Improving Fine-tuning on Low-resource Corpora with Information Bottleneck

Abstract

Large-scale pre-trained language models act as general-purpose feature extractors, but not all the features are relevant for a given target task. This can cause problems in low-resource scenarios, where fine-tuning such large-scale models often over-fits on the small training set. We propose to use the information bottleneck principle to improve generalization in this scenario. We apply the variational information bottleneck method to remove task-irrelevant and redundant features from sentence embeddings during the fine-tuning of BERT. Evaluation on seven low-resource datasets for different tasks shows that our method significantly improves transfer learning in low-resource scenarios and obtains better generalization on 11 out of 13 out-of-domain textual entailment datasets.

1 Introduction

Transfer learning has emerged as the de facto standard technique in natural language processing (NLP), where large-scale language models are pre-trained on an immense amount of text to learn a general-purpose representation, which is then transferred to the target domain. This method has exhibited state-of-the-art results on various NLP benchmarks (Devlin et al., 2019; Radford et al., 2018; Liu et al., 2019; Radford et al., 2019). However, the universal nature of large-scale pre-trained representations poses a problem in low-resource scenarios, which is an important topic in NLP (Cherry et al., 2019). Much of the information in such sentence embeddings is irrelevant to the target task, and it can be difficult to distinguish relevant from irrelevant information when fine-tuning the language-models with a large number of parameters on a small amount of target task data, resulting in over-fitting. For many real-world applications, it can be difficult and expensive to solve this problem by collecting sufficient annotated data for these large neural models to excel.

In this paper, we propose to use the Information Bottleneck (IB) principle (Tishby et al., 1999) to improve transfer learning in resource-limited scenarios. We apply a Variational Information Bottleneck (VIB) (Alemi et al., 2017) approach, as illustrated in Figure 1, to remove task-irrelevant and redundant information from the sentence embeddings. Removing task-irrelevant information helps to mitigate over-fitting when fine-tuning on the small target task training set. Additionally, removing redundant information helps to avoid learning superficial correlations that do not generalize well to out-of-domain datasets.

We evaluate the effectiveness of our method on fine-tuning BERT (Devlin et al., 2019) on seven different datasets for text classification, natural language inference (NLI), similarity, and paraphrase tasks. Our method shows greater robustness to over-fitting than conventional fine-tuning, improving the accuracy on low-resource datasets. Moreover, our method obtains better generalization to out-of-domain NLI datasets.

Our approach is highly effective and simple to implement, involving a small additional MLP classifier on top of the sentence embeddings. It is model agnostic and end-to-end trainable. In summary, we make the following contributions: 1) Proposing the IB principle, and specifically a VIB model, as an effective approach to address over-fitting when fine-tuning large-scale pre-trained models on low-resource scenarios. 2) An empirical evaluation of the method on seven low-resource benchmarks, obtaining substantial generalization gains in all settings, including over 2 point gains on the STS-B and MRPC tasks. 3) Our approach is highly effective and simple to implement, involving a small additional MLP classifier on top of the sentence embeddings. It is model agnostic and end-to-end trainable.
datasets. 3) Providing analysis on how VIB reduces over-fitting. 4) Demonstrating significantly improved generalization by our trained NLI models to 11 out of 13 out-of-domain NLI datasets. To facilitate future work, we will release our code.

2 Fine-tuning on Low-resource Settings

Problem Formulation We consider a general multi-class classification problem with a dataset \( D = \{ (x_i, y_i) \}_{i=1}^{N} \) consisting of inputs \( x_i \in \mathcal{X} \), and labels \( y_i \in \mathcal{Y} \). We assume we are also given a pre-trained encoder \( f_{\varphi}(\cdot) \) parameterized by \( \varphi \) that computes sentence embeddings for the input \( x_i \). Our goal is to improve the fine-tuning of powerful pre-trained models \( f_{\varphi}(\cdot) \) on a low-resource target dataset \( \mathcal{D} \).

2.1 Mitigating Over-fitting

In general, pre-trained language models are powerful general-purpose feature extractors with a large number of parameters. We hypothesize that only a small subset of this encoded information is needed to perform the target task, meaning that there are many irrelevant and redundant features in the embedding. When the dataset for the target task is small, it is hard for fine-tuning to distinguish the required information from the irrelevant one, resulting in over-fitting on the target dataset.

The key idea of our approach is to specifically optimize for the removal of irrelevant and redundant information from the input embeddings. As depicted in Figure 1, we introduce a variational information bottleneck module on top of the pre-trained input embeddings. During fine-tuning, this VIB module tries to compress the sentence embeddings \( x \) into an embedding \( z \) that keeps only the information necessary to predict \( y \).

2.2 Information Bottleneck

To specifically optimize for the removal of irrelevant and redundant information from the input embeddings, we adopt the Information Bottleneck principle. The objective of IB is to find a maximally compressed representation \( Z \) of the input representation \( X \) (compression loss) that maximally preserves information about the output \( Y \) (prediction loss), by minimizing:

\[
L_{IB} = \beta I(X;Z) - I(Z;Y),
\]

where \( I(\cdot;\cdot) \) is the mutual information, and \( \beta \geq 0 \) controls the balance between compression and prediction.

Variational Information Bottleneck Alemi et al. (2017) derive an efficient variational estimate of (1):

\[
L_{\text{VIB}} = \beta \mathbb{E}_{z} [\mathbb{E}_{x} [\text{KL}(p_{\theta}(z|x), q_{\phi}(y|z))]] + \mathbb{E}_{z \sim p_{\theta}(z|x)} [-\log q_{\phi}(y|z)],
\]

where \( q_{\phi}(y|z) \) is an estimate of \( p(y|z) \), \( r(z) \) is an estimate of the prior \( p(z) \), and \( p_{\theta}(z|x) \) is an estimate of the posterior probability of \( z \). During training, the compressed sentence representation \( z \) is sampled from the distribution \( p_{\theta}(z|x) \), making dimensions with high variance less informative for predicting the output class. Thus, the VIB module can learn to block the trained output classifier \( q_{\theta}(y|z) \) from using specific dimensions of \( z \). At test time, the expected value of \( z \) is used for predicting labels with \( q_{\phi}(y|z) \). We assume parametric Gaussian distributions for \( r(z) \) and \( p_{\theta}(z|x) \) to allow an analytic computation for their Kullback-Leibler divergence, namely \( r(z) = \mathcal{N}(z|\mu_0, \Sigma_0) \) and \( p_{\theta}(z|x) = \mathcal{N}(z|\mu(x), \Sigma(x)) \), where \( \mu \) and \( \mu_0 \) are \( K \)-dimensional mean vectors, the \( \Sigma \) and \( \Sigma_0 \) are diagonal covariance matrices, and \( K \) is the dimensionality of \( z \). For \( p_{\theta}(z|x) \), \( \mu(x) \) and \( \Sigma(x) \) are estimated using an MLP from the input sentence representations \( f_{\varphi}(x) \), and \( q_{\phi}(y|z) \) is another MLP.

3 Experiments

Datasets We evaluate the performance on seven benchmarks on different tasks, including text classification, natural language inference, similarity, and paraphrase detection. For NLI, we experiment with the SNLI (Bowman et al., 2015) and MNLI (Williams et al., 2018) benchmarks. For text classification, we evaluate on two sentiment analysis datasets, namely IMDB (Maas et al., 2011) and Yelp2013 (YELP) (Zhang et al., 2015). We additionally evaluate on three low-resource datasets in the GLUE benchmark (Wang et al., 2019).\(^1\) These include paraphrase detection using MRPC (Dolan and Brockett, 2005), semantic textual similarity using STS-B (Cer et al., 2017), and textual entailment using RTE (Dagan et al., 2006). We evaluate on the standard validation and test splits. Since the test sets are not available for MNLI, we tune on the matched dev set and evaluate on the mismatched dev set (MNLI-M) or vice versa (see Appendix A for datasets statistics and more details on the experimental setups).

Baselines We use the BERT base uncased (Devlin et al., 2019) implementation of Wolf et al. (2019) as

\(^1\) We did not evaluate on WNLI and CoLA due to the existing irregularities in these datasets and the reported instability during the fine-tuning. https://gluebenchmark.com/faq
Table 1: Average results over 3 runs on low-resource data in GLUE. We report Pearson/Spearman correlation for STS-B, F1/accuracy for MRPC, and accuracy for RTE.

<table>
<thead>
<tr>
<th>Model</th>
<th>MRPC</th>
<th>STS-B</th>
<th>RTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE</td>
<td>87.8/83.20</td>
<td>84.93/83.53</td>
<td>67.93</td>
</tr>
<tr>
<td>Dropout</td>
<td>87.33/81.9</td>
<td>84.33/82.73</td>
<td>65.80</td>
</tr>
<tr>
<td>VIB</td>
<td>89.23/85.23</td>
<td>87.63/86.5</td>
<td>70.53</td>
</tr>
</tbody>
</table>

Table 1 shows results on the low-resource datasets in GLUE. Our VIB model substantially outperforms the baselines on all the datasets, improving the results on MRPC, STS-B, and RTE by 1.43/2.03, 2.70/2.97, and 2.60 absolute points respectively, demonstrating the effectiveness of the proposed method. We find that dropout decreases the performance compared to CE on low-resource datasets. We conjecture that regularization techniques relying on stochasticity without considering the relevance to the output, in contrast to VIB, can make it more difficult for learning to extract relevant information from a small amount of data. Igl et al. (2019) observe similar effects in another application. See Appendix B for more details and the selected hyper-parameters.

3.1 Results on the GLUE Benchmark

Table 1 shows results on the low-resource datasets in GLUE. Our VIB model substantially outperforms the baselines on all the datasets, improving the results on MRPC, STS-B, and RTE by 1.43/2.03, 2.70/2.97, and 2.60 absolute points respectively, demonstrating the effectiveness of the proposed method. We find that dropout decreases the performance compared to CE on low-resource datasets. We conjecture that regularization techniques relying on stochasticity without considering the relevance to the output, in contrast to VIB, can make it more difficult for learning to extract relevant information from a small amount of data. Igl et al. (2019) observe similar effects in another application. See Appendix B for more details and the selected hyper-parameters.

3.2 Varying-resource Results

To analyze the performance of our method as a function of dataset size, we use large-resource datasets and subsample the training data with varying sizes. Results are shown in Figure 2. VIB consistently improves the fine-tuning of BERT and outperforms dropout on low-resource scenarios, but the advantages are reduced or eliminated as we approach a medium-resource scenario. Also, the improvements are generally larger when the datasets are smaller. See detailed test, validation results, and hyper-parameters in Appendix C.

3.3 Out-of-domain Generalization

Besides improving fine-tuning on low-resource data by removing irrelevant features, we expect VIB to improve on out-of-domain data because it removes redundant features. In particular, annotation artifacts in a specific dataset are known to create shortcut features, which are superficial cues correlated with a label (Gururangan et al., 2018). We hypothesize that these shortcuts are easy to learn (especially when the amount of data is not sufficient), but are actually redundant with deeper features, which capture the true generalizations in the task. By compressing the input embeddings, VIB encourages learning these desirable general features (Shamir et al., 2010; Tishby and Zaslavsky, 2015). By removing the redundant features, the trained model generalizes better to out-of-domain datasets that lack these shortcut features (Belinkov et al., 2019). To evaluate out-of-domain generalization, we take NLI models trained on medium-sized 6K subsampled SNLI and MNLI in Section 3.2 and test their generalization to several NLI datasets.  

Datasets: We consider a total of 12 different NLI datasets used in Mahabadi et al. (2020), including SICK (Marelli et al., 2014), ADD1 (Pavlick and Callison-Burch, 2016), JOCI (Zhang et al., 2017), MPE (Lai et al., 2017), MNLI, SNLI, SciTail (Khot et al., 2018), and three datasets from White et al. (2017) namely DPR (Rahman and Ng, 2012), FN+ (Pavlick et al., 2015), and SPR (Reisinger et al., 2015). We also consider SNLI hard set (Gururangan et al., 2018), a subset of SNLI that avoids its known biases, and Quora Question Pairs (QQP) interpreted as an NLI task as by Gong et al. (2017). We use the same split used in Wang et al. (2017). Since the target datasets have different label spaces, during the evaluation, we map predictions to each target dataset’s space. Following prior work (Belinkov et al., 2019), we perform model selection on the development set of each

See Appendix F for results from models trained on the full SNLI and MNLI datasets.

Figure 2: Results for varying training size resources. We report the mean and standard deviation over three runs, each with a different seed and training samples.
Table 2: Test accuracy results of all models transferring to new target datasets. All models are trained on SNLI or MNLI and tested on the target datasets. \( \Delta \) are absolute differences between IB and CE.

<table>
<thead>
<tr>
<th>Data</th>
<th>SNLI</th>
<th>MNLI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CE</td>
<td>IB</td>
</tr>
<tr>
<td>SICK</td>
<td>48.47</td>
<td>53.58</td>
</tr>
<tr>
<td>ADD1</td>
<td>78.81</td>
<td>84.75</td>
</tr>
<tr>
<td>DPR</td>
<td>50.78</td>
<td>50.14</td>
</tr>
<tr>
<td>SPR</td>
<td>50.21</td>
<td>65.68</td>
</tr>
<tr>
<td>FN+</td>
<td>50.78</td>
<td>53.44</td>
</tr>
<tr>
<td>JOCI</td>
<td>42.03</td>
<td>50.66</td>
</tr>
<tr>
<td>MPE</td>
<td>58.30</td>
<td>58.10</td>
</tr>
<tr>
<td>SCITAIL</td>
<td>62.32</td>
<td>74.84</td>
</tr>
<tr>
<td>QQP</td>
<td>65.19</td>
<td>70.67</td>
</tr>
<tr>
<td>SNLIHard</td>
<td>65.72</td>
<td>68.35</td>
</tr>
<tr>
<td>SNLI</td>
<td>80.54</td>
<td>81.81</td>
</tr>
<tr>
<td>MNLI-M</td>
<td>60.51</td>
<td>64.88</td>
</tr>
<tr>
<td>MNLI</td>
<td>61.79</td>
<td>66.76</td>
</tr>
</tbody>
</table>

Table 3: Ablation results on low-resource datasets in GLUE. We report Pearson/Spearman correlation for STS-B, F1/accuracy for MRPC, and accuracy for RTE.

<table>
<thead>
<tr>
<th>Model</th>
<th>MRPC</th>
<th>STS-B</th>
<th>RTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIB (( \beta = 0 ))</td>
<td>88.57/84.27</td>
<td>87.10/86.0</td>
<td>69.63</td>
</tr>
<tr>
<td>VIB</td>
<td>89.23/85.23</td>
<td>87.63/86.5</td>
<td>70.53</td>
</tr>
</tbody>
</table>

Ablation Study As an ablation, we consider our model without the compression loss (VIB (\( \beta = 0 \))), which reduces to deterministic dimensionality reduction with an MLP since the nonzero variance of embedding \( z \) is only useful for the compression loss. Despite optimizing the reduced dimensionality \( K \) as a hyper-parameter for both methods, this ablation does reduce performance on all considered datasets of GLUE, demonstrating the usefulness of the compression loss of VIB (see details in Appendix D).

5 Conclusion and Future Directions

We propose to apply a VIB module to reduce over-fitting when fine-tuning large-scale pre-trained language models on low-resource datasets. VIB finds the simplest sentence embedding, predictive of the target labels, by removing task-irrelevant and task-redundant information. Our approach is model agnostic, simple to implement, and highly effective. Extensive experiments show that our method substantially improves transfer performance in low-resource scenarios, including a 2.97 point gain on STS-B and a 2.03 point gain on MRPC. Furthermore, we demonstrate that our model results in a better generalization to out-of-domain NLI datasets. Future work includes exploring incorporating VIB on multiple layers of pre-trained language models and using it to jointly learn relevant features and relevant layers.
References

Alex Alemi, Ian Fischer, Josh Dillon, and Kevin Murphy. 2017. Deep variational information bottleneck. ICLR.


Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In EMNLP.


Yichen Gong, Heng Luo, and Jian Zhang. 2017. Natural language inference over interaction space. ICLR.


Alice Lai, Yonatan Bisk, and Julia Hockenmaier. 2017. Natural language inference from multiple premises. In HCNLP.


Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In ACL.

Karimi Rabeeh Mahabadi, Yonatan Belinkov, and James Henderson. 2020. End-to-end bias mitigation by modelling biases in corpora. ACL.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A sick cure for the evaluation of compositional distributional semantic models. In LREC.

Ellie Pavlick and Chris Callison-Burch. 2016. Most “babies” are “little” and most “problems” are “huge”: Compositional entailment in adjective-nouns. In ACL.

Ellie Pavlick, Travis Wolfe, Pushpendre Rastogi, Chris Callison-Burch, Mark Dredze, and Benjamin Van Durme. 2015. Framenet+: Fast paraphrastic tripling of framenet. In ACL.


Altaf Rahman and Vincent Ng. 2012. Resolving complex cases of definite pronouns: the winograd schema challenge. In EMNPL.

Drew Reisinger, Rachel Rudinger, Francis Ferraro, Craig Harman, Kyle Rawlins, and Benjamin Van Durme. 2015. Semantic proto-roles. TACL.

Ohad Shamir, Sivan Sabato, and Naftali Tishby. 2010. Learning and generalization with the information bottleneck. TCS.


Aaron Steven White, Pushpendre Rastogi, Kevin Duh, and Benjamin Van Durme. 2017. Inference is everything: Recasting semantic resources into a unified evaluation framework. In *IJCNLP*.


A Experimental Details

Datasets Statistics Table 4 shows the statistics of the datasets used in our experiments.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Labels</th>
<th>Train</th>
<th>Val.</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMDB</td>
<td>2</td>
<td>20K</td>
<td>5K</td>
<td>25K</td>
</tr>
<tr>
<td>YELP</td>
<td>5</td>
<td>62.5K</td>
<td>7.8K</td>
<td>8.7K</td>
</tr>
</tbody>
</table>

Inference Tasks

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Labels</th>
<th>Train</th>
<th>Val.</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNLI</td>
<td>3</td>
<td>550K</td>
<td>10K</td>
<td>10K</td>
</tr>
<tr>
<td>MNLI</td>
<td>3</td>
<td>393K</td>
<td>9.8K</td>
<td>9.8K</td>
</tr>
<tr>
<td>RTE</td>
<td>2</td>
<td>2.5K</td>
<td>0.08K</td>
<td>3K</td>
</tr>
</tbody>
</table>

Similarity and Paraphrase Tasks

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Labels</th>
<th>Train</th>
<th>Val.</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRPC</td>
<td>2</td>
<td>3.7K</td>
<td>0.4K</td>
<td>1.7K</td>
</tr>
<tr>
<td>STS-B</td>
<td>1 (Similarity score)</td>
<td>5.8K</td>
<td>1.5K</td>
<td>1.4K</td>
</tr>
</tbody>
</table>

Table 4: Datasets used in our experiments.

Data Processing For IMDB, SNLI, MNLI, and tasks in the GLUE benchmark, we do not use any pre-processing. For the considered target NLI datasets in Section 3.3, we use the data in Mahabadi et al. (2020), where the data are first processed with an NLTK tokenizer. We use the Yelp dataset in Zhang et al. (2015), which is also preprocessed with an NLTK tokenizer.

Computing Infrastructure We run all experiments on one GTX1080Ti GPU with 12 GB of RAM.

Base Model We use the BERT base uncased model with default hyper-parameters through all experiments, i.e., we use a sequence length of 128, with batch size 8 and Adam optimizer with a learning rate of 2e-5. We do not use warm-up or weight decay.

VIB We compute the compressed sentence representations using $p_0(z|x) = \mathcal{N}(z|\mu(x),\Sigma(x))$. For computing $\mu(x)$ and $\Sigma(x)$, we first feed sentence embeddings $f_\theta(x)$ through a shallow nonlinear classifier with $768$, $\frac{2^{2044}+K}{4}$, $\frac{768+K}{2}$ hidden units with a ReLU non-linearity. It is then followed by two linear layers, each with K hidden units to compute $\mu(x)$ and $\Sigma(x)$ (after a softplus transform to ensure non-negativity). We average over 5 posterior samples, i.e., we compute $p(y|x) = \frac{1}{5} \sum_{i=1}^{5} q_\phi(z_i)$, where $z_i \sim p_\theta(z|x)$. We consider Gaussian distributions for prior $r(z)$ and $p_\theta(z|x)$ because the Kullback-Leibler divergence between two Gaussian distributions has a closed-form solution. We use the reparameterization trick (Kingma and Welling, 2013) to estimate the gradients, namely $z = \mu(x) + \Sigma(x) \odot \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$. Similar to Bowman et al. (2016), we use a linear annealing schedule for $\beta$ and set it as $\min(1, \text{epoch} \times \beta_0)$ in each epoch, where $\beta_0$ is the initial value.

Dropout As a convention, we apply dropout to the last hidden layer of the BERT encoder.

Similar Training Time Table 5 shows the fine-tuning time for all models. Our VIB method almost has the same training time as the baselines, but our method and dropout require hyper-parameter tuning.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>CE</th>
<th>Dropout</th>
<th>VIB</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRPC</td>
<td>7.18</td>
<td>7.12</td>
<td>7.37</td>
</tr>
<tr>
<td>STS-B</td>
<td>11.62</td>
<td>11.72</td>
<td>11.76</td>
</tr>
<tr>
<td>RTE</td>
<td>4.99</td>
<td>5.06</td>
<td>5.10</td>
</tr>
</tbody>
</table>

Table 5: Fine-tuning times (in minutes) for all models.

Number of Parameters Our VIB model requires $(768+K)K + \frac{(2^{2044}+K)^2}{8}$ parameters for the MLP used to estimate $\mu(x)$ and $\Sigma(x)$ and an additional 2K parameters for the parameters of the prior $r(z)$, namely $\mu_0$ and the diagonal covariance $\Sigma_0$. We consider $K_{\text{min}} = 12$ and $K_{\text{max}} = 384$ in this work; for the BERT base model with 110M trainable parameters, it results in 0.61% - 1.22% of the BERT encoder’s parameters.

B Results on the GLUE Benchmark

Hyper-parameters We fine-tune BERT for 6 epochs and use early stopping for all models by choosing the model performing the best on the validation set. We sweep $\beta$ over $\{1e-4, 1e-5, 1e-6\}$ and K over $\{144, 192, 288, 384\}$. We use the dropout with dropping probabilities of $\{0.25, 0.45, 0.65, 0.85\}$. Dropout with probability 0.25 performs the best on all datasets. Table 6 shows the selected hyper-parameters for VIB, Table 7 shows the corresponding validation accuracies, and Table 8 demonstrates the standard deviations for the reported average results in Table 1.

<table>
<thead>
<tr>
<th>Hyper-parameters</th>
<th>MRPC</th>
<th>STS-B</th>
<th>RTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>288</td>
<td>192</td>
<td>192</td>
</tr>
<tr>
<td>$\beta$</td>
<td>1e-6</td>
<td>1e-5</td>
<td>1e-6</td>
</tr>
</tbody>
</table>

Table 6: Selected hyper-parameters of VIB on GLUE.

C Varying-resource Results

Hyper-parameters We fine-tune all models for 25 epochs. We use early stopping for all models
Table 7: Validation accuracy averaged over 3 runs on the GLUE benchmark. We report F1/accuracy for MRPC, Pearson/Spearman correlation for STS-B, and accuracy for RTE.

<table>
<thead>
<tr>
<th>Method</th>
<th>MRPC</th>
<th>STS-B</th>
<th>RTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE</td>
<td>91.25(0.3)/87.50(0.5)</td>
<td>89.71(0.2)/89.26(0.2)</td>
<td>70.52(1.8)</td>
</tr>
<tr>
<td>Dropout</td>
<td>89.85(0.7)/85.05(1.2)</td>
<td>89.31(0.1)/88.83(0.1)</td>
<td>68.23(1.4)</td>
</tr>
<tr>
<td>VIB</td>
<td>92.06(0.1)/88.81(0.2)</td>
<td>90.36(0.1)/89.88(0.1)</td>
<td>73.41(0.7)</td>
</tr>
</tbody>
</table>

Based on the performance on the validation set, Hyper-parameter tuning is done on the validation set. Since we consider datasets of a different number of training samples, we need to account for a suitable range of dimensions and we sweep K over \{12,18,24,36,48,72,96,144,192,288,384\} and β over \{1e⁻⁴,1e⁻⁵\}. For dropout, we consider dropping probabilities of \{0.25,0.45,0.65,0.85\}. Dropout probability of 0.25 results in the best validation performance on all datasets. Hyper-parameters used to report the final results of VIB on each dataset are shown in Table 9.

Table 8: Standard deviations for the reported results in Table 1 on low-resource datasets in GLUE. We report standard deviation across 3 runs for Pearson/Spearman correlation for MRPC, and accuracy for RTE.

<table>
<thead>
<tr>
<th>Model</th>
<th>MRPC</th>
<th>STS-B</th>
<th>RTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE</td>
<td>0.5/0.6</td>
<td>0.1/0.0</td>
<td>1.5</td>
</tr>
<tr>
<td>Dropout</td>
<td>0.2/0.7</td>
<td>0.9/1.0</td>
<td>1.5</td>
</tr>
<tr>
<td>VIB</td>
<td>0.0/0.2</td>
<td>0.3/0.4</td>
<td>0.5</td>
</tr>
</tbody>
</table>

The results of VIB reported in Table 3 are obtained given the described setup in Appendix B. For VIB (β=0), we sweep K over the same range of values as VIB, i.e., \{12,18,24,36,48,72,96,144,192,288,384\}. Table 12 shows the standard deviations over 3 runs for the reported results in Table 3, Table 13 shows the validation results, and Table 14 shows the selected hyper-parameters.

Table 9: Hyper-parameters used to report the results of the VIB method.

<table>
<thead>
<tr>
<th>Hyper-parameters</th>
<th>200</th>
<th>500</th>
<th>800</th>
<th>1000</th>
<th>3000</th>
<th>6000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset: SNLI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K</td>
<td>48</td>
<td>192</td>
<td>48</td>
<td>192</td>
<td>384</td>
<td>48</td>
</tr>
<tr>
<td>β</td>
<td>1e⁻⁵</td>
<td>1e⁻⁵</td>
<td>1e⁻⁵</td>
<td>1e⁻⁴</td>
<td>1e⁻⁵</td>
<td>1e⁻⁴</td>
</tr>
<tr>
<td>Dataset: MNLI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K</td>
<td>36</td>
<td>24</td>
<td>48</td>
<td>36</td>
<td>288</td>
<td>384</td>
</tr>
<tr>
<td>β</td>
<td>1e⁻⁵</td>
<td>1e⁻⁵</td>
<td>1e⁻⁴</td>
<td>1e⁻⁴</td>
<td>1e⁻⁵</td>
<td>1e⁻⁴</td>
</tr>
<tr>
<td>Dataset: IMDB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K</td>
<td>288</td>
<td>48</td>
<td>192</td>
<td>36</td>
<td>384</td>
<td>18</td>
</tr>
<tr>
<td>β</td>
<td>1e⁻⁴</td>
<td>1e⁻⁴</td>
<td>1e⁻⁴</td>
<td>1e⁻⁴</td>
<td>1e⁻⁴</td>
<td>1e⁻⁴</td>
</tr>
<tr>
<td>Dataset: Yelp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K</td>
<td>72</td>
<td>72</td>
<td>384</td>
<td>384</td>
<td>384</td>
<td>96</td>
</tr>
<tr>
<td>β</td>
<td>1e⁻⁴</td>
<td>1e⁻⁴</td>
<td>1e⁻⁴</td>
<td>1e⁻⁴</td>
<td>1e⁻⁵</td>
<td>1e⁻⁵</td>
</tr>
</tbody>
</table>

Table 10: Selected K for VIB (β=0) on low-resource datasets in GLUE.

Table 11: Validation results for VIB (β=0) on low-resource datasets in GLUE. We report Pearson/Spearman correlation for STS-B, F1/accuracy for MRPC, and accuracy for RTE.

<table>
<thead>
<tr>
<th>MRPC</th>
<th>STS-B</th>
<th>RTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIB</td>
<td>0.0/0.2</td>
<td>0.3/0.4</td>
</tr>
<tr>
<td>VIB (β=0)</td>
<td>0.6/0.7</td>
<td>0.4/0.5</td>
</tr>
</tbody>
</table>

Table 12: Standard deviations for average ablation results on low-resource datasets in GLUE reported in Table 3. We report standard deviation for Pearson/Spearman correlation for STS-B, F1/accuracy for MRPC, and accuracy for RTE.

Table 13: Validation results for VIB (β=0) on low-resource datasets in GLUE. We report Pearson/Spearman correlation for STS-B, F1/accuracy for MRPC, and accuracy for RTE.

<table>
<thead>
<tr>
<th>MRPC</th>
<th>STS-B</th>
<th>RTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIB</td>
<td>91.9 (0.3)/88.65 (0.3)</td>
<td>90.31 (0.1)/89.8 (0.1)</td>
</tr>
</tbody>
</table>

Table 14: Selected K for VIB (β=0) on low-resource datasets in GLUE.

**E Out-of-domain Generalization**

**Mapping:** We train all models on SNLI or MNLI datasets and evaluate their performance on other target datasets. The SNLI and MNLI datasets contain three labels of contradiction, neutral, and entailment. However, some of the considered target datasets experiment with three random sampled training instances.
have only two labels, such as DPR or SciTail. When the target dataset has two labels of entailed and not-entailed, as in DPR, we consider the predicted contradiction and neutral labels as the not-entailed label. In the case the target dataset has two labels of entailment and neutral, as in SciTail, we consider the predicted contradiction label as neutral.

**Hyper-parameters** We consider the models trained on 6k subsampled SNLI and MNLI explained in Appendix C. Table 15 shows the selected hyper-parameters for each target dataset, and Table 16 shows their corresponding validation accuracies.

### F High-resource Out-of-domain Generalization

**Hyper-parameters** we train all models for the default 3 epochs as in Devlin et al. (2019). Hyper-parameter tuning is done on the validation set. We sweep K over \{12, 18, 24, 36, 48, 72, 96, 144, 192, 288, 384\} and \(\beta\) over \{1e-4, 1e-5\}.

**Results** Table 17 shows the transfer performance on the high-resource setting. Models are trained on the full SNLI or MNLI datasets and transferred to
the various target datasets. The model trained on SNLI with VIB improves the transfer on average by 1.54 absolute points; the improvement for the model trained on MNLI with VIB is lower and is 0.25 points. We discuss the transfer performance from the medium-sized SNLI and MNLI datasets in Section 3.3 and observe that VIB is effective in this scenario. We conjecture that when the available data is not sufficient for the models to learn more general features, it is easier for them to learn easy-to-learn shortcut features; learning these shortcuts is reduced by motivating the network towards learning more general features through a VIB module. However, in high-resource datasets, there are already enough data available to learn the general features; therefore, the impact of VIB is reduced, and one needs other methods specifically targeting bias reduction like Mahabadi et al. (2020) and Clark et al. (2019) to address this issue. Investigating this phenomenon further is future work. We report the hyper-parameters in Table 19 and their corresponding validation accuracies in Table 18.

<table>
<thead>
<tr>
<th>Data</th>
<th>SNLI</th>
<th>MNLI</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>β</td>
<td>K</td>
</tr>
<tr>
<td>SICK</td>
<td>48</td>
<td>1e-5</td>
</tr>
<tr>
<td>ADD1</td>
<td>12</td>
<td>1e-4</td>
</tr>
<tr>
<td>DPR</td>
<td>192</td>
<td>1e-5</td>
</tr>
<tr>
<td>SPR</td>
<td>48</td>
<td>1e-5</td>
</tr>
<tr>
<td>FN+</td>
<td>192</td>
<td>1e-5</td>
</tr>
<tr>
<td>JOCI</td>
<td>72</td>
<td>1e-5</td>
</tr>
<tr>
<td>MPE</td>
<td>48</td>
<td>1e-5</td>
</tr>
<tr>
<td>SCITAIL</td>
<td>36</td>
<td>1e-4</td>
</tr>
<tr>
<td>QQP</td>
<td>12</td>
<td>1e-4</td>
</tr>
<tr>
<td>SNLIHard</td>
<td>48</td>
<td>1e-4</td>
</tr>
</tbody>
</table>

Table 15: Hyper-parameters used to report the results of the out-of-domain generalization experiment.

<table>
<thead>
<tr>
<th>Data</th>
<th>SNLI</th>
<th>MNLI</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE</td>
<td>VIB</td>
<td>Δ</td>
</tr>
<tr>
<td>SICK</td>
<td>51.2</td>
<td>90.80</td>
</tr>
<tr>
<td>ADD1</td>
<td>73.73</td>
<td>79.22</td>
</tr>
<tr>
<td>DPR</td>
<td>50.41</td>
<td>50.41</td>
</tr>
<tr>
<td>SPR</td>
<td>50.16</td>
<td>64.95</td>
</tr>
<tr>
<td>FN+</td>
<td>50.85</td>
<td>54.54</td>
</tr>
<tr>
<td>JOCI</td>
<td>42.03</td>
<td>50.66</td>
</tr>
<tr>
<td>MPE</td>
<td>57.46</td>
<td>58.66</td>
</tr>
<tr>
<td>SCITAIL</td>
<td>67.25</td>
<td>78.02</td>
</tr>
<tr>
<td>QQP</td>
<td>65.29</td>
<td>71.55</td>
</tr>
</tbody>
</table>

Table 16: Validation accuracy for all models transferring to new target datasets. All models are trained on 6K subsampled SNLI or MNLI and evaluated on the target datasets. Δ are absolute differences between VIB and the baseline CE.

<table>
<thead>
<tr>
<th>Data</th>
<th>SNLI</th>
<th>MNLI</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE</td>
<td>VIB</td>
<td>Δ</td>
</tr>
<tr>
<td>SICK</td>
<td>57.13</td>
<td>57.5</td>
</tr>
<tr>
<td>ADD1</td>
<td>87.34</td>
<td>88.37</td>
</tr>
<tr>
<td>DPR</td>
<td>50.34</td>
<td>50.23</td>
</tr>
<tr>
<td>SPR</td>
<td>75.29</td>
<td>75.43</td>
</tr>
<tr>
<td>FN+</td>
<td>51.35</td>
<td>55.45</td>
</tr>
<tr>
<td>JOCI</td>
<td>50.66</td>
<td>51.48</td>
</tr>
<tr>
<td>MPE</td>
<td>68.00</td>
<td>70.78</td>
</tr>
<tr>
<td>SCITAIL</td>
<td>66.13</td>
<td>67.83</td>
</tr>
<tr>
<td>QQP</td>
<td>67.35</td>
<td>67.78</td>
</tr>
</tbody>
</table>

Table 17: Test accuracy for all models transferring to new target datasets. All models are trained on SNLI or MNLI and tested on the target datasets. Δ are absolute differences between VIB and the baseline CE.

<table>
<thead>
<tr>
<th>Data</th>
<th>SNLI</th>
<th>MNLI</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE</td>
<td>VIB</td>
<td>Δ</td>
</tr>
<tr>
<td>SICK</td>
<td>53.40</td>
<td>62.20</td>
</tr>
<tr>
<td>ADD1</td>
<td>77.25</td>
<td>77.45</td>
</tr>
<tr>
<td>DPR</td>
<td>50.21</td>
<td>50.62</td>
</tr>
<tr>
<td>SPR</td>
<td>57.37</td>
<td>57.34</td>
</tr>
<tr>
<td>FN+</td>
<td>52.49</td>
<td>55.33</td>
</tr>
<tr>
<td>JOCI</td>
<td>50.66</td>
<td>51.48</td>
</tr>
<tr>
<td>MPE</td>
<td>66.87</td>
<td>68.07</td>
</tr>
<tr>
<td>SCITAIL</td>
<td>69.63</td>
<td>73.31</td>
</tr>
<tr>
<td>QQP</td>
<td>67.11</td>
<td>67.91</td>
</tr>
</tbody>
</table>

Table 18: Validation accuracy for all models transferring to new target datasets. All models are trained on SNLI or MNLI and evaluated on the target datasets. Δ are absolute differences between VIB and the baseline CE.
<table>
<thead>
<tr>
<th>Data</th>
<th>SNLI</th>
<th>MNLI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>K</td>
<td>β</td>
</tr>
<tr>
<td>SICK</td>
<td>96</td>
<td>1e-4</td>
</tr>
<tr>
<td>ADD1</td>
<td>288</td>
<td>1e-4</td>
</tr>
<tr>
<td>DPR</td>
<td>18</td>
<td>1e-5</td>
</tr>
<tr>
<td>SPR</td>
<td>72</td>
<td>1e-5</td>
</tr>
<tr>
<td>FN+</td>
<td>72</td>
<td>1e-4</td>
</tr>
<tr>
<td>JOCI</td>
<td>72</td>
<td>1e-5</td>
</tr>
<tr>
<td>MPE</td>
<td>12</td>
<td>1e-4</td>
</tr>
<tr>
<td>SCITAIL</td>
<td>96</td>
<td>1e-4</td>
</tr>
<tr>
<td>QQP</td>
<td>192</td>
<td>1e-5</td>
</tr>
<tr>
<td>SNLIHard</td>
<td>144</td>
<td>1e-5</td>
</tr>
</tbody>
</table>

Table 19: Hyper-parameters used to report the results on the high-resource out-of-domain generalization experiment.