from various classes. Since our loss function incorporates either contrastive loss or triplet loss, we expect to see a hyperplane (or nonlinear equivalent) separating classes in the cartesian space.

5.3 Experimental details

We trained all models for 40 epochs with a 10^{-4} learning rate using both a linear-rate scheduler and L2 regularization implemented via a weight decay of 0.01. These parameters were empirically chosen to minimize the average number of epochs until validation loss converged. Our three-term scalarized objective re-weighted pair-wise distance losses (either contrastive or triplet loss) by 0.1 (to prioritize optimization for classification accuracy). We trained on 1 sample of each class, varying the number of augments/class (see A.2 for more on generating robust samples for training). The remaining data was divided into validation and testing with a 30:70 split. Our model was adapted from DistillBert, Sanh et al. (2020), due to its low-latency, condensed architecture which better facilitated high-frequency training. For experiments discussed below, we freeze the intermediate layers of BERT, allowing for only the last 2 hidden layers to be updated via backpropagation. This was to preserve the semantic structure and textual meaning that is represented in the latent space of BERT’s robust, pretrained model. We add a classification head which produced unnormalized scores for each class. Because overfitting is an inherent issue with few-shot learning, we introduce dropout of 0.1 both after the classification head and within the pretrained BERT model. A softmax over these logits gave label predictions. Entropy regularization, weighted by 0.1, was computed from the softmaxed logits.

5.4 Results

5.4.1 Multicriteron Loss and Siamese Networks

We show that Siamese networks with both contrastive and triplet loss have marginal improvements compared to baseline models in few-shot settings (Table. 1). In fact, exlcuding methods for dealing with sparse data, baseline models seem to outperform pairwise methods in multi-class settings. We interpret these results as an inability to generalize pairwise comparisons from just one example of each class. The resulting overfitting is reflected by non-convergence of validation loss. Semantic textual similarity, however, is marginally reflected in down-projected clusters by the spatial separation of samples from different classes (A.5). There remains occlusion between classes in these down-projected clusters suggesting that under one-shot constraints, neither classification loss nor euclidean comparison of embeddings is optimized. Little distinction is explicit between accuracy/f1 and clustering for triplet and contrastive methods although triplet seems to outperform both baseline and contrastive methods by a wide margin. Under limited data constraints, pairwise methods seem to under-perform baseline and random chance.

Average Accuracy with and without Data Augmentation						
	2	**2, +1**	**3**	**3, +1**	**5**	**5, +1**
Baseline	0.59	0.65	0.42	0.48	0.31	0.24
Contrastive Loss	0.5	0.73	0.45	0.51	0.30	0.30
Triplet Loss	0.79	0.80	0.39	0.64	0.09	0.34

Table 1: Performance without versus with 1 augmented sample (columns labeled "+1"). Note that accuracy declined across all architectures as number of classes increased, and that accuracy for our custom loss functions generally fell short of the baseline without data augmentation. 1 additional (from augmentation) sample per class helped pairwise architectures outperform the baseline models and score far above random chance. Triplet loss yielded the highest accuracy evaluated on all amounts of target classes.

Average F1 with and without Data Augmentation						
	2	**2, +1**	**3**	**3, +1**	**5**	**5, +1**
Baseline	0.26	0.32	0.18	0.23	0.11	0.07
Contrastive Loss	0.24	0.43	0.18	0.25	0.09	0.11
Triplet Loss	0.47	0.48	0.12	0.35	0.01	0.13

Table 2: Performance without data augmentation versus with (columns labeled "+1"). All models showed increases in average F1 with one augmented sample. Triplet and Siamese networks, for the most part, outperformed baseline models when data augmentation was present.

5.4.2 Data Augmentation

Interestingly, we observe differential effects of data augmentation on baseline versus pairwise models. From an information theoretic perspective, there are different utilization efficiencies of augmented data based on the parameters of the objective and model. Specifically, when the augmented samples equal or outnumber the desired number of predicted classes, we empirically observe declining performance (Table 1). This can be understood in the context of Signal to Noise (SNR) ratio. When the signal of ground-truth data is made obscure by a greater proportion of augmented, noisy data we see less accurate predictions from the model. We also note that augmented data seems better utilized by pairwise-learning (either Contrastive or Triplet loss) than by the baseline model (Table 1). We reason that for the method of augmentation selected here, pairwise distances (made robust through noisy augmentation) are more informative than the additional training examples by themselves. In all cases, we see that augmenting the dataset by at least one sample results in significant performance improvements both in terms of F1 (Table 2, Figure 3) and confusion (Figure 4). These results extend to more complex classification tasks, namely 5-class prediction (A.4).

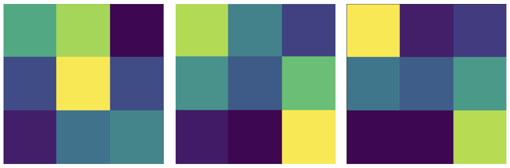

Figure 4: From left to right: baseline, Contrastive, and Triplet confusion matrices for 3-class prediction with 1 "true" and 1 augmented sample per class. Lighter colors along the diagonal and a darker off-diagonal indicates lower confusion. Pairwise architectures trade off high confusion at intermediate classes (column 2) for decreased confusion at the boundaries (column 1, 3) compared with baselines models. Color intensity (increasing in terms of lightness) of (i,j) entry signifies the number of predictions of type j made for class i.

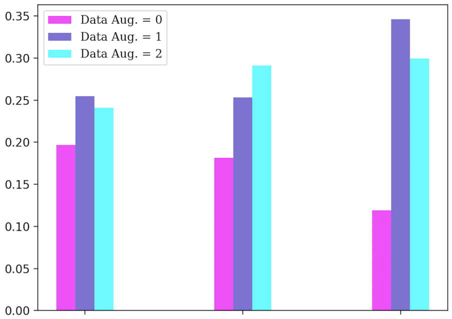

Figure 5: From left to right: Baseline, Contrastive, Triplet architectures. Note that in none of our trials did we see the absence of data augmentation outperform its inclusion. In general, data augmentation benefited F1 with diminishing returns after the addition of 1 sample per class.

5.4.3 Recurrent Architectures

Irrespective of loss function, our results (Table 3) demonstrate a significant improvement in BERT+LSTM+Siamese accuracy over the baseline of random chance, as well as prior results obtained for our BERT and Siamese model. Averaged across classes and loss functions, we see that inclusion of the recurrent architecture leads to an 6% improvement compared to using only BERT and Siamese networks. This result underscores the importance of LSTMs in capturing sequential dependencies that span the essay which BERT may not be adequately identifying. It is also notable that a bigger model performs better in the data-sparse context of training. Rather than overfitting, we see that the mapping of long-term dependencies does help with the STS objective.

RNNs Average Accuracy with and without Transfer Learning						
	2	**2, +Tr**	**3**	**3, +Tr**	**5**	**5, +Tr**
Baseline	0.84	0.83	0.60	0.60	0.35	0.33
Contrastive Loss	0.56	0.5	0.57	0.66	0.37	0.38
Triplet Loss	0.54	0.51	0.42	0.52	0.21	0.30

Table 3: Performance without transfer learning versus with transfer learning (columns labeled "+Tr"). Observe the general improvement in accuracy across all loss functions (compared with Table 1, no data augmentation). We observe, even without transfer learning, higher accuracy compared to standard baseline, Contrastive, and Triplet models trained with 0 data augmentation. We observe increased average accuracy by incorporating transfer learning with recurrent networks, especially for Triplet models.

5.4.4 Transfer Learning

When loading the frozen weights trained with the LDC dataset into our model (and keeping the RNN architecture), we see marked improvement over both the baseline BERT+Siamese and BERT+LSTM+Siamese (Table 3). Compared to BERT and the Siamese Network, we see an 8% improvement in accuracy. With respect to the prior model (which didn't incorporate transfer learning, but held all else constant) there is a 2% improvement. These results demonstrate that transfer learning noticeably augments accuracy in data-sparse contexts. Regardless, it is clear that transfer learning has a positive effect on model weights being closer to their ideal parameter space from the onset of training.

6 Analysis

6.1 Rationalizing multicriterion objectives

Here, we discuss ensemble methods for learning textual representations and assessing similarity in data-sparse contexts. We show pairwise distance metrics better capture semantic textual similarity that is beneficial towards downstream tasks such as classification. A multicriteron loss was empirically observed to perform better than individually optimizing for pairwise distances or classification. We reason that this phenomenon occurs uniquely when the losses are complementary in nature with respect to properties of the text. Thus, they motivate optimizer steps in parallel directions contributing to faster and more robust convergence.

6.2 Interpreting gradient-informed augmented data

Our experiments further reveal the presence of information tradeoffs associated with pairwise learning methods. We reason that our model performs well with less data supplemented by an equivalent amount of augmented samples because the additional samples incorporate noise that adversarially updates weights. To this end, we avoid overfitting and help generalization. Interestingly, we note different efficiencies of information utilization based on the architecture used. Namely, we observe that pairwise losses such as Contrastive and Triplet loss were better suited to dealing with adversarial training data and saw significant performance improvements from augmented samples compared with the baseline model. This, we reason, is because pairwise architectures discriminate between samples, having a sort of implicit "suspicion" that emerges from weighting loss terms based on the euclidean distance of documents in their embedding-spaces. Baseline models on the other hand, must assume the class labels of noisy data as ground truth without any context of how similar/different this augmented data is from the actual training set.

6.3 RNNs and Transfer Learning

Our experiments denote a significant increase in accuracy with the inclusion of RNNs into the few-shot model architecture. Since BERT's transformer architecture processes input in parallel, it struggles to capture the sequential relationships and long-term dependencies that define an essay. The inclusion of an LSTM layer builds on the contextualized embeddings produced by BERT to capture the sequential relationships between words. Additionally, we observe that transfer learning on the TOEFL dataset also has an improvement in model classification. We reason that this dataset providing a number of examples written by English learners helps train weights in earlier layers to represent the more abstract, domain-agnostic aspects of essays. Thus, transfer learning presents a viable means of improving model performance while still being useful in a data-sparse, context-specific domain.

7 Conclusion

7.1 Findings and Implications

In this paper we investigated the various avenues for automated essay scoring in data-sparse contexts using Distill-BERT and Siamese networks. We found that for just a single sample per class (or in the case of triplet loss, two per class), we were able to reach 84% accuracy in binary classification, 66% for three classes, and 38% for five classes. These results are superior to random classification, and indicate our model learned to disambiguate classes to a significant degree. Our results indicate high levels of success for samples at both ends of the scoring range, and middling to poor success for intermediate samples. This could be seen as a potential downside to pairwise learning: since the distance between samples of different classes are maximized, their embeddings are pushed more closely to the other classes on the margins of their own, and the disambiguation of samples from their class neighbors becomes difficult. In sum, our ensemble approach to essay classification serves as a series of findings that establish how to optimize learning in domain-specific, data-sparse environment.

7.2 Future Work

We aim to experiment with rationale choice of training samples, adopting greedy strategies to maximize the variance of those essays chosen for our few-shot training method. Determining the success of ensembling higher-order embeddings (those produced by RNNs) with data augmentation and pairwise loss functions could improve performance under current data scarcity constraints. Additionally, our work used DistilBERT for rapid iteration and testing. Given that the latest Large Language Models are orders of magnitude more powerful, it would be interesting to see if our results still hold for the current state of the field, and to what degree classification can be improved further by using a newer pretrained model.

References

Valmadre J. Henriques J.F. Vedaldi A. Bertinetto, L. 2016. Fully-convolutional siamese networks for object tracking. In *Computer Vision – ECCV 2016*.

Rich Caruana. 1997. *Machine Learning*, 28(1):41–75.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings.

The Hewlett Foundation. 2012. Automated scoring algorithm for student-written essays. In *Kaggle*.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment.

Daming Lu. 2022. daminglu123 at semeval-2022 task 2: Using bert and lstm to do text classification.

Jiahua Luo, Chi-Man Vong, and Pak-Kin Wong. 2014. Sparse bayesian extreme learning machine for multi-classification. *IEEE Transactions on Neural Networks and Learning Systems*, 25(4):836–843.

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning.

Arnab Kumar Mondal, Jose Dolz, and Christian Desrosiers. 2018. Few-shot 3d multi-modal medical image segmentation using generative adversarial learning.

David Oniani, Sonish Sivarajkumar, and Yanshan Wang. 2022. Few-shot learning for clinical natural language processing using siamese neural networks.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

David Roa. 2018. Analysis of short text classification strategies using out of-domain vocabularies.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Anna L. Shparberg. 2023. Linguistic data consortium. *The Charleston Advisor*, 24(3):41–44.

A Appendix

A.1 Sample training and validation convergence

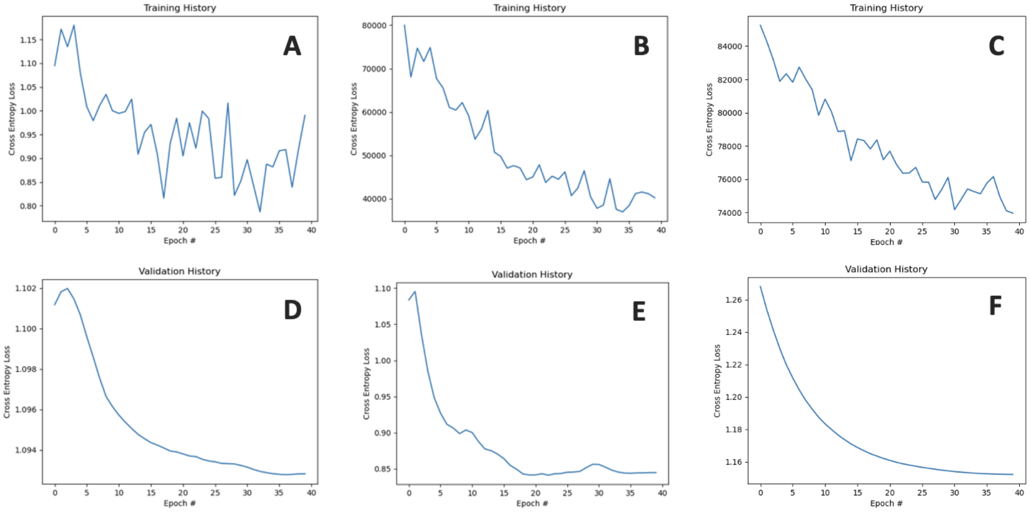

Figure 6: (A-C) shows training loss for baseline, Contrastive, and Triplet architectures respectively. We observe that over a constant number of epochs with identical learning rates, pairwise architectures converge faster and to a lower training loss value. Training is halted after 40 epochs to prevent overfitting. (D-F) shows validation loss for baseline, Contrastive and Triplet architectures respectively. Contrastive networks converged the fastest with all three showing validation loss converge around 20-40 epochs thereby validating are empirical choices for hyperparameters as detailed above.

A.2 Pair-wise batching for training

A benefit of pair-wise learning for data-sparse contexts is the ability to train on several combinations of samples from same limited dataset - doing more with less. For instance, with contrastive loss we train on every possible combination of samples in the dataset. Since the objective is to maximize distance between samples of different classes and minimize distance between samples of the same class, we must evaluate every sample with respect to every other sample. This results in training on significantly more pairings than just the number of samples in the dataset. As an example, when training on five samples per class, for six classes, we see the total number of pairs trained on comes out to $\binom{30}{2} = 435$ samples. For triplet loss, the order of samples in the triplet does matter, so we instead determine all permutations of the dataset into groups of three, and exclude any redundancies where the anchor and positive samples are the same. In either case, pairwise learning leads to significantly more training samples than the limited number of samples necessary for few-shot learning.

A.3 Auxiliary prompt dataset

A.3.1 Discussing data and possible reasons for discrepancies in results

We also evaluate our methods on prompt "7". This prompt asks students to produce a narrative essay about a time they or someone they know exhibited patience. The average essay length on average was about 250 words, nearly 100 words shorter than those analyzed above from "prompt 1". In the context of scoring essays from few samples, this means our embeddings are naturally less dense and there are fewer contextual options for inclusion. More feature-sparse essays means that similar scoring samples would then be indistinguishable from each other in the representational space, thus rendering the utility of pairwise architectures diminished. Furthermore, whereas prompt "1" required an argument, a narrative essay is inherently more subjective and thus conserved elements (such as specific tokens) may not be readily indicative of score. This largely explains the decreased average accuracy and F1 we observe on this prompt. However, our general observations about information utilization efficiency and effect of augmentation remain.

A.3.2 Accuracy

Loss Architecture vs Number of Classes for Prompt "7"						
	2	**2, +1**	**3**	**3, +1**	**5**	**5, +1**
Baseline (CE Loss Only)	0.51	0.50	0.33	0.33	0.19	0.18
Contrastive Loss	0.5	0.73	0.30	0.5	0.16	0.20
Triplet Loss	0.73	0.74	0.38	0.42	0.27	0.23

Table 4: Accuracy (with varying data augmentation) on prompt "7". We observe that with no data augmentation, baseline models and pairwise networks perform comparably, each producing random chance accuracy on class prediction. However upon augmenting the training samples by one (columns with "+1"), we observe dramatic increased in the performance of Contrastive and Triplet networks. We see the best utilization of the added data by Contrastive architectures in the prediction of 2 and 3 classes.

A.3.3 Attention via gradient weights

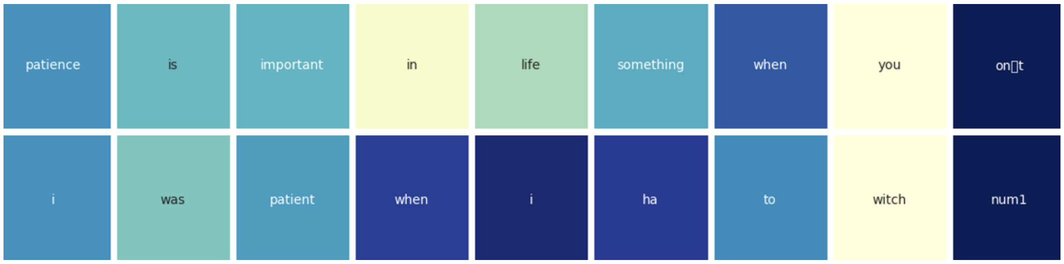

Figure 7: Two excerpts from sample essays taken from prompt "7." As expected there is less interpretability to the heavily-weighted words based on gradients with respect to model outputs. This seems inherently because the essays are narrative in nature and thus there are not set of conserved words that would be indicative of a score - as there may be be for persuasive or argumentative essays. However, we can see some differential weighting is placed on words directly relevant to the prompt such as "patient." Furthermore, we can also observe the importance of indications of narrative elements such as "when" suggesting that the model explicitly observes aspects of story (of course we are anthropomorphizing the model here).

A.4 Multi-class Generalization

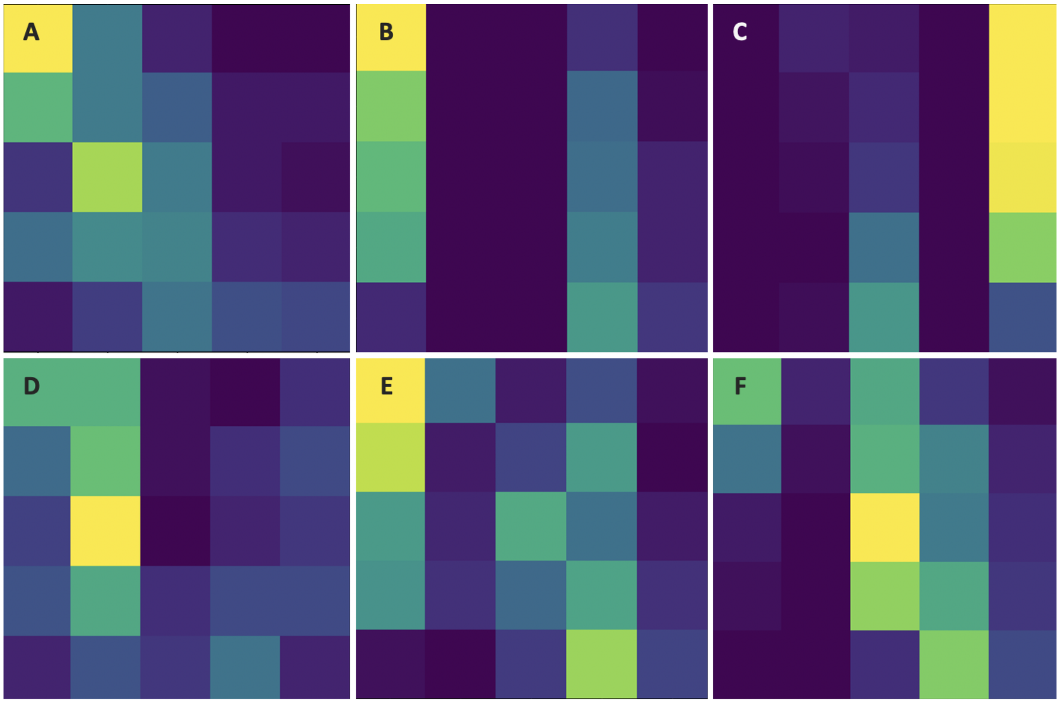

Figure 8: (A-C) Confusion matrices for 5-class classification with baseline, Contrastive, and Triplet models (from left to right), respectively. (D-C) is the same as above but with 1 augmented data sample in the training set. We clearly observe that with data augmentation, false classification severely decreases for pairwise models while there is little effect on the baseline. Overall, highest performing confusion for each model shows that pairwise architectures do a better job of learning to make diverse sets of predictions instead of overfitting to a single class.

A.5 Semantic Textual Similarity expressed through down-projected clustering

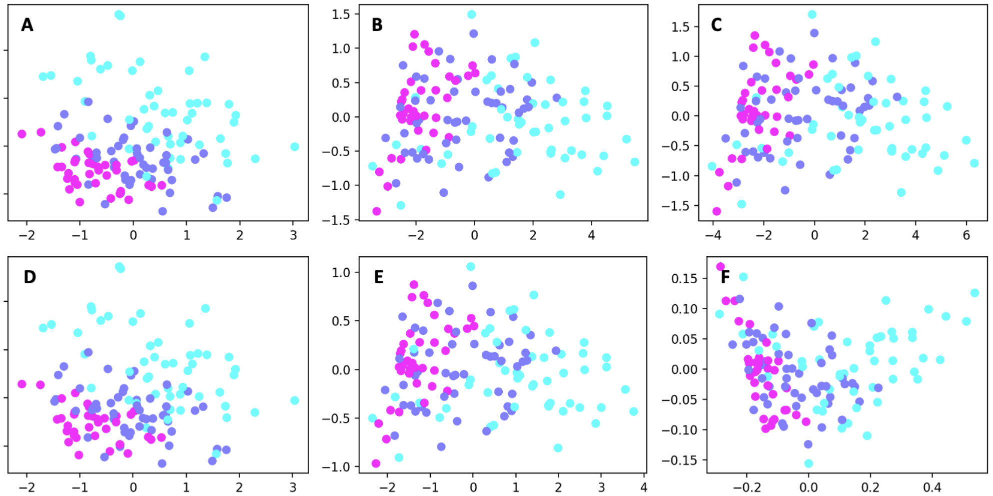

Figure 9: From left to right, Baseline, Contrastive, and Triplet clusters for 3 classes with one-shot learning. (A-C) show down-projected embeddings after training on one sample. While there is inherent ambiguity in the separating hyperplanes, it is clear that contrastive and triplet networks better identify distinction between classes compared with the baseline models. (B-F) Clusters for respective architectures shown with a single augmented sample. We observe that Triplet loss networks better adapt euclidean representations of document embeddings from added data compared with Contrastive and Baseline models. Notice the pink cluster (representing one class of rubric scores) is condensed after the addition of an augmented sample hence showing a greater sensitivity to added data in the representational space. For all architectures there exists an intermediate class that is hard to distinguish from the others (above shown in purple).Unfreezing a greater number of layers in our DistilBERT model may result in greater magnitude changes to representations, in the embedding space, of documents.