Forging Multiple Training Objectives for Pre-trained Language Models via Meta-Learning

Anonymous ACL submission

Abstract

Multiple pre-training objectives fill the vacancy of the understanding capability of single-objective language modeling, which serves the ultimate purpose of pre-trained language models (PrLMs), generalizing well on a mass of scenarios. However, learning multiple training objectives in a single model is challenging due to the unknown relative significance as well as the potential contrariety between them. Empirical studies have shown that the current objective sampling in an ad-hoc manual setting makes the learned language representation barely converge to the desired optimum. Thus, we propose *MOMETAS*, a novel adaptive sampler based on meta-learning, which learns the latent sampling pattern on arbitrary pre-training objectives. The design is lightweight with little additional training overhead. To validate our approach, we adopt five objectives and conduct continual pre-training with BERT-base, BERT-large models, where MOMETAS demonstrates universal performance gain over other rule-based sampling strategies on 14 natural language processing tasks.

1 Introduction

It is appealing for deep neural language models to generalize well on multiple downstream tasks through large-scale language pre-training, e.g. BERT (Devlin et al., 2019), ELECTRA (Clark et al., 2020), DeBERTa (He et al., 2021) and GPT (Brown et al., 2020). Most pre-trained language models (PrLMs) rely on only one or two pre-training objectives, from Masked Language Modeling (MLM), Next Sentence Prediction (NSP) (Devlin et al., 2019), Sentence Order Prediction (SOP) (Lan et al., 2020) and Permutation Language Modeling (PLM) (Yang et al., 2019). Even though PrLMs are intended for high generalization, studies show that they are not always all-rounded and tend to be particularly weak in some aspects (Li and Zhao, 2021; Li et al., 2020; Yang et al., 2019), while an ultimate PrLM for panoramic adaption of language understanding must be able to stand for the nice initialization onto a mass of scenarios simultaneously and effectively (Chen et al., 2018).

With the birth of more and more pre-training objectives, a number of specific ones beyond are found of great benefit to enhance task-level understanding capability, e.g. contrastive learning (Gao et al., 2021), knowledge injection (Xiong et al., 2020), algorithmic difference (Li and Zhao, 2021). To enjoy the merits of all worlds and let the model generalize better on more seen or perhaps unseen tasks, there naturally comes a need to combine all these objectives in an organic manner.

However, learning multiple pre-training objectives simultaneously in a single model is challenging (Chen et al., 2018; Yu et al., 2020). A well-known issue is negative transfer (Wang et al., 2019b) in which learning well on one objective impairs another. More importantly, the relative significance between all objectives is supposed to be scheduled. For instance, NSP can take little effect on the model due to its simpleness in the mature stage of training. However, it is of great difficulty to heuristically tune such a ratio considering the large amounts of compute to pre-train once. In most cases we tentatively treat all of them equally (Liu et al., 2019; Lewis et al., 2020), which makes the learned language representation barely converge to the optimal point and limits the model performance.

To forge multiple training objectives for PrLMs, this paper presents to learn an optimal sampling strategy so that the more informative objective is more likely to be chosen. The backbone is meta-learning (Thrun and Pratt, 1998) and thus we call it ***M**ulti-**O**bjective **META**-**S**ampler* (*MOMETAS*). In the proposed framework, we redesign the pre-training process into two phases, meta-train and meta-test. The model is trained alternately on one sampled objective at each step during meta-train, while the sampling distribution is then updated during meta-test by measuring the relative contribution of each objective. The training design is lightweight with little additional overhead to guarantee the pre-training efficiency. To validate our approach, we consider five pre-training objectives (e.g. for sentence embedding, knowledge capture, syntactic understanding) and continue to pre-train with BERT-base, BERT-large, where MOMETAS demonstrates universal performance gain over other rule-based sampling strategies on 14 natural language processing tasks.

2 Related Work

2.1 Multiple Pre-training Objectives

Our work is dedicated to improvement of learning multiple pre-training objectives on a single language model (Liu et al., 2019; Lewis et al., 2020). Language pre-training is well-studied in recent years and there are various potential objectives proposed, e.g. to enhance general language representation (Lewis et al., 2020), text generation (Yang et al., 2019; Dong et al., 2019), sentence embedding (Gao et al., 2021; Li and Zhao, 2021), dialogue understanding (Xu and Zhao, 2021). MOMETAS is designed to bring them together organically.

Our work is related to balancing training in multi-task networks, e.g. gradient normalization (Chen et al., 2018), projecting conflicting gradients (Yu et al., 2020), weighting training loss based on uncertainty (Kendall et al., 2018). For PrLMs, it is explored more on fine-tuning (Stickland and Murray, 2019; Raffel et al., 2020; Poth et al., 2021). In practice, BERT-style pre-training like MLM (Devlin et al., 2019) establishes self-supervised objectives through certain transformations on text data. From this point of view, our work is similar to reweighting training samples (Alain et al., 2015; Ren et al., 2018) or data selection (Schulman et al., 2016; Wang et al., 2020a).

A related application in natural language processing is to train multilingual models (Arivazhagan et al., 2019; Wang et al., 2020b,c; Zhou et al., 2021; Wang et al., 2021b). For instance, MultiDDS (Wang et al., 2020b) learns a data scorer to balance the data usage of languages. However, designing pre-training is more challenging for lack of prior knowledge, e.g. data size (Johnson et al., 2017), data resource (Neubig and Hu, 2018). Besides, one can not access to real downstream tasks. All these can lead to so different optimization designs.

2.2 Meta Learning

Meta-Learning (Learning to Learn) (Thrun and Pratt, 1998) has a long history with vast contributing literature, whereas we could only mention several related works here. Ravi and Larochelle (2017) designs an LSTM-based meta-learner to learn the update rule for few shot learning. Finn et al. (2017) proposes MAML to learn an optimized initialization ready for fast adaption to new tasks. The idea also emerges in recent natural language processing, e.g. generating the text mask for MLM (Kang et al., 2020), optimizing the first-order approximation of dropout to learn dynamic attention pattern (Wu et al., 2021), leveraging MAML-inspired pre-training to find a global representation of downstream tasks (Lv et al., 2020; Ke et al., 2021).

3 Multi-Objective Meta-Sampler

In this section, we first take an overview of our meta-learning framework. What follows is the preliminaries of the pre-training setting as well as a number of ruled-based samplers. Then we discuss the details of our meta-sampler.

3.1 Overview

As depicted in Figure 1, we learn the problem in two phases, meta-train and meta-test. In meta-train, the model is trained and updated on a series of pre-training objectives sampled through MOMETAS one by one. After a number of steps, it goes through meta-test, where we evaluate the model over all objectives in one shot. The evaluation is done on a clean validation set in addition to the training one. Based on the evaluation feedback, MOMETAS is then updated. We repeat such train-test cycles until the end of pre-training.

3.2 Multi-Objective Pre-training

In our multi-objective pre-training, the model is trained on m different objectives. The input text of each objectives passes a common encoder to obtain the shared language representation and then output through a specific layer (or head). We denote all objectives as $\{\mathcal{T}^1, \mathcal{T}^2, \cdots, \mathcal{T}^m\}$, the sampling of which is subject to the latent distribution $P_{\mathcal{D}}$. At each training step t, a single objective $\mathcal{T}_t \in \{\mathcal{T}^1, \mathcal{T}^2, \cdots, \mathcal{T}^m\}$ is sampled from $P_{\mathcal{D}}$.

3.3 Rule-based Samplers

We first consider several rule-based samplers:

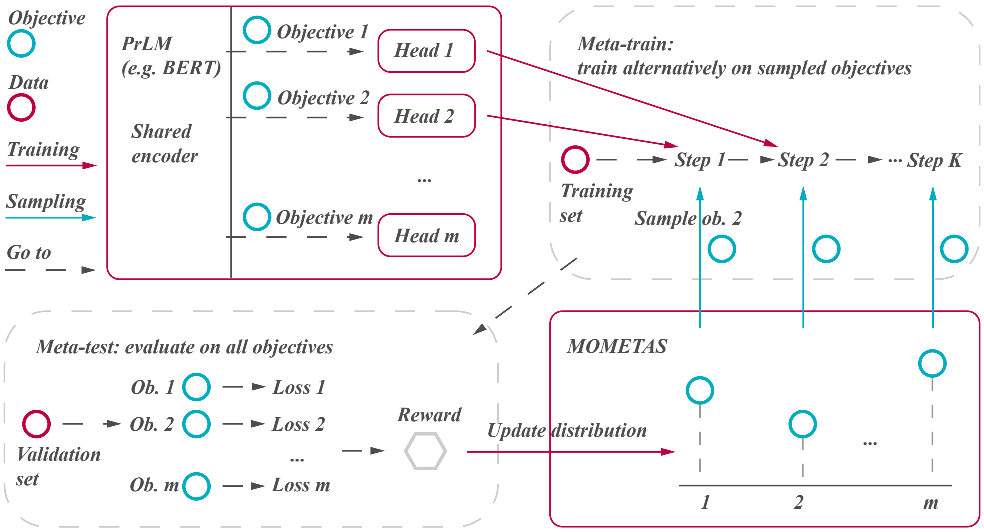

Figure 1: An overview of the meta-learning framework of training PrLMs with MOMETAS, where "ob." serves the short for "objective". We only show the first two and the last samplings for simplicity.

• *Uniform-based*: The most straightforward and simplest approach is to make uniform sampling over all objectives. It equals conventional multi-objective training and multi-task learning. However, when the number of objectives is up, it is hard to guarantee the training efficiency, since some simpler objectives come close to convergence early, while some more difficult ones still require a large number of steps to learn well.

• *Gradient-based*: Gradient acts as a contributing signal of the training state of a network when making gradient descent (Ravi and Larochelle, 2017; Wang et al., 2020b; Yu et al., 2020). Larger gradient may have a greater impact on updating its parameters. An intuitive idea is to sample more on those objectives with large gradients, while less on those with small gradients which tend to take minimal impacts on the network. Computationally, we may take the norm of gradients over all encoder parameters (Ravi and Larochelle, 2017).

• *Loss-based*: Similar as above, loss acts as another contributing signal of how well a certain objective is learned (Kendall et al., 2018). More specifically, we may compute the inverse training rate (IR) by dividing the current loss by its initial value, so that lower IR corresponds to a faster training rate for the objective. Thus, the idea is to sample more on those objectives with higher inverse training rates.

3.4 Meta-Sampler

Both gradient-based and loss-based approaches merely focus on the state of a single objective in an ad-hoc manner but do not take into account the coupling between them, which makes it hard to achieve the optimal point across all objectives.

Thus, we propose to learn a meta-sampler MOMETAS parametrized as $\psi = P_{\mathcal{D}}$, based on meta-learning. Suppose that we sample a single objective at each step t from $P_{\mathcal{D}}$ during meta-train and obtain a sequence of objectives:

$$\tau = \{\mathcal{T}_1, \mathcal{T}_2, \cdots, \mathcal{T}_K\}, \tau \sim P_{\mathcal{D}}$$

where K refers to the number of steps of meta-train (we call it meta length in the paper). In the following meta-test, we evaluate the model over all objectives $\mathcal{T}_{1:K}$ on an additional validation set $\mathcal{V}$. The goal of MOMETAS is to learn well or earn more gain on all objectives, that is to maximize:

$$J(\psi) = E_{\tau \sim P_{\mathcal{D}}}[R(\tau)] \tag{1}$$

where $R(\tau)$ refers to the overall gain given τ.

Since $J(\psi)$ is non-differentiable, it is impossible to apply normal gradient-based methods to update MOMETAS which makes sampling from different objectives. Following REINFORCE (Sutton et al., 1999), we take a number of policy gradient steps to accommodate the non-differentiable operations of sampling, that is:

$$\psi \leftarrow \psi + \beta \sum_{t=1}^{K} \nabla_{\psi} \log P(\mathcal{T}_t; \psi) R(\tau) \tag{2}$$

where β refers to the meta step size. From this perspective, $R(\tau)$ can be viewed as a rewarding function of training gain. Note that $R(\tau)$ is only obtained at the end of meta-train ($t = K$).

Meta length K indicates the accumulation of meta knowledge. Intuitively, larger K comes to more training samples until each meta update step, which stabilizes the training process but lowers down the sensitivity of MOMETAS.

3.4.1 Individual Rewarding

We further explore the details of the rewarding function $R(\tau)$. We first let r^i be the individual gain on each objective ($i = 1 \sim m$) so that $R(\tau) = \sum_{i=1}^{m} r^i$. However, our empirical results show that simply letting r^i be the opposite of each evaluation loss merely leads to limited performance. This is caused by the problem that it cannot address the issue of negative transfer. Suppose that there is a dominant objective, trained well so that the loss of it is continually down. The real situation can be that the overall loss is declining, while the individual losses of certain objectives are still rising, even though MOMETAS is positively rewarded.

To destroy such confusion, we let r^i be the **loss drop** of each objective. Specifically, to compute each loss drop, we always maintain the last loss value as the baseline b^i (the evaluation loss from last meta-test). Then we compare the current loss value a^i (from current meta-test) with it. Because the magnitude of loss differs from objectives, we further compute the relative loss drop by dividing it by the baseline b^i. Hence, the final rewarding function can be formulated as:

$$R(\tau) = \sum_{i=1}^{m} \frac{b^i - a^i}{b^i} \tag{3}$$

where b^i and a^i refer to the loss values of the last meta-test and current meta-test respectively. Such rewarding function forces MOMETAS to explore the optimal sampling pattern which is useful across all pre-training objectives.

3.4.2 Entropy Regularization

To further escape from the local optimum, we impose maximum entropy regularization as an additional constraint (Haarnoja et al., 2018), which is widely used in stochastic reinforcement learning. The idea behind this is that smaller entropy means more deterministic sampling from the distribution and MOMETAS will be punished in this situation,

Algorithm 1 Pre-train with MOMETAS

Input: Model θ, m pre-training objectives $\{\mathcal{T}^1, \mathcal{T}^2, \cdots, \mathcal{T}^m\}$, meta length K, MOMETAS distribution $P_{\mathcal{D}}$, validation set $\mathcal{V}$

1: Initialize $\mathcal{D}$ with uniform distribution
2: **while** not converged **do**
3:     Empty τ
4:     **for** $t = 1$ to K **do**
5:         Sample one objective $\mathcal{T}_t \sim P_{\mathcal{D}}$
6:         Update model parameters θ_t
7:         Append $\mathcal{T}_t$ into τ
8:     **end for**
9:     Fetch data for each objective from $\mathcal{V}$
10:    Evaluate with model parameters θ_K
11:    Compute reward via Eq. 3
12:    Update $P_{\mathcal{D}}$ via Eq. 2
13: **end while**

which encourages MOMETAS to explore and allows it to step out of the local optimal point. Hence, the training objective of MOMETAS comes to:

$$J(\psi) = E_{\tau \sim P_{\mathcal{D}}}[R(\tau) + \lambda H(\psi)] \tag{4}$$

where $H(\psi)$ refers to the entropy regularization term. We find good performances when the temperature parameter λ is set to $1 \sim 3$.

3.4.3 Algorithm

Then we present our meta-learning algorithm, which is summarized in Algorithm A. Specifically, we first initialize MOMETAS distribution $P_{\mathcal{D}}$ with uniform distribution. In meta-train, the model is fed with K sampled pre-training objectives one by one. At each step t, we need to record every single sampling $\mathcal{T}_t$ in order to update MOMETAS later. What follows is meta-test, where the model is evaluated on the validation set $\mathcal{V}$. MOMETAS will be rewarded based on the evaluation feedback and then updated so as to be ready for the next meta-train. We repeat such a train-test cycle for times until model convergence. Note that we fetch the validation samples from $\mathcal{V}$ through random sampling to guarantee the training efficiency.

When pre-training with MOMETAS, the additional time consumption mainly comes from doing evaluation in meta-test. Though it will rise as the number of objectives increases, the evaluation is done only once every K steps (e.g. 100) and is inherently fast with no backward passes. Thus, the overhead brought by MOMETAS is minimal.

	CoLA (Mcc)	SST-2 (Acc)	MRPC (Acc)	QNLI (Acc)	MNLI-m/mm (Acc)	QQP (F1)	RTE (Acc)	STS-B (Spc)	Avg
BERT$_{base}$	51.9	93.5	88.9	90.5	84.6/83.4	71.2	66.4	85.8	79.6
BERT$_{base}$ (Ours)	52.1	92.9	88.7	90.2	84.6/83.4	71.3	67.4	84.6	79.5
+ *Ub*	52.0	93.0	89.1	90.6	84.7/83.7	71.5	66.7	85.0	79.7
+ *Gb*	52.0	93.6	89.2	**90.7**	84.5/84.0	71.8	66.9	85.9	79.8
+ *Lb*	53.1	93.3	89.7	90.5	84.8/**84.4**	71.8	67.3	86.0	80.1
+ *MOMETAS*	**55.9**	**93.7**	**90.0**	**90.7**	**85.2**/84.3	**72.1**	**68.4**	**86.9**	**80.8**

Table 1: GLUE test results under different sampling strategies. BERT$_{base}$ refers to the reported results in Devlin et al. (2019) while BERT$_{base}$ (Ours) refers to our rerun results. Due to limited number of submissions per day, we do not report the results over multiple runs in Table 1 (for multiple runs, please refer to Table 2).

4 Experimental Setup

In this section, we present our experimental setup. Our implementations are based on PyTorch using *transformers* (Wolf et al., 2020).

4.1 Pre-training Objectives

We adopt five pre-training objectives in our experiments. The details of them are listed below.

• *General Language Representation* - **Masked Language Modeling (MLM)**: Following BERT (Devlin et al., 2019), we randomly sample 15% of the tokens in each input sequence and replace them with special `[MASK]` elements. **Added Token Detection (ATD)**: We randomly sample 15% of the positions in each sequence and insert random tokens in them. The model is required to decide which positions are superfluous. Different from MLM, ATD expands the context of text.

• *Sentence Embedding* - **Contrastive Learning of Sentence Embeddings (CSE)**: Following SimCSE (Gao et al., 2021), we feed the same sequence twice by applying different dropout masks and extract the `[CLS]` elements as their sentence representations. The model is required to predict the input sentence itself from in-batch negatives.

• *Syntax* - **Dependency Head Prediction (DHP)**: Following K-adapter (Wang et al., 2021a), we parse each sentence into a dependency tree and let the model predict the head of each token[1].

• *Entity & Knowledge* - **Replaced Entity Detection (RED)**: Following WKLM (Xiong et al., 2020), we randomly replace half of the entities in each sequence and replace them with random ones within the same types.

Though we are unable to cover all alternatives in this paper, the experimental results are of great potential to be extended to other pre-training setups.

[1] https://github.com/stanfordnlp/stanza

4.2 Dataset

Based on our pre-training setup, we validate our approach on a wide range of downstream benchmarks (14 tasks in total). In what follows, we summarize them as well as describe how the chosen ones relate to our pre-training objectives.

General Natural Language Understanding We adopt **GLUE** benchmark (Wang et al., 2019a), a collection of eight natural language understanding tasks, including natural language inference, sentiment analysis and semantic similarity. We exclude problematic WNLI as in Devlin et al. (2019)). In addition, we adopt **SICK** (Marelli et al., 2014), another natural language inference benchmark as a complement.

Semantic Similarity We further adopt **PAWS-QQP** (Zhang et al., 2019), which adds adversarial examples to QQP for evaluating model robustness. Following the zero-shot setting in Zhang et al. (2019), we train the model on QQP and directly evaluate it on PAWS-QQP.

Named Entity Recognition (NER) We adopt two benchmarks, **CoNLL-2003** (Sang and Meulder, 2003) and **WNUT-2017** (Derczynski et al., 2017). Of these, WNUT-2017 contains a large number of rare entities, which therefore requires the model with stronger generalization.

Multi-choice Machine Reading Comprehension (MRC) Two challenging benchmarks are adopted, **DREAM** (Sun et al., 2019) for multi-turn dialogue understanding, and **aNLI** (Bhagavatula et al., 2020) for commonsense reasoning, both of which are in format of multi-choice MRC.

Model	Language Inference		Semantic Similarity		NER		Multi-Choice MRC	
	MNLI (Acc)	SICK (Acc)	P-QQP (Acc)	STS-B (Spc)	CoNLL (F1)	WNUT (F1)	DREAM (Acc)	aNLI (Acc)
BERT-base								
Base	$83.9_{(.3)}$	$87.0_{(.2)}$	$33.4_{(.6)}$	$84.8_{(.6)}$	$91.2_{(.1)}$	$48.8_{(1.0)}$	$62.5_{(.6)}$	$63.8_{(.5)}$
Ub	$84.2_{(.1)}$	$87.5_{(.2)}$	$35.6_{(.8)}$	$85.2_{(.5)}$	$91.6_{(.0)}$	$50.8_{(.7)}$	$63.2_{(.5)}$	$64.6_{(.8)}$
Meta	$\mathbf{84.8}_{(.1)}$	$\mathbf{87.9}_{(.3)}$	$\mathbf{36.5}_{(.9)}$	$\mathbf{86.5}_{(.2)}$	$\mathbf{92.0}_{(.2)}$	$\mathbf{52.1}_{(.7)}$	$\mathbf{64.5}_{(.0)}$	$\mathbf{65.8}_{(.3)}$
BERT-large								
Base	$86.1_{(.2)}$	$87.6_{(.9)}$	$36.2_{(.9)}$	$86.4_{(.3)}$	$91.9_{(.1)}$	$50.2_{(1.5)}$	$66.3_{(1.3)}$	$66.9_{(.8)}$
Ub	$86.1_{(.1)}$	$88.2_{(.1)}$	$40.6_{(.5)}$	$87.5_{(.2)}$	$92.3_{(.3)}$	$50.9_{(1.8)}$	$65.8_{(.8)}$	$67.7_{(.7)}$
Meta	$\mathbf{86.5}_{(.1)}$	$\mathbf{88.6}_{(.1)}$	$\mathbf{41.8}_{(.5)}$	$\mathbf{88.5}_{(.6)}$	$\mathbf{92.4}_{(.2)}$	$\mathbf{52.9}_{(1.2)}$	$\mathbf{68.5}_{(.7)}$	$\mathbf{69.1}_{(.5)}$

Table 2: Results on more different tasks over five runs, where we report the mean as well as the standard deviation. Respectively, *Base*, *Ub* and *Meta* refer to original models, and multi-objective trained models with uniform-based sampling and MOMETAS. For MNLI, we average the two scores of in-distribution and out-of-distribution divisions.

Notably for DREAM and aNLI, there are no straightforward objectives adopted. However, it is desirable that the model is able to learn the inter-disciplinary knowledge and generalize better on tasks not seen during pre-training through jointly learning multiple objectives.

4.3 Baseline Strategies

We compare MOMETAS with several earlier discussed sampling strategies, including *Uniform-based* (Ub), *Gradient-based* (Gb), and *Loss-based* (Lb). Experiments are made on BERT$_{base}$ models.

Except for Ub, the rest two are based on proportion, that is we sample the objectives as proportional to the magnitudes of concerned values. To implement, we compute the average gradient (L2 norm of gradients over encoder parameters) or loss of each objective for every certain number of training steps (to keep in pace with MOMETAS, also K steps). At the same point as meta-test, we update the distribution. However, we find some large values (e.g. big gradient at the start of training) will make the probabilities of other objectives close to zero. Following Andrychowicz et al. (2016), we use Sigmoid function to scale them properly.

4.4 Training Details

Pre-training Inherited from the released checkpoints, `bert-base-uncased` and `bert-large-uncased`[2], we continue to pre-train our models following multi-objective setting. For training corpus, we use a subset of Colossal Clean Crawled Corpus (Raffel et al., 2020) (we use nearly 100GB of it and randomly sample 1GB for validation). Each single model is trained with 512 batch size and for 50K steps (nearly one epoch). Unless otherwise specified, we fix meta length K to 100 and meta step size to 1e-1. Training a base/large-size model takes about 12/36 hours on 8 V100 GPUs with FP16 for both uniform-based sampling and MOMETAS.

Fine-tuning For all GLUE sub-tasks, we follow the hyperparameters shared in Lan et al. (2020) and fine-tune for 3 epochs, except 10 epochs for RTE and STS-B. For other tasks, we merely sweep through learning rates and batch sizes for efficiency, excluding dropout probabilities or weight decay rates. Readers can refer to Appendix A for details.

5 Empirical Results

GLUE Table 1 reports the test results on GLUE benchmark under different sampling strategies, all of which are based BERT$_{base}$. Intuitively, simple uniform multi-objective pre-training (Ub) merely leads to limited performance gain (79.5 → 79.7). Besides, we find that Gb is also not effective, while Lb brings nice gain (**79.5** → **80.1**). However, more powerful performance gain can be seen on MOMETAS-empowered one (**79.5** → **80.8**). Compared to Ub, MOMETAS outperforms it on all eight sub-tasks (**3.9** points absolute gain on CoLA, **0.7** on SST-2, **0.9** on MRPC, **1.7** on RTE, **2.3** on STS-B), which indicates the strength of our meta-learning-based sampling.

[2] https://github.com/huggingface/transformers/

More tasks We make further experiments on more different tasks as in Table 2. Generally, MOMETAS better facilitates multi-objective pre-training compared to uniform-based sampling. We first focus on semantic similarity task (STS-B and PAWS-QQP), for which we adopt CSE to improve the performance. According to Gao et al. (2021), single CSE trained BERT can achieve significant improvement. When the number of objectives increases, however, the situation can be difficult. It does not work well with Ub (84.8 $\rightarrow$ 85.2). Contrarily, MOMETAS brings a huge performance boost on BERT$_{base}$ (**84.8** $\rightarrow$ **86.5** on STS-B, **33.4** $\rightarrow$ **36.5** on P-QQP), even surpasses BERT$_{large}$. Similar situation can be found on NER comparing Ub with MOMETAS (**50.8** $\rightarrow$ **52.1** on WNUT). It demonstrates that **MOMETAS helps maintain the benefit of a single objective in the multi-objective scenario**. Additionally, **MOMETAS-empowered BERT$_{base}$ is able to outperform BERT$_{large}$** on five tasks (SICK, P-QQP, STS-B, CoNLL and WNUT), which indicates the great potential of multi-objective pre-training. On the other hand, because of the attempt to learning cross knowledge from other objectives, MOMETAS also enables the model to learn well on MRC tasks, even though there are no related objectives adopted.

6 Visualization

Probability distribution Figure 2 depicts the sampling distribution of all pre-training objectives learned by MOMETAS. Intuitively, the distribution looks more volatile when $\lambda = 2$ (bottom), while more clustered when $\lambda = 3$ (upper), which indicates the role of entropy regularization. From both cases, we may find some common clues. ATD always stands a high picking weight up to 0.4 in the early stage of training. It uncovers the potential of adding corruption when learning denoising encoder. However, the significance of MLM is lower. It is because MLM has previously been well-trained so that the loss drop is less considerable than the other new ones. Then we look at CSE, a sentence-level objective. Though it is much easier than the other token-level ones, it has never been underweight.

Reward We observe the respective reward curves of MOMETAS and Ub to access to their training gain for multi-objective pre-training. To make intuitive, we depict the difference of them (the former minus the latter) as in Figure 3. Intuitivey, we see slight differences at the beginning of training since

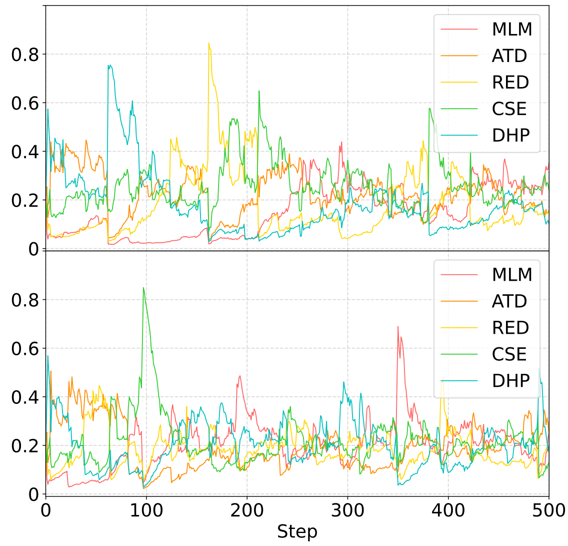

Figure 2: Sampling distribution learned by MOMETAS, upper for $\lambda = 2$, bottom for $\lambda = 3$.

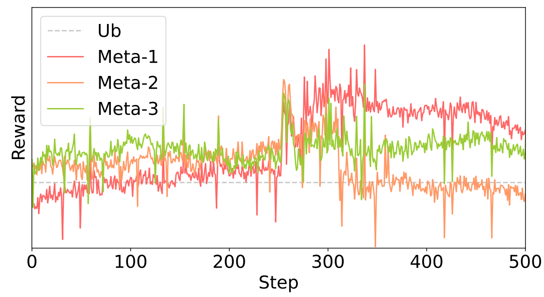

Figure 3: Difference of the total reward, where Ub (a horizontal line of 0) and Meta-x refer to the uniform-based sampling and MOMETAS with entropy regularization $\lambda = x$. To make more intuitive, we smooth the curves by convolution.

MOMETAS is initialized with uniform distribution. However, all three curves are positive for majority of the time. When $\lambda = 1$ for instance, we see a rising trend of the curve, from negative to positive, while when $\lambda = 3$, the curve is always above zero, which implies that MOMETAS learns to achieve more evaluation scores than Ub in meta-test.

7 Ablation Studies

This section reports our ablation studies over a number of factors of MOMETAS in order to better understand their roles. For all experiments, we report the results over five runs.

7.1 Comparison between Rewarding Functions

We compare different rewarding functions $R(\tau)$ on three GLUE sub-tasks, SST-2, QNLI and STS-

	SICK	STS-B	WNUT
Overall	$87.2_{(.2)}$	$85.2_{(.5)}$	$50.0_{(.8)}$
Hard indiv.	$87.6_{(.0)}$	$86.2_{(.2)}$	$51.0_{(1.1)}$
Relative indiv.	$\mathbf{87.9}_{(.3)}$	$\mathbf{86.5}_{(.2)}$	$\mathbf{52.1}_{(.6)}$

Table 3: Comparison between rewarding functions of MOMETAS on BERT$_{base}$. We keep K and λ the same.

	MNLI-m	STS-B	WNUT
Base ($\lambda = 0$)	$84.7_{(.0)}$	$85.8_{(.3)}$	$51.0_{(.6)}$
$\lambda = 1$	$85.1_{(.1)}$	$86.2_{(.2)}$	$51.7_{(.6)}$
$\lambda = 2$	$\mathbf{85.3}_{(.2)}$	$86.2_{(.5)}$	$50.8_{(.2)}$
$\lambda = 3$	$85.2_{(.2)}$	$\mathbf{86.5}_{(.2)}$	$\mathbf{52.1}_{(.7)}$

Table 4: Effect of entropy regularization on BERT$_{base}$. The base model is trained with no regularization.

B: (1) **overall loss rewarding**: we optimize the summation of all losses; (2) **relative individual rewarding**: exactly what we use in MOMETAS, we optimize the summation of all relative loss drops as Eq. 3; (3) **hard individual rewarding**: similar as the relative one, we replace the individual loss drop with ± 1 when it is down or up respectively and optimize the summation of them.

As shown in Table 3, slight improvement can be seen when simply rewarding MOMETAS with overall loss compared to uniform-based sampling in Table 1. In this situation, it is hard to learn the balance between all objectives. However, individual rewarding can achieve stronger performances in both hard and relative cases.

7.2 Effect of Entropy Regularization

When optimizing MOMETAS, we apply maximum entropy regularization to encourage exploration in the hope of seeking out the global optima. Table 4 demonstrates the effect of different degrees of entropy regularization on pre-training performances. We can see general gain compared to original BERT in Table 1 even if there is no regularization applied. However, regularization further boosts the performances. The best case occurs when $\lambda = 3$, which the model outperforms the base one by 0.5, 0.7 and 1.1 points on all three tasks, respectively.

7.3 Effect of Meta Length

In our pre-training framework, MOMETAS is designed to be updated every K steps. K refers to the number of steps of meta-train and meanwhile reflects the knowledge accumulation before meta-test. Generally, when K becomes larger, MOMETAS tends to be less sensitive and pay more attention to long-term benefits. Contrarily, when K is close to 1, it is greedy and only cares about the current moment. In practical, it cannot be smaller than the number of objectives.

	MNLI-m	SICK	STS-B	WNUT
$K = 25$	84.6	87.5	**86.9**	51.3
$K = 50$	85.1	87.5	86.2	51.7
$K = 100$	**85.2**	**87.9**	86.5	52.1
$K = 200$	85.0	87.7	86.3	**52.4**

Table 5: Effect of meta length on BERT$_{base}$. Note that the results are based on five runs but we do not list the variances for space limitation.

Table 5 shows the pre-training performances under a number of values of K. We can see a too small K may lead to worse results (e.g. $K = 25$). It can be presumed that **long-sight helps to find the global optimum**. For example, we cannot acquire sufficient meta knowledge to justify all objectives when K is too small. This can be supported by another fact that **MOMETAS is found more uniform-distributed when K becomes smaller** under the same degree of entropy regularization. On the other hand, we can see nice overall results when K is larger (e.g. $K = 100, 200$). It hints that we can choose a properly larger K to speedup pre-training since there are less meta-test steps.

8 Conclusion

This paper concentrates on multi-objective pre-training of PrLMs and presents Multi-Objective Meta-Sampler (MOMETAS) in the hope of combining arbitrary pre-training objectives organically. We adopt five objectives and conduct experiments on base-size and large-size models. The empirical results demonstrate that MOMETAS largely outperforms other rule-based sampling strategies and unlocks more powerful language models on a wide range of natural language processing tasks.

Our work is limited in not considering the role of the validation set. The most challenging point is the disconnection between pre-training and fine-tuning. Therefore, it can be positive to introduce signals that are more related to the downstream tasks. We will leave this part for our future work.

References

Guillaume Alain, Alex Lamb, Chinnadhurai Sankar, Aaron C. Courville, and Yoshua Bengio. 2015. Variance reduction in SGD by distributed importance sampling. *CoRR*, abs/1511.06481.

Marcin Andrychowicz, Misha Denil, Sergio Gomez Colmenarejo, Matthew W. Hoffman, David Pfau, Tom Schaul, and Nando de Freitas. 2016. Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3981–3989.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George F. Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *CoRR*, abs/1907.05019.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen-tau Yih, and Yejin Choi. 2020. Abductive commonsense reasoning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. 2018. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 793–802. PMLR.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the WNUT2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text, NUT@EMNLP 2017, Copenhagen, Denmark, September 7, 2017*, pages 140–147. Association for Computational Linguistics.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

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 *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13042–13054.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6894–6910. Association for Computational Linguistics.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1856–1865. PMLR.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. {DEBERTA}: {DECODING}-{enhanced} {bert} {with} {disentangled} {attention}. In *International Conference on Learning Representations*.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Trans. Assoc. Comput. Linguistics*, 5:339–351.

Minki Kang, Moonsu Han, and Sung Ju Hwang. 2020. Neural mask generator: Learning to generate adaptive word maskings for language model adaptation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6102–6120. Association for Computational Linguistics.

Zhen Ke, Liang Shi, Songtao Sun, Erli Meng, Bin Wang, and Xipeng Qiu. 2021. Pre-training with meta learning for chinese word segmentation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 5514–5523. Association for Computational Linguistics.

Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 7482–7491. Computer Vision Foundation / IEEE Computer Society.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 9119–9130. Association for Computational Linguistics.

Yian Li and Hai Zhao. 2021. Pre-training universal language representation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 5122–5133. Association for Computational Linguistics.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4487–4496. Association for Computational Linguistics.

Shangwen Lv, Yuechen Wang, Daya Guo, Duyu Tang, Nan Duan, Fuqing Zhu, Ming Gong, Linjun Shou, Ryan Ma, Daxin Jiang, Guihong Cao, Ming Zhou, and Songlin Hu. 2020. Pre-training text representations as meta learning. *CoRR*, abs/2004.05568.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014*, pages 216–223. European Language Resources Association (ELRA).

Graham Neubig and Junjie Hu. 2018. Rapid adaptation of neural machine translation to new languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 875–880. Association for Computational Linguistics.

Clifton Poth, Jonas Pfeiffer, Andreas Rücklé, and Iryna Gurevych. 2021. What to pre-train on? efficient intermediate task selection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 10585–10605. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Sachin Ravi and Hugo Larochelle. 2017. Optimization as a model for few-shot learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. 2018. Learning to reweight examples for robust deep learning. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 4331–4340. PMLR.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003*, pages 142–147. ACL.

John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. 2016. High-dimensional continuous control using generalized advantage estimation. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Asa Cooper Stickland and Iain Murray. 2019. BERT and pals: Projected attention layers for efficient adaptation in multi-task learning. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5986–5995. PMLR.

Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. 2019. DREAM: A challenge dataset and models for dialogue-based reading comprehension. *Trans. Assoc. Comput. Linguistics*, 7:217–231.

Richard S. Sutton, David A. McAllester, Satinder P. Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems 12, [NIPS Conference, Denver, Colorado, USA, November 29 - December 4, 1999]*, pages 1057–1063. The MIT Press.

Sebastian Thrun and Lorien Y. Pratt. 1998. Learning to learn: Introduction and overview. In Sebastian Thrun and Lorien Y. Pratt, editors, *Learning to Learn*, pages 3–17. Springer.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019a. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2021a. K-adapter: Infusing knowledge into pre-trained models with adapters. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 1405–1418. Association for Computational Linguistics.

Xinyi Wang, Hieu Pham, Paul Michel, Antonios Anastasopoulos, Jaime G. Carbonell, and Graham Neubig. 2020a. Optimizing data usage via differentiable rewards. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 9983–9995. PMLR.

Xinyi Wang, Yulia Tsvetkov, and Graham Neubig. 2020b. Balancing training for multilingual neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8526–8537. Association for Computational Linguistics.

Zirui Wang, Zihang Dai, Barnabás Póczos, and Jaime G. Carbonell. 2019b. Characterizing and avoiding negative transfer. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 11293–11302. Computer Vision Foundation / IEEE.

Zirui Wang, Zachary C. Lipton, and Yulia Tsvetkov. 2020c. On negative interference in multilingual models: Findings and A meta-learning treatment. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4438–4450. Association for Computational Linguistics.

Zirui Wang, Yulia Tsvetkov, Orhan Firat, and Yuan Cao. 2021b. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 38–45. Association for Computational Linguistics.

Hongqiu Wu, Hai Zhao, and Min Zhang. 2021. Not all attention is all you need. *CoRR*, abs/2104.04692.

Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. 2020. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yi Xu and Hai Zhao. 2021. Dialogue-oriented pretraining. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 2663–2673. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems 33:*

Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1298–1308. Association for Computational Linguistics.

Chunting Zhou, Daniel Levy, Xian Li, Marjan Ghazvininejad, and Graham Neubig. 2021. Distributionally robust multilingual machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 5664–5674. Association for Computational Linguistics.

A Training Details

	BERT$_{base}$	BERT$_{large}$
Number of hidden layers	12	24
Hidden size	768	1024
Intermediate size	3072	4096
Number of attention heads	12	16
Dropout	0.1	0.1
Batch size	512	512
Learning rate	5e-5	5e-5
Weight Decay	0.01	0.01
Max sequence length	256	256
Warmup proportion	0.06	0.06
Max steps	50K	50K
Gradient clipping	1.0	1.0
FP16	Yes	Yes
Number of GPUs	8	8
Training period	12 hours	36 hours

Table 6: Hyperparameters for pre-training.

	MNLI	SICK	QQP	STS-B	CoNLL	WNUT	DREAM	aNLI
Dropout	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Batch size	128	32	128	16	32	16	16	64
Learning rate	3e-5	5e-5	5e-5	5e-5	5e-5	5e-5	3e-5	5e-5
Weight Decay	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Max sequence length	128	128	128	128	128	64	128	128
Warmup proportion	0.06	0.06	0.06	0.06	0.1	0.1	0.06	0.06
Max epochs	3	3	3	10	3	5	6	3
FP16	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Table 7: Hyperparameters for fine-tuning.