

# HANDLING CLASS IMBALANCE IN LOW-RESOURCE DIALOGUE SYSTEMS BY COMBINING FEW-SHOT CLASSIFICATION AND INTERPOLATION

Vishal Sunder and Eric Fosler-Lussier

The Ohio State University

## ABSTRACT

Utterance classification performance in low-resource dialogue systems is constrained by an inevitably high degree of data imbalance in class labels. We present a new end-to-end pairwise learning framework that is designed specifically to tackle this phenomenon by inducing a few-shot classification capability in the utterance representations and augmenting data through an interpolation of utterance representations. Our approach is a general purpose training methodology, agnostic to the neural architecture used for encoding utterances. We show significant improvements in macro-F1 score over standard cross-entropy training for three different neural architectures, demonstrating improvements on a Virtual Patient dialogue dataset as well as a low-resourced emulation of the Switchboard dialogue act classification dataset.

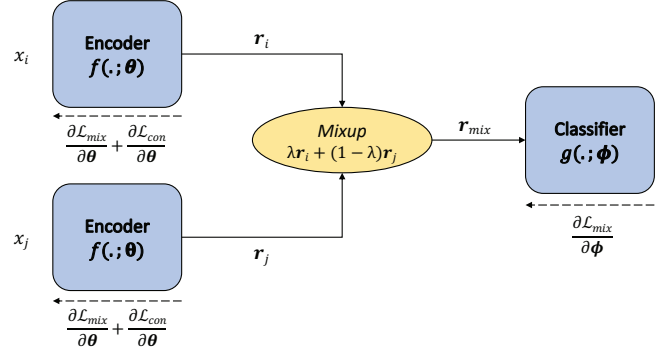
**Index Terms**— Dialogue systems, Low-resource, Class imbalance, Few-shot learning, Data augmentation

## 1. INTRODUCTION

In recent years, there has been a lot of interest in the deployment of question answering (QA) dialogue agents for specialized domains [1, 2, 3]. A simple yet effective approach to this application has been to treat question answering as a utterance classification task. Dialogue datasets annotated for this purpose typically have a large number of classes catering to very fine grained user queries. Due to the limited amount of data that can be realistically collected for specialized domains, the dataset becomes highly class-imbalanced, following a Zipfian-style distribution of utterance classes.

This challenge is very evident in the Virtual Patient dialogue agent [1, 2] used to train medical students at Ohio State to take patient histories. Student-posed questions are classified as one of 348 question types, ranging from the frequent “How are you today?” to the infrequently asked “Are you nervous?” The simple dialog strategy allows medical professors to author QA pairs for particular virtual patients without requiring them to become dialogue system experts.

Various methods have been proposed to handling rare classes in this low-resource dataset, including memory and paraphrasing [4], text-to-phonetic data-augmentation [5] and an ensemble of rule-based and deep learning based models



**Fig. 1:** Model overview. Sentences  $x_i$  and  $x_j$  are fed into the same encoder and interpolated thereafter. Solid lines indicate forward-propagation. Dashed lines indicate gradient flow.

[6]. Recently, self-attention has shown to work particularly well for rare class classification [7].

We propose a novel end-to-end pairwise learning framework (Figure 1, described in Section 2) which augments few-shot classification [8] with an interpolation based data augmentation technique [9] to tackle the said problem of class-imbalance.<sup>1</sup> Few-shot classification helps in identifying classes with as few as one example in the training set with a nearest neighbor search. Further, pairs of training instances are interpolated using *mixup* [10] for augmenting the data. A classifier trained with augmented data helps to maintain performance of the model on the frequent classes, unlike other pairwise learning frameworks [4] that require additional model ensembling to maintain overall performance.

The effectiveness of this method is demonstrated both in the virtual patient dialogue domain as well as a low-resource version of the Switchboard dialogue act classification task that has a similar class imbalance. Our training approach considerably improves performance of three neural encoding systems over standard cross-entropy training on both datasets.

## 2. PAIRWISE LEARNING FRAMEWORK

Our pairwise learning framework seeks to create representations of sentence pairs,  $r_i$  and  $r_j$ , that are closer together when the sentences are of the same class, and further apart

<sup>1</sup>Code and data available at <https://github.com/OSU-slatelab/vp-pairwise>

when of different classes, while still retaining good classification ability. From an original training set of (sentence, class) pairs  $\{(x_i, y_i)\}_{i=1}^N$ , we sample a paired training set  $\{(x_{i,1}, x_{i,2}, y_{i,1}, y_{i,2})\}_{i=1}^{N_s}$  using a strategy explained in Section 3.1. As illustrated in Figure 1, our model uses one of three previously-developed encoders  $f(\cdot; \theta)$  to transform sentences  $x_i$  into representations  $\mathbf{r}_i$ ; pairs of representations are then augmented through a *mixup* strategy that tries to disentangle classes during classification.

## 2.1. Encoder

The encoder is a deep neural network,  $f(\cdot; \theta)$ , which takes the input  $x_i$  and returns its vector representation  $\mathbf{r}_i = f(x_i; \theta)$ . We apply this transformation to both utterances in an instance of the paired training data to obtain  $\mathbf{r}_i$  and  $\mathbf{r}_j$ . To model  $f(\cdot; \theta)$  we use three different neural architectures: Text-CNN, Self-attentive RNN and BERT. The first two encoders can run in realtime for concurrent users; the last demonstrates a relatively recent state-of-the-art encoding technique.

**Text-CNN encoder [6, 11]:** The Text-CNN encoder utilizes convolutions on 300 dimensional GloVe embeddings [12] of  $x_i$  with 300 filters of size 1, 2 and 3. Each filter’s output is max-pooled over time; the concatenated poolings are fed into a fully connected layer with *tanh* activation to obtain the final encoded representation  $\mathbf{r}_i \in \mathbb{R}^{600}$ . This operation is performed for both  $x_i$  and  $x_j$  in a paired training instance using the same CNN to obtain  $\mathbf{r}_i$  and  $\mathbf{r}_j$ .

**Self-attentive RNN encoder [7, 13]:** The Self-attentive RNN is a bidirectional Gated Recurrent Unit (GRU) [14] with self attention. The embedding representations of  $x_i$  and  $x_j$  are passed separately through a self-attentive bidirectional-GRU to obtain the  $n$ -head representation matrices,  $\mathbf{M}_i, \mathbf{M}_j \in \mathbb{R}^{600 \times n}$ . Each column  $\mathbf{m}_{ik}$  (and  $\mathbf{m}_{jk}$ ) of  $\mathbf{M}_i$  (and  $\mathbf{M}_j$ ) is an attention head representation.

To give more importance to the attention heads with similar representations in a paired instance, we perform a novel *2<sup>nd</sup> order attention* on the attention head representations to obtain final representations  $\mathbf{r}_i$  and  $\mathbf{r}_j$ . Mathematically,

$$\begin{aligned} score_k &= \tanh(\mathbf{m}_{ik}^T \mathbf{W} \mathbf{m}_{jk}) \\ \mathbf{s}_k &= \frac{\exp(score_k)}{\sum_{i=1}^n \exp(score_i)} \\ \mathbf{r}_i &= \mathbf{M}_i \mathbf{s} \\ \mathbf{r}_j &= \mathbf{M}_j \mathbf{s} \end{aligned} \quad \left| \quad \begin{array}{c} \mathbf{M}_i \begin{bmatrix} \mathbf{m}_{i1} & \mathbf{m}_{i2} & \mathbf{m}_{i3} & \dots & \mathbf{m}_{in} \end{bmatrix} \mathbf{r}_i \\ \mathbf{M}_j \begin{bmatrix} \mathbf{m}_{j1} & \mathbf{m}_{j2} & \mathbf{m}_{j3} & \dots & \mathbf{m}_{jn} \end{bmatrix} \mathbf{r}_j \end{array} \right.$$

**Fig. 2:** *2<sup>nd</sup> order attention* highlights attention-head similarities in matched pairs.

with  $\mathbf{W} \in \mathbb{R}^{600 \times 600}$  a matrix of parameters learned during the training process;  $\mathbf{s} \in \mathbb{R}^n$  is a probability distribution that gives the weight of each attention-head representation.

**BERT encoder [15]:** We fine-tune the pretrained BERT model *bert-based-uncased* using default WordPiece embeddings.<sup>2</sup> The final layer output of this encoder is a matrix  $\mathbf{B}_i \in \mathbb{R}^{768 \times l}$  where  $l$  is the length of the sequence. The encoded representations  $\mathbf{r}_i$  and  $\mathbf{r}_j$  are found by mean pooling over the columns of  $\mathbf{B}_i$  and  $\mathbf{B}_j$  respectively.

## 2.2. Classifier

The classifier ( $g(\cdot; \phi)$ ) is a 3-layer fully-connected MLP with *tanh* activations. We use a *mixup* strategy [10] to combine the representations,  $\mathbf{r}_i$  and  $\mathbf{r}_j$  in a paired training instance and feed it to the classifier. The idea is to create new training instances of data by interpolating between two instances of training examples. Formally,

$$\begin{aligned} \mathbf{r}_{mix} &= \lambda \mathbf{r}_i + (1 - \lambda) \mathbf{r}_j \\ \mathbf{y}_{mix} &= \lambda \mathbf{y}_i + (1 - \lambda) \mathbf{y}_j \end{aligned}$$

Here,  $\mathbf{y}_i$  and  $\mathbf{y}_j$  are one-hot representations of  $y_i$  and  $y_j$  respectively.  $\lambda$  is sampled from  $\lambda \sim \text{Beta}(\alpha, \alpha)$ . Higher values of the hyperparameter  $\alpha$  result in  $\lambda$  being close to 0.5. For rare classes, we tune  $\alpha \in \{10.0, 20.0\}$  to generate more novel data instances around the rare class. To preserve the frequent class distribution, we tune  $\alpha \in \{0.05, 0.1\}$  for frequent classes generating  $\lambda$  close to 1.0. We found that this strategy performed better than using a fixed  $\alpha$  value.

The classifier’s output representation  $\mathbf{c}_{mix}^{class} = g(\mathbf{r}_{mix}; \phi) \in \mathbb{R}^C$  is the pre-softmax output of the classifier.

## 2.3. Pairwise Loss Function

### 2.3.1. Contrastive Loss

To learn rich utterance representations for few shot classification, we use the contrastive loss [16] on encoder outputs which helps to separate classes in the semantic space more effectively. For a paired training instance  $(x_i, x_j, y_i, y_j)$  with  $D$  being the euclidean distance between the normalized encoded representations of  $x_i$  ( $\frac{\mathbf{r}_i}{\|\mathbf{r}_i\|_2}$ ) and  $x_j$  ( $\frac{\mathbf{r}_j}{\|\mathbf{r}_j\|_2}$ ):

$$\begin{aligned} \mathcal{L}_{con}(\theta) &= \frac{1}{2} y_{pair} \{ \max(0, D - m_{pos}) \}^2 \\ &\quad + \frac{1}{2} (1 - y_{pair}) \{ \max(0, m_{neg} - D) \}^2 \end{aligned}$$

where  $y_{pair}$  is set to 1 if  $y_i = y_j$  and 0 otherwise.  $m_{pos}$  and  $m_{neg}$  are the positive and negative margins respectively. We use  $m_{pos} = 0.8$  and  $m_{neg} = 1.2$ .

### 2.3.2. Mixup Loss

We train the classifier to predict the mixed class representation  $\mathbf{y}_{mix}$  (defined in section 2.2) by using the KL-divergence

<sup>2</sup>[https://huggingface.co/transformers/model\\_doc/bert.html](https://huggingface.co/transformers/model_doc/bert.html)

between  $\mathbf{y}_{mix}$  and the classifier predicted distribution as the second component of the loss function i.e.,

$$\mathcal{L}_{mix}(\theta, \phi) = \text{KL}(\mathbf{y}_{mix} \| \text{softmax}(\mathbf{c}_{mix}^{class}))$$

We found that KL-divergence works better than cross-entropy on mixed labels for the datasets we used. We combine the two losses using a hyperparameter  $\beta \in [0, 1]$ :

$$\mathcal{L}_{pair}(\theta, \phi) = \beta \mathcal{L}_{con}(\theta) + (1 - \beta) \mathcal{L}_{mix}(\theta, \phi)$$

## 2.4. Testing

At test time, an utterance  $x_{test}$  is encoded to obtain  $\mathbf{r}_{test} = f(x_{test}; \theta)$ .<sup>3</sup> For each class, we perform a 1-nearest-neighbor search<sup>4</sup> on the training set using  $\mathbf{r}_{test}$  and set the corresponding elements of the class score  $\mathbf{c}_{test}^{nn} \in \mathbb{R}^C$  to be the inverse distance to  $\mathbf{r}_{test}$ . We also compute the classifier class scores on the unmixed test utterance,  $\mathbf{c}_{test}^{class} = g(\mathbf{r}_{test}; \phi)$ .

Each of these confidence scores have distinct advantages for rare class classification.  $\mathbf{c}_{test}^{nn}$  incorporates a *few-shot* classification capability on the rarest classes [8] and  $\mathbf{c}_{test}^{class}$  incorporates a capability of using the classifier trained using the augmented data [10]. We combine the two by first normalizing them and then interpolating:

$$\mathbf{c}_{test}^{final} = \gamma \bar{\mathbf{c}}_{test}^{nn} + (1 - \gamma) \bar{\mathbf{c}}_{test}^{class}$$

$\gamma \in [0, 1]$  is tuned on the validation data. The maximal element of  $\mathbf{c}_{test}^{final}$  is used to make the prediction.

## 3. EXPERIMENTAL SETUP

### 3.1. Sampling of pairs

We randomly sample 50,000 positive pairs ( $y_i = y_j$ ) and 100,000 negative pairs ( $y_i \neq y_j$ ) to create sets  $\mathcal{S}^+$  and  $\mathcal{S}^-$  respectively. Once per epoch, for every pair  $(x_i^+, x_j^+) \in \mathcal{S}^+$  we compute  $d_{ij}^+ = \|f(x_i^+; \theta) - f(x_j^+; \theta)\|_2$  and select the top 25,000 pairs with the *highest* corresponding  $d_{ij}^+$ . Similarly, we compute  $d_{ij}^-$  for pairs  $(x_i^-, x_j^-) \in \mathcal{S}^-$  and select the top 50,000 pairs with the *lowest* corresponding  $d_{ij}^-$ . This gives us a paired training set size of 75,000.

### 3.2. Datasets

**Virtual Patient Corpus [6, 7]:** The Virtual Patient (VP) dataset is a collection of dialogues between medical students and a virtual patient experiencing back pain. Each student query was mapped to a question from set of 348 canonical

<sup>3</sup>For the self-attentive RNN, we just average the attention head representations during test time as we don't have a paired counterpart to perform the  $2^{nd}$  order attention.

<sup>4</sup>For efficient search, we utilize the FAISS toolkit [17] (<https://github.com/facebookresearch/faiss>)

Quintile #	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>	5 <sup>th</sup>
Virtual Patient	1.8%	3.5%	6.9%	17.7%	70.1%
Switchboard	0.3%	0.9%	3.0%	6.6%	89.2%

**Table 1:** % of data in the training set per class quintile

questions whose answers are already known. The data consists of a total of 9,626 question utterances over 259 dialogues (6,827 training, 2,799 test). The data are highly imbalanced across classes (Table 1), with the top fifth of classes comprising 70% of the examples in the training set.

**Switchboard Dialog Act corpus [18]:** The Switchboard Dialog Act (SwDA) dataset is a collection of telephone conversations which are transcribed and annotated for 43 dialog act classes; it exhibits a class imbalance similar to the Virtual Patient data (see Table 1). SwDA has a training set size of 193k, validation set size of 23k and test set size of 5k. We experiment with several subsets to simulate data growth; we also create a VP-style low-resource setting which has 5 subsets of 6850 instances each from the entire SwDA training set by random sampling. All the models are trained on these five data subsets and the mean  $\pm$  standard deviation is reported.

### 3.3. Training details

We train using 90/10 train/dev splits, using the Adam optimizer [19] with a learning rate  $4e - 5$  for BERT and  $1e - 3$  for the rest. BERT is trained for 6 epochs; other models are trained for 15 epochs. The model-epoch with the best dev set Macro-F1 performance is retained for testing.

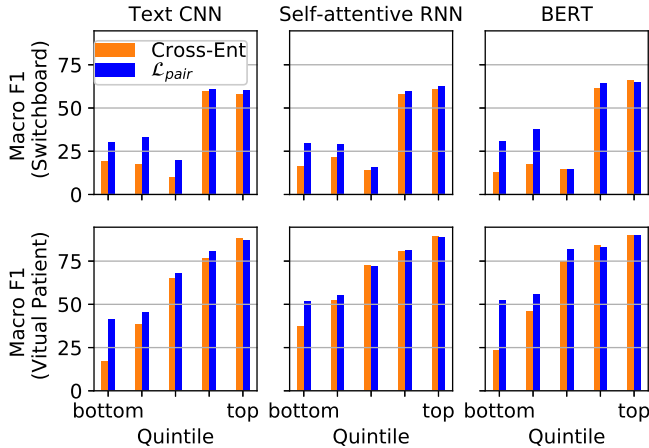
## 4. RESULTS AND ANALYSIS

We compare the performance of the proposed pairwise training against the conventional cross-entropy training performance (Table 2).<sup>5</sup> For each of the three neural architectures used for modeling the encoder, we train a corresponding

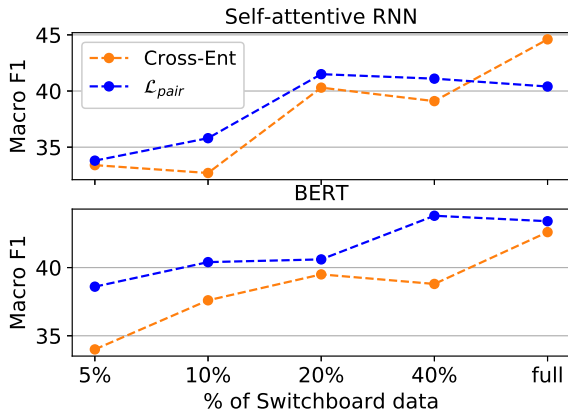
<sup>5</sup>In the virtual patient test set, a few classes are absent. Previous work [7] does not take this into account when computing the macro-F1 score and hence report a slightly underestimated value. We correct for this.

Encoder Model	Training	SwDA		Virtual Patient	
		Acc(%)	F1(%)	Acc(%)	F1(%)
Text-CNN	Cross-Entropy[6]	59.3 $\pm$ 0.7	26.4 $\pm$ 1.2	75.7	51.4
	$\mathcal{L}_{pair}$	<b>60.8</b> $\pm$ 0.8	<b>32.4</b> $\pm$ 0.9	<b>75.8</b>	<b>57.1</b>
Self-attentive RNN	Cross-Entropy[7]	61.0 $\pm$ 1.2	29.4 $\pm$ 1.2	<b>79.2</b>	59.8
	$\mathcal{L}_{pair}$	<b>61.9</b> $\pm$ 0.9	<b>32.7</b> $\pm$ 1.0	<b>79.2</b>	<b>63.9</b>
BERT	Cross-Entropy	<b>65.2</b> $\pm$ 0.6	30.4 $\pm$ 1.7	79.4	57.4
	$\mathcal{L}_{pair}$	64.6 $\pm$ 0.5	<b>35.5</b> $\pm$ 1.7	<b>81.0</b>	<b>66.8</b>

**Table 2:** Comparing performance of  $\mathcal{L}_{pair}$  training with cross-entropy training using 3 different models on the two datasets. For SwDA, we perform experiments with 5 smaller subsets of the training data. We report *mean  $\pm$  std* of the performance on these subsets for SwDA. Bold represents the best performance.



**Fig. 3:** Quintile-wise performance of different models on the two datasets. Pairwise training (blue) usually helps rare classes over cross-entropy training (orange) and provides similar performance for frequent classes across decoders.



**Fig. 4:** Effect on F1 of varying the amount of training data. We test using the self-attentive RNN and BERT as they were the best performing models for the Virtual Patient data.

model with the same architecture using a cross-entropy loss. Macro-F1 scores improve for all models on all datasets.

We also plot macro-F1 performance for each class quintile separately (Figure 3). It is clear that pairwise training does extremely well compared to cross-entropy training in the lower quintiles, while retaining the performance in the top quintile. In particular, the BERT based pairwise training yields an improvement of almost 2x on the bottom quintile. This is especially useful given that a major drawback of BERT has been its poor performance on rare classes ([7], [20]).

We also compare the performance of the proposed pairwise training with cross-entropy training as we increase the training data for SwDA (Figure 4). Up through 40% of the SwDA data (train set size of 77k), pairwise training gives better performance compared to cross-entropy training. With the full data (train set size of 193k), pairwise training does not

perform quite as well for self-attentive RNN with minor improvements using BERT. That pairwise training does better for a data size as high as 77k utterances is encouraging, as task specific dialog datasets for real-world deployment will usually start at a much smaller size.

Finally, we perform ablation studies to see the effectiveness of each component of  $\mathcal{L}_{pair}$ . We trained the self-attentive RNN by removing one component at a time (Table 3). Both  $\mathcal{L}_{con}$  and  $\mathcal{L}_{mix}$  contribute to the final performance;  $\mathcal{L}_{con}$  helps more on the VP dataset and not as much on SwDA. This may be because virtual patient has a lot more classes spanning across a similar training set size. Hence, many classes have as few as one instance in the training set. Therefore, using  $\mathcal{L}_{con}$  with a 1-nearest-neighbor classification helps more in these few-shot cases. As our 2<sup>nd</sup> order attention method is new, we note through ablation that it helps with the VP data while making no difference on SwDA; we attribute this to two possible hypotheses: the classes in Virtual Patient are more semantically fine-grained in contrast to speech-act classes in SwDA, so attending to specific attention heads may be more crucial in Virtual Patient, or since SwDA utterances are relatively shorter, the information contained in different attention heads may be correlated. This will be investigated more fully in subsequent work.

## 5. CONCLUSION

We proposed an end-to-end pairwise learning framework which mitigate class imbalance issues in low-resource dialogue systems, by generalizing well to the rare classes while maintaining performance on the frequent ones. By using a combination of a contrast based and an interpolation based loss function, we show considerable improvements over cross-entropy training. Effectively incorporating dialogue context in the proposed pairwise training is a subject of future work.

## 6. ACKNOWLEDGEMENTS

This material builds upon work supported by the National Science Foundation under Grant No. 1618336. We gratefully acknowledge the OSU Virtual Patient team, especially Adam Stiff, for their assistance.

	SwDA		Virtual Patient	
	Acc(%)	F1(%)	Acc(%)	F1(%)
Full	<b>61.9 ± 0.9</b>	<b>32.7 ± 1.0</b>	<b>79.2</b>	<b>63.9</b>
$-\mathcal{L}_{con}(\beta = 0; \gamma = 0)$	60.5 ± 1.0	32.1 ± 1.5	79.1	62.6
$-\mathcal{L}_{mix}(\beta = 1; \gamma = 1)$	38.6 ± 0.7	22.9 ± 0.6	76.8	61.1
$-\mathcal{2}^{nd} \text{ order attention}$	<b>61.9 ± 0.8</b>	<b>32.7 ± 1.0</b>	78.6	61.3

**Table 3:** Ablation studies on self-attentive RNN on the two datasets, removing one either part of  $\mathcal{L}_{pair}$ . Results on SwDA are the *mean ± std* on 5 small subsets of the training data.

## 7. REFERENCES

- [1] Douglas R Danforth, Mike Procter, Richard Chen, Mary Johnson, and Robert Heller, “Development of virtual patient simulations for medical education,” *Journal For Virtual Worlds Research*, vol. 2, no. 2, 2009.
- [2] DR Danforth, A Price, Kellen Maicher, D Post, Beth Liston, Daniel Clinchot, Cynthia Ledford, D Way, and Holly Cronau, “Can virtual standardized patients be used to assess communication skills in medical students,” in *Proceedings of the 17th Annual IAMSE Meeting, St. Andrews, Scotland*, 2013.
- [3] Prerna Khurana, Puneet Agarwal, Gautam Shroff, Lovekesh Vig, and Ashwin Srinivasan, “Hybrid bilstm-siamese network for faq assistance,” in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 537–545.
- [4] Lifeng Jin, David King, Amad Hussein, Michael White, and Douglas Danforth, “Using paraphrasing and memory-augmented models to combat data sparsity in question interpretation with a virtual patient dialogue system,” in *Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications*, 2018, pp. 13–23.
- [5] Adam Stiff, Prashant Serai, and Eric Fosler-Lussier, “Improving human-computer interaction in low-resource settings with text-to-phonetic data augmentation,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 7320–7324.
- [6] Lifeng Jin, Michael White, Evan Jaffe, Laura Zimmerman, and Douglas Danforth, “Combining cnns and pattern matching for question interpretation in a virtual patient dialogue system,” in *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, 2017, pp. 11–21.
- [7] Adam Stiff, Qi Song, and Eric Fosler-Lussier, “How self-attention improves rare class performance in a question-answering dialogue agent,” in *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 2020, pp. 196–202.
- [8] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov, “Siamese neural networks for one-shot image recognition,” in *ICML deep learning workshop*. Lille, 2015, vol. 2.
- [9] Jiaao Chen, Zichao Yang, and Diyi Yang, “Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification,” *arXiv preprint arXiv:2004.12239*, 2020.
- [10] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz, “mixup: Beyond empirical risk minimization,” *arXiv preprint arXiv:1710.09412*, 2017.
- [11] Yoon Kim, “Convolutional neural networks for sentence classification,” *arXiv preprint arXiv:1408.5882*, 2014.
- [12] Jeffrey Pennington, Richard Socher, and Christopher D Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.
- [13] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio, “A structured self-attentive sentence embedding,” *arXiv preprint arXiv:1703.03130*, 2017.
- [14] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.
- [15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.
- [16] Raia Hadsell, Sumit Chopra, and Yann LeCun, “Dimensionality reduction by learning an invariant mapping,” in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*. IEEE, 2006, vol. 2, pp. 1735–1742.
- [17] Jeff Johnson, Matthijs Douze, and Hervé Jégou, “Billion-scale similarity search with gpus,” *arXiv preprint arXiv:1702.08734*, 2017.
- [18] Daniel Jurafsky and Elizabeth Shriberg, “Switchboard swbd-damsl shallow-discourse-function annotation coders manual, draft 13 daniel jurafsky\*, elizabeth shriberg+, and debra biasca\*\* university of colorado at boulder &+ sri international,” 1997.
- [19] Diederik P Kingma and Jimmy Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [20] Abhijit Mahabal, Jason Baldridge, Burcu Karagol Ayan, Vincent Perot, and Dan Roth, “Text classification with few examples using controlled generalization,” *arXiv preprint arXiv:2005.08469*, 2020.