

TADA: Efficient Task-Agnostic Domain Adaptation for Transformers

Chia-Chien Hung^{1,2,3*}, Lukas Lange³, Jannik Strötgen^{3,4}

¹NEC Laboratories Europe GmbH, Heidelberg, Germany

²Data and Web Science Group, University of Mannheim, Germany

³Bosch Center for Artificial Intelligence, Renningen, Germany

⁴Karlsruhe University of Applied Sciences, Karlsruhe, Germany

Chia-Chien.Hung@necclab.eu

Lukas.Lange@de.bosch.com

jannik.stroetgen@h-ka.de

Abstract

Intermediate training of pre-trained transformer-based language models on domain-specific data leads to substantial gains for downstream tasks. To increase efficiency and prevent catastrophic forgetting alleviated from full domain-adaptive pre-training, approaches such as adapters have been developed. However, these require additional parameters for each layer, and are criticized for their limited expressiveness. In this work, we introduce TADA, a novel task-agnostic domain adaptation method which is modular, parameter-efficient, and thus, data-efficient. Within TADA, we retrain the embeddings to learn domain-aware input representations and tokenizers for the transformer encoder, while freezing all other parameters of the model. Then, task-specific fine-tuning is performed. We further conduct experiments with meta-embeddings and newly introduced meta-tokenizers, resulting in one model per task in multi-domain use cases. Our broad evaluation in 4 downstream tasks for 14 domains across single- and multi-domain setups and high- and low-resource scenarios reveals that TADA is an effective and efficient alternative to full domain-adaptive pre-training and adapters for domain adaptation, while not introducing additional parameters or complex training steps.

1 Introduction

Pre-trained language models (Radford et al., 2018; Devlin et al., 2019) utilizing transformers (Vaswani et al., 2017) have emerged as a key technology for achieving impressive gains in a wide variety of natural language processing (NLP) tasks. However, these pre-trained transformer-based language models (PTLMs) are trained on massive and heterogeneous corpora with a focus on generalizability without addressing particular domain-specific concerns. In practice, the absence of such domain-relevant information can severely hurt performance

in downstream applications as shown in numerous studies (i.a., Zhu and Goldberg, 2009; Ruder and Plank, 2018; Friedrich et al., 2020).

To impart useful domain knowledge, two main methods of domain adaptation leveraging transformers have emerged: (1) *Massive pre-training from scratch* (Beltagy et al., 2019; Wu et al., 2020) relies on large-scale domain-specific corpora incorporating various self-supervised objectives during pre-training. However, the extensive training process is time- and resource-inefficient, as it requires a large collection of (un)labeled domain-specialized corpora and massive computational power. (2) *Domain-adaptive intermediate pre-training* (Gururangan et al., 2020) is considered more light-weight, as it requires only a small amount of in-domain data and fewer epochs continually training on the PTLM from a previous checkpoint. However, *fully pre-training* the model (i.e., updating all PTLM parameters) may result in catastrophic forgetting and interference (McCloskey and Cohen, 1989; Houlsby et al., 2019), in particular for longer iterations of adaptation. To overcome these limitations, alternatives such as *adapters* (Rebuffi et al., 2017; Houlsby et al., 2019), and *sparse fine-tuning* (Guo et al., 2021; Ben Zaken et al., 2022) have been introduced. These approaches, however, are still parameter- and time-inefficient, as they either add additional parameters or require complex training steps and/or models.

In this work, we propose **Task-Agnostic Domain Adaptation** for transformers (TADA), a novel domain specialization framework. As depicted in Figure 1, it consists of two steps: (1) We conduct intermediate training of a pre-trained transformer-based language model (e.g., BERT) on the unlabeled domain-specific text corpora in order to inject domain knowledge into the transformer. Here, we *fix* the parameter weights of the encoder while updating only the weights of the embeddings (i.e., embedding-based domain-adaptive pre-training).

*Research work conducted during internship at Bosch Center for Artificial Intelligence.

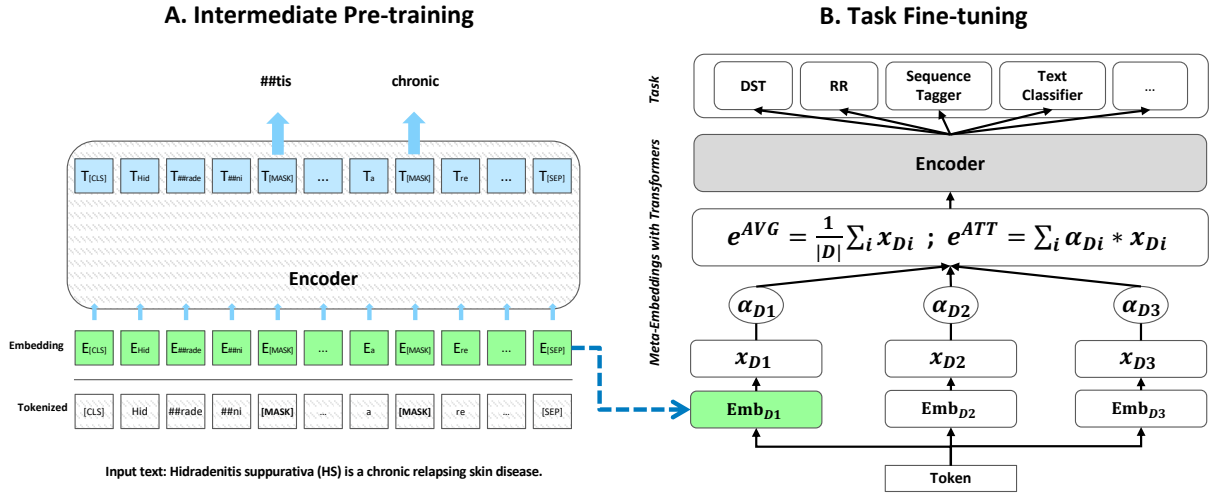


Figure 1: Overview of the TADA framework consisting of two steps. Part A: Domain specialization is performed via embedding-based domain-adaptive intermediate pre-training with Masked Language Modeling (MLM) objective on in-domain data. Part B: The domain-specialized embeddings are then fine-tuned for downstream tasks in single- or multi-domain scenarios with two meta-embeddings methods: average (AVG) and attention-based (ATT).

As a result, we obtain domain-specialized embeddings for each domain with the *shared* encoder from the original PTLM without adding further parameters for domain adaptation. (2) The obtained domain-specialized embeddings along with the encoder can then be fine-tuned for downstream tasks in single- or multi-domain scenarios (Lange et al., 2021b), where the latter is conducted with meta-embeddings (Coates and Bollegala, 2018; Kiela et al., 2018) and a novel meta-tokenization method for different tokenizers.

Contributions. We advance the field of domain specialization with the following contributions:

- (i) We propose a modular, parameter-efficient, and task-agnostic domain adaptation method (TADA) without introducing additional parameters for intermediate training of PTLMs.
- (ii) We demonstrate the effectiveness of our specialization method on four heterogeneous downstream tasks – dialog state tracking (DST), response retrieval (RR), named entity recognition (NER), and natural language inference (NLI) across 14 domains.
- (iii) We propose modular domain specialization via meta-embeddings and show the advantages in multi-domain scenarios.
- (iv) We introduce the concept of meta-tokenization to combine sequences from different tokenizers in a single transformer model and perform the first study on this promising topic.
- (v) We release the code and resources for TADA publicly.¹

¹<https://github.com/boschresearch/TADA>

2 Methods for Domain Specialization

To inject domain-specific knowledge through domain-adaptive pre-training into PTLMs, these models are trained on unlabeled in-domain text corpora. For this, we introduce a novel *embedding-based* intermediate training approach as an alternative to *fully pre-training* and *adapters* (§ 2.1), and further study the effects of domain-specific tokenization (§ 2.2). We then utilize multiple domain-specialized embeddings with our newly proposed meta-tokenizers and powerful meta-embeddings in multi-domain scenarios (§ 2.3 and § 2.4).

2.1 Domain Specialization

Following successful work on *intermediate pre-training* leveraging language modeling for domain-adaptation (Gururangan et al., 2020; Hung et al., 2022a) and language-adaptation (Glavaš et al., 2020; Hung et al., 2022b), we investigate the effects of training with masked language modeling (MLM) on domain-specific text corpora (e.g., clinical reports or academic publications). For this, the MLM loss L_{mlm} is commonly computed as the negative log-likelihood of the true token probability (Devlin et al., 2019; Liu et al., 2019).

$$L_{mlm} = - \sum_{m=1}^M \log P(t_m), \quad (1)$$

where M is the total number of masked tokens in a given text and $P(t_m)$ is the predicted probability of the token t_m over the vocabulary size.

Fully pre-training the model requires adjusting all of the model’s parameters, which can be undesirable due to time- and resource-inefficiency and can dramatically increase the risk of catastrophic forgetting of the previously acquired knowledge (McCloskey and Cohen, 1989; Ansell et al., 2022). To alleviate these issues, we propose a parameter-efficient approach without adding additional parameters during intermediate domain-specialized adaptation: we freeze most of the PTLM parameters and only update the input embeddings weights of the first transformer layer (i.e., the parameters of the embeddings layer) during MLM. With this, the model can learn domain-specific input representations while preserving acquired knowledge in the frozen parameters. As shown in Figure 1, the encoder parameters are fixed during intermediate pre-training while only the embeddings layer parameters are updated.

As a result, after intermediate MLM, multiple embeddings specialized for different domains are all applicable with the *same* shared encoder. As these trained domain-specialized embeddings are easily *portable* to any downstream task, we experiment with their combination in multi-domain scenarios via meta-embeddings methods (Yin and Schütze, 2016; Kiela et al., 2018). We discuss this in more detail in Section § 2.3.

2.2 Domain-Specific Tokenization

Inspired by previous work on domain-specialized tokenizers and vocabularies for language model pre-training (Beltagy et al., 2019; Lee et al., 2019; Yang et al., 2020), we study the domain adaptation of tokenizers for transformers and train domain-specialized variants with the standard WordPiece algorithm (Schuster and Nakajima, 2012) analogously to the BERT tokenizer. As a result, the domain-specialized tokenizers cover more in-domain terms compared to the original PTLM tokenizers. In particular, this reduces the number of out-of-vocabulary tokens, i.e., words that have to be split into multiple subwords, whose embedding quality often does not match the quality of word-level representations (Hedderich et al., 2021).

2.3 Meta-Embeddings

Given n embeddings from different domains D , each domain would have an input representation $x_{Di} \in \mathbb{R}^E$, $1 \leq i \leq n$, where n is the number of domains and E is the dimension of the input embeddings. Here, we consider two variants: *aver-*

aging (Coates and Bollegala, 2018) and *attention-based* meta-embeddings (Kiela et al., 2018).

Averaging merges all embeddings into one vector without training additional parameters by taking the unweighted average:

$$e^{AVG} = \frac{1}{n} \sum_i x_{Di}, \quad (2)$$

In addition, a weighted average with dynamic attention weights α_{Di} can be used. For this, the attention weights are computed as follows:

$$\alpha_{Di} = \frac{\exp(V \cdot \tanh(Wx_{Di}))}{\sum_{k=1}^n \exp(V \cdot \tanh(Wx_{Dk}))}, \quad (3)$$

with $W \in \mathbb{R}^{H \times E}$ and $V \in \mathbb{R}^{1 \times H}$ being parameters that are randomly initialized and learned during training and H is the dimension of the attention vector which is a predefined hyperparameter.

The domain embeddings x_{Di} are then weighted using the learned attention weights α_{Di} into one representation vector:

$$e^{ATT} = \sum_i \alpha_{Di} \cdot x_{Di}, \quad (4)$$

As *Averaging* simply merges all information into one vector, it cannot focus on valuable domain knowledge in specific embeddings. In contrast, the *attention-based* weighting allows for dynamic combinations of embeddings based on their importance depending on the current input token.

As shown in related works, these meta-embeddings approaches suffered from critical mismatch issues when combining embeddings of different sizes and input granularities (e.g., character- and word-level embeddings) that could be addressed by learning additional mappings to the same dimensions on word-level to force all the input embeddings towards a common input space (Lange et al., 2021a).

Our proposed method prevents these issues by (a) keeping the input granularity fixed, which alleviates the need for learning additional mappings, and (b) locating all domain embeddings in the same space immediately after pre-training by freezing the subsequent transformer layers. We compare the results of two variants in Section § 4. More information on meta-embeddings can be found in the survey of Bollegala and O’Neill (2022).

	Domain Text: Acetaminophen is an analgesic drug	=> TOK-1: Ace #ta #mino #phen is an anal #gesic dr #ug (10 subwords)	
		=> TOK-2: Aceta #minophen is an anal #gesic drug (7 subwords)	
Aggregation:	SPACE	DYNAMIC	TRUNCATION
TOK-1	[Ace #ta #mino #phen] is an [anal #gesic] [dr #ug]	[Ace #ta] [#mino #phen] is an anal #gesic [dr #ug]	[Ace] [#mino] is an anal #gesic [dr]
TOK-2	[Aceta #minophen] is an [anal #gesic] drug	Aceta #minophen is an anal #gesic drug	Aceta #minophen is an anal #gesic drug

Table 1: Examples of our proposed aggregation approaches for meta-tokenization: SPACE, DYNAMIC, TRUNCATION for a given text and two different tokenizers (TOK-1, TOK-2). The bottom of the table shows the results after aggregation. $[a\ b \dots z]$ denotes the average of all embedding vectors corresponding to subword tokens a, b, \dots, z .

Task	Dataset	Domain	Background	Train / Dev / Test	License†
DST, RR	MultiWOZ 2.1 (Eric et al., 2020)	Taxi	200 K	1,654 / 207 / 195	MIT
		Restaurant	200 K	3,813 / 438 / 437	
		Hotel	200 K	3,381 / 416 / 394	
		Train	200 K	3,103 / 484 / 494	
		Attraction	200 K	2,717 / 401 / 395	
NLI	MNLI (Williams et al., 2018)	Government	46.0 K	77,350 / 2,000 / 2,000	OANC
		Travel	47.4 K	77,350 / 2,000 / 2,000	OANC
		Slate	214.8 K	77,306 / 2,000 / 2,000	OANC
		Telephone	234.6 K	83,348 / 2,000 / 2,000	OANC
		Fiction	299.5 K	77,348 / 2,000 / 2,000	CC-BY-SA-3.0; CC-BY-3.0
NER	CoNLL (Tjong Kim Sang and De Meulder, 2003)	News	51.0 K	14,987 / 3,466 / 3,684	DUA
	I2B2-CLIN (Uzuner et al., 2011)	Clinical	299.9 K	13,052 / 3,263 / 27,625	DUA
	SEC (Salinas Alvarado et al., 2015)	Financial	4.8 K	825 / 207 / 443	CC-BY-3.0
	LITBANK (Bamman et al., 2019)	Fiction	299.5 K	5,548 / 1,388 / 2,973	CC-BY-4.0
	SOFC (Friedrich et al., 2020)	Science	300.1 K	489 / 123 / 263	CC-BY-4.0

Table 2: Overview of the selected datasets for 4 tasks (DST, RR, NLI, NER) on 14 domains. For each domain, we report the number of collected in-domain texts for domain-adaptive pre-training, as well as the size and license of the downstream dataset. All selected datasets are applicable for *commercial* usage. †License: Open American National Corpus (OANC), Direct Universal Access (DUA), Creative Commons Attribution Share-Alike (CC-BY-SA), Creative Commons Attribution International License (CC-BY).

2.4 Meta-Tokenization for Meta-Embeddings

To utilize our domain-adapted tokenizers in a single model with meta-embeddings, we have to align different output sequences generated by each tokenizer for the same input. This is not straightforward due to mismatches in subword token boundaries and sequence lengths. We thus introduce three different aggregation methods to perform the meta-tokenization:

(a) **SPACE:** We split the input sequence on white-spaces into tokens and aggregate for each tokenizer all subword tokens corresponding to a particular token in the original sequence.

(b) **DYNAMIC:** The shortest sequence from all tokenizers is taken as a reference. Subwords from longer sequences are aggregated accordingly. This assumes that word-level knowledge is more useful than subword knowledge and that fewer word splitting is an indication of in-domain knowledge.

(c) **TRUNCATION:** This method is similar to the DYNAMIC aggregation, but it uses only the first subword for each token instead of computing the average when a token is split into more subwords.

Once the token and subword boundaries are determined, we retrieve the subword embeddings

from the embedding layer corresponding to the tokenizer and perform the aggregation if necessary, in our case averaging all subword embeddings. Examples for each method are shown in Table 1.

3 Experimental Setup

This section introduces four downstream tasks with their respective datasets and evaluation metrics. We further provide details on our models, their hyper-parameters, and the baseline systems.

3.1 Tasks and Evaluation Measures

We evaluate our domain-specialized models and baselines on four prominent downstream tasks: dialog state tracking (DST), response retrieval (RR), named entity recognition (NER), and natural language inference (NLI) with five domains per task. Table 2 shows the statistics of all datasets.

DST is cast as a multi-classification dialog task. Given a dialog history (sequence of utterances) and a predefined ontology, the goal is to predict the output state, i.e., (domain, slot, value) tuples (Wu et al., 2020) like (*restaurant*, *pricerange*, *expensive*). The standard joint goal accuracy is adopted as the evaluation measure: at each dialog turn, it

compares the predicted dialog states against the annotated ground truth. The predicted state is considered accurate if and only if all the predicted slot values match exactly to the ground truth.

RR is a ranking task, relevant for retrieval-based task-oriented dialog systems (Henderson et al., 2019; Wu et al., 2020). Given the dialog context, the model ranks N dataset utterances, including the *true response* to the context (i.e., the candidate set covers one *true* response and $N - 1$ *false* responses). Following Henderson et al. (2019), we report the recall at top rank given 99 randomly sampled false responses, denoted as $R_{100}@1$.

NER is a sequence tagging task, aiming to detect named entities within a sentence by classifying each token into the entity type from a predefined set of categories (e.g., PERSON, ORGANIZATION) including a neutral type (O) for non-entities. Following prior work (Tjong Kim Sang and De Meulder, 2003; Nadeau and Sekine, 2007), we report the strict micro F_1 score.

NLI is a language understanding task testing the reasoning abilities of machine learning models beyond simple pattern recognition. The task is to determine if a *hypothesis* logically follows the relationship from a *premise*, inferred by ENTAILMENT (true), CONTRADICTION (false), or NEUTRAL (undefined). Following Williams et al. (2018), accuracy is reported as the evaluation measure.

3.2 Background Data for MLM-Specialization

We take unlabeled background datasets from the original or related text sources to specialize our models with domain-adaptive pre-training (details are available in Appendix C). For MLM training, we randomly sample up to 200K domain-specific sentences² and dynamically mask 15% of the subword tokens following Liu et al. (2019).

3.3 Models and Baselines

We experiment with the most widely used PTLM: BERT (Devlin et al., 2019) for NER and NLI. For DST and RR as dialog tasks, we experiment with BERT and TOD-BERT (Wu et al., 2020) following Hung et al. (2022a) for comparing general- and task-specific PTLMs.³ We want to highlight that

²Except for four low-resource domains. For these, we randomly sample 44K (GOVERNMENT, TRAVEL, NEWS) and 4.5K (FINANCIAL) respectively.

³We use the pre-trained models from HuggingFace: bert-base-uncased (NLI, NER) and bert-base-cased, TODBERT/TOD-BERT-JNT-V1 (RR, DST).

our proposed method can be easily applied to any existing PTLM. As baselines, we report the performance of the non-specialized variants and compare them against (a) full pre-training (Gururangan et al., 2020), (b) adapter-based models (Houlsby et al., 2019), and (c) our domain-specialized PTLM variants trained with TADA.

3.4 Hyperparameters and Optimization

During MLM training, we fix the maximum sequence length to 256 (DST, RR) and 128 (NER, NLI) subwords and do lowercasing. We train for 30 epochs in batches of 32 instances and search for the optimal learning rate among the following values: $\{5 \cdot 10^{-5}, 1 \cdot 10^{-5}, 1 \cdot 10^{-6}\}$. Early stopping is applied on the development set performance (patience: 3 epochs) and the cross-entropy loss is minimized using AdamW (Loshchilov and Hutter, 2019). For DST and RR, we follow the hyperparameter setup from Hung et al. (2022a). For NLI, we train for 3 epochs in batches of 32 instances. For NER, we train 10 epochs in batches of 8 instances. Both tasks use a fixed learning rate of $5 \cdot 10^{-5}$.

4 Evaluation Results

For each downstream task, we first conduct experiments in a single-domain scenario, i.e., training and testing on data from the same domain, to show the advantages of our proposed approach of task-agnostic domain-adaptive embedding-based pre-training and tokenizers (§ 4.1). We further consider the combination of domain-specialized embeddings with meta-embeddings variants (Coates and Bollegala, 2018; Kiela et al., 2018) in a multi-domain scenario, where we jointly train on data from all domains of the respective task (§ 4.2).

4.1 Single-Domain Evaluation

We report downstream performance for the single-domain scenario in Table 3, with each subtable being segmented into three parts: (1) at the top, we show baseline results (BERT, TOD-BERT) without any domain specialization; (2) in the middle, we show results of domain-specialized PTLMs via full domain-adaptive training and the adapter-based approach; (3) the bottom of the table contains results of our proposed approach specializing only the embeddings and the domain-specific tokenization.

In both DST and RR, TOD-BERT outperforms BERT due to its training for conversational knowledge. By further domain-adaptive pre-training with

Model	DST						RR					
	Taxi	Restaurant	Hotel	Train	Attraction	Avg.	Taxi	Restaurant	Hotel	Train	Attraction	Avg.
BERT	23.87	35.44	30.18	41.93	29.77	32.24	23.25	37.61	38.97	44.53	48.47	38.57
TOD-BERT	30.45	43.58	36.20	48.79	42.70	40.34	45.68	57.43	53.84	60.66	60.26	55.57
BERT (MLM-FULL)	23.74	37.09	32.77	40.96	36.66	34.24	31.37	53.08	45.41	51.66	52.23	46.75
TOD-BERT (MLM-FULL)	29.94	43.14	36.11	47.61	41.54	39.67	41.77	55.27	50.60	55.17	54.62	51.49
BERT (MLM-ADAPT)	22.52	40.49	31.90	42.17	35.05	34.43	32.84	44.01	39.15	38.43	45.05	39.90
TOD-BERT (MLM-ADAPT)	32.06	44.06	36.74	48.84	43.50	41.04	49.08	58.18	55.55	59.46	60.26	56.51
BERT (MLM-EMB)	22.39	31.26	25.75	41.00	34.02	30.88	40.89	54.24	47.30	52.18	56.50	50.22
TOD-BERT (MLM-EMB)	32.00	43.47	36.67	47.34	42.80	40.46	47.08	57.71	55.65	60.72	60.39	56.31
TOD-BERT (MLM-EMBTOK-S)	33.03	41.14	36.77	47.50	40.77	39.84	50.41	58.97	56.48	62.63	59.56	57.61
TOD-BERT (MLM-EMBTOK-X)	32.55	44.60	36.92	47.27	43.58	40.98	50.77	60.40	56.87	62.11	60.89	58.21

Model	NLI					NER						
	Government	Telephone	Fiction	Slate	Travel	Avg.	Financial	Fiction	News	Clinical	Science	Avg.
BERT	79.07	78.18	76.63	73.40	77.33	76.92	90.56	72.09	90.04	85.91	78.23	83.44
BERT (MLM-FULL)	80.82	81.43	76.43	71.97	77.78	77.69	90.53	72.33	90.62	86.18	78.19	83.57
BERT (MLM-ADAPT)	75.58	73.70	72.33	67.11	72.42	72.23	76.62	63.82	89.17	80.64	61.65	74.38
BERT (MLM-EMB)	80.77	80.42	79.27	73.50	77.94	78.38	90.38	71.79	90.67	85.82	78.82	83.50
BERT (MLM-EMBTOK-S)	80.57	79.15	78.51	72.94	77.28	77.69	87.49	69.90	89.55	85.53	79.39	82.37
BERT (MLM-EMBTOK-X)	81.08	80.16	78.97	73.15	77.68	78.21	89.27	69.77	89.21	85.31	77.33	82.18

Table 3: Results of our single-domain models with domain-specialized embeddings and tokenizers on four tasks.

full MLM training (MLM-FULL), TOD-BERT’s performance decreases (i.e., -4% for RR and -0.8% for DST compared to TOD-BERT). It is argued that full MLM domain specialization has negative interference: while TOD-BERT is being trained on domain data during intermediate pre-training, the model is forgetting the conversational knowledge obtained during the initial dialogic pre-training stage (Wu et al., 2020). The hypothesis is further supported by the observations for the adapter-based method which gains slight performance increases.

Our proposed embedding-based domain-adaptation (MLM-EMB) yields similar performance gains as specialization with adapters for TOD-BERT on average. Inspired by previous work on domain-specialized subtokens for language model pre-training (Beltagy et al., 2019; Yang et al., 2020), we additionally train domain-specific tokenizers (MLM-EMBTOK) with the WordPiece algorithm (Schuster and Nakajima, 2012). The training corpora are either obtained from only background corpora (S) or from the combination of background and training set of each domain (X). Further, our domain-specialized tokenizers coupled with the embedding-based domain-adaptive pre-training exhibit similar average performance for DST and outperform the state-of-the-art adapters and all other methods for RR.

Similar findings are observed for NLI and NER. MLM-EMB compared to MLM-FULL results in +0.7% performance gains in NLI and reaches similar average gains in NER. Especially for NLI, the domain-specialized tokenizers (MLM-EMBTOK)

are beneficial in combination with our domain-specialized embeddings, while having considerably fewer trainable parameters. Given that TADA is substantially more efficient and parameter-free (i.e., without adding extra parameters), this promises more sustainable domain-adaptive pre-training.

4.2 Multi-Domain Evaluation

In practice, a single model must be able to handle multiple domains because the deployment of multiple models may not be feasible. To simulate a multi-domain setting, we utilize the domain-specialized embeddings from each domain (§ 4.1) and combine them with meta-embeddings (§ 2.3).

To train a single model for each task applicable to all domains, we concatenate the training sets of all domains for each task. As baselines for DST and RR, we report the performance of BERT and TOD-BERT and a version fine-tuned on the concatenated multi-domain training sets (MLM-FULL). We test the effect of multi-domain specialization in two variants: *averaging* (AVG) and *attention-based* (ATT) meta-embeddings. We conduct experiments to check whether including general-purpose embeddings from TOD-BERT (EMB+MLM-EMBs) is beneficial compared to the one without (MLM-EMBs). The results in Table 4 show that combining domain-specialized embeddings outperforms TOD-BERT in both tasks. In particular, averaging meta-embeddings performs better in RR while attention-based ones work better in DST by 3.8% and 2.2% compared to TOD-BERT, respectively. It is further suggested that combining

Model	DST						RR					
	Taxi	Restaurant	Hotel	Train	Attraction	Avg.	Taxi	Restaurant	Hotel	Train	Attraction	Avg.
BERT	29.10	39.92	36.67	47.63	42.32	39.13	44.87	51.98	49.11	50.15	54.81	50.18
TOD-BERT	34.65	44.24	39.54	51.66	44.24	42.87	50.99	61.53	56.09	58.94	62.76	58.06
BERT (MLM-FULL)	31.94	42.16	38.48	45.37	41.48	39.89	49.59	55.76	54.66	55.59	59.85	55.09
TOD-BERT (MLM-FULL)	32.26	45.70	39.51	51.31	45.92	42.94	53.51	64.44	59.22	62.14	66.49	61.16
(AVG) TOD-BERT (EMB+MLM-EMBs)	37.65	46.06	39.61	51.95	46.95	44.44	52.84	62.56	58.54	60.79	64.87	59.92
(ATT) TOD-BERT (EMB+MLM-EMBs)	35.13	46.86	40.73	51.10	44.76	43.72	53.06	63.18	56.94	60.45	64.13	59.55
(AVG) TOD-BERT (MLM-EMBs)	35.42	46.71	40.82	52.34	47.30	44.52	55.20	64.58	60.39	62.84	66.11	61.82
(ATT) TOD-BERT (MLM-EMBs)	37.35	46.98	41.32	51.92	47.88	45.09	53.73	64.00	59.89	61.54	65.05	60.84

Model	NLI						NER					
	Government	Telephone	Fiction	Slate	Travel	Avg.	Financial	Fiction	News	Clinical	Science	Avg.
BERT	82.88	82.10	80.69	76.01	80.11	80.36	87.68	69.11	89.96	85.76	76.14	81.73
BERT (MLM-FULL)	83.29	81.79	81.11	76.32	79.66	80.43	88.71	69.92	89.69	85.61	80.03	82.79
(AVG) BERT (MLM-EMBs)	83.80	80.87	81.70	77.60	81.30	81.05	87.72	68.78	90.16	85.68	78.22	82.11
(ATT) BERT (MLM-EMBs)	83.50	81.64	81.74	76.68	80.36	80.78	88.89	69.05	90.56	85.43	80.55	82.90

Table 4: Results of our multi-domain models leveraging meta-embeddings on four downstream tasks.

only domain-specialized embeddings (i.e., without adding general-purpose embeddings) works better for both meta-embeddings variants.

These findings are confirmed by NLI and NER experiments. The meta-embeddings applied in our multi-domain scenarios outperform BERT by 0.7 points for NLI and 1.2 points for NER, respectively. An encouraging finding is that two domains (FINANCIAL, SCIENCE) with the smallest number of training resources benefit the most compared to the other domains in the NER task. Such few-shot settings are further investigated in § 5.1.

Overall, we find that the meta-embeddings provide a simple yet effective way to combine several domain-specialized embeddings, alleviating the need of deploying multiple models.

5 Analysis

To more precisely analyze the advantages of our proposed embedding-based domain-adaptive pre-training methods and tokenizers, we study the following: few-shot transfer capability (§ 5.1), the effect of domain-specialized tokenizers on subword tokens (§ 5.2), and the combinations of multiple domain-specialized tokenizers with meta-tokenizers in multi-domain scenarios (§ 5.3).

5.1 Few-Shot Learning

We report few-shot experiments in Table 5 using 1% and 20% of the training data for NLI. We run three experiments with different random seeds to reduce variance and report the mean and standard deviation for these limited data scenarios. MLM-EMB on average outperforms MLM-FULL by 1% in the single-domain scenario, especially for SLATE and TRAVEL domains with

the largest improvements (i.e., 3.3% and 2.7%, respectively). In contrast, the adapter-based models (MLM-ADAPT) perform worse in this few-shot setting. This demonstrates the negative interference (-10%) caused by the additional parameters that cannot be properly trained given the scarcity of task data for fine-tuning. In multi-domain settings, attention-based meta-embeddings on average surpass the standard BERT model in both few-shot setups. Overall, these findings demonstrate the strength of our proposed embedding-based domain-adaptive pre-training in limited data scenarios.

5.2 Domain-Specific Tokenizers

To study whether domain-specialized tokenizers better represent the target domain, we select the development sets and count the number of words that are split into multiple tokens for each tokenizer. The assumption is that the domain-specialized tokenizers allow for word-level segmentation, and thus, word-level embeddings, instead of fallbacks to lower-quality embeddings from multiple subword tokens.

We compare three different tokenizers for each setting: (a) TOK-O: original tokenizer from PTLMs without domain specialization; (b) TOK-S: domain-specialized tokenizer trained on the in-domain background corpus; (c) TOK-X: domain-specialized tokenizer trained on the concatenated in-domain background corpus plus the training set.

Table 6 shows the results on all four tasks averaged across domains. It is evident that TOK-X compared to TOK-O in general significantly reduces the number of tokens split into multiple subwords (-42.6% in DST, RR; -31.7% in NLI; -20.5% in NER). This indicates that the domain-specialized

Model	Government		Telephone		Fiction		Slate		Travel		Avg.	
	1%	20%	1%	20%	1%	20%	1%	20%	1%	20%	1%	20%
BERT	57.62±5.4	75.21±4	49.20±1.9	74.45±3	43.76±2.2	72.90±3	46.70±2.1	67.71±5	54.05±4.0	71.55±4	50.27±2.4	72.36±1
BERT (MLM-FULL)	61.92±1.8	76.07±7	54.53±1.6	75.07±7	49.32±1.4	73.21±6	45.81±0.7	67.26±6	56.56±3.5	72.50±4	53.63±0.5	72.82±4
SD BERT (MLM-ADAPT)	42.88±1.8	67.93±2	41.27±1.1	65.80±2	38.12±1.7	59.53±4	38.91±2.1	54.71±7	40.74±2.8	65.89±6	40.38±1.5	62.78±7
BERT (MLM-EMB)	61.66±1.0	76.61±3	49.86±0.8	75.33±3	48.35±4.1	72.22±6	49.10±2.5	68.26±3	60.27±1.6	72.73±6	53.85±1.7	73.03±1
BERT (MLM-EMBTOK-X)	61.27±1.8	75.75±5	49.20±5.5	74.11±1	49.74±0.8	72.26±8	49.10±1.9	66.51±8	58.99±2.3	72.15±8	53.66±2.0	72.16±1
MD BERT	69.56±3.2	79.49±7	64.80±2.0	77.72±2	61.53±2.5	76.84±7	61.43±2.0	72.64±4	66.40±2.9	76.42±5	64.74±1.8	76.62±2
(AVG) BERT (MLM-EMBs)	70.13±1.3	80.00±2	64.39±1.3	78.28±2	62.24±1.7	76.94±4	62.61±1.6	71.61±3	66.45±1.4	76.21±4	65.16±1.3	76.61±1
(ATT) BERT (MLM-EMBs)	71.21±1.1	79.90±3	65.56±1.4	78.48±1	61.33±1.3	77.34±3	61.99±1.3	72.69±4	66.24±1.7	76.32±5	65.27±1.6	76.95±2

Table 5: Few-shot learning results on NLI task for 1% and 20% of the training data size in single-domain (SD) and multi-domain (MD) scenarios. We report mean and standard deviation of 3 runs with different random seeds.

Dialog State Tracking and Response Retrieval							
Model	Taxi	Restaur.	Hotel	Train	Attract.	Avg.	Diff.
TOK-O	856	1597	1530	1659	1310	1390.4	-
TOK-S	715	1338	951	951	946	1048.2	-24.6%
TOK-X	465	959	753	753	740	798.4	-42.6%
Natural Language Inference							
Model	Govern.	Tele.	Fiction	Slate	Travel	Avg.	Diff.
TOK-O	4095	4221	3379	5094	5883	4534.3	-
TOK-S	1874	3517	3568	3597	3685	3248.2	-28.4%
TOK-X	1873	3522	2426	3683	3984	3097.6	-31.7%
Named Entity Recognition							
Model	Financ.	Fiction	News	Clinical	Science	Avg.	Diff.
TOK-O	397	1930	6357	5121	832	2927.4	-
TOK-S	695	1958	8526	3744	653	3115.2	+6.4%
TOK-X	600	1822	5818	2939	463	2328.4	-20.5%

Table 6: The number of words that have to be split into multiple tokens (\geq subwords) for different tokenizers.

tokenizers cover more tokens on the word-level, and thus, convey more domain-specific information. For domains with smaller background datasets, e.g., FINANCIAL and NEWS, the tokenizers are not able to leverage more word-level information. For example, TOK-S that was trained on the background data performs worse in these domains, as the background data is too small and the models overfit on background data coming from a similar, but not equal source. Including the training corpora helps to avoid overfitting and/or shift the tokenizers towards the dataset word distribution, as TOK-X improves for both domains over TOK-S. The finding is well-aligned with the results in Table 3 (see § 4.1) and supports our hypothesis that word-level tokenization is beneficial.

5.3 Study on Meta-Tokenizers

In Section § 4.2, we experiment with multiple domain-specialized embeddings inside meta-embeddings. These embeddings are, however, based on the original tokenizers and not on the domain-specialized ones. While the latter are considered to contain more domain knowledge and

Model	DST	RR	NLI	NER
(AVG) BERT \ddagger (MLM-EMBs)	44.52	61.82	81.05	82.11
(ATT) BERT \ddagger (MLM-EMBs)	45.09	60.84	80.78	82.90
(AVG) BERT \ddagger (MLM-EMBTOKs-X) dyn	42.16	<u>59.87</u>	79.10	70.73
(AVG) BERT \ddagger (MLM-EMBTOKs-X) space	41.57	58.54	79.51	70.63
(AVG) BERT \ddagger (MLM-EMBTOKs-X) trun	40.26	58.07	79.47	66.66
(ATT) BERT \ddagger (MLM-EMBTOKs-X) dyn	<u>42.73</u>	59.22	79.32	<u>70.83</u>
(ATT) BERT \ddagger (MLM-EMBTOKs-X) space	41.45	58.95	<u>79.93</u>	70.71
(ATT) BERT \ddagger (MLM-EMBTOKs-X) trun	40.82	59.09	79.67	68.41

Table 7: Results of meta-tokenizers in multi-domain experiments with meta-embeddings. Here **bold** indicates the best performance and underline indicates the best-performing meta-tokenization aggregation method. \ddagger BERT variants: TOD-BERT (DST, RR) and BERT (NLI, NER).

achieve better downstream single-domain performance (§ 4.1), it is not straightforward to combine tokenized output by different tokenizers for the same input due to mismatches in subword boundaries and sequence lengths.

Therefore, we further conduct experiments with meta-tokenizers in the meta-embeddings setup following § 2.4. We compare the best multi-domain models with our proposed aggregation approaches. The averaged results across domains are shown in Table 7 (per-domain results are available in Appendix D). Overall, it is observed that the SPACE and DYNAMIC approaches work better than TRUNCATION. However, there is still a performance gap between using multiple embeddings sharing the same sequence from the original tokenizer compared to the domain-specialized tokenizers. Nonetheless, this study shows the general applicability of meta-tokenizers in transformers and suggests future work toward leveraging the domain-specialized tokenizers in meta-embeddings.

6 Related Work

Domain Adaptation. Domain adaptation is a type of transfer learning that aims to enable the

trained model to be generalized into a specific domain of interest (Farahani et al., 2021). Recent studies have focused on neural unsupervised or self-supervised domain adaptation leveraging PTLMs (Ramponi and Plank, 2020), which do not rely on large-scale labeled target domain data to acquire domain-specific knowledge. Gururangan et al. (2020) proposed domain-adaptive intermediate pre-training, continually training PTLM on MLM with domain-relevant unlabeled data, leading to improvements in downstream tasks in both high- and low-resource setups. The proposed approach has been applied to multiple tasks (Glavaš et al., 2020; Lewis et al., 2020) across languages (Hung et al., 2023; Wang et al., 2023), however, requires *fully* pre-training (i.e., update all PTLM parameters) during domain adaptation, which can potentially result in catastrophic forgetting and negative interference (Houlsby et al., 2019; He et al., 2021).

Parameter-Efficient Training. Parameter-efficient methods for domain adaptation alleviate these problems. They have shown robust performance in low-resource and few-shot scenarios (Fu et al., 2022), where only a small portion of parameters are trained while the majority of parameters are frozen and shared across tasks. These lightweight alternatives are shown to be more stable than their corresponding fully fine-tuned counterparts and perform *on par with* or better than expensive fully pre-training setups, including *adapters*, *prompt-based fine-tuning*, and *sparse subnetworks*. *Adapters* (Rebuffi et al., 2017; Houlsby et al., 2019) are additional trainable neural modules injected into each layer of the otherwise frozen PTLM, including their variants (Pfeiffer et al., 2021), have been adopted in both single-domain (Bapna and Firat, 2019) and multi-domain (Hung et al., 2022a) scenarios. *Sparse subnetworks* (Hu et al., 2022; Ansell et al., 2022) reduce the number of training parameters by keeping only the most important ones, resulting in a more compact model that requires fewer parameters for fine-tuning. *Prompt-based fine-tuning* (Li and Liang, 2021; Lester et al., 2021; Goswami et al., 2023) reduces the need for extensive fine-tuning with fewer training examples by adding prompts or cues to the input data. These approaches, however, are still parameter- and time-inefficient, as they add additional parameters, require complex training steps, are less intuitive to the expressiveness, or are limited to the multi-domain scenario for domain

adaptation. A broader overview and discussion of recent domain adaptation methods in low-resource scenarios is given in the survey of Hedderich et al. (2021).

7 Conclusions

In this paper, we introduced TADA – a novel task-agnostic domain adaptation method which is modular and parameter-efficient for pre-trained transformer-based language models. We demonstrated the efficacy of TADA in 4 downstream tasks across 14 domains in both single- and multi-domain settings, as well as high- and low-resource scenarios. An in-depth analysis revealed the advantages of TADA in few-shot transfer and highlighted how our domain-specialized tokenizers take the domain vocabularies into account. We conducted the first study on meta-tokenizers and showed their potential in combination with meta-embeddings in multi-domain applications. Our work points to multiple future directions, including advanced meta-tokenization methods and the applicability of TADA beyond the studied tasks in this paper.

Acknowledgements

We would like to thank the members of the NLP and Neuro-Symbolic AI research group at the Bosch Center for Artificial Intelligence (BCAI) and the anonymous reviewers for their feedback.

Limitations

In this work, we have focused on the efficiency concerns of task-agnostic domain adaptation approaches leveraging pre-trained transformer-based language models. The experiments are conducted on four tasks across 14 domains in both high- and low-resource scenarios. We only consider the methods utilizing pre-collected in-domain unlabeled text corpora for domain-adaptive pre-training. It is worth pointing out that the selected domains are strongly correlated to the selected tasks, which does not reflect the wide spectrum of domain interests. Besides, the datasets are covered only in English to magnify the domain adaptation controlling factors and use cases, while multilinguality would be the next step to explore. We experimented on encoder-only PTLM based on the downstream classification tasks, where the encoder-decoder PTLM would be applicable to different tasks (e.g., natural language generation, summarization, etc.) requiring more computational resources. We hope that

future research builds on top of our findings and extends the research toward more domains, more languages, more tasks, and specifically with the meta-tokenizers for efficiency concerns of domain adaptation approaches.

Ethics Statement

We utilized the pre-collected in-domain unlabeled text corpora to explore the domain-adaptation pre-training approaches with efficiency concerns in this work. Although we carefully consider the data distribution and the selection procedures, the pre-collected background sets for each domain might introduce the potential risk of sampling biases. Moreover, (pre)training, as well as fine-tuning of large-scale PTLMs, might pose a potential threat to the environment (Strubell et al., 2019): in light of the context, the task-agnostic domain adaptation approaches we introduced are aimed at mitigating towards the directions of reducing the carbon footprint of pretrained language models.

References

- Alan Ansell, Edoardo Ponti, Anna Korhonen, and Ivan Vulić. 2022. [Composable sparse fine-tuning for cross-lingual transfer](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1778–1796, Dublin, Ireland. Association for Computational Linguistics.
- David Bamman, Sejal Papat, and Sheng Shen. 2019. [An annotated dataset of literary entities](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2138–2144, Minneapolis, Minnesota. Association for Computational Linguistics.
- Ankur Bapna and Orhan Firat. 2019. [Simple, scalable adaptation for neural machine translation](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1538–1548, Hong Kong, China. Association for Computational Linguistics.
- Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. [SciBERT: A pretrained language model for scientific text](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.
- Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. [BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, Dublin, Ireland. Association for Computational Linguistics.
- Danushka Bollegala and James O’Neill. 2022. [A survey on word meta-embedding learning](#). In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 5402–5409. International Joint Conferences on Artificial Intelligence Organization. Survey Track.
- Joshua Coates and Danushka Bollegala. 2018. [Frustratingly easy meta-embedding – computing meta-embeddings by averaging source word embeddings](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 194–198, New Orleans, Louisiana. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. 2020. [MultiWOZ 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 422–428, Marseille, France. European Language Resources Association.
- Abolfazl Farahani, Sahar Voghoei, Khaled Rasheed, and Hamid R Arabnia. 2021. [A brief review of domain adaptation](#). *Advances in data science and information engineering*, pages 877–894.
- Annemarie Friedrich, Heike Adel, Federico Tomazic, Johannes Hingerl, Renou Benteau, Anika Marusczyk, and Lukas Lange. 2020. [The SOFC-exp corpus and neural approaches to information extraction in the materials science domain](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1255–1268, Online. Association for Computational Linguistics.
- Zihao Fu, Haoran Yang, Anthony Man-Cho So, Wai Lam, Lidong Bing, and Nigel Collier. 2022. [On the effectiveness of parameter-efficient fine-tuning](#). *arXiv preprint arXiv:2211.15583*.

- Goran Glavaš, Mladen Karan, and Ivan Vulić. 2020. [XHate-999: Analyzing and detecting abusive language across domains and languages](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6350–6365, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Koustava Goswami, Lukas Lange, Jun Araki, and Heike Adel. 2023. [SwitchPrompt: Learning domain-specific gated soft prompts for classification in low-resource domains](#). In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2689–2695, Dubrovnik, Croatia. Association for Computational Linguistics.
- Demi Guo, Alexander Rush, and Yoon Kim. 2021. [Parameter-efficient transfer learning with diff pruning](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4884–4896, Online. Association for Computational Linguistics.
- Suchin Gururangan, Ana Marasović, 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*, Online.
- Ruidan He, Linlin Liu, Hai Ye, Qingyu Tan, Bosheng Ding, Liying Cheng, Jiawei Low, Lidong Bing, and Luo Si. 2021. [On the effectiveness of adapter-based tuning for pretrained language model adaptation](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2208–2222, Online. Association for Computational Linguistics.
- Michael A. Hedderich, Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. 2021. [A survey on recent approaches for natural language processing in low-resource scenarios](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2545–2568, Online. Association for Computational Linguistics.
- Matthew Henderson, Ivan Vulić, Daniela Gerz, Iñigo Casanueva, Paweł Budzianowski, Sam Coope, Georgios Spithourakis, Tsung-Hsien Wen, Nikola Mrkšić, and Pei-Hao Su. 2019. [Training neural response selection for task-oriented dialogue systems](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5392–5404, Florence, Italy. Association for Computational Linguistics.
- 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*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [Lora: Low-rank adaptation of large language models](#). In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.
- Chia-Chien Hung, Anne Lauscher, Dirk Hovy, Simone Paolo Ponzetto, and Goran Glavaš. 2023. [Can demographic factors improve text classification? revisiting demographic adaptation in the age of transformers](#). In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1565–1580, Dubrovnik, Croatia. Association for Computational Linguistics.
- Chia-Chien Hung, Anne Lauscher, Simone Paolo Ponzetto, and Goran Glavaš. 2022a. [DS-TOD: Efficient domain specialization for task-oriented dialog](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 891–904, Dublin, Ireland. Association for Computational Linguistics.
- Chia-Chien Hung, Anne Lauscher, Ivan Vulić, Simone Ponzetto, and Goran Glavaš. 2022b. [Multi2WOZ: A robust multilingual dataset and conversational pre-training for task-oriented dialog](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3687–3703, Seattle, United States. Association for Computational Linguistics.
- Douwe Kiela, Changhan Wang, and Kyunghyun Cho. 2018. [Dynamic meta-embeddings for improved sentence representations](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1466–1477, Brussels, Belgium. Association for Computational Linguistics.
- Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. 2021a. [FAME: Feature-based adversarial meta-embeddings for robust input representations](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8382–8395, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. 2022. [CLIN-X: pre-trained language models and a study on cross-task transfer for concept extraction in the clinical domain](#). *Bioinformatics*, 38(12):3267–3274.
- Lukas Lange, Jannik Strötgen, Heike Adel, and Dietrich Klakow. 2021b. [To share or not to share: Predicting sets of sources for model transfer learning](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages

- 8744–8753, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. [BioBERT: a pre-trained biomedical language representation model for biomedical text mining](#). *Bioinformatics*, 36(4):1234–1240.
- Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. [The power of scale for parameter-efficient prompt tuning](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Patrick Lewis, Myle Ott, Jingfei Du, and Veselin Stoyanov. 2020. [Pretrained language models for biomedical and clinical tasks: Understanding and extending the state-of-the-art](#). In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pages 146–157, Online. Association for Computational Linguistics.
- Xiang Lisa Li and Percy Liang. 2021. [Prefix-tuning: Optimizing continuous prompts for generation](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. 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. 2019. [Roberta: A robustly optimized bert pretraining approach](#). *arXiv preprint arXiv:1907.11692*.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). In *International Conference on Learning Representations*.
- Pekka Malo, Ankur Sinha, Pekka Korhonen, Jyrki Walenius, and Pyry Takala. 2014. [Good debt or bad debt: Detecting semantic orientations in economic texts](#). *Journal of the Association for Information Science and Technology*, 65(4):782–796.
- Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.
- David Nadeau and Satoshi Sekine. 2007. [A survey of named entity recognition and classification](#). *Linguistic Investigations*, 30(1):3–26.
- Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. [AdapterFusion: Non-destructive task composition for transfer learning](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503, Online. Association for Computational Linguistics.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. [Improving language understanding by generative pre-training](#).
- Alan Ramponi and Barbara Plank. 2020. [Neural unsupervised domain adaptation in NLP—A survey](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6838–6855, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. [Learning multiple visual domains with residual adapters](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Sebastian Ruder and Barbara Plank. 2018. [Strong baselines for neural semi-supervised learning under domain shift](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1044–1054, Melbourne, Australia. Association for Computational Linguistics.
- Julio Cesar Salinas Alvarado, Karin Verspoor, and Timothy Baldwin. 2015. [Domain adaption of named entity recognition to support credit risk assessment](#). In *Proceedings of the Australasian Language Technology Association Workshop 2015*, pages 84–90, Parramatta, Australia.
- Mike Schuster and Kaisuke Nakajima. 2012. [Japanese and korean voice search](#). In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152.
- 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.
- 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 at HLT-NAACL 2003*, pages 142–147.
- Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2011. [2010 i2b2/va challenge on concepts, assertions, and relations in clinical text](#). *Journal of the American Medical Informatics Association*, 18(5):552–556.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

- Mingyang Wang, Heike Adel, Lukas Lange, Jan-nik Strötgen, and Hinrich Schütze. 2023. [NL-NDE at SemEval-2023 task 12: Adaptive pretraining and source language selection for low-resource multilingual sentiment analysis](#). *arXiv preprint arXiv:2305.00090*.
- Adina Williams, Nikita Nangia, and Samuel 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, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.
- Chien-Sheng Wu, Steven C.H. Hoi, Richard Socher, and Caiming Xiong. 2020. [TOD-BERT: Pre-trained natural language understanding for task-oriented dialogue](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 917–929, Online. Association for Computational Linguistics.
- Yi Yang, Mark Christopher Siy UY, and Allen Huang. 2020. [Finbert: A pretrained language model for financial communications](#). *arXiv preprint arXiv:2006.08097*.
- Wenpeng Yin and Hinrich Schütze. 2016. [Learning word meta-embeddings](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1351–1360, Berlin, Germany. Association for Computational Linguistics.
- Xiaojin Zhu and Andrew B Goldberg. 2009. [Introduction to semi-supervised learning](#). *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130.
- Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. [Aligning books and movies: Towards story-like visual explanations by watching movies and reading books](#). In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

A Computational Information

All the experiments are performed on Nvidia Tesla V100 GPUs with 32GB VRAM and run on a carbon-neutral GPU cluster. The number of parameters and the total computational budget for domain-adaptive pre-training (in GPU hours) are shown in Table 8.

Model	# Trainable Parameters	MLM Budget (in GPU hours)
BERT _‡ (MLM-FULL)	~110 M	~5.5h (NER and NLI), 7.5h (DST and RR)
BERT _‡ (MLM-ADAPT)	~0.9 M	~2.5h (NER and NLI), 3.5h (DST and RR)
BERT _‡ (MLM-EMB)	~24 M	~3.5h (NER and NLI), 4.5h (DST and RR)

Table 8: Overview of the computational information for the domain-adaptive pre-training. ‡BERT variants: BERT (NLI, NER) and TOD-BERT (DST, RR).

B Hyperparameters

Detailed explanations of our hyperparameters are provided in the main paper in Section § 3.4. In our conducted experiments, we only search for the learning rate in domain-adaptive pre-training. The best learning rate depends on the selected domains and methods for each task.

C In-domain Unlabeled Text Corpora

We provide more detailed information on the background datasets that are used for domain-adaptive pre-training in Table 9.

Task	Domain	Background dataset	# Sentences
DST, RR	Taxi	DomainCC corpus from Hung et al. (2022a).	200 K
	Restaurant		200 K
	Hotel		200 K
	Train		200 K
	Attraction		200 K
NLI	Government	The respective part of the OANC corpus.	46.0 K
	Travel		47.4 K
	Slate		214.8 K
	Telephone		234.6 K
	Fiction	The books corpus (Zhu et al., 2015), used as the pre-training data of BERT (Devlin et al., 2019).	299.5 K
NER	News	The Reuters news corpus in NLTK (nltk.corpus.reuters). Similar to the training data of CoNLL (Tjong Kim Sang and De Meulder, 2003).	51.0 K
	Clinical	Pubmed abstracts from clinical publications filtered following Lange et al. (2022).	299.9 K
	Financial	The financial phrase bank from Malo et al. (2014).	4.8 K
	Fiction	Same as NLI FICTION, described above.	299.5 K
	Science	Randomly sampled SemanticScholar abstracts from Biology (70%) and Computer Science (30%). Similar to the pre-training data of SciBERT (Beltagy et al., 2019).	300.1 K

Table 9: Overview of the background datasets and their sizes as reported in Table 2 in the background column. The background datasets are used to train domain-specific tokenizers and domain-adapted embeddings layer.

D Per-Domain Results for Meta-Tokenizers

We provide the results for each domain in our multi-domain experiments with meta-tokenizers and meta-embeddings in Table 10 for DST and RR, and in Table 11 for NLI and NER.

Model	DST						RR					
	Taxi	Restaurant	Hotel	Train	Attraction	Avg.	Taxi	Restaurant	Hotel	Train	Attraction	Avg.
(AVG) TOD-BERT (MLM-EMBs)	35.42	46.71	40.82	52.34	47.30	44.52	55.20	64.58	60.39	62.84	66.11	61.82
(ATT) TOD-BERT (MLM-EMBs)	37.35	46.98	41.32	51.92	47.88	45.09	53.73	64.00	59.89	61.54	65.05	60.84
(AVG) TOD-BERT (MLM-EMBTOKs-X) dyn	32.06	44.12	40.54	49.89	44.21	42.16	52.84	62.54	58.26	61.24	64.46	59.87
(AVG) TOD-BERT (MLM-EMBTOKs-X) space	31.35	44.89	37.27	49.47	44.86	41.57	51.59	62.46	56.44	60.21	61.99	58.54
(AVG) TOD-BERT (MLM-EMBTOKs-X) trun	33.61	43.88	38.20	44.24	41.35	40.26	52.55	61.19	55.55	58.58	62.47	58.07
(ATT) TOD-BERT (MLM-EMBTOKs-X) dyn	34.06	45.01	39.73	50.11	44.73	42.73	51.22	62.08	58.04	61.39	63.35	59.22
(ATT) TOD-BERT (MLM-EMBTOKs-X) space	30.19	42.57	40.23	49.84	44.41	41.45	51.51	61.64	57.30	60.91	63.41	58.95
(ATT) TOD-BERT (MLM-EMBTOKs-X) trun	31.45	43.44	37.08	48.13	44.02	40.82	51.59	62.63	57.97	60.66	62.62	59.09

Table 10: Results of meta-tokenizers in multi-domain experiments with meta-embeddings on two downstream tasks: DST and RR, with joint goal accuracy (%) and $R_{100}@1$ (%) as evaluation metric, respectively. Three meta-tokenization aggregation methods: dynamic (dyn), space (space), truncation (trun), are combined with two meta-embeddings approaches: average (AVG), attention-based (ATT).

Model	NLI						NER					
	Government	Telephone	Fiction	Slate	Travel	Avg.	Financial	Fiction	News	Clinical	Science	Avg.
(AVG) BERT (MLM-EMBs)	83.80	80.87	81.70	77.60	81.30	81.05	87.72	68.78	90.16	85.68	78.22	82.11
(ATT) BERT (MLM-EMBs)	83.50	81.64	81.74	76.68	80.36	80.78	88.89	69.05	90.56	85.43	80.55	82.90
(AVG) BERT (MLM-EMBTOKs-X) dyn	81.08	79.81	80.44	75.35	78.80	79.10	83.26	59.70	75.93	70.42	64.33	70.73
(AVG) BERT (MLM-EMBTOKs-X) space	81.90	81.33	80.49	75.14	78.69	79.51	83.68	61.68	76.39	70.78	60.61	70.63
(AVG) BERT (MLM-EMBTOKs-X) trun	81.44	81.38	79.17	75.86	79.50	79.47	77.99	53.53	74.37	67.08	60.33	66.66
(ATT) BERT (MLM-EMBTOKs-X) dyn	81.70	80.62	80.33	74.78	79.15	79.32	84.64	59.98	76.08	71.30	62.17	70.83
(ATT) BERT (MLM-EMBTOKs-X) space	83.34	81.43	80.23	74.83	79.81	79.93	83.70	62.03	76.04	71.54	60.22	70.71
(ATT) BERT (MLM-EMBTOKs-X) trun	82.37	81.64	78.81	75.65	79.90	79.67	80.33	58.80	74.49	66.92	61.51	68.41

Table 11: Results of meta-tokenizers in multi-domain experiments with meta-embeddings on two downstream tasks: NLI and NER, with accuracy (%) and F_1 (%) as the evaluation metric, respectively. Three meta-tokenization aggregation methods: dynamic (dyn), space (space), truncation (trun), are combined with two meta-embeddings approaches: average (AVG), attention-based (ATT).

ACL 2023 Responsible NLP Checklist

A For every submission:

- A1. Did you describe the limitations of your work?
Limitations section
- A2. Did you discuss any potential risks of your work?
Ethics Statement
- A3. Do the abstract and introduction summarize the paper’s main claims?
Abstract
- A4. Have you used AI writing assistants when working on this paper?
Left blank.

B Did you use or create scientific artifacts?

Section 3 Experimental Setup

- B1. Did you cite the creators of artifacts you used?
Section 3 Experimental Setup
- B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section 3 Experimental Setup
- B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section 3 Experimental Setup
- B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.
- B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 3 Experimental Setup
- B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 3 Experimental Setup

C Did you run computational experiments?

Appendix

- C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

- C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?

Section 3 Experimental Setup and Appendix

- C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?

Section 4 Evaluation Results and Section 5 Analysis

- C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?

Section 3 Experimental Setup

D Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

- D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?

No response.

- D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?

No response.

- D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?

No response.

- D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?

No response.

- D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?

No response.