CONDITIONALLY ADAPTIVE MULTI-TASK LEARNING: IMPROVING TRANSFER LEARNING IN NLP USING FEWER PARAMETERS & LESS DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-Task Learning (MTL) has emerged as a promising approach for transferring learned knowledge across different tasks. However, MTL must deal with challenges such as: overfitting to low resource tasks, catastrophic forgetting, and negative task transfer, or learning interference. Additionally, in Natural Language Processing (NLP), MTL alone has typically not reached the performance level possible through per-task fine-tuning of pretrained models. However, many fine-tuning approaches are both parameter inefficient, e.g. potentially involving one new model per task, and highly susceptible to losing knowledge acquired during pretraining. We propose a novel transformer based architecture consisting of a new conditional attention mechanism as well as a set of task conditioned modules that facilitate weight sharing. Through this construction we achieve more efficient parameter sharing and mitigate forgetting by keeping half of the weights of a pretrained model fixed. We also use a new multi-task data sampling strategy to mitigate the negative effects of data imbalance across tasks. Using this approach we are able to surpass single-task fine-tuning methods while being parameter and data efficient. With our base model, we attain 2.2% higher performance compared to a full fine-tuned BERT large model on the GLUE benchmark, adding only 5.6% more trained parameters per task (whereas naive fine-tuning potentially adds 100% of the trained parameters per task) and needing only 64.6% of the data. We show that a larger variant of our single multi-task model approach performs competitively across 26 NLP tasks and yields state-of-the-art results on a number of test and development sets. Our code is publicly available[1].

1 INTRODUCTION

The introduction of deep, contextualized Masked Language Models (MLM)[2] trained on massive amounts of unlabeled data has led to significant advances across many different Natural Language Processing (NLP) tasks (Peters et al., 2018; Liu et al., 2019a). Much of these recent advances can be attributed to the now well known BERT approach (Devlin et al., 2018). Substantial improvements over previous state-of-the-art results on the GLUE benchmark[3] (Wang et al., 2018) have been obtained by multiple groups using BERT models with task specific fine-tuning. The "BERT-variant + fine-tuning" formula has continued to improve over time with newer work constantly pushing the state-of-the-art forward on the GLUE benchmark. The use of a single neural architecture for multiple NLP tasks has shown promise long before the current wave of BERT inspired methods (Collobert & Weston, 2008) and recent work has argued that autoregressive language models (ARLMs) trained on large scale datasets – such as the GPT family of models (Radford et al., 2018), are in practice multi-task learners (Brown et al., 2020). However, even with MLMs and ARLMs trained for multi-tasking, single-task fine-tuning is usually also employed to achieve state of the art performance on specific tasks of interest. Typically this fine tuning process may entail: creating a task specific fine tuned model (Devlin et al., 2018), training specialized model components for task specific predictions (Houlsby et al., 2019) or fine-tuning a single multi-task architecture (Liu et al., 2019b).

[1] `https://github.com/CAMTL/CA-MTL`
[2] For reader convenience, all acronyms in this paper are summarized in section A.1 of the Appendix.
[3] https://gluebenchmark.com/tasks

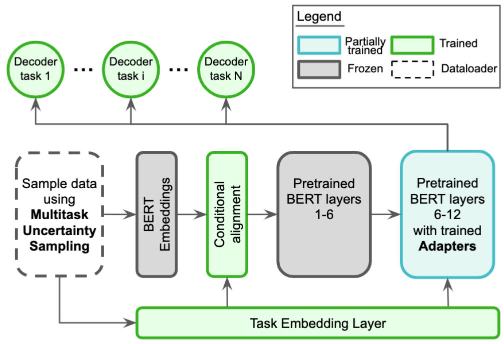


Figure 1: CA-MTL$_{\text{BASE}}$ architecture first uses our uncertainty-based sampling algorithm to choose task data for batching. Then, the input tokens go through a frozen embedding layer, followed by a Conditional Alignment layer. The rest contains frozen BERT-based Transformer layers and trainable adapters.

Single-task fine-tuning over all of the pretrained model's parameters may have other issues. Recent analysis of such MLM have shed light on the linguistic knowledge that is captured in the hidden states and attention maps (Clark et al., 2019b; Tenney et al., 2019a; Merchant et al., 2020). Particularly, BERT has middle Transformer (Vaswani et al., 2017) layers that are typically the most transferable to a downstream task (Liu et al., 2019a). The model proxies the steps of the traditional NLP pipeline in an localizable way (Tenney et al., 2019a) — with basic syntactic information appearing earlier in the network, while high-level semantic information appearing in higher level layers. Since pretraining is usually done on large scale datasets, it may be useful, for a variety of downstream tasks, to conserve that knowledge. However, single-task fine-tuning causes catastrophic forgetting of the knowledge learned during MLM (Howard & Ruder, 2018).

Inspired by the human ability to transfer learned knowledge from one task to another new task, Multi-Task Learning (MTL) in a general sense (Caruana, 1997; Rajpurkar et al., 2016b; Ruder, 2017) has been applied in many fields outside of NLP. Caruana (1993) showed that a model trained in a *multi-task* manner can take advantage of inductive transfer between tasks, achieving a better generalization performance. MTL has the advantage of computational/storage efficiency (Zhang & Yang, 2017), but training models in a multi-task setting is a balancing act; particularly with datasets that have different: **(a)** dataset sizes, **(b)** task difficulty levels, and **(c)** different types of loss functions. In practice, learning multiple tasks at once is challenging since negative transfer (Wang et al., 2019a), task interference (Wu et al., 2020; Yu et al., 2020) and catastrophic forgetting (Serrà et al., 2018) can lead to worse data efficiency, training stability and test performance across tasks compared to single-task fine-tuning.

One of our objectives here is to understand if it is possible to outperform individually fine-tuned BERT-based models using only MTL. Towards that end, we seek to improve pretraining knowledge retention and multi-task inductive knowledge transfer. Our contributions consist of the following:

1. A new *multi-task* Transformer Attention Module using block-diagonal Conditional Attention (section 2.1) that allows the original query-key based attention to account for task specific biases.
2. A new set of modules that adapt a pretrained MLM Transformer to new tasks, facilitate weight sharing in MTL, using:
   - A Conditional Alignment method that aligns the data of diverse tasks and that performs better than its unconditioned and higher capacity predecessor (section 2.2).
   - A Conditional Layer Normalization module that adapts layer normalization statistics to specific tasks (section 2.3) .
   - A Conditional Adapter that facilitates weight sharing and task-specific information flow from lower layers (Section 2.4).
3. A novel way to prioritize tasks with an uncertainty based multi-task data sampling method that helps balance the sampling of tasks during MTL to avoid catastrophic forgetting (see Section 2.5).

Our Conditional Adaptive Multi-Task Learning (CA-MTL) approach is illustrated in Figure 1. To the best of our knowledge our work is the first to explore the use of a latent representation of tasks to modularize and adapt pretrained architectures. Further, we believe our work is also the first to examine uncertainty sampling for large-scale multi-task learning in NLP. We show the efficacy of CA-MTL by: **(a)** testing on 26 different tasks and **(b)** presenting state-of-the-art results on a number of test/development sets. Moreover, we further demonstrate that our method has advantages over **(c)** other adapter networks, and **(d)** other MTL sampling methods.

2 METHODOLOGY

This section is organized according to the two main MTL problems that we will tackle: (1) How to modularize a pretrained network with task latent representations? (2) How to balance different tasks in MTL? We define each task as: $\mathcal{T}_i \triangleq \{p_i(\mathbf{y}_i|\mathbf{x}_i, \mathbf{z}_i), \mathcal{L}_i, \tilde{p}_i(\mathbf{x}_i)\}$, where $\mathbf{z}_i$ is task i's embedding, $\mathcal{L}_i$ is the task loss, and $\tilde{p}_i(\mathbf{x}_i)$ is the empirical distribution of the training data pair $\{\mathbf{x}_i, \mathbf{y}_i\}$, for $i \in \{1, \dots, T\}$ and T the number of supervised tasks. The MTL objective is:

$$\min_{\phi(\mathbf{z}),\theta_1,\dots,\theta_T} \sum_{i=1}^{T} \mathcal{L}_i(f_{\phi(\mathbf{z}_i),\theta_i}(\mathbf{x}_i), \mathbf{y}_i) \tag{1}$$

where f is the predictor function (includes encoder model and decoder heads), $\phi(\mathbf{z})$ are learnable generated weights conditioned on $\mathbf{z}$, and θ_i are task specific parameters for the output decoder heads. We now present *five different modifications and extensions* that we have made to the generic Transformer architecture. In our ablation study of Table 5, we outline the effects of each component by reporting the average GLUE score for various configurations.

2.1 CONDITIONAL ATTENTION

Given d, the input dimensions, the query $\mathbf{Q}$, the key $\mathbf{K}$, and the value $\mathbf{V}$ as defined in Vaswani et al. (2017), we redefine the attention operation:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}, \mathbf{z}_i)) = \text{softmax}\left[\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}} + M(\mathbf{z}_i)\right]\mathbf{V}$$

$$M(\mathbf{z}_i) = \bigoplus_{n=1}^{N} A'_n(\mathbf{z}_i), \quad A'_n(\mathbf{z}_i) = A_n\gamma_i(\mathbf{z}_i) + \beta_i(\mathbf{z}_i),$$

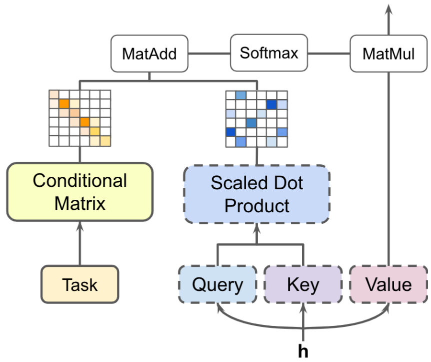


Figure 2: Conditional Matrix $M(\mathbf{z}_i)$ and a Transformer Attention Matrix from the Query/Key dot product are added before being applied to Value. The Conditional Matrix is not dependent on $\mathbf{h}$, the input hidden state, but only on $\mathbf{z}_i$, the task embedding.

where L is the input sequence, N the number of block matrices $A_n \in \mathbb{R}^{(L/N)\times(L/N)}$ along the diagonal of the attention matrix, and $M(\mathbf{z}_i) = \text{diag}(A'_1, \dots, A'_N)$ a block diagonal conditional matrix. While the original attention matrix depends on the hidden states h, $M(\mathbf{z}_i)$ is a learnable weight matrix that only depends on the task embedding $\mathbf{z}_i \in \mathbb{R}^d$. $\gamma_i, \beta_i : \mathbb{R}^d \mapsto \mathbb{R}^{L^2/N^2}$ are Feature Wise Linear Modulation (Perez et al., 2018) functions. We also experimented with full-block Conditional Attention $\in \mathbb{R}^{L\times L}$. Not only did it have N^2 more parameters compared to the block-diagonal variant, but it also performed significantly worse on the GLUE development set. It is possible that GLUE tasks derive a certain benefit from localized attention that is a consequence of $M(\mathbf{z}_i)$. With $M(\mathbf{z}_i)$, each element in a sequence can only attend to other elements in its sub-sequence of length L/N. In our experiments we used $N = d/L$. The full Conditional Attention mechanism used in our experiments is illustrated in Figure 2.

2.2 CONDITIONAL ALIGNMENT

Wu et al. (2020) showed that in MTL having T separate alignment modules $R_1, \dots, R_T$ increases $\text{BERT}_{\text{LARGE}}$ scores on five GLUE tasks by 2.35 (Avg of CoLA, MRPC, QNLI, RTE, SST-2. Individual scores are not reported in Wu et al. (2020)). Inspired by this work, we found that adding a task conditioned alignment layer between the input embedding and the BERT Transformer layers improved multi-task model performance. However, instead of having T separate alignment matrices R_i for each of the T task, one alignment matrix $\hat{R}$ is generated as a function of the task embedding z_i. Inserting $\hat{R}$ into BERT, yields the following encoder function $\hat{f}$:

$$\hat{f} = \sum_{t=1}^{T} g_{\theta_i}(E(\mathbf{x}_i)\hat{R}(\mathbf{z}_i)B), \qquad \hat{R}(\mathbf{z}_i) = R\gamma_i(\mathbf{z}_i) + \beta_i(\mathbf{z}_i) \tag{2}$$

where $\mathbf{x}_i \in \mathbb{R}^d$ is the layer input, g_{θ_i} is the decoder head function for task i with weights θ_i, E the frozen BERT embedding layer, B the BERT Transformer layers and R the linear weight matrix of a single task conditioned alignment matrix. $\gamma_i, \beta_i : \mathbb{R}^d \mapsto \mathbb{R}^d$ are Feature Wise Linear Modulation functions. We tested this module on 5 GLUE tasks and with $\text{BERT}_{\text{LARGE}}$ as in Wu et al. (2020). Enabling task conditioned weight sharing across covariance alignment modules allows us to outperforms $\text{BERT}_{\text{LARGE}}$ by 3.61% average GLUE score.

2.3 Conditional Layer Normalization (CLN)

We extend the Conditional Batch Normalization idea from de Vries et al. (2017) to Layer Normalization (Ba et al., 2016). For task $\mathcal{T}_i$, $i \in \{1, \ldots, T\}$:

$$\mathbf{h}_i = \frac{1}{\sigma} \odot (\mathbf{a}_i - \mu) * \hat{\gamma}_i(\mathbf{z}_i) + \beta_i(\mathbf{z}_i), \qquad \hat{\gamma}_i(\mathbf{z}_i) = \boldsymbol{\gamma}' \gamma_i(\mathbf{z}_i) + \boldsymbol{\beta}' \tag{3}$$

where $\mathbf{h}_i$ is the CLN output vector, $\mathbf{a}_i$ are the preceding layer activations associated with task i, μ and σ are the mean and the variance of the summed inputs within each layer as defined in Ba et al. (2016). Conditional Layer Normalization is initialized with BERT's Layer Normalization affine transformation weights and bias $\boldsymbol{\gamma}'$ and $\boldsymbol{\beta}'$ from the original formulation: $\mathbf{h} = \frac{1}{\sigma} \odot (\mathbf{a} - \mu) * \boldsymbol{\gamma}' + \boldsymbol{\beta}'$. During training, the weight and bias functions of $\gamma_i(*)$ and $\beta_i(*)$ are always trained, while the original Layer Normalization weight may be kept fixed. This module was added to account for task specific re-scaling of individual training cases. Layer Normalization normalizes the inputs across features. The conditioning introduced in equation 2.3 allows us to modulate the normalization's output based on a task's latent representation.

2.4 Conditional Adapters

We created a task conditioned two layer feed-forward neural network (called a Conditional Feed Forward or CFF in Figure 3) with a bottleneck. The conditional bottleneck layer follows the same transformation as in equation 2. The adapter in Figure 3a is placed inside a Transformer layer. The conditional bottleneck layer is also the main building block of the skip connection seen in Figure 3b. This Conditional Adapter allows lower layer information to flow upwards depending on the task. Our intuition for introducing this component is related to recent studies (Tenney et al., 2019a) that showed that the "most important layers for a given task appear at specific positions". As with the other modules described so far, each task's adapters is created from the weights of a single shared adapter that is modulated by the task embedding.

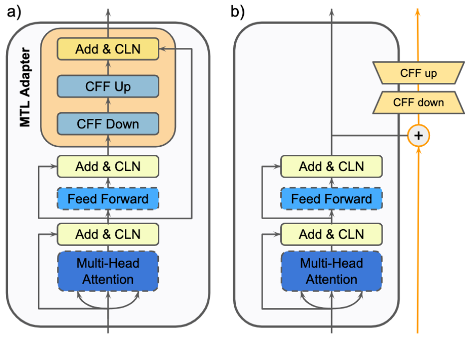


Figure 3: The Conditional Adapter in Figure a) is added to the top most Transformer layer of $\text{CA-MTL}_{\text{BASE}}$ and uses a CLN[1] and a conditional bottleneck. The Conditional Adapter in Figure b) is added along side all Transformer layers in $\text{CA-MTL}_{\text{LARGE}}$. The connection at layer j takes in the matrix sum of the Transformer layer output at j and the previous connection's output at $j-1$.

[1]CFF=Conditional Feed-Forward
CLN=Conditional Layer Norm

2.5 Multi-Task Uncertainty Sampling

MT-Uncertainty Sampling is a task selection strategy that is inspired by Active Learning techniques. Our algorithm 1 in the Appendix, Section A.2. MT-Uncertainty Sampling uses Shannon Entropy, an uncertainty measure, to choose training examples by first doing forward pass through the model with $b \times T$ input samples. For an output classification prediction with C_i possible classes and probabilities $(p_{i,1}, \ldots, p_{i,C_i})$, the Shannon Entropy H_i, for task $\mathcal{T}_i$ and $i \in \{1, \ldots, T\}$, our uncertainty measure $\mathcal{U}(\mathrm{x})$ are given by:

$$H_i = H_i(f_{\phi(\mathbf{z}_i),\theta_i}(\mathrm{x})) = -\sum_{c=1}^{C_i} p_c \log p_c, \qquad \mathcal{U}(x_i) = \frac{H_i(f_{\phi(\mathbf{z}_i),\theta_i}(\mathrm{x}))}{\hat{H} \times H_i'}, \tag{4}$$

$$\hat{H} = \max_{i \in \{1,\ldots,T\}} \bar{H}_i = \max \left[\frac{1}{b} \sum_{\mathrm{x} \in \mathbf{x}_i} H_i \right], \qquad H_i' = -\sum_{c=1}^{C_i} \frac{1}{C_i} \log \left[\frac{1}{C_i} \right], \tag{5}$$

where $\bar{H}_i$ is the average Shannon Entropy across b samples of task t, H_i', the Shannon entropy of choosing classes with uniform distribution and $\hat{H}$, the maximum of each task's average entropy over

b samples. H_i' is normalizing factor that accounts for differing number of prediction classes (Without the normalizing factor H_i', tasks with a binary classification $C_i = 1$ were rarely chosen). Further, to limit high entropy outliers and to favor tasks with highest uncertainty, we normalize with $\hat{H}$. The measure in eq. 4 allows Algorithm 1 to chose b samples from $b \times T$ candidates to train the model.

3 Related Work

Multi-Tasking in NLP To take advantage of the potential positive transfer of knowledge from one task to another, several work have proposed carefully choosing which tasks to train as an intermediate step in NLP before single task fine-tuning (Bingel & Søgaard, 2017; Kerinec et al., 2018; Wang et al., 2019a; Standley et al., 2019; Pruksachatkun et al., 2020). In this work, all tasks are trained *jointly* and evaluated from a *single model*. Such MTL approach has not yet been successful in NLP. For example, on the GLUE benchmark, MTL baseline models perform significantly worse than single task models (Wang et al., 2018). Also, the MTL model in McCann et al. (2018) performs better when each task is trained individually. Recently, approaches in MTL have tackled the problem by designing task specific decoders on top of a shared model (Liu et al., 2019b) or distilling multiple single-task models into one (Clark et al., 2019c). Nonetheless, the final step in the training process always involved single-task finetuning. In this paper, we show that it is possible to achieve high performance without single task fine-tuning or task-specific architecture design.

Adapters There are two common transfer learning methods in NLP. One popular approach is to encode text with pretrained text embeddings and then use a task specific downstream decoder (Subramanian et al., 2018). The other is single task fine-tuning of a pretrained model. Another transfer method has emerged with *Adapters*. Adapters are trainable modules that are attached in specific locations of a pretrained network. This approach is useful with pretrained MLM models that have rich linguistic information (Tenney et al., 2019b; Clark et al., 2019b; Liu et al., 2019a; Tenney et al., 2019a). Recently, both Houlsby et al. (2019) and Stickland et al. (2019) added an adapters to a pretrained BERT model by fine-tuning the layer norms and adding feed forward bottlenecks in every Transformer layer. However, such methods adapted each task individually during the fine-tuning process. Unlike prior work, our method harnesses the vectorized representations of tasks to modularize a single pretrained model across all tasks. We use five new architectural components that modify the Transformer. We show in the results section that our approach outperforms other adapters.

Active Learning, Task Selection and Sampling Our sampling technique is similar to the ones found in several active learning algorithms (Chen et al., 2006) that are based on Shannon entropy estimations. Reichart et al. (2008) and Ikhwantri et al. (2018) examined Multi-Task Active Learning (MTAL) using a two task annotation scenario and showed performance gains while needing less labelled data. Our approach is a substantially different variant of MTAL since it was developed for task selection. Instead of choosing one informative sample for T different learners (or models) for each T tasks, we choose T tasks samples for *one model* to learn all tasks. Our algorithm differs in three ways: a) we use uncertainty sampling to maximize large scale MTL ($\gg 2$ tasks) performance via the modularization of a shared neural architecture; b) the algorithm weights each sample by the corresponding task average score; c) the Shannon entropy is normalized to account for different types of losses (see equation 5). Recently, Glover & Hokamp (2019) explored task selection in MTL using learning policies based on counterfactual estimations (Charles et al., 2013). However, such method considers only fixed stochastic parameterised policies while our method *adapts* its selection criterion based on model uncertainty throughout the training process. In the results section below, we demonstrate that our Uncertainty MT-Sampling scheme outperforms other task selection strategies, and we are able to achieve this with 35% less data.

4 Experiments

We show that the five parameter sharing modules of section 2 achieve parameter efficient transfer for 26 NLP tasks. On the GLUE test benchmark, CA-MTL$_{\text{BASE}}$ outperforms a fully fine-tuned single task BERT$_{\text{BASE}}$ model by 2.8%. Similarly, CA-MTL$_{\text{LARGE}}$ outperforms a fully fine-tuned single task BERT$_{\text{LARGE}}$ model by 0.9%. In both cases, CA-MTL only adds 5.6% of the number of parameters trained by fine-tuning. We demonstrate that our MT-Uncertainty sampling strategy performs better than competing methods and increases the performance of *multi-task* models.

Our implementation of CA-MTL is based on the PyTorch implementation of BERT from HuggingFace. We used Adam (Kingma & Ba, 2015) as the optimizer with a learning rate of 2e-5. We applied a learning rate decay with warm up over the first 10% of the training steps. Unless otherwise specified, we used a batch size of 32, a maximum epoch of 5, a seed of 12 following Dodge et al. (2020) and a maximum sequence length of 128. Our data prepossessing and linear decoder heads are the same as in Devlin et al. (2018). We used the same dropout rate of 0.1 in all layers. To run our experiments, we used either four NVIDIA P100 GPU for base models or four NVIDIA V100 GPU for larger ones. We did not perform parameter search. We do not use ensemble of models or task-specific tricks (Devlin et al., 2018; Liu et al., 2019b; Clark et al., 2019c). All models are either 12 layers for base and 24 layers for large. To preserve the weights of the pretrained model as much as possible, we froze the bottom half of the layers in all experiments. We also tested freezing all layers and freezing no layers but found that freezing half the layers worked best.

4.1 Multi-Task Uncertainty Sampling

Our MT-Uncertainty sampling strategy, from section 2.5, is compared to 3 other task selection schemes: a) Counterfactual b) Task size c) Random. We used a $\text{BERT}_{\text{BASE}}$ on 200k iterations and with the same hyperparameters as in Glover & Hokamp (2019). For more information on Counterfactual task selection, we invite the reader to consult the full explanation in Glover & Hokamp (2019). For T tasks and the dataset D_i for tasks $i \in \{1, \ldots, T\}$, we rewrite the definitions of Random π_{rand} and Task size $\pi_{|task|}$ sampling:

$$\pi_{rand} = 1/T, \quad \pi_{|task|} = |D_i| \left[\sum_{i=1}^{T} |D_i| \right]^{-1} \tag{6}$$

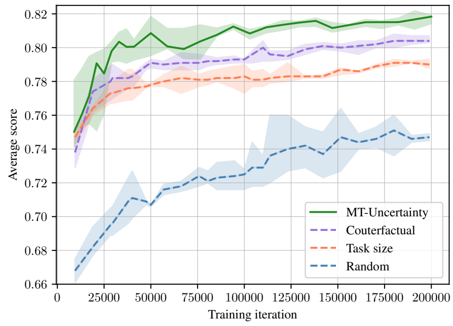


Figure 4: Avg dev set scores on 8 GLUE tasks (excl. WNLI) for each sampling strategy and using $\text{BERT}_{\text{BASE}}$. Each line is the median over 3 seeds. CA-MTL converges faster than other methods. Data for the Counterfactual and Task Size policy $\pi_{|task|}$ (eq. 6) were drawn from Glover & Hokamp (2019).

Our experimental results are presented in Figure 4. We see from the results that MT-Uncertainty converges by reaching the 80% average GLUE score line before other methods. Further, MT-Uncertainty maximum score on 200k iterations is at 82.2, which is 1.7 percentage points higher than Counterfactual sampling. Additional analysis of our MTL sampling method is left to the Appendix. We provide evidence in Figure 6 of Section A.2 that MT-Uncertainty is able to manage task difficulty — by choosing the most difficult tasks first. We also validated that MT-Uncertainty is able to avoid catastrophic forgetting in Figure 7 of Section A.4 — by sampling tasks before performance drops.

4.2 Transfer to New Tasks

Table 1: Domain adaptation results on dev. sets for BASE models. Results from: [1]Liu et al. (2019b), [2]Jiang et al. (2020)

% data used	SciTail				SNLI			
	0.1%	1%	10%	100%	0.1%	1%	10%	100%
$\text{BERT}_{\text{BASE}}$[1]	51.2	82.2	90.5	94.3	52.5	78.1	86.7	91.0
MT-DNN[1]	81.9	88.3	91.1	95.7	81.9	88.3	91.1	95.7
$\text{MT-DNN}_{\text{SMART}}$[2]	82.3	88.6	91.3	**96.1**	82.7	86.0	**88.7**	**91.6**
$\text{CA-MTL}_{\text{BERT}}$	**83.2**	**88.7**	**91.4**	95.6	**82.8**	**86.2**	88.0	91.5

In Table 1 we examine the ability of our method to quickly adapt to new tasks. We performed domain adaptation on SciTail (Khot et al., 2018) and SNLI (Bowman et al., 2015) datasets using a $\text{CA-MTL}_{\text{BASE}}$ model trained on GLUE and a new linear decoder head. We tested several pretrained and randomly initialized task embeddings (The complete set of experiments with all task embeddings can be found in the Appendix, Section A.5) in a zero-shot setting. We then selected the best task embedding for our results in Table 1. STS-B and MRPC MTL-trained task embeddings performed best on SciTail and SNLI respectively. $\text{CA-MTL}_{\text{BERT-BASE}}$ has faster adaption than $\text{MT-DNN}_{\text{SMART}}$ (Jiang et al., 2020) as evidenced by higher performances in low resource regimes (0.1% and 1% of the data). When trained on the complete dataset, $\text{CA-MTL}_{\text{BERT-BASE}}$ is on par with $\text{MT-DNN}_{\text{SMART}}$. Unlike $\text{MT-DNN}_{\text{SMART}}$ however, we use a standard optimizer and we do not add context from a semantic similarity model – $\text{MT-DNN}_{\text{SMART}}$ is built off HNN (He et al., 2019). Nonetheless, with a larger model, CA-MTL surpasses $\text{MT-DNN}_{\text{SMART}}$ on the full SNLI and SciTail datasets in Table 4.

4.3 JOINTLY TRAINING ON 24 TASKS: GLUE/SUPER-GLUE, MRQA AND WNUT2017

In Table 2 we evaluated the performance of CA-MTL against single task fine-tuned models as well as the latest BERT-based adapters. Our results indicate that CA-MTL outperforms other adapters on both development and test sets. Furthermore, CA-MTL$_{\text{RoBERTa-BASE}}$ attains a score of 85.65 on the development set score. This score is 2% higher than a single task BERT$_{\text{LARGE}}$ model that has almost twice the number of parameters. Single task fine tuning methods in Table 2 needs $9\times$ BERT based models, one for each tasks. As in Houlsby et al. (2019), MNLI$_{\text{m}}$ and MNLI$_{\text{mm}}$ are treated as separate tasks. Since our approach is based on reusing weights for multiple tasks, CA-MTL requires only $1.12\times$ the number of parameters, which is even smaller than $1.13\times$ for the adapter proposed in Stickland et al. (2019).

Table 2: GLUE development set and evaluation server test results. F1 scores are reported for QQP and MRPC, Spearman's correlation for STS-B, accuracy on the matched/mismatch sets for MNLI, Matthew's correlation for CoLA and accuracy for other tasks. * Individual scores not available.
Results from: [1]Liu et al. (2019b) [2]Stickland et al. (2019). [3]Devlin et al. (2018) [4]Houlsby et al. (2019) .

Method	Total num params	Trained params / task	GLUE CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	Avg
Development set results											
Single Task BERT$_{\text{LARGE}}$ [1]	9.0×	100%	61.8	86.3/86.2	87.6	90.5	89.5	71.1	93.5	89.4	84.01
PALs+Anneal Samp. [3]	1.13×	12.5%	–	–	–	–	–	–	–	–	81.70*
CA-MTL$_{\text{BERT-BASE}}$ (ours)	1.12×	5.6%	60.9	82.7/83.1	88.9	90.7	90.3	79.1	91.9	88.8	**84.03**
CA-MTL$_{\text{RoBERTa-BASE}}$ (ours)	1.12×	5.6%	59.8	86.4/86.4	92.3	92.0	87.7	83.1	93.9	89.3	**85.65**
Base Models — Test Server Results											
Single Task BERT$_{\text{BASE}}$ [3]	9.0×	100%	52.1	84.6/83.4	88.9	90.5	71.2	66.4	93.5	85.8	79.6
PALs+Anneal Samp. [2]	1.13×	12.5%	51.2	84.3/83.5	88.7	90.0	71.5	76.0	92.6	85.8	80.4
CA-MTL$_{\text{BERT-BASE}}$ (ours)	1.12×	5.6 %	53.1	85.9/85.8	88.6	90.5	69.2	76.4	93.2	85.3	**80.9**
CA-MTL$_{\text{ROBERTA-BASE}}$ (ours)	1.12×	5.6%	53.1	86.2/85.7	89.7	91.7	69.9	76.9	95.0	87.7	**81.8**
Large Models — Test Server Results											
Single Task BERT$_{\text{LARGE}}$ [1]	9.0×	100%	60.5	86.7/85.9	89.3	92.7	72.1	70.1	94.9	86.5	82.1
Adapters-256 [4]	1.3×	3.6%	59.5	84.9/85.1	89.5	90.7	71.8	71.5	94.0	86.9	80.0
Adapters-64 [4]	1.2×	2.1%	56.9	85.3/84.6	89.6	91.4	71.8	68.8	94.2	87.3	79.6
CA-MTL$_{\text{BERT-LARGE}}$ (ours)	1.12×	5.6%	59.5	85.9/85.4	89.3	92.6	71.4	79.0	94.7	87.7	**82.8**

24-task MTL. We *jointly* trained a CA-MTL$_{\text{RoBERTa-LARGE}}$ model on 9 GLUE tasks, 8 Super-GLUE tasks (Wang et al., 2019b), 6 MRQA tasks (Fisch et al., 2019) , and on WNUT2017 (Derczynski et al., 2017). Benchmarks, datasets and their corresponding tasks are described in section A.3 of the Appendix. For MRQA tasks, we increased the input sequence length to 256. We trained the model on 8 epochs. Our main point of comparison in Table 3(a) is the T5 (Raffel et al., 2019b) baseline model which uses close to 5 × more parameters per task and has a sequence length of 512. Our model outperforms T5 by 6 percentage points on the Super-GLUE development set. Our main point of comparison in Table 3(b) is SpanBERT (Joshi et al., 2019) which uses close to 7 × more parameters per task and has a sequence length of 512. Our MRQA results show that our approach is above SpanBERT performance on 3 out 6 tasks and competitive overall. For the Named Entity Recognition task, we set a new SOTA result on WNUT2017 with a 58.0 test score (see section A.6 and table 12).

Table 3: Performance on various tasks. ST = single task fine-tuning, A = trained with adapter modules. Results from: [1]Pfeiffer et al. (2020) [2]baseline in Raffel et al. (2019a) [3]Joshi et al. (2019)

(a) Super-GLUE Task Dev set	Fusion ST-A[1]	T5 ST[2]	T5 ST-A$_{2048}$[2]	**CA-MTL**
Params per task	—	220M	115M	47M
BoolQ	76.2	76.6	74.5	**86.0**
CB	92.1	91.2	88.0	**95.1**
COPA	—	66.2	58.0	**89.0**
MultiRC	—	66.1	61.1	**67.4**
ReCoRD	—	**69.0**	66.7	68.7
RTE	—	76.3	76.6	**91.0**
WSC	—	**78.6**	76.0	70.2
WiC	—	68.0	71.1	**72.3**
Super-GLUE$_{\text{avg}}$	—	74.0	71.5	**80.0**

(b) MRQA Task Test set	T5 ST[2]	Span-BERT[3]	**CA-MTL**
Params per task	220M	340M	47M
HotpotQA	—	**83.0**	81.1
Natural Questions	—	82.5	**82.7**
NewsQA	—	73.6	**73.7**
TriviaQA	—	**83.6**	81.0
SearchQA	—	**84.8**	83.2
SQuAD v2	**88.8**	88.7	**88.8**
MRQA$_{\text{avg}}$	—	**82.7**	81.8

In Table 4 (left) we present results from the GLUE benchmark test server, we compare our 24-task CA-MTL$_{\text{RoBERTa-LARGE}}$ against other models that do not use assemble methods. We see that our approach has greater overall performance when compared to MT-DNN and SpanBERT. However, the

Table 4: Our 24-task MTL model vs. other large models on GLUE, SNLI and SciTail. Results from leaderboards. References: [1]Liu et al. (2019b) [2]baseline in Joshi et al. (2019) [3]Raffel et al. (2019b) [4]Jiang et al. (2020) [5]Zhang et al. (2019) [6]Liu et al. (2020). * Some scores not in leaderboard.

Methods without assemble/tricks	GLUE — Test Server Results								
	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	Avg
MT-DNN[1]	62.5	86.7/86.0	91.1	93.1	72.7	81.4	95.6	88.8	84.2
SpanBERT[2]	64.3	88.1/87.7	90.9	94.3	71.9	79.0	94.8	89.1	84.5
$T5_{LARGE}$[3]	61.2	89.9/89.6	92.4	94.8	73.9	87.2	96.3	89.2	**86.1**
CA-MTL (ours)	62.2	89.0/88.4	92.0	94.7	72.3	86.2	96.3	89.8	85.7

SOTA Methods	SNLI test	SciTail test
MT-DNN[4]	91.6	94.1
$\text{MT-DNN}_{\text{SMART}}$[4]	91.7	95.2
SemBERT[5]*	91.9	—
ALUM[6]*	—	**96.8**
CA-MTL (ours)	**92.1**	**96.8**

single-task fine-tuning approach of $T5_{LARGE}$ surpasses our method by 0.4 percentage points. Finally, as shown in Table 4 (right) we reach a new state of the art result on the SNLI and SciTail test sets. On the SNLI test set, CA-MTL performs 0.2 percentage points higher than SemBERT (Zhang et al., 2019) which incorporates structured semantic information in its input.

4.4 Ablation Analysis and the Effects of Scaling Task Count

In Table 5, we present the results of an ablation study to determine which elements of $\text{CA-MTL}_{\text{BERT-BASE}}$ had the largest positive gain on average GLUE scores. Each component of CA-MTL not only lifted overall performance, it also decreased variations in scores across tasks by 30%. Moreover, training with only 64% of the data, MT-Uncertainty sampling provides a 1% increase in average GLUE scores. The largest gain in performance and the biggest drop in variance comes from the Conditional Attention module, which suggests the importance of task specific attention maps when working with Transformer models in MTL. Compared to a multi-task $BERT_{BASE}$ model with Random task sampling π_{rand}, this result provides evidence that tasks are better able to share weights when an architecture is modularized with learned task embeddings.

Table 5: Model ablation study[a] on the GLUE development set. Each CA-MTL component increases average GLUE scores and reduces MTL performance variance.

Model changes	Avg GLUE	σ GLUE	% data used
$BERT_{BASE}$ with Random Samp.	80.61	14.41	100
+ Conditional Attention	82.41	10.67	100
+ Conditional Adapter	82.90	11.27	100
+ CA and CLN	83.12	10.91	100
+ MT-Uncertainty (CA-MTL)	84.03	**10.02**	66.3
$\text{CA-MTL}_{\text{RoBERTa-BASE}}$	**85.65**	10.69	**64.6**

[a]MT=multi-task, CA=Conditional Alignment, CLN=Conditional Layer Normalization, σ=scores standard deviation across tasks.

In Figure 5 we further investigate the effects of our task conditioned adaptive modules and MT-Uncertainty sampling on task interference by measuring GLUE average scores when progressively adding additional tasks. The results show that adding 23 tasks drops the performance of a multi-task $BERT_{BASE}$ model with Random task sampling π_{rand}. Without CA-MTL, adding SuperGLUE makes a vanilla BERT model drop by 1.73 percentage points on average GLUE scores. The opposite is true when the task conditioned modules of CA-MTL are integrated. The Conditional modules play an important role in weight sharing across tasks by limiting task interference. MT-Uncertainty sampling provides even better gains in a large scale MTL setting as seen in Table 5. Indeed, MT-Uncertainty provides a close to 4 percentage point rise on avg. GLUE scores on 23 tasks.

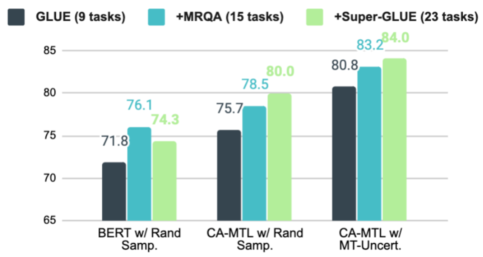


Figure 5: Effects of adding more datasets on avg GLUE scores. Experiments conducted on 3 epochs. When 23 tasks are trained jointly, performance of $\text{CA-MTL}_{\text{BERT-BASE}}$ continues to improve.

5 Conclusion

In large scale multi-task experiments, CA-MTL attains improvements over fully-tuned single task models while adding only 5.6% more parameters and using only 64% of the data. Our task conditioned adaptive modules and sampling scheme are extensible and adapt better than other methods to new tasks with small amounts of data. CA-MTL surpasses current state of the art results on a variety of test and development sets thus highlighting the effectiveness of our approach.

REFERENCES

Gustavo Aguilar, Suraj Maharjan, Adrián Pastor López-Monroy, and Thamar Solorio. A multi-task approach for named entity recognition in social media data. *CoRR*, abs/1906.04135, 2019. URL http://arxiv.org/abs/1906.04135.

Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016. URL http://arxiv.org/abs/1607.06450.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009.

Joachim Bingel and Anders Søgaard. Identifying beneficial task relations for multi-task learning in deep neural networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 164–169, Valencia, Spain, April 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/E17-2026.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2015.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv*, pp. arXiv–2005, 2020.

Rich Caruana. Multitask learning. *Mach. Learn.*, 28(1):41–75, July 1997. ISSN 0885-6125. doi: 10.1023/A:1007379606734. URL https://doi.org/10.1023/A:1007379606734.

Richard Caruana. Multitask learning: A knowledge-based source of inductive bias. In *Proceedings of the Tenth International Conference on Machine Learning*, pp. 41–48. Morgan Kaufmann, 1993.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 1–14, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/S17-2001. URL https://www.aclweb.org/anthology/S17-2001.

Denis Charles, Max Chickering, and Patrice Simard. Counterfactual reasoning and learning systems: The example of computational advertising. *Journal of Machine Learning Research*, 14:3207–3260, November 2013.

Jinying Chen, Andrew Schein, Lyle Ungar, and Martha Palmer. An empirical study of the behavior of active learning for word sense disambiguation. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pp. 120–127, New York City, USA, June 2006. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N06-1016.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. 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)*, pp. 2924–2936, Minneapolis, Minnesota, June 2019a. Association for Computational Linguistics. doi: 10.18653/v1/N19-1300. URL https://www.aclweb.org/anthology/N19-1300.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? an analysis of bert's attention. *CoRR*, abs/1906.04341, 2019b. URL http://arxiv.org/abs/1906.04341.

Kevin Clark, Minh-Thang Luong, Urvashi Khandelwal, Christopher D. Manning, and Quoc V. Le. Bam! born-again multi-task networks for natural language understanding. *CoRR*, abs/1907.04829, 2019c. URL http://arxiv.org/abs/1907.04829.

Edward Collins, Nikolai Rozanov, and Bingbing Zhang. Evolutionary data measures: Understanding the difficulty of text classification tasks. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pp. 380–391, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/K18-1037. URL https://www.aclweb.org/anthology/K18-1037.

Ronan Collobert and Jason Weston. A unified architecture for natural language processing: deep neural networks with multitask learning. In *ICML*, pp. 160–167, 2008. URL https://doi.org/10.1145/1390156.1390177.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, July 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.acl-main.747.

Marie-Catherine de Marneffe, M. Simons, and J. Tonhauser. The commitmentbank: Investigating projection in naturally occurring discourse. 2019.

Harm de Vries, Florian Strub, Jeremie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C Courville. Modulating early visual processing by language. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 30*, pp. 6594–6604. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7237-modulating-early-visual-processing-by-language.pdf.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. Results of the WNUT2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pp. 140–147, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4418. URL https://www.aclweb.org/anthology/W17-4418.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL http://arxiv.org/abs/1810.04805.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*, 2020.

William B. Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005. URL https://www.aclweb.org/anthology/I05-5002.

Matthew Dunn, Levent Sagun, Mike Higgins, V. Ugur Güney, Volkan Cirik, and Kyunghyun Cho. Searchqa: A new q&a dataset augmented with context from a search engine. *CoRR*, abs/1704.05179, 2017. URL http://arxiv.org/abs/1704.05179.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. Mrqa 2019 shared task: Evaluating generalization in reading comprehension, 2019.

John Glover and Chris Hokamp. Task selection policies for multitask learning. *CoRR*, 2019. URL http://arxiv.org/abs/1907.06214.

Andrew Gordon, Zornitsa Kozareva, and Melissa Roemmele. SemEval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pp. 394–398, Montréal, Canada, 7-8 June 2012. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/S12-1052.

Michelle Guo, Albert Haque, De-An Huang, Serena Yeung, and Li Fei-Fei. Dynamic task prioritization for multitask learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

Pengcheng He, Xiaodong Liu, Weizhu Chen, and Jianfeng Gao. A hybrid neural network model for commonsense reasoning. In *Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing*, pp. 13–21, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-6002. URL https://www.aclweb.org/anthology/D19-6002.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. *CoRR*, abs/1902.00751, 2019. URL http://arxiv.org/abs/1902.00751.

Jeremy Howard and Sebastian Ruder. 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)*, pp. 328–339, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1031. URL https://www.aclweb.org/anthology/P18-1031.

Fariz Ikhwantri, Samuel Louvan, Kemal Kurniawan, Bagas Abisena, Valdi Rachman, Alfan Farizki Wicaksono, and Rahmad Mahendra. Multi-task active learning for neural semantic role labeling on low resource conversational corpus. In *Proceedings of the Workshop on Deep Learning Approaches for Low-Resource NLP*, pp. 43–50, 2018.

Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. SMART: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2177–2190, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.197. URL https://www.aclweb.org/anthology/2020.acl-main.197.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://www.aclweb.org/anthology/P17-1147.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. Spanbert: Improving pre-training by representing and predicting spans. *CoRR*, abs/1907.10529, 2019. URL http://arxiv.org/abs/1907.10529.

Emma Kerinec, Chloé Braud, and Anders Søgaard. When does deep multi-task learning work for loosely related document classification tasks? In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 1–8, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5401. URL https://www.aclweb.org/anthology/W18-5401.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 252–262, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1023. URL https://www.aclweb.org/anthology/N18-1023.

Tushar Khot, A. Sabharwal, and Peter Clark. Scitail: A textual entailment dataset from science question answering. In *AAAI*, 2018.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*, 2019.

Hector J. Levesque. The winograd schema challenge. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*. AAAI, 2011. URL http://dblp.uni-trier.de/db/conf/aaaiss/aaaiss2011-6.html#Levesque11.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. Linguistic knowledge and transferability of contextual representations. *CoRR*, abs/1903.08855, 2019a. URL http://arxiv.org/abs/1903.08855.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-task deep neural networks for natural language understanding. *CoRR*, abs/1901.11504, 2019b. URL http://arxiv.org/abs/1901.11504.

Xiaodong Liu, Hao Cheng, Pengcheng He, Weizhu Chen, Yu Wang, Hoifung Poon, and Jianfeng Gao. Adversarial training for large neural language models, 2020.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019c. URL http://arxiv.org/abs/1907.11692.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*, 2018.

Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. What happens to bert embeddings during fine-tuning? *arXiv preprint arXiv:2004.14448*, 2020.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. Bertweet: A pre-trained language model for english tweets. *arXiv preprint arXiv:2005.10200*, 2020.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI*, 2018.

Matthew E. Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. Dissecting contextual word embeddings: Architecture and representation. *CoRR*, abs/1808.08949, 2018. URL http://arxiv.org/abs/1808.08949.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. Adapterfusion: Non-destructive task composition for transfer learning, 2020.

Adam Poliak, Aparajita Haldar, Rachel Rudinger, J. Edward Hu, Ellie Pavlick, Aaron Steven White, and Benjamin Van Durme. Collecting diverse natural language inference problems for sentence representation evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 67–81, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1007. URL https://www.aclweb.org/anthology/D18-1007.

Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel R Bowman. Intermediate-task transfer learning with pretrained models for natural language understanding: When and why does it work? *arXiv preprint arXiv:2005.00628*, 2020.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2019a.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2019b.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, Texas, November 2016a. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL `https://www.aclweb.org/anthology/D16-1264`.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, Texas, November 2016b. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264.

Roi Reichart, Katrin Tomanek, Udo Hahn, and Ari Rappoport. Multi-task active learning for linguistic annotations. In *Proceedings of ACL-08: HLT*, pp. 861–869, 2008.

Sebastian Ruder. An overview of multi-task learning in deep neural networks. *ArXiv*, abs/1706.05098, 2017.

Joan Serrà, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *ICML*, pp. 4555–4564, 2018. URL `http://proceedings.mlr.press/v80/serra18a.html`.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/D13-1170`.

Trevor Standley, Amir Roshan Zamir, Dawn Chen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? *CoRR*, abs/1905.07553, 2019. URL `http://arxiv.org/abs/1905.07553`.

Asa Cooper Stickland, Iain Murray, someone, and someone. BERT and PALs: Projected attention layers for efficient adaptation in multi-task learning. volume 97 of *Proceedings of Machine Learning Research*, pp. 5986–5995, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL `http://proceedings.mlr.press/v97/stickland19a.html`.

Sandeep Subramanian, Adam Trischler, Yoshua Bengio, and Christopher J. Pal. Learning general purpose distributed sentence representations via large scale multi-task learning. *CoRR*, abs/1804.00079, 2018. URL `http://arxiv.org/abs/1804.00079`.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. *CoRR*, abs/1905.05950, 2019a. URL `http://arxiv.org/abs/1905.05950`.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. What do you learn from context? probing for sentence structure in contextualized word representations. *CoRR*, abs/1905.06316, 2019b. URL `http://arxiv.org/abs/1905.06316`.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pp. 191–200, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-2623. URL `https://www.aclweb.org/anthology/W17-2623`.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL `http://arxiv.org/abs/1706.03762`.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 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*, pp. 353–355, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5446. URL `https://www.aclweb.org/anthology/W18-5446`.

Alex Wang, Jan Hula, Patrick Xia, Raghavendra Pappagari, R. Thomas McCoy, Roma Patel, Najoung Kim, Ian Tenney, Yinghui Huang, Katherin Yu, Shuning Jin, Berlin Chen, Benjamin Van Durme, Edouard Grave, Ellie Pavlick, and Samuel R. Bowman. Can you tell me how to get past sesame street? sentence-level pretraining beyond language modeling. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019a.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *CoRR*, abs/1905.00537, 2019b. URL http://arxiv.org/abs/1905.00537.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. Neural network acceptability judgments. *CoRR*, abs/1805.12471, 2018. URL http://arxiv.org/abs/1805.12471.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL https://www.aclweb.org/anthology/N18-1101.

Sen Wu, Hongyang R. Zhang, and Christopher Ré. Understanding and improving information transfer in multi-task learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SylzhkBtDB.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://www.aclweb.org/anthology/D18-1259.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *arXiv preprint arXiv:2001.06782*, 2020.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. Record: Bridging the gap between human and machine commonsense reading comprehension. *CoRR*, abs/1810.12885, 2018. URL http://arxiv.org/abs/1810.12885.

Yu Zhang and Qiang Yang. A survey on multi-task learning. *CoRR*, abs/1707.08114, 2017. URL http://arxiv.org/abs/1707.08114.

Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. Semantics-aware bert for language understanding. *arXiv preprint arXiv:1909.02209*, 2019.

Joey Tianyi Zhou, Hao Zhang, Di Jin, Hongyuan Zhu, Meng Fang, Rick Siow Mong Goh, and Kenneth Kwok. Dual adversarial neural transfer for low-resource named entity recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, July 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P19-1336.

A Appendix

A.1 Summary of Acronyms

Acronyms of datasets and descriptions can be found below in section A.3.

Table 6: List of acronyms used in this paper.

Acronym	Description
ARLM	Autoregressive Language Models
CA-MTL	Conditional Adaptive Multi-Task Learning: our architecture
CFF	Conditional Feed-Forward: a feed forward layer modulated by a conditioning vector
CLN	Conditional Layer Normalization in section 2.3
EDM	Evolutionary Data Measures (Collins et al., 2018): a task difficulty estimate
GLUE	General Language Understanding Evaluation Wang et al. (2018): a benchmark with multiple datasets
QA	Question Answering
MT	Multi-Task
MTAL	Multi-Task Active Learning: finding the most informative instance for multiple learners (or models)
MLM	Masked Language Model: BERT Devlin et al. (2018) is an example of an MLM
MTL	Multi-Task Learning: "learning tasks in parallel while using a shared representation" (Caruana, 1997)
MRQA	Machine Reading for Question Answering Fisch et al. (2019): a benchmark with multiple datasets
NLP	Natural Language Processing
SOTA	State of the art
ST	Single Task finetuning: all weights are typically updated
ST-A	ST with Adapter modules: one adapter per task is trained and pretrained weights are optionally updated

A.2 Uncertainty Sampling: Algorithm and Additional Results

Algorithm 1: Multi-task Uncertainty Sampling

Input: Training data D_t for task $t \in [1, \dots, T]$; batch size b; C_t possible output classes for task t; $f := f_{\phi(\mathbf{z}_i), \theta_i}$ our model with weights ϕ, θ_i;

Output: $\mathcal{B}'$ - multi-task batch of size b

$\mathcal{B} \leftarrow \emptyset$
for $t \leftarrow 1$ **to** T **do**
  Generate $\mathbf{x}_t := \{x_{t,1}, \dots, x_{t,b}\} \overset{\text{i.i.d.}}{\sim} D_t$
  for $i \leftarrow 1$ **to** b **do**
    $\mathcal{H}_{t,i} \leftarrow - \sum_{c=1}^{C_i} p_c(f(x_{t,i})) \log p_c(f(x_{t,i}))$ ▷ Entropy of each sample
  end
  Compute $\bar{\mathcal{H}}_t \leftarrow \frac{1}{b} \sum_{x \in \mathbf{x}_i} \mathcal{H}_{t,i}$ ▷ Average entropy for task t
  Compute $H'_t \leftarrow - \sum_{c=1}^{C_t} \frac{1}{C_t} \log \left[\frac{1}{C_t}\right]$ ▷ Max entropy (uniform distribution)
  $\mathcal{B} \leftarrow \mathcal{B} \cup \mathbf{x}_t$ and $D_t \leftarrow D_t \setminus \mathbf{x}_t$
  if $D_t = \emptyset$ **then**
    Reload D_t
  end
  for $i \leftarrow 1$ **to** b **do**
    Compute: $\mathcal{U}_{t,i} \leftarrow \mathcal{H}_{t,i} / H'_t$ ▷ Uncertainty normalized with max entropy
  end
end
Compute $\hat{\mathcal{H}} \leftarrow \max_{i \in \{1,\dots,T\}} [\bar{\mathcal{H}}_t]$ ▷ Entropy of task with highest average entropy
Update $\mathcal{U}_{t,i} \leftarrow \mathcal{U}_{t,i} / \hat{\mathcal{H}}$ ▷ Normalize each sample's uncertainty measure
$\mathcal{B}' \leftarrow \text{top\_b}(\{\mathcal{U}_{t,i} | t \in [1, \dots, T], i \in [1, \dots, b]\})$ ▷ b samples w/ highest uncertainty

Return: With $\mathcal{B}'$, solve eq. 1 with gradient descent; updated model f

An advantage of our MT-Uncertainty Sampling approach is its ability to manage task difficulty. This is highlighted in Figure 6. In this experiment, we estimated task difficulty using the Evolutionary

Data Measures (EDM)[4] proposed by Collins et al. (2018). The task difficulty estimate relies on multiple dataset statistics such as the data size, class diversity, class balance and class interference. Interestingly, estimated task difficulty correlates with the first instance that the selection of a specific task occurs. Supposing that QNLI is an outlier, we notice that peaks in the data occur whenever tasks are first selected by MT Uncertainty sampling. This process follows the following order: 1. MNLI 2. CoLA 3. RTE 4. QQP 5. MRPC 6.SST-2, which is the order from highest task difficulty to lowest task difficulty using EDM. As opposed to Curriculum Learning (Bengio et al., 2009), MT-Uncertainty dynamically prioritizes the most difficult tasks. As also discovered in MTL vision work (Guo et al., 2018), this type of prioritization on more difficult tasks may explain MT-Uncertainty's improved performance over other task selection methods.

While the EDM difficulty measure, is shown to correlate well with model performance, it lacks precision. As reported in Collins et al. (2018), the average score achieved on the Yahoo Answers dataset is 69.9% and its difficulty is 4.51. The average score achieved on Yelp Full is 56.8%, 13.1% less than Yahoo Answers and its difficulty is 4.42. The authors mention that "This indicates that the difficulty measure in its current incarnation may be more effective at assigning a class of difficulty to datasets, rather than a regression-like value".

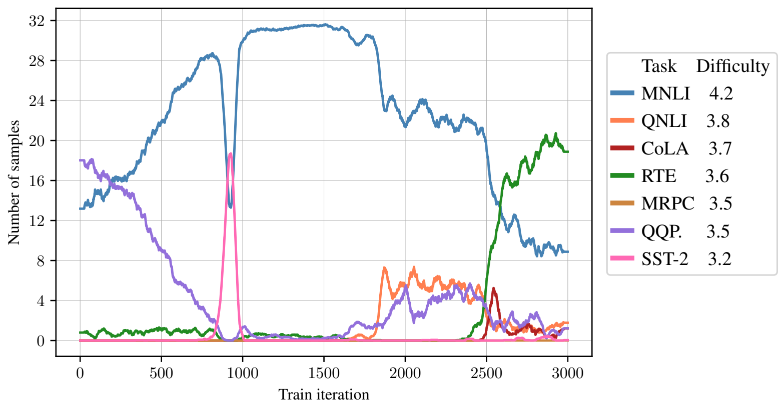


Figure 6: Task composition of MT-Uncertainty sampling and estimated task difficulty using EDM: number of training samples per task at each iteration for batch size of 32. The occurrence of first peaks and estimated difficulty follow the same order: From highest to lowest: MNLI > CoLA > RTE > QQP = MRPC > SST-2.

A.3 Dataset Description

The datasets that were used for the domain adaption experiments were SciTail[5] and SNLI[6]. We *jointly* trained a CA-MTL$_{\text{RoBERTa-LARGE}}$ model on 9 GLUE tasks, 8 Super-GLUE[7] tasks, 6 MRQA[8] tasks, and on WNUT2017[9] (Derczynski et al., 2017).

All GLUE tasks are binary classification, except STS-B (regression) and MNLI (three classes). We used the same GLUE data preprocessing as in Devlin et al. (2018).

[4]https://github.com/Wluper/edm
[5]https://allenai.org/data/scitail; Leaderboard can be found at: https://leaderboard.allenai.org/scitail/submissions/public
[6]https://nlp.stanford.edu/projects/snli/
[7]https://super.gluebenchmark.com/tasks
[8]https://github.com/mrqa/MRQA-Shared-Task-2019
[9]https://github.com/leondz/emerging_entities_17

Table 7: GLUE (Wang et al., 2018) dataset description.
References: [1]Warstadt et al. (2018), [2]Socher et al. (2013), [3]Dolan & Brockett (2005), [4]Cer et al. (2017), [5]Williams et al. (2018), [6]Wang et al. (2018), [7]Levesque (2011)

Acronym	Corpus	|Train|	Task	Domain
CoLA[1]	Corpus of Linguistic Acceptability	8.5K	acceptability	miscellaneous
SST-2[2]	Stanford Sentiment Treebank	67K	sentiment detection	movie reviews
MRPC[3]	Microsoft Research Paraphrase Corpus	3.7K	paraphrase detection	news
STS-B[4]	Semantic Textual Similarity Benchmark	7K	textual similarity	miscellaneous
QQP	Quora Question Pairs	364K	paraphrase detection	online QA
MNLI[5]	Multi-Genre NLI	393K	inference	miscellaneous
RTE[6]	Recognition Textual Entailment	2.5K	inference/entailment	news, Wikipedia
WNLI[7]	Winograd NLI	634	coreference	fiction books

Table 8: Super-GLUE (Wang et al., 2019b) dataset description. References: [1]Clark et al. (2019a), [2]de Marneffe et al. (2019), [3]Gordon et al. (2012), [4]Khashabi et al. (2018), [5]Zhang et al. (2018), [6]Wang et al. (2019b), [7]Poliak et al. (2018), [8]Levesque (2011)

Acronym	Corpus	|Train|	Task	Domain
BoolQ[1]	Boolean Questions	9.4K	acceptability	Google queries, Wikipedia
CB[2]	CommitmentBank	250	sentiment detection	miscellaneous
COPA[3]	Choice of Plausible Alternatives	400	paraphrase detection	blogs, encyclopedia
MultiRC[4]	Multi-Sentence Reading Comprehension	5.1K	textual similarity	miscellaneous
ReCoRD[5]	Reading Comprehension and Commonsense Reasoning	101K	paraphrase detection	news
RTE[6]	Recognition Textual Entailment	2.5K	inference	news, Wikipedia
WiC[7]	Word-in-Context	6K	word sense disambiguation	WordNet, VerbNet
WSC[8]	Winograd Schema Challenge	554	coreference resolution	fiction books

Table 9: MRQA (Fisch et al., 2019) dataset description. References: [1]Rajpurkar et al. (2016a), [2]Trischler et al. (2017), [3]Joshi et al. (2017), [4]Dunn et al. (2017), [5]Yang et al. (2018), [6]Kwiatkowski et al. (2019)

Acronym	Corpus	|Train|	Task	Domain
SQuAD[1]	Stanford QA Dataset	86.6K	crowdsourced questions	Wikipedia
NewsQA[2]	NewsQA	74.2K	crowdsourced questions	news
TriviaQA[3]	TriviaQA	61.7K	trivia QA	web snipets
SearchQA[4]	SearchQA	117.4K	Jeopardy QA	web snipets
HotpotQA[5]	HotpotQA	72.9K	crowdsourced questions	Wikipedia
Natural Questions[6]	Natural Questions	104.7K	search logs	Wikipedia

SuperGLUE has a more diverse task format than GLUE, which is mostly limited to sentence and sentence-pair classification. We follow the same preprocessing procedure as in Wang et al. (2019b). All tasks are binary classification tasks, except CB (three classes). Also, WiC and WSC are span based classification tasks. We used the same modified MRQA dataset and preprocessing steps that were used in Joshi et al. (2019). All MRQA tasks are span prediction tasks which seeks to identify start and end tokens of an answer span in the input text.

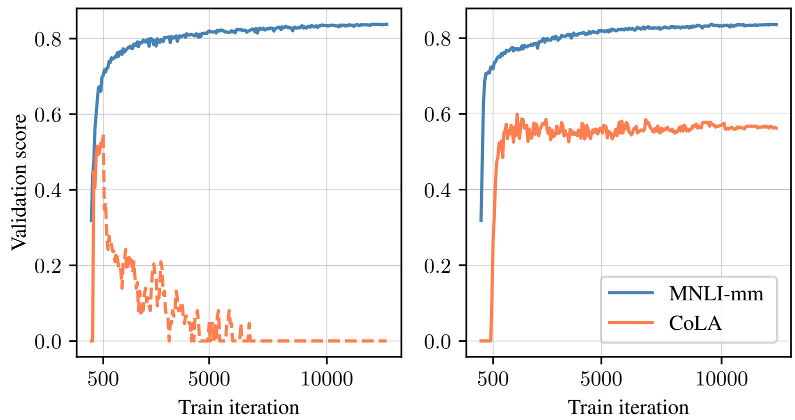
Table 10: SNLI (Bowman et al., 2015) and SciTail (Khot et al., 2018) datasets description.

Acronym	Corpus	|Train|	Task	Domain
SNLI[1]	Stanford Natural Language Inference	550.2k	inference	human-written English sentence pairs
SciTail[2]	Science and Entailment	23.5K	entailment	Science question answering

SNLI is a natural inference task where we predict three classes. Examples of three target labels are: Entailment, Contradiction, and Neutral (irrelevant). SciTail is a textual entailment dataset. The hypotheses in SciTail are created from multiple-choice science exams and the answers candidates (premise) are extracted from the web using information retrieval tools. SciTail is a binary true/false classification tasks that seeks to predict whether the premise entails the hypothesis. The two datasets are used only for domain adaptation in this study (see section A.5 for the details of our approach).

A.4 Catastrophic Forgetting

The datasets in the GLUE benchmark offers a wide range of dataset sizes. In MTL, heuristics to balance tasks during training is typically done by weighting each task's loss differently. We have investigated in preceding section MT-Uncertainty was able to prioritize task difficulty. Now, we see if MT-Uncertainty can help keep a low resource task performance steady and avoid catastrophic forgetting. Our experimental set-up is the same as in section 4.1. In Figure 6, we compare our method with Random sampling (see equation 6). With Random sampling, CoLA's dataset is seen completely by iteration 500 and the task performance starts to decrease. On the other hand, MT-Uncertainty samples the task whenever it's Shannon Entropy is high.



(a) Random (b) MT-Uncertainty

Figure 7: Illustrating catastrophic forgetting with two tasks in the first epoch: With a random sampling strategy, all of CoLA's tasks are sampled by iteration 500, at which point the larger MNLI dataset overtakes the learning process. With MT-Uncertainty sampling, CoLA is sampled whenever Shannon entropy is high but not necessarily at every iteration, allowing lower resource tasks to avoid catastrophic forgetting.

A.5 Zero-Shot Results on SciTail and SNLI

Table 11: CA-MTL is flexible and extensible to new tasks. However, CA-MTL is sensitive to the new task's embedding. We tested multiple task embeddings that worked best on either SciTail or SNLI by checking performance in a zero shot setting or using 0% of the data.

Initialization of new task embedding layer	SciTail 0% of data	SNLI 0% of data
CoLA init	43.0	34.0
MNLI init	24.2	33.0
MRPC init	34.5	**45.5**
STS-B init	**46.9**	33.2
SST-2 init	25.8	34.2
QQP init	31.7	37.3
QNLI init	32.0	38.0
RTE init	32.3	40.6
WNLI init	29.0	30.4
Average init	28.7	37.7
Random init	46.8	34.0
Xavier init	29.8	37.6

Before testing models on domain adaptation in section 4.2, we ran zero-shot evaluations on the development set of SciTail and SNLI. Table 11 outlines CA-MTL$_{\text{BERT-BASE}}$'s zero-shot transfer abilities when pretrained on GLUE with our MTL approach. We expand the task embedding layer to accommodate an extra task and explore various embedding initialization. We found that reusing STS-B and MRPC task embeddings worked best for SciTail and SNLI respectively.

A.6 NAMED ENTITY RECOGNITION (NER) RESULTS

We report NER task results on the WNUT2017 dataset in table 12. As with the other 23 tasks (see section 4.2) that we *jointly* trained our 24-task CA-MTL$_{\text{RoBERTa-LARGE}}$ model, we did not use fine-tuning or assemble methods on WNUT2017. We compare with the latest state-of-the art models. Note that Nguyen et al. (2020) used RoBERTa$_{\text{LARGE}}$ (Liu et al., 2019c) and XLM-R$_{\text{LARGE}}$ (Conneau et al., 2020) as large model baselines. CA-MTL$_{\text{RoBERTa-LARGE}}$ outperforms XLM-R$_{\text{LARGE}}$ by 1.6% WNUT2017 F1 score. Except for the BLSTM-CRF-MTL Aguilar et al. (2019) model and our method, all methods use single task fine-tuning.

Table 12: WNUT2017 test F1 results (entity level) on the NER task. Results taken from: [1]Aguilar et al. (2019), [2]Zhou et al. (2019), [3]Nguyen et al. (2020)

SOTA Models	F1
BLSTM-CRF-MTL[1]	41.9
DATNet[2]	42.3
BERTweet[3]	56.5
RoBERTa$_{\text{LARGE}}$[3]	56.9
XLM-R$_{\text{LARGE}}$[3]	57.1
CA-MTL$_{\text{RoBERTa-LARGE}}$ (ours)	**58.0**

A.7 MORE EXPERIMENTAL DETAILS

For Figure 5 and Table 5, all BERT-based model have half their layers frozen (untrained) for a fair comparison of ablation results.