What to Pre-Train on?
Efficient Intermediate Task Selection

Clifton Poth, Jonas Pfeiffer, Andreas Rücklé,*[*] and Iryna Gurevych
Ubiquitous Knowledge Processing Lab, Technical University of Darmstadt
www.ukp.tu-darmstadt.de

Abstract

Intermediate task fine-tuning has been shown to culminate in large transfer gains across many NLP tasks. With an abundance of candidate datasets as well as pre-trained language models, it has become infeasible to experiment with all combinations to find the best transfer setting. In this work, we provide a comprehensive comparison of different methods for efficiently identifying beneficial tasks for intermediate transfer learning. We focus on parameter and computationally efficient adapter settings, highlight different data-availability scenarios, and provide expense estimates for each method. We experiment with a diverse set of 42 intermediate and 11 target English *classification*, *multiple choice, question answering*, and *sequence tagging* tasks. Our results demonstrate that efficient embedding based methods, which rely solely on the respective datasets, outperform computational expensive few-shot fine-tuning approaches. Our best methods achieve an average *Regret@3* of 1% across all target tasks, demonstrating that we are able to efficiently identify the best datasets for intermediate training. [1]

1 Introduction

Large pre-trained language models (LMs) are continuously pushing the state of the art across various NLP tasks. The established procedure performs self-supervised pre-training on a large text corpus and subsequently fine-tunes the model on a specific target task (Devlin et al., 2019; Liu et al., 2019b). The same procedure has also been applied to adapter-based training strategies, which achieve on-par task performance to full model fine-tuning while being considerably more parameter efficient (Houlsby et al., 2019) and faster to train (Rücklé et al., 2021).[2] Besides being more efficient, adapters are also highly modular, enabling a wider range of transfer learning techniques (Pfeiffer et al., 2020b, 2021a,b; Üstün et al., 2020; Vidoni et al., 2020; Rust et al., 2021; Ansell et al., 2021).

Extending upon the established two-step learning procedure, incorporating *intermediate* stages of knowledge transfer can yield further gains for fully fine-tuned models. For instance, Phang et al. (2018) sequentially fine-tune a pre-trained language model on a compatible *intermediate* task before *target* task fine-tuning. It has been shown that this is most effective for low-resource target tasks, however, not all task combinations are beneficial and many yield decreased performances (Phang et al., 2018; Wang et al., 2019a; Pruksachatkun et al., 2020). The abundance of diverse labeled datasets as well as the continuous development of new pre-trained LMs calls for methods that efficiently identify *intermediate* dataset that benefit the target task.

So far, it is unclear how adapter-based approaches behave with intermediate fine-tuning. In the **first part of this work**, we thus establish that this setup results in similar gains for adapters, as has been shown for full model fine-tuning (Phang et al., 2018; Pruksachatkun et al., 2020; Gururangan et al., 2020). Focusing on a low-resource target task setup, we find that only a subset of intermediate adapters yield positive gains, while others hurt the performance considerably (see Table 1 and Figure 2). Our results demonstrate that it is necessary to obtain methods that efficiently identify beneficial intermediately trained adapters.

In the **second part**, we leverage the transfer results from part one to automatically rank and identify beneficial intermediate tasks. With the rise of large publicly accessible repositories for NLP

[*] Contributions made prior to joining Amazon.

[1] Code released at https://github.com/Adapter-Hub/efficient-task-transfer.

[2] Adapters are new weights at every layer of a pre-trained transformer model. To fine-tune a model on a downstream task, all pre-trained transformer weights are frozen and only the newly introduced adapter weights are trained.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 10585–10605
November 7–11, 2021. ©2021 Association for Computational Linguistics

models (Wolf et al., 2020; Pfeiffer et al., 2020a), the chances of finding pre-trained models that yield positive transfer gains are high. However, it is infeasible to brute-force the identification of the best intermediate task. Existing approaches have focused on beneficial task selection for multi-task learning (Bingel and Søgaard, 2017), full fine-tuning of intermediate and target transformer-based LMs for NLP tasks (Vu et al., 2020), adapter-based models for vision tasks (Puigcerver et al., 2021) and unsupervised approaches for zero-shot transfer for community question answering (Rücklé et al., 2020). Each of these works require different types of data, such as intermediate task data and/or intermediate model weights, which, depending on the scenario, are potentially not accessible.[3]

In this work we thus aim to address the **efficiency** aspect of transfer learning in NLP from multiple different angles, resulting in the following **contributions**: **1)** We focus on adapter-based transfer learning which is considerably more parameter (Houlsby et al., 2019) and computationally efficient than full model fine-tuning (Rücklé et al., 2021), while achieving on-par performance; **2)** We evaluate sequential fine-tuning of adapter-based approaches on a diverse set of 42 intermediate and 11 target tasks (i.e. classification, multiple choice, question answering, and sequence tagging); **3)** We identify the best intermediate task for transfer learning, *without* the necessity of *computational expensive, explicit training* on all potential candidates. We compare different selection techniques, consolidating previously proposed and new methods; **4)** We provide a thorough analysis of the different techniques, available data scenarios, and task-, and model types, thus presenting deeper insights into the best approach for each respective setting; **5)** We provide *computational cost estimates*, enabling informed decision making for trade-offs between *expense* and downstream task performance.

2 Related Work

2.1 Transfer between tasks

Phang et al. (2018) show that training on intermediate tasks results in performance gains for many target tasks. Subsequent work further explores the effects on more diverse sets of tasks (Wang et al., 2019a; Talmor and Berant, 2019; Liu et al., 2019a; Sap et al., 2019; Pruksachatkun et al., 2020; Vu et al., 2020). Wang et al. (2019a), Yogatama et al. (2019), and Pruksachatkun et al. (2020) emphasizes the risks of catastrophic forgetting and negative transfer results, finding that the success of sequential transfer varies largely when considering different intermediate tasks.

While previous work has shown that intermediate task training improves the performance on the target task in full fine-tuning setups, we establish that the same holds true for adapter-based training.

2.2 Predicting Beneficial Transfer Sources

Automatically selecting intermediate tasks that yield transfer gains is critical when considering the increasing availability of tasks and models.

Proxy estimators have been proposed to evaluate the transferability of pre-trained models towards a target task. Nguyen et al. (2020), Li et al. (2021) and Deshpande et al. (2021) estimate the transferability between classification tasks by building an empirical classifier from the source and target task label distribution. Puigcerver et al. (2021) experiment with multiple model selection methods, including kNN proxy models to estimate the target task performance. In a similar direction, Renggli et al. (2020) study proxy models based on kNN and linear classifiers, finding that a hybrid approach combination of *task-aware* and *task-agnostic* strategies yields the best results.

Bingel and Søgaard (2017) find that gradients of the learning curves correlate with multi-task learning success. Zamir et al. (2018) build a taxonomy of vision tasks, giving insights into non-trivial transfer relations between tasks. Multiple works propose using embeddings that capture statistics, features, or the domain of a dataset. Edwards and Storkey (2017) leverage variational autoencoders (Kingma and Welling, 2014) to encode all samples of a dataset. Jomaa et al. (2019) train a dataset meta-feature extractor that can successfully capture the domain of a dataset. Vu et al. (2020) encode each training example of a dataset by averaging over BERT's representations of the last layer. Rücklé et al. (2020) capture domain similarity by embedding dataset examples using a sentence embedding model. Achille et al. (2019) and Vu et al. (2020) compute task embeddings based on the Fisher Information Matrix of a probe network.

While many different methods have been pro-

[3] Bingel and Søgaard (2017) and Vu et al. (2020) require access to both intermediate task data and models, Puigcerver et al. (2021) require access to only the intermediate model, and Rücklé et al. (2020) only to the intermediate task data.

posed, there lacks a direct comparison among them. Additionally, previous work has only focus on BERT, which we find to behave considerably different to other model types such as RoBERTa for some methods. In this work we aim to consolidate all methods and experiment with newer model types to provide a more thorough perspective.

3 Adapter-Based Sequential Transfer

We present a large-scale study on *adapter-based* sequential fine-tuning, finding that around half of the task combinations yield no positive gains. This demonstrates the importance of finding approaches that efficiently identify suitable intermediate tasks.

3.1 Tasks

We select *QA* tasks from the MultiQA repository (Talmor and Berant, 2019) and sequence *tagging* tasks from Liu et al. (2019a). Most of our *classification* tasks are available in the (Super)GLUE (Wang et al., 2018, 2019b) benchmarks. We experiment with *multiple choice* commonsense reasoning tasks to cover a broader range of different types, and domains. In total, we experiment with 53 tasks, divided into 42 intermediate and 11 target tasks.[4]

3.2 Experimental Setup

We experiment with BERT-base (Devlin et al., 2019) and RoBERTa-base (Liu et al., 2019b), training adapters with the configuration proposed by Pfeiffer et al. (2021a). We adopt the two-stage sequential fine-tuning setup of Phang et al. (2018), splitting the tasks in two disjoint subsets $\mathcal{S}$ and $\mathcal{T}$, denoted as intermediate and target tasks, respectively. For each pair (s, t) with $s \in \mathcal{S}$ and $t \in \mathcal{T}$, we first train a randomly initialized adapter on s (keeping the base model's parameters fixed). We then fine-tune the trained adapter on t.[5]

For target task fine-tuning, we simulate a **low-resource setup** by limiting the maximum number of training examples on t to 1000. This choice is motivated by the observation that smaller target tasks benefit the most from sequential fine-tuning while at the same time revealing the largest performance variances (Phang et al., 2018; Vu et al., 2020). Low-resource setups, thus, reflect the most beneficial application setting for our transfer learning strategy and also allow us to more thoroughly study different transfer relations.

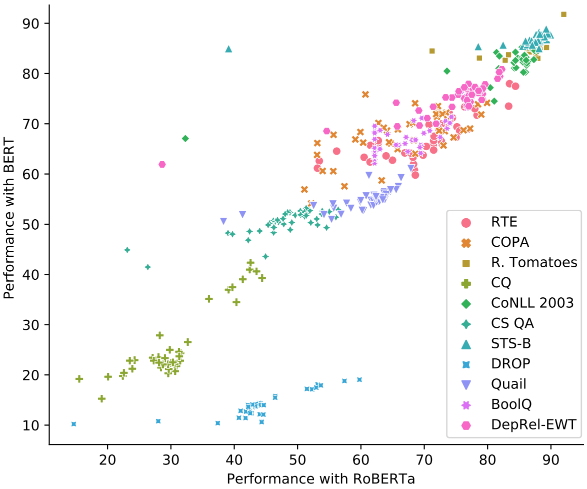

Figure 1: Comparison of transfer performance between BERT and RoBERTa for the respective target tasks, pre-trained on the 42 intermediate tasks.

3.3 Results

Figure 2 shows the relative transfer gains and Table 1 lists the absolute scores of all intermediate and target task combinations for RoBERTa.[6] We observe large variations in transfer gains (and losses) across the different combinations. Even though larger variances may be explained by a higher task difficulty (see 'No Transfer' in Table 1), they also illustrate the heterogeneity and potential of sequential fine-tuning in our adapter-based setting. At the same time, we find several cases of transfer *losses*—with up to 60% lower performances (see Figure 2)—potentially occurring due to catastrophic forgetting.

Overall, for RoBERTa, 243 (53%) transfer combinations yield positive transfer gains whereas 203 (44%) yield losses. The mean of all transfer gains is 2.3%. However, from our eleven target tasks *only five* benefit on average (see 'Avg. Transfer' in Table 1). This illustrates the high risk of choosing the wrong intermediate tasks. Avoiding such hurtful combinations and efficiently identifying the best ones is necessary; evaluating all combinations is inefficient and often not feasible.

We further find that the best performing intermediate tasks for BERT and RoBERTa overlap considerably as illustrated in Figure 1, with transfer performances correlating with a Spearman correlation of 0.94 when averaged over all settings, and 0.68 when averaged per target task.

[4] The choice for our intermediate and target task split was motivated by previous work (Sap et al., 2019; Vu et al., 2020, *inter alia*). For more details see Appendix A.

[5] For more details please refer to Appendix B.

[6] We list the corresponding transfer results for BERT in Table 10 of the Appendix.

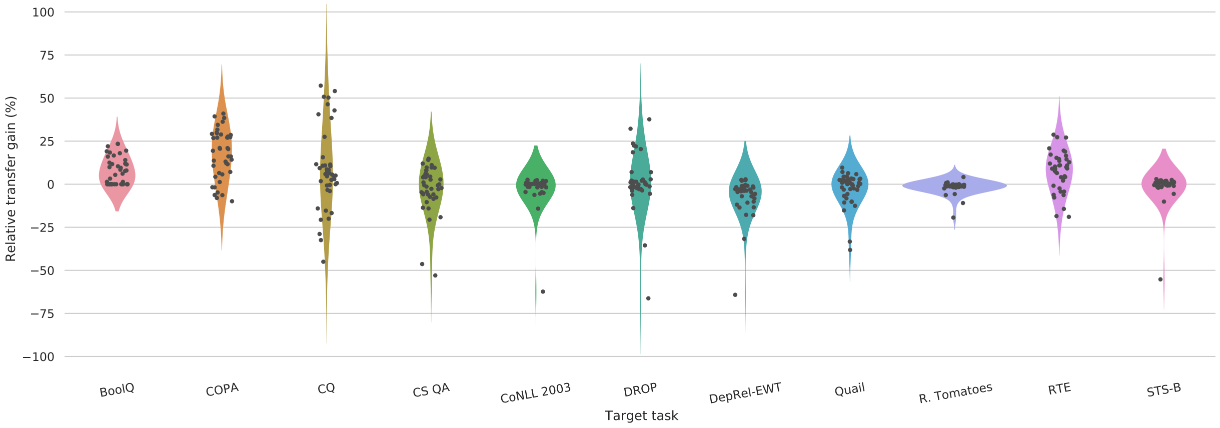

Figure 2: Transfer gains/losses between all intermediate and target tasks with RoBERTa as base model. Each violin represents one target task where each dot represents the *relative transfer gain* (y-axis) from one intermediate task.

Task	**BoolQ**	**COPA**	**CQ**	**CS QA**	**CoNLL 2003**	**DROP**	**DepRel-EWT**	**Quail**	**R. Tomatoes**	**RTE**	**STS-B**
No Transfer	**62.17**	**56.68**	**28.24**	**49.12**	**85.74**	**43.42**	**79.96**	**61.94**	**88.35**	**65.56**	**87.35**
Avg. Transfer	**66.93**	**66.49**	**30.60**	**47.88**	**83.80**	**43.96**	**74.59**	**60.50**	**86.95**	**70.38**	**86.31**
ANLI	**76.75**	73.64	32.67	51.35	85.82	44.66	76.77	66.34	87.64	**84.40**	88.24
ART	70.87	79.00	27.12	53.02	84.48	41.01	74.39	64.00	87.73	73.43	88.45
CoLA	63.01	63.60	31.27	51.11	85.53	43.34	79.24	63.46	88.14	66.93	87.45
CoNLL’00	62.17	57.40	30.32	44.95	87.30	44.33	81.58	54.17	87.19	61.52	88.11
Cosmos QA	74.28	78.48	29.83	53.81	84.17	43.66	74.37	65.56	88.12	79.21	88.00
DuoRC-p	62.17	64.76	39.11	49.63	85.42	51.48	76.33	60.20	86.98	71.55	88.54
DuoRC-s	62.17	67.68	42.45	51.29	85.66	52.29	75.46	61.94	86.92	72.78	88.51
EmoContext	68.64	64.12	23.52	46.96	86.37	42.34	77.69	62.01	88.03	71.48	86.91
Emotion	65.54	60.36	22.58	45.54	84.41	42.22	77.03	59.99	88.01	63.97	87.59
GED-FCE	62.17	55.68	30.05	46.26	86.19	42.52	79.61	59.92	87.64	60.51	86.49
Hellaswag	69.44	77.24	30.67	56.05	85.87	42.86	75.34	62.06	87.64	76.82	88.67
HotpotQA	62.17	59.12	42.58	39.72	73.59	53.68	54.60	55.36	83.30	62.74	86.26
IMDb	68.09	62.76	28.04	46.18	85.90	43.30	77.55	61.60	88.97	68.45	86.38
MIT Movie	62.17	53.12	31.32	44.01	87.43	44.59	78.58	58.42	87.56	67.94	87.05
MNLI	75.86	76.20	31.52	48.80	85.59	43.95	71.41	61.27	88.31	83.47	89.25
MRPC	64.19	68.56	29.06	48.44	86.18	43.43	77.05	62.95	87.35	72.71	88.97
MultiRC	74.05	71.88	30.59	51.35	81.86	43.89	73.38	64.70	87.17	77.98	88.06
NewsQA	62.17	68.24	43.52	49.09	81.41	53.16	65.57	60.44	87.22	72.06	87.66
POS-Co.’03	62.17	51.08	20.09	23.11	87.14	40.74	81.84	41.31	71.24	53.14	78.53
POS-EWT	62.17	53.96	31.21	46.22	87.76	44.31	**82.25**	55.71	86.81	56.17	87.55
QNLI	73.61	72.00	36.01	51.94	85.96	46.47	76.91	64.67	87.64	70.76	89.07
QQP	63.03	72.16	24.27	48.85	81.77	41.79	69.14	61.57	87.58	72.71	89.51
QuaRTz	69.44	68.60	27.30	50.48	86.01	42.69	78.56	62.47	88.16	68.30	88.27
Quoref	62.17	60.72	40.34	46.83	85.86	52.95	76.98	62.75	87.04	71.48	88.19
RACE	76.72	72.04	23.91	53.15	81.05	42.46	65.74	**67.89**	87.19	83.32	88.13
ReCoRD	62.17	63.28	28.25	46.50	84.21	41.85	71.87	63.63	86.96	71.41	86.98
SICK	72.53	73.48	29.66	53.89	85.45	44.11	79.23	63.63	87.94	75.02	88.26
SNLI	69.61	72.36	19.08	42.21	32.29	14.66	28.62	52.53	82.78	76.39	85.90
SQuAD	62.17	68.36	39.71	51.09	86.48	57.40	76.33	61.99	86.21	72.27	87.82
SQuAD 2.0	68.34	73.24	**44.40**	50.43	86.14	**59.78**	76.99	63.79	87.90	75.60	89.06
SST-2	67.14	65.84	28.43	47.81	85.66	42.73	77.28	60.82	**92.03**	69.82	86.75
ST-PMB	62.17	52.16	15.53	26.36	87.47	28.00	81.94	38.31	78.71	53.43	39.12
SWAG	69.30	74.64	28.96	55.00	85.61	43.30	76.97	63.71	88.35	74.44	89.32
SciCite	65.98	64.44	28.75	47.71	86.07	42.18	78.95	62.40	87.77	68.59	86.63
SciTail	72.13	72.00	29.91	53.89	84.74	43.74	77.72	63.77	87.82	73.94	**89.93**
Social IQA	73.36	**79.92**	30.88	**56.41**	86.00	43.34	78.38	65.91	88.27	78.34	88.57
TREC	67.41	59.92	31.47	48.04	86.09	42.86	78.02	62.38	88.14	70.32	86.94
WNUT17	62.17	53.08	28.13	45.36	**87.96**	44.03	77.51	55.59	87.54	61.44	86.00
WiC	62.17	72.88	29.55	52.09	85.73	42.74	79.08	63.16	87.92	68.23	88.00
WikiHop	62.18	55.64	41.35	39.02	84.40	46.45	70.47	56.87	86.59	62.74	85.56
WinoGrande	69.93	73.04	29.58	54.58	86.16	43.54	76.75	63.93	87.94	75.16	87.88
Yelp Polarity	66.97	65.80	22.41	42.42	80.42	37.40	69.25	57.74	89.31	64.98	82.45

Table 1: Target task performances for transferring between intermediate tasks (rows) and target tasks (columns) with RoBERTa as base model. The first row ‘*No Transfer*’ shows the baseline performance when training only on the target task without transfer. All scores are mean values over five random restarts.

4 Methods for the Efficient Selection of Intermediate Tasks

We now present different model selection methods, and later in §5, study their effectiveness in our setting outlined above. We group the different methods based on the assumptions they make with regard to the availability of intermediate task data D_S and intermediate models M_S. Access to both can be expensive when considering large pre-trained model repositories with hundreds of tasks.

4.1 Metadata: Educated Guess

A setting in which there exist neither access to the intermediate task data D_S nor models trained on the data M_S, can be regarded as an *educated guess* scenario. The selection criterion can only rely on metadata available for an intermediate task dataset.

Dataset *Size*. Under the assumption that *more data* implies better transfer performance, the selection criterion denoted as *Size* ranks all intermediate tasks in descending order by the training data size.

Task *Type*. Under the assumption that similar objective functions transfer well, we pre-select the subset of tasks of the same type. This approach may be combined with a *random* selection of the remaining tasks, or with ranking them by *size*.

4.2 Intermediate Task Data

With an abundance of available datasets,[7] and the continuous development of new LMs, fine-tuned versions for every task-model combination are not (immediately) available. The following methods, thus, leverage the intermediate task data D_S *without* requiring the respective fine-tuned models M_S.

Text Embeddings (*TextEmb*). Vu et al. (2020) pass each example through a LM and average over the output representations of the final layer (across all examples and all input tokens). Assuming that similar embeddings imply positive transfer gains, they rank the intermediate tasks according to their embeddings' cosine similarity to the target task.

SBERT Embeddings (*SEmb*). Sentence embedding models such as Sentence-BERT (SBERT; Reimers and Gurevych, 2019) may be better suited to represent the dataset examples. Similar to *TextEmb*, we rank the intermediate tasks according to their embedding cosine similarity.

4.3 Intermediate Model

Scenarios in which we only have access to the trained intermediate models (M_S) occur when the training data is proprietary or if implementing all dataset is too tedious. With the availability of model repositories (Wolf et al., 2020; Pfeiffer et al., 2020a) such approaches can be implemented without requiring additional data during model upload (i.e. in contrast to *TaskEmbs*, where the training dataset information needs to be made available). The following describes methods only requiring access to the intermediate models M_S.

Few-Shot Fine-Tuning (*FSFT*). Fine-tuning of all available intermediate task models on the entire target task is infeasible. As an alternative, we can train models for a few steps on the target task to approximate the final performance. After N steps on the target task, we rank the intermediate models based on their respective transfer performance.

Proxy Models. Following Puigcerver et al. (2021), we leverage simple proxy models to obtain a performance estimation of each trained model M_S on on the *target* dataset D_T. Specifically, we experiment with **k-Nearest Neighbors (*kNN*)**, with $k = 1$ and Euclidian distance, and **logistic/ linear regression (*linear*)** as proxy models. For both, we first compute $\mathbf{h}_{x_i}^{M}$, the token-wise averaged output representations of M_S, for each training input $x_i \in D_T$. Using these, we define $D_T^M = \{(\mathbf{h}_{x_i}^{M}, y_i)\}_{i=1}^{N}$ as the target dataset embedded by M_S. In the next step, we apply the proxy model on D_T^M and obtain its performance using cross-validation. By repeating this process for each intermediate task model, we obtain a list of performance scores which we leverage to rank the intermediate tasks.

4.4 Intermediate Model and Task Data

Access to both intermediate dataset D_S and intermediates model M_S provides a wholesome depiction of the intermediate task, as all previously mentioned methods are applicable in this scenario. Further methods which require access to both are:

Task Embeddings (*TaskEmb*). Achille et al. (2019) and Vu et al. (2020) obtain task embeddings via the Fisher Information Matrix (FIM). The FIM captures how sensitive the loss function is towards small perturbations in the weights of the model and thus gives an indication on the importance of certain weights towards solving a task.

Given the model weights θ and the joint distribu-

[7] e.g. via https://huggingface.co/datasets.

tion of task features and labels $P_\theta(X, Y)$, we can define the FIM as the expected covariance of the gradients of the log-likelihood w.r.t. θ:

$$F_\theta = \mathbb{E}_{x,y\sim P_\theta(X,Y)} \left[\nabla_\theta \log P_\theta(x, y) \cdot \nabla_\theta \log P_\theta(x, y)^\intercal\right]$$

We follow the implementation details given in Vu et al. (2020). For a dataset D and a model M fine-tuned on D, we compute the empirical FIM based on D's examples. The task embeddings are the diagonal entries of the FIM.

Few-Shot Task Embeddings (*FS-TaskEmb*). We also leverage task embeddings in our few-shot scenario outlined above (see *FSFT*), where we fine-tune intermediate models for a few steps on the target dataset. With very few training instances, the accuracy scores of FSFT (alone) may not be reliable indicators of the final transfer performances. As an alternative, we compute the *TaskEmb* similarity of each intermediate model before and after training N steps on the target task. We then rank all intermediate models in decreasing order of this similarity.

5 Experimental Setup

We evaluate the approaches of §4, each having the objective to rank the intermediate adapters $s \in |\mathcal{S}|$ with respect to their performance on $t \in \mathcal{T}$ when applied in a sequential adapter training setup. We leverage the transfer performance results of our 462 experiments obtained in §3 for our ranking task.

5.1 Hyperparameters

If not otherwise mentioned, we follow the experimental setup as described in §3. We describe method specific hyperparameters in the following.

SEmb. We use a Sentence-(Ro)BERT(a)-*base* models, fine-tuned on NLI and STS tasks, in concordance with the respective target model type.

FSFT. We fine-tune each intermediate adapter on the target task for one full epoch and rank them based on their target task performances.[8]

Proxy Models. For both *kNN* and *linear*, we obtain performance scores with 5-fold cross-validation on each target task. The architectures slightly vary across task types. For classification, regression, and multiple-choice target tasks, proxy models predict the label or answer choice. For sequence tagging tasks, each token in a sequence represents a training instance of D_T^M, with the tag being the class label. Since this would increase the total number of training examples, we randomly select 1000 embedded examples from D_T, to maintain equal sizes of D_T^M across all target tasks. We do not study proxy models on extractive QA tasks as they cannot directly be transformed into classification tasks.

TaskEmb. We perform standard fine-tuning of randomly initialized adapter modules within the pre-trained LM to obtain task embeddings.

FS-TaskEmb. We follow the setup of FSFT by training for one epoch (50 update steps).

5.2 Metrics

We compute the *NDCG* (Järvelin and Kekäläinen, 2002), a widely used information retrieval metric that evaluates a ranking with attached relevances (which correspond to our transfer results of §3).

Furthermore, we calculate *Regret@k* (Renggli et al., 2020), which measures the relative performance difference between the top k selected intermediate tasks and the optimal intermediate task:

$$\text{Regret}_k = \frac{\overbrace{\max_{s\in\mathcal{S}} \mathbf{E}[T(s,t)]}^{O(\mathcal{S},t)} - \overbrace{\max_{\hat{s}\in\mathcal{S}_k} \mathbf{E}[T(\hat{s},t)]}^{M_k(\mathcal{S},t)}}{O(\mathcal{S},t)}$$

where $T(s,t)$ is the performance on target task t when transferring from intermediate task s. $O(\mathcal{S},t)$ denotes the expected target task performance of an optimal selection. $M_k(\mathcal{S},t)$ is the highest performance on t among the k top-ranked intermediate tasks of the tested selection method. We take the difference between both measures and normalize it by the optimal target task performance to obtain our final relative notion of regret.[9]

6 Experimental Results

Table 2 shows the results when selecting among all available intermediate tasks for BERT and RoBERTa.[10] As expected the *Random* and *Size* baselines do not yield good rankings when selecting among all intermediate tasks.

Access to only $\mathbf{D_S}$ or $\mathbf{M_S}$. These methods typically perform better than our baselines.

[8] As this represents a rather optimistic estimate of the few-shot transfer performance, in Appendix D we also investigate settings in which we train for only 5, 10, or 25 update steps.

[9] We provide more details about our selection of metrics in Appendix C.

[10] Table 5 in the appendix shows results when preferring tasks of the same *type* for BERT and RoBERTa.

		Classification			M. Choice			QA			Tagging			All		
		NDCG	R@1	R@3	NDCG	R@1	R@3	NDCG	R@1	R@3	NDCG	R@1	R@3	NDCG	R@1	R@3
-	Random	0.49	11.09	**5.55**	**0.43**	**13.80**	**6.30**	0.33	**26.95**	**18.98**	**0.60**	**7.65**	**2.82**	**0.47**	**14.09**	**7.70**
	Size	**0.53**	**10.80**	6.26	0.39	14.89	11.10	**0.36**	33.18	33.18	0.48	8.44	8.44	0.45	15.55	12.87
D_S	SEmb	**0.82**	**0.27**	**0.27**	**0.79**	4.47	**0.00**	**0.49**	**17.04**	**7.59**	0.80	**0.47**	0.47	**0.75**	**4.50**	**1.56**
	TextEmb	0.72	2.54	1.20	0.74	**2.94**	**0.00**	0.48	**17.04**	15.34	**0.88**	**0.47**	**0.11**	0.71	4.91	3.25
M_S	FSFT	**0.89**	**0.28**	**0.00**	**0.89**	**0.00**	**0.00**	**0.28**	**21.21**	**18.20**	**0.97**	**0.00**	**0.00**	**0.79**	**3.96**	**3.31**
	kNN	0.83	2.49	0.12	0.76	1.91	1.57	-	-	-	0.88	1.44	0.11	-	-	-
	linear	0.79	2.51	1.00	0.89	**0.00**	**0.00**	-	-	-	0.92	0.28	0.28	-	-	-
D_S, M_S	FS-TaskEmb	**0.87**	**0.19**	**0.19**	**0.73**	**3.03**	**0.83**	**0.28**	**12.90**	**10.38**	**0.88**	**0.19**	**0.19**	**0.73**	**3.28**	**2.22**
	TaskEmb	0.71	14.04	3.08	0.67	6.70	1.92	0.24	30.02	22.40	0.78	31.84	**0.19**	0.63	18.18	5.75

(a) RoBERTa

		Classification			M. Choice			QA			Tagging			All		
		NDCG	R@1	R@3	NDCG	R@1	R@3	NDCG	R@1	R@3	NDCG	R@1	R@3	NDCG	R@1	R@3
-	Random	0.45	**8.61**	6.04	**0.50**	**8.40**	**5.03**	**0.44**	**29.35**	**18.65**	**0.56**	**7.26**	**2.95**	0.48	**12.08**	**7.50**
	Size	**0.51**	11.34	**5.87**	0.50	11.85	7.51	0.42	33.80	33.80	0.48	9.37	9.37	**0.48**	15.20	12.03
D_S	SEmb	0.78	**0.75**	**0.53**	0.59	7.93	**1.25**	**0.88**	**2.98**	**0.00**	**0.79**	**0.56**	0.56	0.75	3.08	**0.63**
	TextEmb	**0.81**	1.26	0.75	**0.60**	**6.77**	1.46	0.86	**2.98**	2.42	0.79	**0.56**	**0.51**	**0.76**	**2.95**	1.20
M_S	FSFT	**0.93**	**0.33**	0.33	0.72	4.16	1.64	**0.39**	**17.07**	**17.07**	0.89	0.65	0.50	**0.77**	**4.48**	**3.76**
	kNN	0.90	1.10	**0.00**	0.68	2.82	1.85	-	-	-	0.94	**0.00**	**0.00**	-	-	-
	linear	0.82	3.66	1.86	**0.76**	**1.35**	**0.86**	-	-	-	**0.96**	**0.00**	**0.00**	-	-	-
D_S, M_S	FS-TaskEmb	**0.92**	**0.62**	**0.00**	**0.72**	5.38	**0.93**	0.66	11.17	**2.07**	**0.82**	1.37	**0.50**	**0.80**	3.97	**0.72**
	TaskEmb	0.83	3.89	2.02	0.72	**4.19**	1.17	**0.67**	**3.61**	3.61	0.73	**1.36**	**0.50**	0.75	**3.46**	1.80

(b) BERT

Table 2: Evaluation of intermediate task rankings produced by different methods for RoBERTa (a) and BERT (b). The table shows the mean *NDCG* and *Regret* scores by target task type. The best score in each group is highlighted in bold, best overall score is underlined. For *NDCG*, higher is better; for *Regret*, lower is better.

TextEmb and SEmb perform on par in most cases.[11] While FSFT outperforms the other approaches in most cases, it comes at the high cost of requiring downloading and fine-tuning all intermediate models for a few steps. This can be prohibitive if we consider many intermediate tasks. If we have access to TextEmb or SEmb information of the intermediate task (i.e., individual vectors distributed as part of a model repository), these techniques yield similar performances at a much lower cost.

Access to both $\mathbf{D_S}$ and $\mathbf{M_S}$. Assuming the availability of *both* intermediate models and intermediate data is the most prohibitive setting. Surprisingly, we find BERT and RoBERTa to behave considerably differently, especially evident for QA tasks. As shown by Vu et al. (2020), TaskEmb performs very well for BERT, however we find that the results of this gradient based approach do not translate to RoBERTa. While these approaches perform best or competitively for all task types using BERT, they considerably underperform all methods when leveraging pre-trained RoBERTa weights. Here, the two much simpler domain embedding methods outperform the *TaskEmb* method based on the FIM.

Summary. We find that simple indicators such as domain similarity are suitable for selecting intermediate pre-training tasks for both BERT and RoBERTa based models. Our evaluated methods are able to efficiently select the best performing intermediate tasks with a *Regret@3* of 0.0 in many cases. Our results, thus, show that the selection methods are able to effectively rank the top tasks with relative certainty, thus considerably reducing the number of necessary experiments.[12]

7 Analysis

Computational Costs. Table 3 estimates the computational costs of each transfer source selection method. *Complexity* shows the required data passes through the model.[13] For the embedding-based approaches, we assume pre-computed embeddings for all intermediate tasks. For TaskEmb, we only train an adapter on the target task for e epochs.

In addition to the complexity, we calculate the required Multiply-Accumulate computations (MAC) for 42 intermediate tasks and one target task with 1000 training examples, each with an average sequence length of 128.[14] Following our experi-

[11] The used SBERT model is trained on NLI and STS-B tasks, which are included in our set of intermediate and target tasks, respectively. A direct comparison between TextEmb and SEmb for the respective classification tasks is thus difficult.

[12] We also find that combining domain and task type match indicators often yield the best overall results, outperforming computationally more expensive methods. See Appendix **??** for more experiments with task type pre-selection.

[13] We neglect computations related to embedding similarities and proxy models as they are cheap compared to model forward/ backward passes.

[14] We recorded MAC with the pytorch-OpCounter package.

Method	**Complexity**	**MACs**
Metadata	1	0
TextEmb/ SEmb	f	1.10E+13
TaskEmb	$(e+1)f+eb$	3.30E+14
kNN/ linear	nf	4.61E+14
FSFT/ FS-TaskEmb	$2nef+neb$	1.38E+15

Table 3: Computational cost of transfer source selection. f denotes a forward pass through all target task examples once, b is the corresponding backward pass, n is the number of source models, and e is the number of full training epochs (for FS approaches $e \leq 1$).

	NDCG	R@1	R@3	R@5
SEmb-BERT$_D$	0.72	5.50	2.07	1.69
SEmb-BERT$_B$	0.72	4.99	1.16	**0.07**
SEmb-BERT$_L$	0.70	6.30	2.12	1.01
SEmb-RoBERTa$_D$	**0.77**	4.60	0.82	0.44
SEmb-RoBERTa$_B$	0.75	4.50	1.56	0.48
SEmb-RoBERTa$_L$	0.74	**3.96**	**0.47**	**0.07**

Table 4: Intermediate SEmb rankings for RoBERTa tasks produced by different model-type variants. *D, B,* and *L* stand for *Distill*, *Base*, and *Large*, respectively. The table shows the mean *NDCG* and *Regret* scores. For *NDCG*, higher is better; for *Regret*, lower is better.

mental setup in §5, we set $e = 15$ for TaskEmb and $e = 1$ for FSFT/ FS-TaskEmb. We find that embedding-based methods require two orders of magnitude fewer computations compared to fine-tuning approaches. The difference may be even larger when we consider more intermediate tasks. Since fine-tuning approaches do not yield gains that would warrant the high computational expense (see §6), we conclude that *SEmb* has the most favorable trade-off between efficiency and effectiveness.

SEmb Model Dependency. We compare different pre-trained sentence-embedding model variants to identify the extent to which SEmb is invariant to such changes. We experiment with BERT and RoBERTa variants of sizes *Distill*, *Base*, and *Large*, and present results for RoBERTa tasks in Table 4.[15] We find that all variants perform comparably, demonstrating that SEmb is a computationally efficient, model-type invariant method for selecting beneficial intermediate tasks.

BERT vs RoBERTa TaskEmb Space. To better understand the TaskEmb performance differences between BERT and RoBERTa models, we visualize the respective embedding spaces using T-SNE in Figure 3. We find that BERT embeddings are clustered much more closely in the vector space than

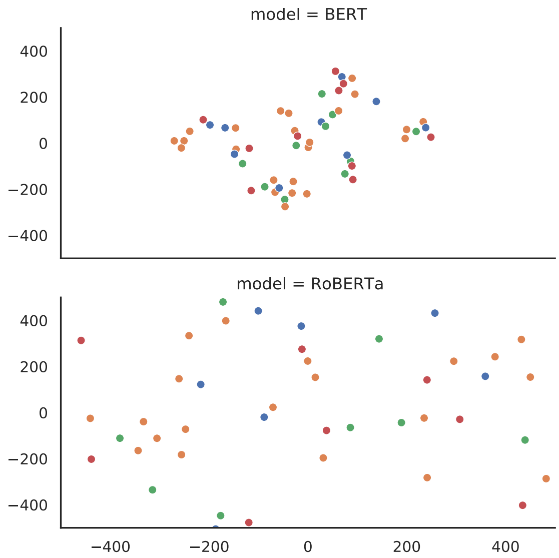

Figure 3: Clustering of BERT and RoBERTa TaskEmbs, respectively, using T-SNE. Colors indicate task types. We compared different random seeds, all of which resulted in similar visualizations.

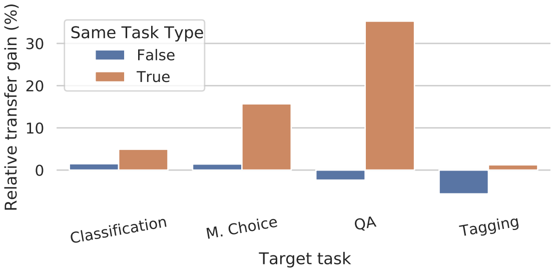

Figure 4: Relative transfer gains for transfer within and across types, split by target task type. Results shown for RoBERTa.

RoBERTa embeddings. While TaskEmbs of BERT also seem to be located in the proximity of related tasks, TaskEmbs of RoBERTa are distributed further apart. This can result in worse performance due to the curse of dimensionality.

Overall, our results and analysis suggest that TaskEmb, unlike SentEmb, considerably depend on the chosen base model.

Within- and Across-Type Transfer. Our experimental setup includes tasks of four different types, i.e. Transformer prediction head structures: *sequence classification/ regression*, *multiple choice*, *extractive question answering* and *sequence tagging*. Figure 4 compares the relative transfer gains within and across these task types for RoBERTa. We see that within-type transfer is consistently stronger across all target tasks. We find the largest differences between within-type and across-type

[15] The full results can be found in Table 7 of the appendix.

transfer for the extractive QA target tasks. These observations may be partly explained by the homogeneity of the included QA intermediate tasks; They overwhelmingly focus on general reading comprehension across multiple domains with paragraphs from Wikipedia or the web as contexts. Tasks of other types more distinctly focus on individual domains and scenarios.

Overall, we find a negative across-type transfer gain (i.e., loss) for *8 out of 11 tested target tasks* (on average). This suggests that task type match between intermediate and target task is a strong indicator for transfer success. Thus, in the next section, we evaluate variants of all methods presented in §4 that prefer intermediate tasks of the same type as the target task.

Pre-Ranking by Task Types. We implement a simple mechanism to ensure that tasks with the same type as the target task are always ranked before tasks of other types during intermediate task selection. Given a task selection method, we first rank all tasks of the same type at the top before ranking tasks of all other types below. Results for applying this mechanism to all presented task selection methods are given for BERT and RoBERTa in Table 5 of the Appendix.

We find that even though the random and *Size* baselines do not yield good rankings when selecting among all intermediate tasks (cf. Table 2), the scores considerably improve when preferring tasks of the same type. In general, we see almost consistent improvements across all task selection methods for both BERT and RoBERTa when implementing pre-ranking by task types. Considering all target tasks and all methods, preferring intermediate tasks of the same type yields improved NDCG scores in 77 of 99 cases.

Further Analysis. We further find that embedding based approaches are sample efficient, while FSFT appproaches are not (§D). We also report results for combining ranking approaches with Rank Fusion, which does not yield consistent improvements over the individual approaches presented before (§E).

8 Conclusion

In this work we have established that intermediate pre-training *can* yield gains in adapter-based setups, however, around 44% of all transfer combinations result in decreased performances. We have consolidated several existing and new methods for efficiently identifying beneficial intermediate tasks. Experimenting with different model types, we find that the previously proposed best performing approaches for BERT do not translate to RoBERTa.

Overall, efficient embedding based methods, such as those relying on pre-computable sentence representations, perform better or often on-par with more expensive approaches. The best methods achieve a *Regret@3* of less than 1% on average, demonstrating that they are effective at efficiently identifying the best intermediate tasks. The approaches evaluated and proposed in this work, thus, enable the automatic identification of beneficial intermediate tasks, deeming exhaustive experimentation on many task-combinations unnecessary. When applied on a broad scale, these methods can contribute to more sustainable (Strubell et al., 2019; Moosavi et al., 2020) and more inclusive (Joshi et al., 2020) natural language processing.

Acknowledgements

Clifton and Jonas are supported by the LOEWE initiative (Hesse, Germany) within the emergenCITY center. Andreas was supported by the German Research Foundation (DFG) as part of the UKP-SQuARE project (grant GU 798/29-1).

We thank Leonardo Ribeiro and the anonymous reviewers for insightful feedback and suggestions on a draft of this paper.

References

Lasha Abzianidze and Johan Bos. 2017. Towards universal semantic tagging. In *IWCS 2017 - 12th International Conference on Computational Semantics - Short Papers, Montpellier, France, September 19 - 22, 2017*. The Association for Computer Linguistics.

Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless C. Fowlkes, Stefano Soatto, and Pietro Perona. 2019. Task2Vec: Task embedding for meta-learning. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 6429–6438. IEEE.

Alan Ansell, Edoardo Maria Ponti, Jonas Pfeiffer, Sebastian Ruder, Goran Glavaš, Ivan Vulić, and Anna Korhonen. 2021. MAD-G: Multilingual Adapter Generation for Efficient Cross-Lingual Transfer. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, Online.

Junwei Bao, Nan Duan, Zhao Yan, Ming Zhou, and Tiejun Zhao. 2016. Constraint-based question answering with knowledge graph. In *COLING 2016, 26th International Conference on Computational*

Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan, pages 2503–2514. Association for Computational Linguistics.

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*.

Joachim Bingel and Anders Søgaard. 2017. 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, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*, pages 164–169. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 632–642. The Association for Computational Linguistics.

Daniel M. Cer, Mona T. Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation, SemEval@ACL 2017, Vancouver, Canada, August 3-4, 2017*, pages 1–14. Association for Computational Linguistics.

Ankush Chatterjee, Kedhar Nath Narahari, Meghana Joshi, and Puneet Agrawal. 2019. SemEval-2019 task 3: EmoContext contextual emotion detection in text. In *Proceedings of the 13th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2019, Minneapolis, MN, USA, June 6-7, 2019*, pages 39–48. Association for Computational Linguistics.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2924–2936. Association for Computational Linguistics.

Arman Cohan, Waleed Ammar, Madeleine van Zuylen, and Field Cady. 2019. Structural scaffolds for citation intent classification in scientific publications. 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 3586–3596. Association for Computational Linguistics.

Gordon V. Cormack, Charles L. A. Clarke, and Stefan Büttcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, Boston, MA, USA, July 19-23, 2009*, pages 758–759. ACM.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL recognising textual entailment challenge. In *Machine Learning Challenges, Evaluating Predictive Uncertainty, Visual Object Classification and Recognizing Textual Entailment, First PASCAL Machine Learning Challenges Workshop, MLCW 2005, Southampton, UK, April 11-13, 2005, Revised Selected Papers*, volume 3944 of *Lecture Notes in Computer Science*, pages 177–190. Springer.

Pradeep Dasigi, Nelson F. Liu, Ana Marasovic, Noah A. Smith, and Matt Gardner. 2019. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 5924–5931. Association for Computational Linguistics.

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.

Aditya Deshpande, Alessandro Achille, Avinash Ravichandran, Hao Li, Luca Zancato, Charless C. Fowlkes, Rahul Bhotika, Stefano Soatto, and Pietro Perona. 2021. A linearized framework and a new benchmark for model selection for fine-tuning. *arXiv preprint*.

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.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases.

In *Proceedings of the Third International Workshop on Paraphrasing, IWP@IJCNLP 2005, Jeju Island, Korea, October 2005, 2005*. Asian Federation of Natural Language Processing.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. 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 2368–2378. Association for Computational Linguistics.

Harrison Edwards and Amos J. Storkey. 2017. Towards a neural statistician. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Andrew S. Gordon, Zornitsa Kozareva, and Melissa Roemmele. 2012. SemEval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *Proceedings of the 6th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2012, Montréal, Canada, June 7-8, 2012*, pages 394–398. The Association for Computer Linguistics.

Suchin Gururangan, Ana Marasovic, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8342–8360.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. 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 2790–2799. PMLR.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 2391–2401. Association for Computational Linguistics.

Shankar Iyer, Nikhil Dandekar, and Kornél Csernai. 2017. First Quora Dataset Release: Question Pairs.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446.

Hadi S. Jomaa, Josif Grabocka, and Lars Schmidt-Thieme. 2019. Dataset2Vec: Learning dataset meta-features. *arXiv preprint*.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. 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, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 252–262. Association for Computational Linguistics.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. SciTaiL: A textual entailment dataset from science question answering. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th Innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5189–5197. AAAI Press.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard H. Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 785–794. Association for Computational Linguistics.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *19th International Conference on Computational Linguistics, COLING 2002, Howard International House and Academia Sinica, Taipei, Taiwan, August 24 - September 1, 2002*.

Yandong Li, Xuhui Jia, Ruoxin Sang, Yukun Zhu, Bradley Green, Liqiang Wang, and Boqing Gong. 2021. Ranking neural checkpoints. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 2663–2673. Computer Vision Foundation / IEEE.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. Linguistic knowledge and transferability of contextual representations. 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 1073–1094. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 142–150. The Association for Computer Linguistics.

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).

Nafise Sadat Moosavi, Angela Fan, Vered Shwartz, Goran Glavaš, Shafiq Joty, Alex Wang, and Thomas Wolf, editors. 2020. *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*. Association for Computational Linguistics, Online.

Cuong Nguyen, Tal Hassner, Matthias W. Seeger, and Cédric Archambeau. 2020. LEEP: A new measure to evaluate transferability of learned representations. 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 7294–7305. PMLR.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4885–4901. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*, pages 115–124. The Association for Computer Linguistics.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021a. AdapterFusion: Non-Destructive Task Composition for Transfer Learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulic, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020a. AdapterHub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 46–54. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020b. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2021b. UNKs Everywhere: Adapting Multilingual Language Models to New Scripts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Online, November , 2021.*

Jason Phang, Thibault Févry, and Samuel R. Bowman. 2018. Sentence encoders on STILTs: Supplementary training on intermediate labeled-data tasks. *arXiv preprint*.

Mohammad Taher Pilehvar and José Camacho-Collados. 2019. WiC: The word-in-context dataset for evaluating context-sensitive meaning representations. 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 1267–1273. Association for Computational Linguistics.

Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel R. Bowman. 2020. Intermediate-task transfer learning with pretrained language models: When and why does it work? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5231–5247. Association for Computational Linguistics.

Joan Puigcerver, Carlos Riquelme Ruiz, Basil Mustafa, Cédric Renggli, André Susano Pinto, Sylvain Gelly, Daniel Keysers, and Neil Houlsby. 2021. Scalable transfer learning with expert models. In *9th International Conference on Learning Representations,* *ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 784–789. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics.

Marek Rei and Helen Yannakoudakis. 2016. Compositional sequence labeling models for error detection in learner writing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics.

Cédric Renggli, André Susano Pinto, Luka Rimanic, Joan Puigcerver, Carlos Riquelme, Ce Zhang, and Mario Lucic. 2020. Which model to transfer? Finding the needle in the growing haystack. *arXiv preprint*.

Anna Rogers, Olga Kovaleva, Matthew Downey, and Anna Rumshisky. 2020. Getting closer to AI complete question answering: A set of prerequisite real tasks. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, the Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, the Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8722–8731. AAAI Press.

Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych. 2021. AdapterDrop: On the Efficiency of Adapters in Transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Online, November , 2021*.

Andreas Rücklé, Jonas Pfeiffer, and Iryna Gurevych. 2020. MultiCQA: Zero-shot transfer of self-supervised text matching models on a massive scale. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 2471–2486. Association for Computational Linguistics.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. How Good is Your Tokenizer? On the Monolingual Performance of Multilingual Language Models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics, ACL 2021, Online, August 1-6, 2021*. Association for Computational Linguistics.

Amrita Saha, Rahul Aralikatte, Mitesh M. Khapra, and Karthik Sankaranarayanan. 2018. DuoRC: Towards complex language understanding with paraphrased reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1683–1693. Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. WinoGrande: An adversarial winograd schema challenge at scale. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, the Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, the Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8732–8740. AAAI Press.

Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 shared task chunking. In *Fourth Conference on Computational Natural Language Learning, CoNLL 2000, and the Second Learning Language in Logic Workshop, LLL 2000, Held in Cooperation with ICGI-2000, Lisbon, Portugal, September 13-14, 2000*, pages 127–132. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4462–4472. Association for Computational Linguistics.

Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. 2018. CARER: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 3687–3697. Association for Computational Linguistics.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel R. Bowman, Miriam Connor, John Bauer, and Christopher D. Manning. 2014. A gold standard dependency corpus for english. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014*, pages 2897–2904. European Language Resources Association (ELRA).

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A Meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1631–1642. Association for Computational Linguistics.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

Oyvind Tafjord, Matt Gardner, Kevin Lin, and Peter Clark. 2019. QuaRTz: An open-domain dataset of qualitative relationship questions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 5940–5945. Association for Computational Linguistics.

Alon Talmor and Jonathan Berant. 2019. MultiQA: An empirical investigation of generalization and transfer in reading comprehension. 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 4911–4921. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. 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 4149–4158. Association for Computational Linguistics.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP, Rep4NLP@ACL 2017, Vancouver, Canada, August 3, 2017*, pages 191–200. Association for Computational Linguistics.

Ahmet Üstün, Arianna Bisazza, Gosse Bouma, and Gertjan van Noord. 2020. UDapter: Language Adaptation for Truly Universal Dependency Parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*.

Marko Vidoni, Ivan Vulić, and Goran Glavaš. 2020. Orthogonal language and task adapters in zero-shot cross-lingual transfer. In *arXiv preprint*.

Tu Vu, Tong Wang, Tsendsuren Munkhdalai, Alessandro Sordoni, Adam Trischler, Andrew Mattarella-Micke, Subhransu Maji, and Mohit Iyyer. 2020. Exploring and predicting transferability across NLP tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 7882–7926. Association for Computational Linguistics.

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. 2019a. Can you tell me how to get past sesame street? Sentence-level pretraining beyond language modeling. 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 4465–4476. Association for Computational Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019b. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. 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 3261–3275.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the Workshop: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2018, Brussels, Belgium, November 1, 2018*, pages 353–355. Association for Computational Linguistics.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments.

Transactions of the Association for Computational Linguistics 2019, 7:625–641.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics 2018*, 6:287–302.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. 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, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

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.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380. Association for Computational Linguistics.

Dani Yogatama, Cyprien de Masson d'Autume, Jerome Connor, Tomás Kociský, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, and Phil Blunsom. 2019. Learning and evaluating general linguistic intelligence. *arXiv preprint*.

Amir Roshan Zamir, Alexander Sax, William B. Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. 2018. Taskonomy: Disentangling task transfer learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 3712–3722. IEEE Computer Society.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 93–104. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? 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 4791–4800. Association for Computational Linguistics.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. ReCoRD: Bridging the gap between human and machine commonsense reading comprehension. *arXiv preprint*.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.

A Tasks

Our experiments cover a diverse set of 53 different tasks, broadly divided into the four task types *sequence classification/ regression*, *multiple choice*, *extractive question answering* and *sequence tagging*. Motivated by previous work, we first select tasks that are either part of widely used benchmarks (Wang et al., 2018, 2019b; Talmor and Berant, 2019) or have been successfully applied to sequential transfer setups previously (Sap et al., 2019; Liu et al., 2019a; Pruksachatkun et al., 2020; Vu et al., 2020). Additionally, we include other recent challenging tasks that fall under the four defined task types (e.g. Bhagavatula et al. (2020); Rogers et al. (2020)) and tasks that extend the range of included dataset sizes and task domains. In general, we focus on tasks with publicly available datasets, e.g. via *HuggingFace Datasets*[16]. Our full set of tasks is split into 42 intermediate tasks, presented in Table 8, and 11 target tasks, presented in Table 9.

B Transfer training details

For all our experiments, we use the PyTorch implementations of BERT and RoBERTa in the *HuggingFace Transformers* library (Wolf et al., 2020) as the basis. The adapter implementation is provided by the *AdapterHub* framework (Pfeiffer et al., 2020a) and integrated into the Transformers library [17].

In the light of the number and variety of different tasks used, we don't perform any extensive

[16] `https://huggingface.co/datasets`

[17] `https://github.com/Adapter-Hub/adapter-transformers`

hyperparameter tuning on each training task. We mostly adhere to the hyperparameter recommendations of the Transformers library and Pfeiffer et al. (2021a) for adapter training. Specifically, we train all adapters for a maximum of 15 epochs, with early stopping after 3 epochs without improvements on the validation set. We use a learning rate of 10^{-4} and batch sizes between 4 and 32, depending on the size of the dataset. These settings apply to the adapter training on each intermediate task as well as the subsequent fine-tuning on the target dataset. Additionally, since performances on the low-resource target tasks can be unstable, we perform multiple random restarts (five restarts for RoBERTa and three restarts for BERT) for all training runs on the target tasks, reporting the mean of all restarts. The final scores on each task are computed on the respective tests set if publicly available, otherwise on the validation sets.

Results for RoBERTa are shown in Table 1 and results for BERT are shown in Table 10.

C Metrics for transfer source selection

C.1 NDCG

Following Vu et al. (2020), we compute the *Normalized Discounted Cumulative Gain (NDCG)* (Järvelin and Kekäläinen, 2002), a widely used information retrieval metric that evaluates a ranking with attached relevances. The NDCG is defined via the *Discounted Cumulative Gain (DCG)*, which represents a relevance score for a set of items, each discounted by its position in the ranking. The DCG of a ranking R, accumulated at a particular rank position p, can be computed as:

$$\mathrm{DCG}_p(R) = \sum_{i=1}^{p} \frac{2^{\mathrm{rel}_i} - 1}{\log_2(i+1)}$$

In our setting, R refers to a ranking of intermediate tasks where the relevance rel_i of the intermediate task with rank i is set to the mean target performance when transferring the adapter trained on this intermediate task, i.e. $\mathrm{rel}_i \in [0, 100]$. We always evaluate the full ranking of intermediate tasks, thus we set $p = |\mathcal{S}|$.

The NDCG finally normalizes the DCG of the ranking predicted by the task selection method (R_{pred}) by the perfect ranking produced by the empirical transfer results (R_{true}). An NDCG of 100% indicates a perfect ranking.

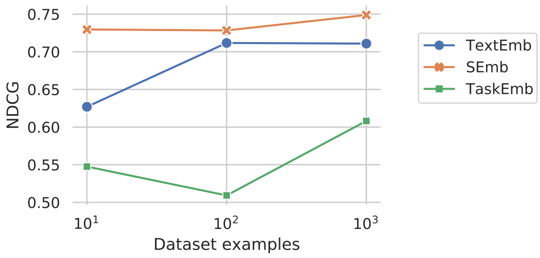

Figure 5: Intermediate task selection performances for feature embedding methods with different data sizes on the target task. Results shown for RoBERTa and averaged over all targets.

$$\mathrm{NDCG}_p(R) = \frac{\mathrm{DCG}_p(R_{pred})}{\mathrm{DCG}_p(R_{true})}$$

C.2 Choice of metrics

Our selection of evaluation metrics combines two measures that both evaluate the quality of the full ranking (*NDCG*) and the top selections of each methods (*Regret*). We prefer this combination of metrics over various other common possible evaluation metrics. We experimented with classical correlation measures such as Spearman rank correlation, finding they give poor indication on the overall quality of a selection method. The Spearman correlation is agnostic to the location within the ranking, thus penalizing mismatches at the bottom of the ranking with the same weight as mismatches at the top. In our setting, the top ranks are more important, making the NDCG which is biased towards correct rankings at the top a better fit. Renggli et al. (2020) further discuss the limitations of correlation as an evaluation metric for task selection.

Vu et al. (2020) use the average predicted rank ρ of the source task with the best target performance as an additional metric. However, this metric does not account for the real target performance difference between the top ranked source tasks across different methods. In a simple example, assume two selection methods A and B assign the top performing source task s_{max} to the same average rank. Further, A ranks a different source task on top which nearly performs on par with s_{max} while B predicts a much weaker source task on top. In this case, we clearly would want to prefer method A over method B. Unlike ρ, our choice of regret as evaluation metric considers these differences.

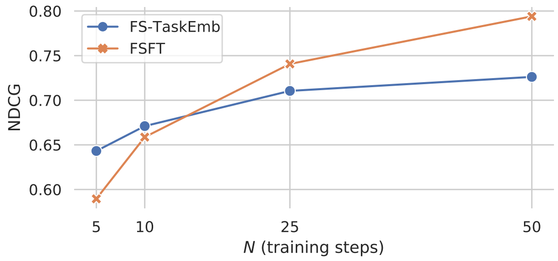

Figure 6: Intermediate task selection performances for fine-tuning methods at different checkpoints. Results shown for RoBERTa and averaged over all targets.

D Sample Efficiency

Embedding-based approaches. Intermediate pre-training can have a larger impact on small target tasks. We therefore analyze and compare the effectiveness of embedding-based approaches with only 10, 100, and 1000 target examples.

Figure 5 plots the results for all feature embedding methods when applied to intermediate task selection for RoBERTa. We find that the quality of the rankings can decrease substantially in the smallest setting with only 10 target examples. *SEmb* is a notable exception, achieving results close to that of the full 1000 examples (73% vs. 74.9% NDCG). With that, SEmb consistently performs above all other methods in all settings.

Few-Shot approaches. We experiment with N $\in \{5, 10, 25, 50\}$ update steps for the fine-tuning methods *FSFT* and *FS-TaskEmb*. Results for RoBERTa are shown in Figure 6. While unsurprisingly, the performance for both methods improves consistently with the number of fine-tuning steps, *FS-TaskEmb* produces superior rankings at earlier checkpoints, however is outperformed by *FSFT* on the long run. The results indicate that updating for < 25 update steps does not provide sufficient evidence to reliably predict the best intermediate tasks.

E Rank Fusion

Vu et al. (2020) use the Reciprocal Rank Fusion algorithm (Cormack et al., 2009) to aggregate the rankings of TextEmb and TaskEmb. further experiment with various combinations of ranks produced by methods of different categories, e.g. *Size* + *SEmb*. Table 6 shows the results for a selection of all possible method combinations when applied to intermediate task selection for RoBERTa.

In a few cases, fusing improves performance over the single-method performances of all included methods (e.g. *TaskEmb+TextEmb*). However, for most cases, rank fusion performance is either roughly on-par with the performance of the best included single method (e.g. *SEmb+TaskEmb*) or even hurts task selection performance sometimes significantly (e.g. *Size+SEmb*). Thus, while adding additional computational overhead to the task selection process, fusing does not yield better performance in general.

F SEmb Model Dependency

The full results of our experiments with sentence-embedding model variants can be found in Table 7. Experiments were conducted on RoBERTa transfer results.

		Classification			M. Choice			QA			Tagging			All		
		NDCG	R@1	R@3	NDCG	R@1	R@3	NDCG	R@1	R@3	NDCG	R@1	R@3	NDCG	R@1	R@3
-	Random-T	**0.54**	**8.81**	**5.09**	0.66	5.50	1.81	0.57	9.46	3.78	0.84	1.54	**0.38**	0.63	6.71	3.10
	Size-T	0.54	10.80	6.26	**0.66**	**4.30**	**1.22**	**0.96**	**0.00**	**0.00**	**0.87**	**0.47**	0.47	**0.71**	**5.18**	**2.69**
D_S	SEmb-T	**0.83**	**0.27**	0.27	**0.92**	**0.00**	**0.00**	0.54	7.76	4.04	0.94	**0.47**	**0.00**	**0.82**	**1.59**	0.83
	TextEmb-T	0.75	2.13	**0.19**	0.89	0.38	**0.00**	**0.62**	**4.04**	**2.05**	**0.95**	**0.47**	**0.00**	0.80	1.70	**0.44**
M_S	FSFT-T	**0.86**	**0.28**	**0.00**	0.93	**0.00**	**0.00**	**0.49**	**10.99**	**10.39**	**0.97**	**0.00**	**0.00**	**0.83**	**2.10**	**1.89**
	kNN-T	0.82	2.49	0.12	0.81	1.91	1.91	-	-	-	0.95	0.11	0.11	-	-	-
	linear-T	0.78	1.84	1.49	**0.96**	**0.00**	**0.00**	-	-	-	0.95	**0.00**	**0.00**	-	-	-
D_S, M_S	FS-TaskEmb-T	**0.88**	**0.19**	**0.00**	0.75	**3.03**	0.83	**0.46**	**12.90**	**4.19**	**0.93**	**0.19**	**0.19**	**0.78**	**3.28**	**1.02**
	TaskEmb-T	0.76	4.82	0.12	**0.76**	3.74	**0.60**	0.45	**12.90**	5.42	0.92	**0.19**	**0.19**	0.73	5.15	1.23

(a) RoBERTa

		Classification			M. Choice			QA			Tagging			All		
		NDCG	R@1	R@3	NDCG	R@1	R@3	NDCG	R@1	R@3	NDCG	R@1	R@3	NDCG	R@1	R@3
-	Random-T	0.50	**8.88**	**5.58**	0.58	6.14	3.31	0.72	8.49	2.95	0.74	2.02	0.91	0.61	6.82	3.63
	Size-T	**0.53**	11.34	5.87	**0.60**	**5.97**	**1.78**	**0.84**	**3.61**	**0.00**	**0.74**	**1.36**	**0.85**	**0.64**	**6.65**	**2.78**
D_S	SEmb-T	0.82	**0.75**	**0.31**	**0.74**	**0.93**	**0.93**	**0.89**	**2.98**	**0.00**	**0.83**	**0.56**	**0.51**	**0.81**	**1.17**	**0.46**
	TextEmb-T	**0.82**	1.26	0.75	0.72	1.95	**0.93**	0.87	**2.98**	2.42	0.82	**0.56**	**0.51**	0.80	1.63	1.06
M_S	FSFT-T	**0.92**	**0.33**	**0.00**	0.73	5.38	1.95	**0.78**	**0.00**	**0.00**	0.88	0.65	**0.50**	**0.84**	**1.70**	**0.62**
	kNN-T	0.91	1.10	**0.00**	0.70	**2.82**	1.46	-	-	-	**0.88**	**0.56**	0.51	-	-	-
	linear-T	0.79	3.00	1.70	**0.73**	2.94	**0.93**	-	-	-	0.85	0.91	0.51	-	-	-
D_S, M_S	FS-TaskEmb-T	**0.95**	**0.00**	**0.00**	0.71	5.38	**0.93**	0.67	11.17	2.07	**0.80**	1.37	**0.50**	**0.81**	3.75	0.72
	TaskEmb-T	0.87	2.13	0.33	**0.72**	**4.19**	**0.93**	**0.77**	**3.61**	**0.00**	0.80	**1.36**	**0.50**	0.80	**2.82**	**0.46**

(b) BERT

Table 5: Evaluation of intermediate task rankings produced by different methods for RoBERTa (a) and BERT (b) when preferring tasks of the same type. The table shows the mean *NDCG* and *Regret* scores by target task type. The best score in each group is highlighted in bold, best overall score is underlined. For *NDCG*, higher is better; for *Regret*, lower is better.

		Classification			M. Choice			QA			Tagging			All		
		NDCG	R@1	R@3	NDCG	R@1	R@3	NDCG	R@1	R@3	NDCG	R@1	R@3	NDCG	R@1	R@3
D_S	Size+SEmb	0.72	5.68	**0.47**	**0.64**	**6.95**	**2.39**	**0.61**	**33.18**	**0.00**	0.60	**7.81**	**1.33**	**0.66**	11.41	**1.06**
	Size+TextEmb	**0.77**	**2.13**	1.13	0.59	6.98	4.86	0.52	**33.18**	2.05	**0.62**	7.94	**1.33**	0.65	**10.15**	2.35
M_S	FSFT+kNN+linear	**0.91**	**0.19**	**0.12**	**0.83**	**1.91**	**0.00**	-	-	-	**0.95**	**0.11**	**0.11**	-	-	-
	Size+FSFT	0.80	1.21	0.19	0.55	9.56	2.39	**0.28**	**57.20**	**18.52**	0.66	5.33	0.47	**0.62**	**14.42**	**4.17**
	Size+kNN	0.73	6.42	**0.12**	0.50	7.07	4.30	-	-	-	0.62	8.44	1.03	-	-	-
	Size+linear	0.70	3.53	2.44	0.61	4.30	2.39	-	-	-	0.66	4.37	0.47	-	-	-
D_S, M_S	FSFT+FS-TaskEmb	0.90	0.46	**0.00**	**0.88**	0.83	**0.00**	0.25	34.25	18.20	**0.93**	**0.19**	0.19	**0.78**	6.66	3.34
	SEmb+TaskEmb	**0.92**	0.27	0.27	0.78	4.88	**0.00**	0.30	28.20	20.15	0.81	0.65	0.19	0.75	6.68	3.80
	Size+FS-TaskEmb	0.83	0.93	0.17	0.61	9.87	1.22	0.31	**16.39**	12.90	0.69	1.54	0.47	0.65	6.29	2.83
	Size+SEmb+FSFT+FS-TaskEmb	0.91	**0.19**	0.19	0.80	4.88	**0.00**	**0.33**	21.21	**10.38**	0.86	0.65	0.47	0.76	**5.38**	**2.04**
	Size+SEmb+linear+TaskEmb	0.85	3.45	0.19	0.79	4.88	**0.00**	-	-	-	0.78	1.54	0.47	-	-	-
	Size+TaskEmb	0.66	5.98	1.15	0.50	9.56	3.37	0.28	43.50	32.70	0.55	64.25	1.53	0.53	24.38	7.56
	Size+TaskEmb+TextEmb	0.81	2.13	0.12	0.67	4.88	2.39	0.32	38.07	32.68	0.68	31.84	0.47	0.66	14.82	6.72
	TaskEmb+FS-TaskEmb	0.79	5.53	0.19	0.71	3.74	1.22	0.25	22.40	22.40	0.85	**0.19**	0.19	0.68	7.14	4.51
	TaskEmb+TextEmb	0.86	1.12	0.12	0.76	4.88	**0.00**	0.30	36.36	28.75	0.83	**0.19**	0.19	0.73	8.39	5.30
	All	0.90	**0.19**	0.19	0.87	**0.38**	**0.00**	-	-	-	0.91	0.65	**0.00**	-	-	-

Table 6: Evaluation of intermediate task rankings produced by method combinations for RoBERTa. The table shows the mean *NDCG* and *Regret* scores by target task type. The best score in each group is highlighted in bold, best overall score is underlined. For *NDCG*, higher is better; for *Regret*, lower is better.

	Classification			M. Choice			QA			Tagging			All		
	NDCG	R@1	R@3	NDCG	R@1	R@3	NDCG	R@1	R@3	NDCG	R@1	R@3	NDCG	R@1	R@3
SEmb-BERT$_D$	0.83	**0.27**	**0.19**	0.67	8.13	2.56	0.48	17.04	7.16	0.84	**0.47**	**0.00**	0.72	5.50	2.07
SEmb-BERT$_B$	0.82	**0.27**	0.24	0.58	12.85	2.56	**0.61**	**7.16**	**2.05**	0.84	**0.47**	**0.00**	0.72	4.99	1.16
SEmb-BERT$_L$	0.82	**0.27**	0.27	0.58	11.07	2.56	0.49	17.04	7.16	0.84	**0.47**	0.11	0.70	6.30	2.12
SEmb-RoBERTa$_D$	**0.84**	**0.27**	**0.19**	0.76	4.86	**0.00**	0.52	17.04	4.04	**0.88**	**0.47**	0.11	**0.77**	4.60	0.82
SEmb-RoBERTa$_B$	0.82	**0.27**	0.27	**0.79**	**4.47**	**0.00**	0.49	17.04	7.59	0.80	**0.47**	0.47	0.75	4.50	1.56
SEmb-RoBERTa$_L$	0.76	3.73	0.27	0.75	**4.47**	**0.00**	0.59	**7.16**	**2.05**	0.82	**0.47**	**0.00**	0.74	**3.96**	**0.47**

Table 7: Evaluation of intermediate task rankings produced by SEmb variations for RoBERTa tasks. *D, B,* and *L* stand for *Distill*, *Base*, and *Large*, respectively. The table shows the mean *NDCG* and *Regret* scores by target task type. The best overall scores are highlighted in bold. For *NDCG*, higher is better; for *Regret*, lower is better.

Name	|Train|	Task	Domain/ Source	Metric(s)	RoBERTa	BERT
Sequence classification/ regression						
MRPC (Dolan and Brockett, 2005)	3.7K	semantic textual similarity	news	acc./ F1	88.48/ 91.53	84.80/ 89.53
SICK (Marelli et al., 2014)	4.4K	NLI	image/ video captions	acc.	89.29	84.24
WiC (Pilehvar and Camacho-Collados, 2019)	5.4K	word sense disambiguation	misc.	acc.	65.52	65.99
TREC (Li and Roth, 2002)	5.5K	question classification	misc.	acc.	96.4	95.60
SciCite (Cohan et al., 2019)	8.2K	citation intents	scientific papers	acc.	84.72	85.26
CoLA (Warstadt et al., 2019)	8.5K	linguistic acceptability	books, journals	Matthews	59.18	62.18
Emotion (Saravia et al., 2018)	16K	emotion classification	Twitter	acc.	94.1	93.5
IMDb (Maas et al., 2011)	25K	sentiment classification	movie reviews	acc.	94.19	91.76
MultiRC (Khashabi et al., 2018)	27K	reading comprehension	misc.	EM/ F1	28.96/ 67.01	18.57/ 66.35
SciTail (Khot et al., 2018)	27K	NLI	science exams	acc.	95.25	93.79
EmoContext (Chatterjee et al., 2019)	30K	emotion classification	crowdsourced	acc.	89	89.74
SST-2 (Socher et al., 2013)	67K	sentiment classification	movie reviews	acc.	94.95	92.20
ReCoRD (Zhang et al., 2018)	101K	commonsense reasoning	news articles	EM/ F1	80.55/ 81.25	64.58/ 65.24
QNLI (Wang et al., 2018)	105K	question-answer NLI	Wikipedia	acc.	92.75	91.14
ANLI (Nie et al., 2020)	163K	NLI	misc.	acc.	41.5	45.42
QQP (Iyer et al., 2017)	364K	semantic textual similarity	Quora	acc./ F1	90.80/ 87.68	90.31/ 87.04
MNLI (Williams et al., 2018)	393K	NLI	misc.	acc. (matched)	87.5	84.20
SNLI (Bowman et al., 2015)	550K	NLI	misc.	acc.	91.13	90.62
Yelp Polarity (Zhang et al., 2015)	560K	sentiment classification	Yelp reviews	acc.	96.61	95.71
Multiple-choice						
QuaRTz (Tafjord et al., 2019)	2.7K	qualitative reasoning	crowdsourced	acc.	79.69	52.86
Cosmos QA (Huang et al., 2019)	25K	commonsense reasoning	crowdsourced	acc.	70.49	60.47
Social IQA (Sap et al., 2019)	33K	commonsense reasoning	knowledge base	acc.	72.21	62.49
HellaSwag (Zellers et al., 2019)	40K	commonsense reasoning	misc.	acc.	62.04	38.20
WinoGrande (Sakaguchi et al., 2020)	41K	coreference resolution	crowdsourced	acc.	63.54	54.38
SWAG (Zellers et al., 2018)	74K	commonsense reasoning	video captions	acc.	83.29	80.06
RACE (Lai et al., 2017)	88K	reading comprehension	English exams	acc.	73.46	65.97
ART (Bhagavatula et al., 2020)	170K	NLI	stories	acc.	73.43	64.36
Extractive question answering						
Quoref (Dasigi et al., 2019)	20K	coreference QA	Wikipedia	EM/ F1	68.73/ 73.22	64.06/ 68.15
WikiHop (Welbl et al., 2018)[18]	51K	multi-hop QA	Wikipedia	EM/ F1	56.48/ 61.71	55.72/ 60.79

DuoRC-s (Saha et al., 2018)[18]	86K	QA	Wikipedia	EM/ F1	59.36/ 67.10	53.19/ 60.73
HotpotQA (Yang et al., 2018)[18]	90K	multi-hop QA	Wikipedia	EM/ F1	57.60/ 71.05	54.81/ 68.49
DuoRC-p (Saha et al., 2018)[18]	100K	QA	IMDb	EM/ F1	49.76/ 53.38	47.76/ 51.31
SQuAD 1.0 (Rajpurkar et al., 2016)[18]	108K	QA	Wikipedia	EM/ F1	84.02/ 91.06	80.26/ 88.08
NewsQA (Trischler et al., 2017)[18]	120K	QA	news articles	EM/ F1	48.70/ 63.93	48.68/ 64.86
SQuAD 2.0 (Rajpurkar et al., 2018)[18]	162K	QA	Wikipedia	EM/ F1	78.39/ 81.47	67.99/ 71.22
Sequence tagging						
NER-WNUT17 (Derczynski et al., 2017)	3.4K	NER	Twitter, forums	F1	55.24	45.27
NER-MITMovie	7.8K	NER	movie reviews	F1	69.29	68.63
Chunk-CoNLL2000 (Sang and Buchholz, 2000)	8.9K	chunking	Penn Treebank	F1	96.35	95.92
POS-EWT (Silveira et al., 2014)	12.5K	POS	web treebank	F1	97.30	96.79
POS-CoNLL2003 (Sang and Meulder, 2003)	14K	POS	news	F1	95.05	93.96
GED-FCE (Rei and Yannakoudakis, 2016)	29K	GED	misc.	$F_{0.5}$	89.79/ 68.12	64.94
ST-PMB (Abzianidze and Bos, 2017)	63K	semantic tagging	meaning bank	acc./ F1	89.50/ 89.38	90.26/ 90.26

Table 8: Overview of intermediate tasks used in our experiments, grouped by task type and ordered by training set size.

Name	**Task**	**Domain/ Source**	**Metric(s)**
Sequence classification/ regression			
BoolQ (Clark et al., 2019)	binary QA	Wikipedia, web queries	acc.
RTE (Dagan et al., 2005)	NLI	news, Wikipedia	acc.
Rotten Tomatoes (Pang and Lee, 2005)	sentiment classification	movie reviews	acc.
STS-B (Cer et al., 2017)	semantic textual similarity	misc.	Spearman
Multiple-choice			
COPA (Gordon et al., 2012)	commonsense reasoning	blogs, encyclopedia	acc.
CS QA (Talmor et al., 2019)	commonsense reasoning	knowledge base	acc.
QuAIL (Rogers et al., 2020)	multiple-choice QA	misc.	acc.
Extractive question answering			
CQ (Bao et al., 2016)[18]	QA	web snippets	EM/ F1
DROP (Dua et al., 2019)[18]	QA	Wikipedia	EM/ F1
Sequence labeling			
DepRel-EWT (Silveira et al., 2014)	relation classification[19]	web treebank	F1
NER-CoNLL2003 (Sang and Meulder, 2003)	NER	news	F1

Table 9: Overview of target tasks used in our experiments, grouped by task type.

[18] We use the version provided in MultiQA (Talmor and Berant, 2019).

[19] Instead of performing full dependency parsing, we only label each token in a sentence with a label corresponding to the dependency relation to its head as this task can be modeled directly as a sequence tagging task.

Task	BoolQ	COPA	CQ	CS QA	CoNLL 2003	DROP	DepRel-EWT	Quail	R. Tomatoes	RTE	STS-B
No Transfer	**63.60**	**67.04**	**23.98**	**51.06**	**83.66**	**14.36**	**76.20**	**54.31**	**85.35**	**60.94**	**86.52**
Avg. Transfer	**67.29**	**67.99**	**25.76**	**50.28**	**82.33**	**13.96**	**75.06**	**54.83**	**85.16**	**67.21**	**86.72**
ANLI	75.84	71.87	26.55	51.73	83.80	13.99	74.66	59.23	84.08	77.38	87.77
ART	68.55	71.27	23.43	51.05	81.21	12.85	73.36	55.82	84.99	69.43	87.13
CoLA	64.96	68.93	24.61	51.57	82.25	13.74	77.02	54.81	85.62	64.26	86.24
CoNLL'00	64.00	57.13	22.49	43.57	82.23	10.62	78.97	51.88	84.99	64.74	84.98
Cosmos QA	71.36	73.73	25.00	52.03	82.29	13.92	75.21	56.84	85.30	75.21	86.92
DuoRC-p	68.17	66.53	36.97	51.79	85.47	17.22	75.03	52.76	85.05	67.63	87.88
DuoRC-s	68.43	70.80	40.97	52.28	85.69	17.13	76.46	53.80	85.21	66.06	87.20
EmoContext	66.79	68.93	22.84	50.42	82.23	13.85	76.88	54.42	85.93	65.82	86.44
Emotion	65.79	66.13	20.42	50.15	82.38	13.65	73.49	54.71	84.74	64.02	86.22
GED-FCE	64.11	68.47	22.37	49.36	82.47	12.37	77.84	54.36	85.55	63.42	86.14
Hellaswag	65.81	68.93	20.74	51.41	81.65	13.56	75.78	55.75	84.93	67.99	86.69
HotpotQA	65.14	65.40	**42.35**	48.02	80.99	17.91	68.55	50.94	83.74	66.67	85.84
IMDb	67.55	69.73	23.56	50.56	82.49	14.06	75.56	54.74	86.62	60.65	86.21
MIT Movie	63.64	66.20	23.69	48.65	84.60	12.10	76.58	53.14	85.99	64.86	85.66
MNLI	**76.26**	69.00	24.32	52.36	82.23	14.45	73.39	59.69	85.99	**78.10**	**88.90**
MRPC	67.88	72.73	23.35	52.36	82.90	14.23	75.59	54.54	85.90	66.91	87.19
MultiRC	70.14	71.73	22.20	**53.18**	83.71	14.05	75.25	56.82	85.58	73.65	87.11
NewsQA	68.71	72.80	40.60	50.34	83.89	18.13	74.18	52.73	84.83	66.79	87.80
POS-Co.'03	63.52	56.93	19.64	44.88	84.72	11.46	80.27	51.85	84.49	61.01	85.25
POS-EWT	64.48	62.87	22.93	48.18	84.86	10.74	**80.84**	53.16	84.08	64.26	86.23
QNLI	69.09	71.73	35.15	52.72	83.47	15.79	76.76	56.45	84.40	69.07	88.08
QQP	68.88	73.80	22.94	48.89	81.34	11.42	72.63	53.77	85.24	70.16	88.02
QuaRTz	64.14	63.40	22.84	51.87	83.45	14.16	76.13	54.37	85.77	60.89	86.40
Quoref	66.92	**75.40**	34.47	50.83	85.45	17.47	77.94	54.50	84.62	67.03	87.46
RACE	73.00	72.80	21.23	49.93	72.29	12.90	69.47	**61.09**	85.87	73.41	88.29
ReCoRD	62.17	61.13	27.86	50.26	81.59	12.69	70.01	56.16	84.83	65.70	85.58
SICK	67.32	71.93	21.18	52.63	82.99	14.37	75.93	55.36	85.24	67.63	87.22
SNLI	71.07	68.80	15.26	46.82	67.23	10.22	61.90	53.68	82.65	73.04	85.36
SQuAD	69.54	68.47	37.45	52.42	**85.73**	18.80	77.24	53.70	84.96	68.95	87.13
SQuAD 2.0	70.18	68.13	39.30	52.63	85.51	**19.05**	77.39	54.87	85.40	68.71	87.34
SST-2	66.99	69.67	21.46	50.31	80.08	12.37	74.37	55.61	**91.78**	63.54	86.36
ST-PMB	64.51	54.47	19.21	41.47	84.84	10.79	79.46	50.52	83.08	62.09	85.03
SWAG	65.26	66.33	22.33	52.63	83.55	13.93	76.50	56.41	85.21	71.36	86.86
SciCite	65.74	67.93	22.08	51.11	83.69	14.00	76.63	55.07	85.46	58.72	86.72
SciTail	70.23	72.00	21.84	52.58	83.11	14.23	76.39	55.99	85.65	72.20	87.70
Social IQA	70.60	74.07	21.74	52.61	78.50	13.77	76.61	57.46	84.96	71.72	87.02
TREC	64.48	67.60	22.83	52.03	82.60	14.09	77.30	55.45	85.58	64.62	86.53
WNUT17	64.20	64.73	22.45	49.85	84.49	12.17	75.78	54.00	84.83	63.30	86.54
WiC	64.04	72.13	22.11	50.31	83.46	14.11	74.76	55.25	84.99	62.58	86.53
WikiHop	62.95	62.07	39.02	48.24	84.16	15.47	71.11	51.96	84.30	65.82	85.47
WinoGrande	67.92	65.47	20.31	49.09	81.93	13.98	73.65	55.14	83.02	69.55	86.91
Yelp Polarity	66.04	63.40	19.79	48.57	76.51	10.40	69.65	54.16	85.18	63.90	85.76

Table 10: Target task performances for transferring between intermediate tasks (rows) and target tasks (columns) with BERT as base model. The first row '*No Transfer*' shows the baseline performance when training only on the target task without transfer. All scores are mean values over three random restarts.