Fine-tuning BERT for Low-Resource Natural Language Understanding via Active Learning

Daniel Grießhaber
Institute for Applied Artificial Intelligence (IAAI)
Hochschule der Medien Stuttgart
Nobelstraße 10
70569 Stuttgart
griesshaber@hdm-stuttgart.de

Johannes Maucher
IAAI
Hochschule der Medien Stuttgart
Nobelstraße 10
70569 Stuttgart
maucher@hdm-stuttgart.de

Ngoc Thang Vu
Institute for Natural Language Processing (IMS)
University of Stuttgart
Pfaffenwaldring 5B
70569 Stuttgart
thangvu@ims.uni-stuttgart.de

Abstract

Recently, leveraging pre-trained Transformer based language models in down stream, task specific models has advanced state of the art results in natural language understanding tasks. However, only a little research has explored the suitability of this approach in low resource settings with less than 1,000 training data points. In this work, we explore fine-tuning methods of BERT - a pre-trained Transformer based language model - by utilizing pool-based active learning to speed up training while keeping the cost of labeling new data constant. Our experimental results on the GLUE data set show an advantage in model performance by maximizing the approximate knowledge gain of the model when querying from the pool of unlabeled data. Finally, we demonstrate and analyze the benefits of freezing layers of the language model during fine-tuning to reduce the number of trainable parameters, making it more suitable for low-resource settings.

1 Introduction

Pre-trained language models have received great interest in the natural language processing (NLP) community in the last recent years (Dai and Le, 2015; Radford, 2018; Howard and Ruder, 2018; Baevski et al., 2019; Dong et al., 2019). These models are trained in a semi-supervised fashion to learn a general language model, for example, by predicting the next word of a sentence (Radford, 2018). Then, transfer learning (Pan and Yang, 2010; Ruder et al., 2019; Moeed et al., 2020) can be used to leverage the learned knowledge for a down-stream task, such as text-classification (Do and Ng, 2006; Aggarwal and Zhai, 2012; Reimers et al., 2019; Sun et al., 2019b).

Devlin et al. (2019) introduced the "Bidirectional Encoder Representations from Transformers" (BERT), a pre-trained language model based on the Transformer architecture (Vaswani et al., 2017). BERT is a deeply bidirectional model that was pre-trained using a huge amount of text with a masked language model objective where the goal is to predict randomly masked words from their context (Taylor, 1953). The fact is, BERT has achieved state of the art results on the "General Language Understanding Evaluation" (GLUE) benchmark (Wang et al., 2018) by only training a single, task-specific layer at the output and fine-tuning the base model for each task. Furthermore, BERT demonstrated its applicability to many other natural language tasks since then including but not limited to sentiment analysis (Sun et al., 2019a; Xu et al., 2019; Li et al., 2019), relation extraction (Baldini Soares et al., 2019; Huang et al., 2019b) and word sense disambiguation (Huang et al., 2019a; Hadiwinoto et al., 2019; Huang et al., 2019a), as well as its adaptability to languages other than English (Martin et al., 2020; Antoun et al.,

Proceedings of the 28th International Conference on Computational Linguistics, pages 1158–1171
Barcelona, Spain (Online), December 8-13, 2020

2020; Agerri et al., 2020). However, the fine-tuning data set often contains thousands of labeled data points. This plethora of training data is often not available in real world scenarios (Tan and Zhang, 2008; Wan, 2008; Salameh et al., 2015; Fang and Cohn, 2017).

In this paper, we focus on the low-resource setting with less than 1,000 training data points. Our research attempts to answer the question if pool-based active learning can be used to increase the performance of a text classifier based on a Transformer architecture such as BERT. That leads to the next question: How can layer freezing techniques (Yosinski et al., 2014; Howard and Ruder, 2018; Peters et al., 2019), i.e. reducing the parameter space, impact model training convergence with fewer data points?

To answer these questions, we explore the use of recently introduced Bayesian approximations of model uncertainty (Gal and Ghahramani, 2016) for data selection that potentially leads to faster convergence during fine-tuning by only introducing new data points that maximize the knowledge gain of the model. To the best of our knowledge, the work presented in this paper is the first demonstration of combining modern transfer learning using pre-trained Transformer-based language model such as the BERT model with active learning to improve performance in low-resource scenarios. Furthermore, we explore the effect of trainable parameters reduction on model performance and training stability by analyzing the layer-wise change of model parameters to reason about the selection of layers excluded from training.

The main findings of our work are summarized as follows: a) we found that the model's classification uncertainty on unseen data can be approximated by using Bayesian approximations and therefore, used to efficiently select data for manual labeling in an active learning setting; b) by analyzing layer-wise change of model parameters, we found that the active learning strategy specifically selects data points that train the first and thus more general natural language understanding layers of the BERT model rather than the later and thus more task-specific layers.

2 Methods

2.1 Base Model

In (Devlin et al., 2019) a simple classification architecture subsequent to the output of the Transformer is used to calculate the cross-entropy of the classifier for a text classification task with C classes. Specifically, first, a dropout operation (Srivastava et al., 2014) is applied to the Transformer's last layer hidden state of the special `[CLS]` token that is inserted at the beginning of each document. The regularized output is then fed into a single fully-connected layer with C output neurons and a softmax activation function to scale the logits of its output to probabilities of class association.

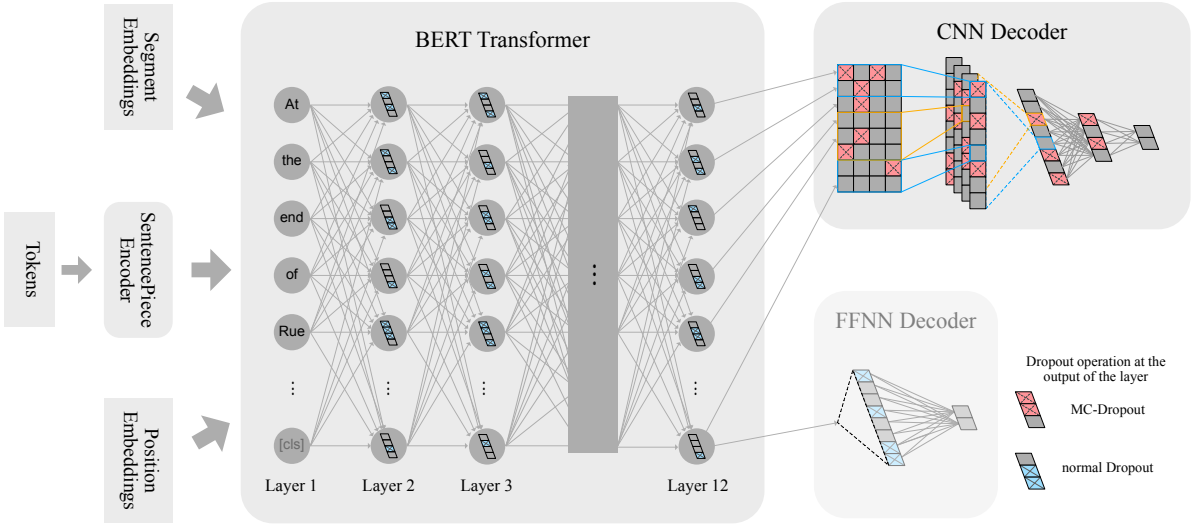


Figure 1: High-level overview of the architecture used in our experiments including a depiction of the simple FFNN decoder used by Devlin et al. (2019) highlighting where our model differs from the original experiment setup.

In contrast, we use a more complex classification architecture based on a convolutional neural network (CNN) following Kim (2014)[1]. All hidden states in the last layer of the BERT model are arranged in a 2-dimensional matrix. Then, convolutional filters of height $(3, 4, 5)$ and length corresponding to the hidden state size (786) are shifted over the input to calculate 64 1-dimensional feature maps per filter size. These feature maps are then batch-normalized (Ioffe and Szegedy, 2015), dropout regularized and global max-pooled before they are concatenated and fed into 2 fully-connected layers, each of which applies another dropout operation on its input [2].

2.2 How to Select Data

When labeled training data is sparse, but unlabeled in-domain data is readily available, manual labeling of all data is often not feasible due to cost. In this scenario, it is advantageous to let the current model select Q data points that it is most confused about from the pool of potential but yet unlabeled training elements (U) to be labeled by an human annotator. Then, they can be included in the training set $T_{new} = T_{old} \cup U_{x \in U_{argmax}(a(x,\mathcal{M}))\langle 1,\ldots,Q\rangle}$ where $a(x, \mathcal{M})$ is a function predicting the knowledge gain of the model $\mathcal{M}$ when trained on the datum x. Thus, the model is specifically trained on data that it can not yet confidently classify to maximize the knowledge gain in each training step, while keeping the cost of labeling new data constant.

We propose to estimate model prediction uncertainty by using Bayesian approximations as presented in Gal and Ghahramani (2016) for data selection process. The main idea is to leverage stochastic regularization layers (e.g. Dropout or Gaussian noises) that can be used to approximate model uncertainty ($\phi(x)$) on any datum (x) by performing multiple stochastic forward passes for each element in U. This is implemented by applying the layer operation not only during training but also during the interference stage of the model. Multiple forward passes of the model $\mathcal{M}$ with the same parameters (θ) and inputs (x) thus yield different model outputs (y), as each pass samples a concrete model from a approximate distribution $q^*_\theta(\omega)$ that minimizes the Kullback-Leibler divergence to the true model posterior $p(\omega|T)$. Gal and Ghahramani (2016) call this Monte-Carlo-Dropout (MC-Dropout) as the repeated, non-deterministic forward passes of the model can be interpreted as a Monte-Carlo process.

To decide which elements in U are chosen, different acquisition functions ($a(x, \mathcal{M})$) could be used. In this work, we focus on the "Bayesian Active Learning by Disagreement (BALD)" (Houlsby et al., 2011) acquisition strategy because it demonstrated very good performance in comparison to other strategies in the experiments by Gal et al. (2017) as well as in our own preliminary experiments. BALD calculates the information gain for the model's parameters that can be achieved with the new data points, that is, the mutual information between predictions and parameters $\mathbb{I}[y, \omega|x, T]$. Hence, this acquisition function has maximum values for inputs that produce disagreeing predictions with high uncertainty. This is equivalent to high entropy in the logits (the unscaled output of the network before the final softmax normalization) in a classifier model, as multiple (stochastic) forward passes of the model with the same input yield different classification results. We use the same approximation of BALD that Gal et al. (2017) used in their work (equation 1) where $\widehat{p}^s_c$ is the probability of class association (i.e. the softmax scaled logits) for input x and class c for one of S samples $\widehat{\omega}_s \sim q^*_\theta(\omega)$ from the model's approximated posterior distribution, i.e. $\widehat{p}^s = \text{softmax}(\mathcal{M}(y|x, \theta, \widehat{\omega}_s))$.

$$
\begin{aligned}
&a_{BALD}(x, \mathcal{M}) = \\
&-\sum_{c \in C} \left(\frac{1}{S} \sum_{s=1}^{S} \widehat{p}^s_c \right) log \left(\frac{1}{S} \sum_{s=1}^{S} \widehat{p}^s_c \right) + \frac{1}{S} \sum_{c \in C, s=1}^{s=S} \widehat{p}^s_c \, log \, \widehat{p}^s_c \approx \mathbb{I}[y, \omega|x, T]
\end{aligned} \tag{1}
$$

As a baseline, we compare the BALD strategy with random sampling of elements in U. That is, $a_{Rand}(x, \mathcal{M}) = unif[0, 1)$, where $unif[a, b)$ is a sample from the uniform distribution in the half-closed interval $[a, b)$.

[1] We found that using a CNN is slightly better than using a simple feed forward neural network in our experiments.

[2] All implementation details can be found in the source code provided in `https://gitlab.mi.hdm-stuttgart.de/griesshaber/bayesian-active-learning`

When using the BALD acquisition function, we sample for $S = 50$ forward passes and add $Q = 100$ data points with the highest model uncertainty according to the calculated BALD scores. When randomly acquiring new data points for the baseline, no forward passes are needed ($S = 0$) while the number of acquisitions Q stays constant. The MC-Dropout layers that apply the dropout operation also during a stochastic forward pass of the model, are only placed in the classification architecture, thus the base model is used unaltered with regular dropout layers only active during the training phase. In our CNN architecture, the dropout is only applied in the penultimate layer of the network. We use the same dropout rate of 0.1 as Devlin et al. (2019) in the decoder. This means that the stochastic forward passes only affect the output of the layers after the first MC-Dropout layer in the model. Thus, only a single and relativley expensive pass through the transformer is needed in each learning step while the required multiple passes though the subsequent classifiers are comparatively cheap.

2.3 How to Fine-tune Models

Reducing the Number of Trainable Parameters Freezing of parameters can be useful when fine-tuning a complex model, as it effectively lowers the number of parameters that need to be tuned during training (Howard and Ruder, 2018). This is especially relevant in the low-resource setting with a small amount of training data, since the pre-trained $\text{BERT}_{\text{BASE}}$ model with $\sim 110M$ parameters is over-parametrized for such small data sets. However, since freezing parameters also lowers the adaptability of a model (Peters et al., 2019), it is crucial to determine which parameters are frozen and which can be fine-tuned during training to not negatively affect the model's performance.

To our best knowledge no prior work has considered the training set size as a dependent parameter. This parameter is especially important in low-resource settings. Therefore, we will conduct experiments to visualize the model's performance in dependence of training data size with different sets of layers that are frozen during training. We denote the number of frozen layers with F. For positive values, the layers in the half-closed integer interval $[0\ ..\ F)$ are frozen, while negative values represent the interval $[12+F\ ..\ 12)$, i.e. the last $-F$ layers of the BERT model. Layers of the classification architecture are always trainable during training because they are initialized randomly and thus need to be tuned. We also visualize the change of model parameters in each layer during fine-tuning to reason about the choice of frozen layers. Results of these experiments are presented in section 4.3.

Analyzing Changes in Layer-Parameters To analyze the changes in model parameters during the fine-tuning of the BERT model, we capture the initial state of all model parameters before starting the training. This includes all pre-trained model weights, as well as the randomly initialized parameters of the CNN denoted by θ_0. We can then snapshot the same parameters after training for 3 epochs denoted by θ_3. The change in the model parameters can thus be described by $\Delta\theta = \theta_3 - \theta_0$. Since the number of parameters is very high ($7,087,872$ for every layer of the BERT model while the exact number of parameters of the CNN depends on the output configuration of the model), we aggregate the change of parameters per layer by calculating the mean absolute difference (MAD) of the parameters. The mean absolute difference in layer x can be described by equation 2 where θ_{exn} is the value of the n-th parameter in layer x after training epoch e.

$$\begin{aligned} \overline{|\Delta\theta_x|} &= \frac{1}{N}\sum_{i=0}^{N}|\Delta\theta_{xn}| \\ &= \frac{1}{N}\sum_{i=0}^{N}|\theta_{0xn} - \theta_{3xn}| \end{aligned} \tag{2}$$

3 Experiment Setup

Base Model The BERT model, as introduced by Devlin et al. (2019), is used to create contextualized embeddings in all experiments. Specifically, the pre-trained language model $\textbf{BERT}_{\textbf{BASE}}$[3] with 12 layers, 768 hidden states and 12 self-attention heads made available online by the authors is used.

[3] Available under
https://storage.googleapis.com/bert_models/2018_10_18/uncased_L-12_H-768_A-12.zip

Data Sets Similar to the experiments presented by Devlin et al. (2019), we use the "GLUE Multi-Task Benchmark" (Wang et al., 2018) consisting of multiple NLP classification tasks including sentiment analysis and textual entailment to evaluate the performance of the different model configurations. Contrary to (Devlin et al., 2019), we report the average accuracy of $N = 3$ runs instead of the maximum achieved value for each setting. By doing so, we are able to analyze training stability by comparing the distribution of the results over different runs.

Low-Resource Scenarios To simulate the low-resource setting in the GLUE tasks, we repeatedly evaluate the model's performance after training for 3 epochs on a subset of the available data. Since data points in the training set provided by the GLUE Benchmark are already shuffled, the subset S_x simply contains the first x data points ($S_x = \{s_1, s_2, \ldots, s_x\}$) to ensure the same data selection between experiments.

4 Results

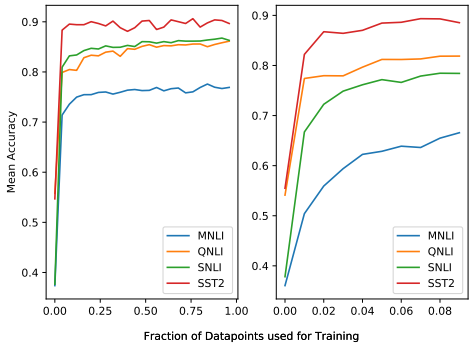


Figure 2: Accuracies of the BERT model when adapted on 100% and 10% (using random aquisition) of the available training data respectively.

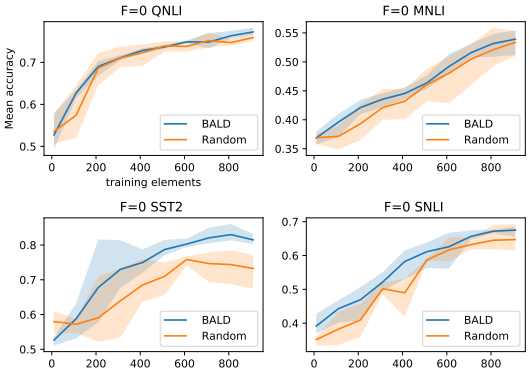


Figure 3: Model accuracies when adapted with data selected with the BALD acquisition strategy vs. random data selection.

4.1 Effect of Training Data Size

As expected, all models benefit from an increased number of training elements until reaching comparable performance to the results reported by (Devlin et al., 2019) when trained on the whole data set. However, Figure 2 shows that, while the accuracy is generally increasing for all experiments when introducing more training data, the model has a much larger increase in performance at the beginning, while adding another partition of the data set to an already large training set can only marginally improve the model's accuracy. Adding more data, however, generally reduces the variance in model performance between runs when randomly initializing the classification architecture's parameters. Due to these observations and for comparability of the results between data sets, the experiments in the following subsections are all performed on an initial training set of S_{10} and reevaluated repeatably for 9 iterations after adding another partition of 100 elements until the final subset S_{910} is reached. While at this point the performance has not yet converged to the final accuracies for any dataset, the performance changes drastically in these early training stages, which is ideal for our experimental comparison of training speed increases.

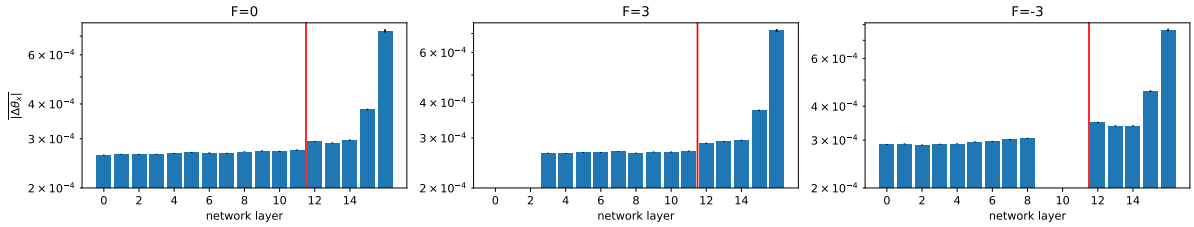
4.2 Active Learning

The active learning setting requires a pool of unlabeled data to pick the next training elements to acquire. Ideally, the knowledge gain $a(x, \mathcal{M})$ of all elements in the dataset would be approximated, however due to the need of multiple forward passes for each element in the pool to approximate a only a subset $U = S_{20,000} \setminus S_{10}$ was chosen for our experiments to seed up training. Thus, the size of U can be seen as a hyperparameter to speed up training with the trade off of the risk to exclude highly relevant data points from the available pool data. The size of U was chosen by increasing it for multiple runs until

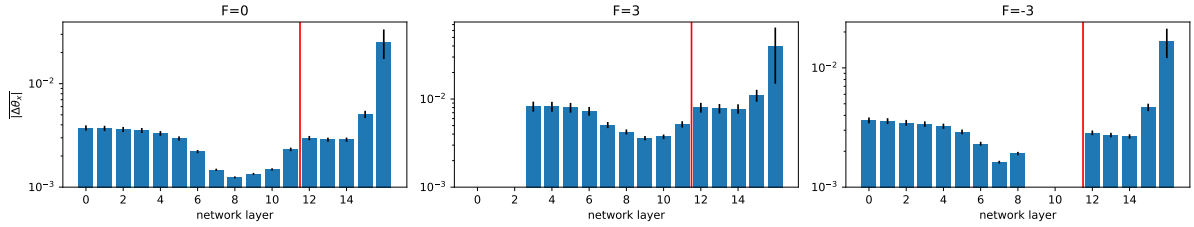
the performance increase leveled off and acquisition times were still acceptable. Since training with the random strategy does not have the same trade-off, all available data points in the pool can be used for the acquisition without lengthening the training.

The results in Figure 3 show a better mean accuracy for all models and generally across all training set sizes when the BALD strategy is used, compared to picking random training elements from U. Another noteworthy observation is the lower spread of model accuracy between experiments when training data is selected based on the active learning strategy. These results show a strong indication that the Monte-Carlo approximation of model uncertainty works for state of the art Transformer architectures like the BERT model and can improve training performance when an active learning scenario in a low-resource setting is feasible.

4.3 Layer Freezing to Reduce Number of Parameters



(a) random sampling of new training elements from U



(b) selective acquisition of elements with maximum BALD score from U

Figure 4: Visualization of the mean absolute difference of parameters when training a classifier model on the QNLI data set. The blue bars indicate the variance over all weight changes. The red line indicates the border between the BERT model and the decoder CNN. Layers 12, 13 and 14 are the parallel convolutional layers, 15 and 16 the fully-connected hidden and output layers.

Figure 4 visualizes the mean absolute difference of model parameters on a per-layer basis when fine-tuning on the QNLI data set. Note that the frozen layers have a mean absolute difference of 0 as the training does not alter any parameters in these layers. When randomly sampling from U is used to choose new data elements, layers closer to the output of the model generally have a higher change in their parameters during training which is in line with the findings by Yosinski et al. (2014), Howard and Ruder (2018) and Hao et al. (2019) which show that, while fine-tuning a Transformer-based language model, the front layers have a more general language understanding while the later layers capture more task specific concepts and thus need to be trained more. However, if the active learning strategy is used, layers closer to the output of the BERT model have a smaller value change while the relative mean absolute difference in the layers of the CNN is comparable between the two strategies. This indicates that the active learning strategy specifically selects data points that train the first and thus more general layers of the BERT model.

Freezing of layers with a smaller overall change in parameters during training may be beneficial in low-resource scenarios, since this reduces the number of parameters in the model equally while potentially enabling more freedom in the layers that need to be tuned more. Based on this hypothesis, we compare the model's performance when freezing different numbers of layers starting on both, the input and output of the BERT model.

Table 1 shows a general increase in model performance when freezing 25% of the BERT's layers, indicating that a reduction of parameters in the low-resource setting is indeed beneficial. However, with 50% of the layers fixed during training, the average performance of the model decreases again, in many cases even below the baseline where all parameters are trainable. This is an indicator that the BERT model needs fine-tuning and freezing a large ratio of layers may result in a model that can not adapt adequately to the task and is in line with the results of Peters et al. (2019).

Figure 5 and Table 2 also show the mean accuracy and bounds of these experiments over a range of different numbers of training elements. The average accuracy of the models after multiple training trials do not show any conclusive advantage of freezing the later BERT layers over the the ones in the front. However, for all data sets the models are more stable during the training over different runs, indicated by a lower average width of the bounds when using $F = -3$.

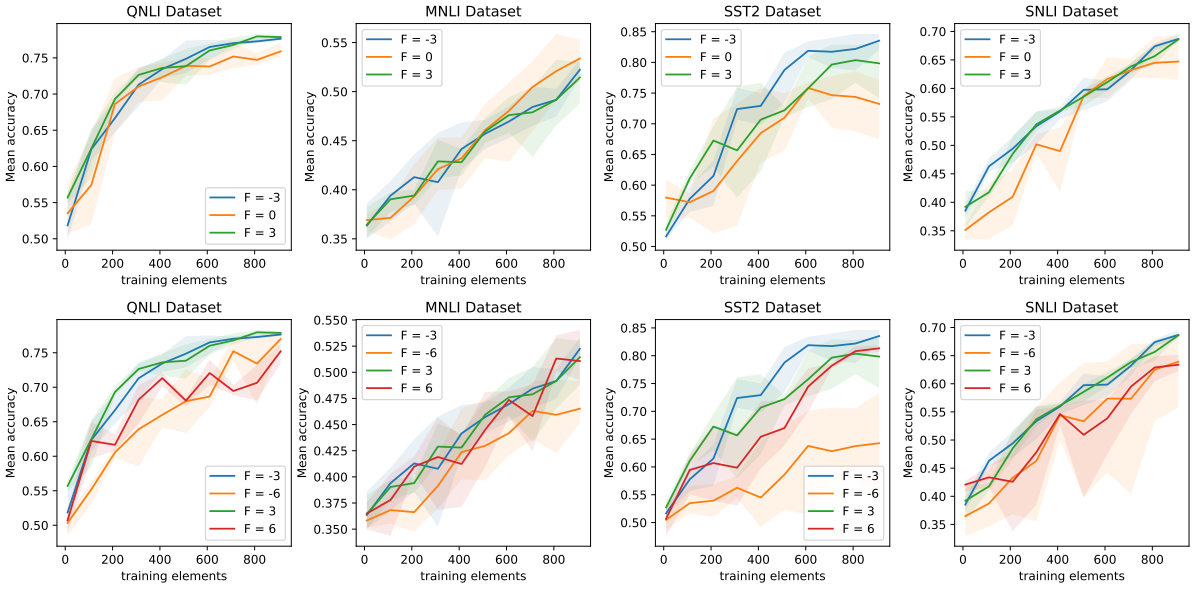


Figure 5: The effect of layer-freezing on the stability of the training during active learning using the BALD strategy. The semi-transparent area in the chart is the confidence interval over multiple runs of the configuration, while the solid line denotes the average accuracy.

F	MNLI	QNLI	SST2	SNLI
0	0.53 ± 0.021	0.76 ± 0.010	0.78 ± 0.059	0.67 ± 0.015
3	0.51 ± 0.021	0.78 ± 0.003	0.80 ± 0.045	0.69 ± 0.002
-3	0.52 ± 0.010	0.78 ± 0.002	0.84 ± 0.013	0.69 ± 0.008
6	0.51 ± 0.024	0.75 ± 0.014	0.81 ± 0.006	0.63 ± 0.014
-6	0.47 ± 0.020	0.77 ± 0.010	0.64 ± 0.094	0.64 ± 0.067

Table 1: Mean accuracies of the model after training on 910 data points using the BALD acquisition strategy.

F	MNLI	QNLI	SST2	SNLI
0	0.054	0.049	0.108	0.061
3	0.043	0.032	0.078	0.032
-3	0.038	0.024	0.047	0.028
6	0.050	0.033	0.071	0.082
-6	0.050	0.050	0.113	0.161

Table 2: Mean width of the confidence intervals, i.e. the difference between upper and lower limit, over different runs using the BALD acquisition strategy.

5 Qualitative Analysis

Overall Observation Figure 6 visualizes the number of training elements in T grouped by their label $_c$ after each acquisition iteration of picking 100 elements from U. As the labels in the data sets and thus the subset U are equally distributed, we expect the same equal distribution when sampling randomly from the pool elements, which is apparent in the lower row of Figure 6 where the training data was actually randomly chosen from the pool of data. In contrast, we observed that when using the BALD acquisition with non-deterministic forward passes, the distribution shows a stronger bias to a particular class. This bias increases when sampling more pool data, whereas the difference between the biggest and smallest class in T stays constant during random acquisition (see table 3). Furthermore, when randomly sampling, the class with the most or least training elements ($\arg\min_c |T_c|$, $\arg\max_c |T_c|$) is changing in many iterations, whereas in the case where active learning is used, this is constant for the most part and changing, if any, only in the first iterations.

Which Class? As the goal of active learning is to maximize the knowledge gain of the model with minimum cost, the choice of data the acquisition strategy selects from U may give further insight into the models understanding of the input data. One characteristic when applying active learning on the MNLI data set, that is apparent in Figure 6, is that samples from the `neutral` class are queried the most. It might indicate that the model is most confused about the samples from this class. This observation may be justified by the fact that the hypothesis stated in that datum may be unrelated to the associated premise, confusing the model.

Another observation is that the model queries data points from the `contradiction` class more often than data points from the `entailment` class at the beginning of the training, while at the end number of queries from the different classes is almost equal. This indicates that, at the start of training, the model is more confused about contradictions, which may be explained by the fact that a contradiction can differ from an entailment only by a single negation at the correct place in the input, making it harder to differentiate from an entailment. An example of negated statements in a contradiction can be seen in the last row of Table 4 which also shows that the model indeed gave this example a high BALD score.

Table 4 shows some of the examples of the MNLI dataset with their calculated BALD scores in the middle of the training after 5 acquisition iterations. These examples show another noteworthy behaviour of the active learner. Samples from the entailment class where many words in the two input sentences match or and have a similar wording in general, get a low BALD score, indicating that the model is already confident that it is able to correctly classify those examples. The same is true for input pairs that differ in wording and contradict themselves. However, entailing pairs where the wording is mostly different between the two inputs, as well as contradicting pairs with similar phrasing get a much higher BALD score and thus are more likely to be sampled during the acquisition phase. The model at this point in training thus seems to already have learned to perform its task confidently on the *simpler* examples and can thus concentrate more on the *non-trivial* data-points.

6 Related Work

Low-resource NLP Previous work in low-resource NLP tasks includes feature-engineering (Tan and Zhang, 2008) which requires a recurring effort when adapting to a new data set. Another approach is to transfer knowledge across domains to increase the amount of data that is available for training (Zoph et al., 2016; Nguyen and Chiang, 2017; Kocmi and Bojar, 2018; Yang et al., 2017; Gupta et al., 2018). One of these approaches relied on adversarial training (Goodfellow et al., 2014) to learn a domain adaptive classifier (Ganin et al., 2016) in another domain or language where training data was plentiful while ensuring that the model generalizes to the low-resource domain (Chen et al., 2018). However, these approaches have not used a pre-trained generic language model, but perform pre-training for each task individually.

Adapting pre-trained models The effectiveness of transfer learning in low-resource settings was previously demonstrated for machine translation (Zoph et al., 2016; Nguyen and Chiang, 2017; Kocmi and Bojar, 2018), sequence tagging (Yang et al., 2017) and sentiment classification (Gupta et al., 2018). However, all this prior work does not use general LMs but transfers knowledge from a pre-trained model to a

		Data set			
		BALD		Random	
		MNLI	QNLI	MNLI	QNLI
$\|T\|$	110	13	30	3	4
	210	19	24	3	2
	310	20	16	1	2
	410	29	18	2	2
	510	30	36	9	4
	610	28	40	6	6
	710	37	44	3	2
	810	34	44	4	4
	910	25	36	9	4

Table 3: Differences in number of elements in the largest and smallest group in the trainset: $\Delta\,|T| = \max(|T_c|) - \min(|T_c|)$

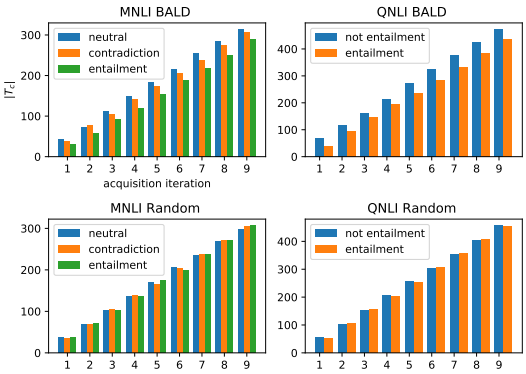


Figure 6: Distribution of training elements in T for different acquisition strategies with the BERT model. $|T_c|$ is the number of elements in T labeled with $_c$

queried	$a_{BALD}(x, \mathcal{M})$	x_1	x_2	y
no	-8.2×10^8	The view of the illuminated mont at night is spectacular from the other side of the bay.	The view of the lit up mont at night is wonderful.	entailment
no	-5.9×10^8	At the end of Rue des Francs-Bourgeois is what many consider to be the city's most handsome residential square, the Place des Vosges, with its stone and red brick facades.	Place des Vosges is constructed entirely of gray marble.	contradiction
yes	6.2×10^8	The grounds, like those of any other private house after nightfall, seemed untenanted.	The grounds were empty.	entailment
yes	8.6×10^8	And Forrest saying: "We don't give medals, Sergeant.	Forrest said they gave medals ever week.	contradiction
yes	5.2×10^8	I did not wonder at John objecting to his beard.	John had no objections to his beard.	contradiction

Table 4: Examples from the MNLI Dataset with the two input sentences and the associated class label. The first column shows whether or not this data point was queried by the active learner during the complete training, $a_{BALD}(x, \mathcal{M})$ is the calculated BALD score of the input after 5 acquisition iterations ($|T| = 510$).

close to target corpus. Some previous work has analyzed the performance behavior of the BERT model in different scenarios. Hao et al. (2019) showed that a classifier that fine-tunes a pre-trained BERT model generally has wider optima on the training loss curves in comparison to models trained for the same task from scratch, indicating a more general classifier (Chaudhari et al., 2017; Li et al., 2018; Izmailov et al., 2018). Peters et al. (2019) examine the adaption phase of LM based classifiers by comparing fine-tuning and feature extraction where the LMs parameters are fixed. In contrast, we focus on the low-resource setting where less than 1,000 data points are available for fine-tuning.

Layer freezing in deep Transformers Experiments by Yosinski et al. (2014) indicated that the first layers of a LM capture a more general language understanding, while later layers capture more task-specific knowledge. With this motivation, Howard and Ruder (2018) introduced *gradual unfreezing* of the Transformer layers during each epoch, beginning with the last layer. Hao et al. (2019) analyzed the *loss surfaces* in dependency of the model parameters before and after training and came to the same conclusion that lower layers contain more transferable features. However, none of the work has considered the training set size as a dependent parameter as our experiments presented in this paper.

Active learning in NLP There is some prior work regarding active learning for NLP tasks using deep neural networks. Zhang et al. (2017) explored pool-based active learning for text classification using a model similar to our setting. However, they used word-level embeddings (Mikolov et al., 2013) and focus on representation learning, querying pool points that are expected to maximize the gradient changes in the embedding layer. Shen et al. (2017) used active learning for named entity recognition tasks. They proposed a acquisition strategy named *Maximum Normalized Log-Probability* which is a normalized form of the *Constrained Forward-Backward* confidence estimation (Culotta and McCallum, 2004; Culotta and McCallum, 2005). Using this strategy, they achieved on-par performance in comparison to a model using the BALD acquisition function and MC Dropout without needing multiple forward passes. However, this approach is not suitable for any arbitrary model architecture but requires conditional random fields (CRFs) for the approximation of model uncertainty.

7 Conclusion

In this paper, we evaluated the performance of a pre-trained Transformer model - BERT - in an active learning scenario for text classification in low-resource settings. We showed that using Monte-Carlo Dropout in the classification architecture is an effective way to approximate model uncertainty on unlabeled training elements. This technique enables us to select data for annotation that maximize the knowledge gain for the model fine-tuning process. Experimental results on GLUE data set show that it improves both model performance and training stability. Finally, in order to improve the efficiency of the fine-tuning process with a small amount of data, we explored the reduction of trainable model parameters by freezing layers of the BERT model up to a certain level of depth. Comparing the exclusion of layers in the front or the back of the BERT model from training, we found it to be advantageous for training stability when freezing the layers closest to the output.

Acknowledgments

This research and development project is funded within the "Future of Work" Program by the German Federal Ministry of Education and Research (BMBF) and the European Social Fund in Germany. It is implemented by the Project Management Agency Karlsruhe (PTKA). The authors are responsible for the content of this publication.

References

Rodrigo Agerri, Iñaki San Vicente, Jon Ander Campos, Ander Barrena, Xabier Saralegi, Aitor Soroa, and Eneko Agirre. 2020. Give your text representation models some love: the case for basque. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4781–4788, Marseille, France, May. European Language Resources Association.

Charu C. Aggarwal and ChengXiang Zhai. 2012. A Survey of Text Classification Algorithms. In *Mining Text Data*, pages 163–222. Springer US, Boston, MA.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France, May. European Language Resource Association.

Alexei Baevski, Sergey Edunov, Yinhan Liu, Luke Zettlemoyer, and Michael Auli. 2019. Cloze-driven Pretraining of Self-attention 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 5359–5368, Hong Kong, China, November. Association for Computational Linguistics.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy, July. Association for Computational Linguistics.

Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer T. Chayes, Levent Sagun, and Riccardo Zecchina. 2017. Entropy-SGD: Biasing Gradient Descent Into Wide Valleys. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Xilun Chen, Yu Sun, Ben Athiwaratkun, Claire Cardie, and Kilian Weinberger. 2018. Adversarial Deep Averaging Networks for Cross-Lingual Sentiment Classification. *Transactions of the Association for Computational Linguistics*, 6:557–570.

Aron Culotta and Andrew McCallum. 2004. Confidence Estimation for Information Extraction. In *Proceedings of HLT-NAACL 2004: Short Papers*, pages 109–112, Boston, Massachusetts, USA, May. Association for Computational Linguistics.

Aron Culotta and Andrew McCallum. 2005. Reducing Labeling Effort for Structured Prediction Tasks. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 2*, AAAI'05, pages 746–751. AAAI Press. event-place: Pittsburgh, Pennsylvania.

Andrew M Dai and Quoc V Le. 2015. Semi-supervised Sequence Learning. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3079–3087. Curran Associates, Inc.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Chuong B. Do and Andrew Y. Ng. 2006. Transfer learning for text classification. In Y. Weiss, B. Schölkopf, and J. C. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 299–306. MIT Press.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified Language Model Pre-training for Natural Language Understanding and Generation. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d\textquotesingle Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 13042–13054. Curran Associates, Inc.

Meng Fang and Trevor Cohn. 2017. Model Transfer for Tagging Low-resource Languages using a Bilingual Dictionary. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 587–593, Vancouver, Canada, July. Association for Computational Linguistics.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA, June. PMLR.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 2017. Deep Bayesian Active Learning with Image Data. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1183–1192, International Convention Centre, Sydney, Australia, August. PMLR.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial Training of Neural Networks. *Journal of Machine Learning Research*, 17(1):2096–2030, January.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc.

Rahul Gupta, Saurabh Sahu, Carol Espy-Wilson, and Shrikanth Narayanan. 2018. Semi-Supervised and Transfer Learning Approaches for Low Resource Sentiment Classification. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5109–5113, Calgary, AB, April. IEEE.

Christian Hadiwinoto, Hwee Tou Ng, and Wee Chung Gan. 2019. Improved word sense disambiguation using pre-trained contextualized word representations. 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 5297–5306, Hong Kong, China, November. Association for Computational Linguistics.

Yaru Hao, Li Dong, Furu Wei, and Ke Xu. 2019. Visualizing and Understanding the Effectiveness of BERT. 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 4141–4150, Hong Kong, China, November. Association for Computational Linguistics.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. 2011. Bayesian Active Learning for Classification and Preference Learning. *arXiv:1112.5745 [cs, stat]*, December. arXiv: 1112.5745.

Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia, July. Association for Computational Linguistics.

Luyao Huang, Chi Sun, Xipeng Qiu, and Xuanjing Huang. 2019a. GlossBERT: BERT for word sense disambiguation with gloss knowledge. 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 3509–3514, Hong Kong, China, November. Association for Computational Linguistics.

Weipeng Huang, Xingyi Cheng, Taifeng Wang, and Wei Chu. 2019b. Bert-based multi-head selection for joint entity-relation extraction. In Jie Tang, Min-Yen Kan, Dongyan Zhao, Sujian Li, and Hongying Zan, editors, *Natural Language Processing and Chinese Computing*, pages 713–723, Cham. Springer International Publishing.

Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 448–456. JMLR.org. event-place: Lille, France.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. 2018. Averaging weights leads to wider optima and better generalization. In Ricardo Silva, Amir Globerson, and Amir Globerson, editors, *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, 34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018, pages 876–885. Association For Uncertainty in Artificial Intelligence (AUAI).

Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October. Association for Computational Linguistics.

Tom Kocmi and Ondřej Bojar. 2018. Trivial Transfer Learning for Low-Resource Neural Machine Translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 244–252, Brussels, Belgium, October. Association for Computational Linguistics.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. 2018. Visualizing the Loss Landscape of Neural Nets. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pages 6391–6401, USA. Curran Associates Inc. event-place: Montréal, Canada.

Xin Li, Lidong Bing, Wenxuan Zhang, and Wai Lam. 2019. Exploiting BERT for end-to-end aspect-based sentiment analysis. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 34–41, Hong Kong, China, November. Association for Computational Linguistics.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online, July. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Abdul Moeed, Gerhard Hagerer, Sumit Dugar, Sarthak Gupta, Mainak Ghosh, Hannah Danner, Oliver Mitevski, Andreas Nawroth, and Georg Groh. 2020. An evaluation of progressive neural networksfor transfer learning in natural language processing. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 1376–1381, Marseille, France, May. European Language Resources Association.

Toan Q. Nguyen and David Chiang. 2017. Transfer Learning across Low-Resource, Related Languages for Neural Machine Translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 296–301, Taipei, Taiwan, November. Asian Federation of Natural Language Processing.

S. J. Pan and Q. Yang. 2010. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, October.

Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. 2019. To Tune or Not to Tune? Adapting Pretrained Representations to Diverse Tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 7–14, Florence, Italy, August. Association for Computational Linguistics.

Alec Radford. 2018. Improving Language Understanding by Generative Pre-Training. Technical report, OpenAI.

Nils Reimers, Benjamin Schiller, Tilman Beck, Johannes Daxenberger, Christian Stab, and Iryna Gurevych. 2019. Classification and Clustering of Arguments with Contextualized Word Embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 567–578, Florence, Italy, 07.

Sebastian Ruder, Matthew E. Peters, Swabha Swayamdipta, and Thomas Wolf. 2019. Transfer learning in natural language processing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, pages 15–18, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Mohammad Salameh, Saif Mohammad, and Svetlana Kiritchenko. 2015. Sentiment after Translation: A Case-Study on Arabic Social Media Posts. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 767–777, Denver, Colorado, May. Association for Computational Linguistics.

Yanyao Shen, Hyokun Yun, Zachary Lipton, Yakov Kronrod, and Animashree Anandkumar. 2017. Deep Active Learning for Named Entity Recognition. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 252–256, Vancouver, Canada, August. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Chi Sun, Luyao Huang, and Xipeng Qiu. 2019a. Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 380–385, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019b. How to fine-tune bert for text classification? In Maosong Sun, Xuanjing Huang, Heng Ji, Zhiyuan Liu, and Yang Liu, editors, *Chinese Computational Linguistics*, pages 194–206, Cham. Springer International Publishing.

Songbo Tan and Jin Zhang. 2008. An Empirical Study of Sentiment Analysis for Chinese Documents. *Expert Systems with Applications*, 34(4):2622–2629, May.

Wilson L. Taylor. 1953. "Cloze Procedure": A New Tool for Measuring Readability. *Journalism Quarterly*, 30(4):415–433, September.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Xiaojun Wan. 2008. Using Bilingual Knowledge and Ensemble Techniques for Unsupervised Chinese Sentiment Analysis. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 553–561, Honolulu, Hawaii, October. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, November. Association for Computational Linguistics.

Hu Xu, Bing Liu, Lei Shu, and Philip Yu. 2019. BERT post-training for review reading comprehension and aspect-based sentiment analysis. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2324–2335, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Zhilin Yang, Ruslan Salakhutdinov, and William W. Cohen. 2017. Transfer Learning for Sequence Tagging with Hierarchical Recurrent Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks? In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3320–3328. Curran Associates, Inc.

Ye Zhang, Matthew Lease, and Byron C. Wallace. 2017. Active Discriminative Text Representation Learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pages 3386–3392. AAAI Press. event-place: San Francisco, California, USA.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer Learning for Low-Resource Neural Machine Translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas, November. Association for Computational Linguistics.