Recycling GPT-2 and Don’t stop pretraining

Feb 17 2021

As good as new. How to successfully recycle English GPT-2 to make models for other languages (Vries & Nissim, 2020)

- Problem
  - Training GPT-2 from scratch is data and computation intensive
  - Can we adapt English GPT-2 in an efficient way to genetically related languages such as Italian and Dutch(same word order SVO, adj-noun)?
- Solution
  - Retrain lexical embeddings of English GPT-2 on Italian/Dutch data without tuning the transformer layers
  - Bonus – Lexical embeddings for Italian/Dutch are aligned with English and can induce a bilingual lexicon
- Caveat – Training a medium/large size model is still time-consuming
  - Transform relearned lexical embeddings of GPT-2 small to GPT-2 medium
- Results – GPT-2 with relearned lexical embeddings on Italian can generate realistic sentences. Gets better with some finetuning.

As good as new. How to successfully recycle English GPT-2 to make models for other languages (Vries & Nissim, 2020)

- GPT-2
  - Auto-regressive Transformer-decoder base language model for English
  - 4 sizes: **small (sml)**, **medium (med)**, large, extra large
- Pre-training data
  - Italian
    - Same pre-training data as Italian GPT-2 (GepPpeTto) – Wiki (2.8GB) + ItWac (11GB)
  - Dutch
    - Wiki (2GB) + news (2.9GB) + books (6.5GB) + Dutch news (2.1GB)
- Evaluation corpora (e.g., perplexity)
  - Italian - Eval corpora from GepPpeTto
  - Dutch – 22-genre, 500M work SoNaR

As good as new. How to successfully recycle English GPT-2 to make models for other languages (Vries & Nissim, 2020)

- Tokenization
  - Dutch – full pre-training data of GePpeTto
    - Vocab size – balance of large vocabulary vs coverage for uncommon tokens
- Computation
  - 8 x V100 32GB GPU
  - 16-bit automatic mixed-precision training (2-3 times reduction)
  - Split each document into a windows of 128 tokens (for retraining lexical embeddings)
  - Bucketed random sampling (decreases amount of padding within minibatches)
  - Maximum batch size that fit into GPU memory

As good as new. How to successfully recycle English GPT-2 to make models for other languages (Vries & Nissim, 2020)

- Cross-language transfer
  - English GPT-2 uses English BPE vocabulary, not usable for new languages
  - Randomly initialize the lexical embedding layer (first and last layer) + freeze rest of the model + relearn the lexical embeddings on language specific data
- Relearning lexical embeddings
  - Still sample training time but converges faster
  - Sml > med in terms of perplexity
- Vocabulary alignment
  - Relearned Italian/Dutch lexical embeddings are aligned to English embeddings

Model	PPL	
	ita	nld
$\texttt{sml}_{\texttt{rle}}$	44.19	48.85
$\texttt{med}_{\texttt{rle}}$*	-	185.02

Table 1: Perplexity scores on test data for relearned lexical embeddings (`rle`). * `med` training from random initialisation is stopped early after two days of training.

English	Italian	Dutch
while	mentre	terwijl
genes	geni	genen
clothes	vestiti	kleren
musicians	composi[...]	artiesten
permitted	ammessa	toegelaten
Finally	infine	Eindelijk
satisfied	soddisfatto	tevreden
Accuracy:	85%	89%

Table 2: Alignment of closest tokens in the lexical embeddings of $\texttt{sml}_{\texttt{rle}}$ for Italian and Dutch. Accuracy scores are based on a manual evaluation by the authors of 200 random aligned tokens. Semantically correct subword matches are included.

As good as new. How to successfully recycle English GPT-2 to make models for other languages (Vries & Nissim, 2020)

- Text generation
  - Sometime English word order, doesn't produce correct Italian prepositional articles. Get's noun-adjective order for Italian
  - Syntactic features of tokens in addition to semantics

Italian	**Literal English translation**
La prima parte del film venne *distribuito* in Giappone con l'aggiunta della colonna sonora.	The first part of the film was *distributed* in Japan with the addition of the soundtrack.
L'unico motivo *di la* mia insoddisfazione fu il fatto che l'inizio della sua attività [...]	The only reason *of the* my unsatisfaction was the fact that the beginning of-the his/her activity [...]
Il suo nome deriva da un vocabolo arabo.	The his/her name derives from a word Arabic.
Dutch	**Literal English translation**
In een artikel in de Journal of Economicologie (1998), *The New York Times schrijft*:	In an article in the Journal of Economicology (1998), *The New York Times writes*:
Ik kan me niet voorstellen dat mensen van mijn generatie *zijn zo boos op mij te wachten*.	I can me not imagine that people of my generation *are so mad at me to wait*.
Ik heb niets gedaan om mijn moeder te helpen.	I have nothing done to my mother to help.

Table 3: A selection of generated sentences by the `sml` model with Italian and Dutch lexical embeddings. Parts in Italic are ungrammatical in the target language.

As good as new. How to successfully recycle English GPT-2 to make models for other languages (Vries & Nissim, 2020)

- Scaling up complexity
  - Relearning for large models are still time consuming
  - Idea: Use the vocabulary alignment between source and target languages for smaller model can be used to initialize embeddings of larger model
- Approach
  - 50K English words
  - Train: Learn a mapping between a English word from small model to medium model
  - Test: Map the Italian words from small model to medium model
- Learning mapping
  - Regression – Learn W that minimizes Euclidean distance between source and target embedding

As good as new. How to successfully recycle English GPT-2 to make models for other languages (Vries & Nissim, 2020)

- Orthogonal Procrustes
  - Constrain the transformation W to be orthogonal matrix
  - Exact solution, only rotations, preserves monolingual invariance
- Weighted K-NN
  - Unknown target language token = k-nn source embedding space for English and use the distance-weighted sum of these tokens in the target space

Model	Italian		Dutch		
	Int@1k	**PPL**	**Int@1k**	**PPL**	**PPL (1 epoch)**
$\texttt{med}_{\texttt{rle}}$ (1 epoch)	0.38	-	185.02	-	-
$\texttt{sml}_{\texttt{rle}} \xrightarrow{proc} \texttt{med}$	**0.61**	8.12×10^{12}	**0.61**	5.02×10^{12}	52.69
$\texttt{sml}_{\texttt{rle}} \xrightarrow{lstsq} \texttt{med}$	0.56	**364.06**	0.56	**293.61**	**47.57**
$\texttt{sml}_{\texttt{rle}} \xrightarrow{1-nn} \texttt{med}$	0.37	2,764.19	0.36	1,101.59	50.25
$\texttt{sml}_{\texttt{rle}} \xrightarrow{10-nn} \texttt{med}$	0.37	20,715.80	0.35	11,871.66	56.88

Table 4: Scores for different transformation methods. Int@1K are the average 1k nearest English neighbours intersection (int) fractions between `sml` and transformed `med` embeddings. *PPL* is the perplexity on the test sets for Italian and Dutch. *PPL (1 epoch)* indicates the perplexity after one epoch of training, indicating closeness of the transformation to a good local optimum.

As good as new. How to successfully recycle English GPT-2 to make models for other languages (Vries & Nissim, 2020)

- Full model finetuning
  - Sml model - relearn + finetune
  - Med model – relearn + align + finetune
- Automatic Evaluation
  - sml + finetuning is best

Model	PPL	
	ita	nld
$\texttt{sml}_{\texttt{rle}}$	44.19	48.85
$\texttt{sml}_{\texttt{rle}}$ + finetuning	**42.45**	**39.59**
$\texttt{med}_{\texttt{rle}}$	42.51	44.68
$\texttt{GePpeTto}_{\texttt{sml}}$	106.84	-

Table 5: Perplexities of the concatenated test data for the final models. The $\texttt{med}_{\texttt{rle}}$ model is in practice the $\texttt{sml}_{\texttt{rle}} \xrightarrow{lstsq} \texttt{med}$ model. The small Dutch model seems to benefit more from full model finetuning.

Model	Proceedings	News	Legal
$\texttt{sml}_{\texttt{rle}}$	44.47	239.14	52.01
$\texttt{sml}_{\texttt{rle}}$ + fine	**36.35**	**171.83**	**42.92**
$\texttt{med}_{\texttt{rle}}$	40.62	234.52	45.01

Table 7: Perplexities for some SoNaR genres in Dutch. Models rankings are consistent across genres.

Model	Social	News	Legal
$\texttt{sml}_{\texttt{rle}}$	134.64	67.14	16.95
$\texttt{sml}_{\texttt{rle}}$ + fine	**118.19**	**55.63**	15.36
$\texttt{med}_{\texttt{rle}}$	123.64	59.18	**14.95**
$\texttt{GePpeTto}_{\texttt{sml}}$	179.47	80.83	34.71

Table 6: Perplexities for different genres within the Italian test data. Rankings are consistent with Table 5 except for the legal domain.

As good as new. How to successfully recycle English GPT-2 to make models for other languages (Vries & Nissim, 2020)

- Human Evaluation
  - Generations from relearn are easily identifiable
  - Generations from relearn + finetune are not so.

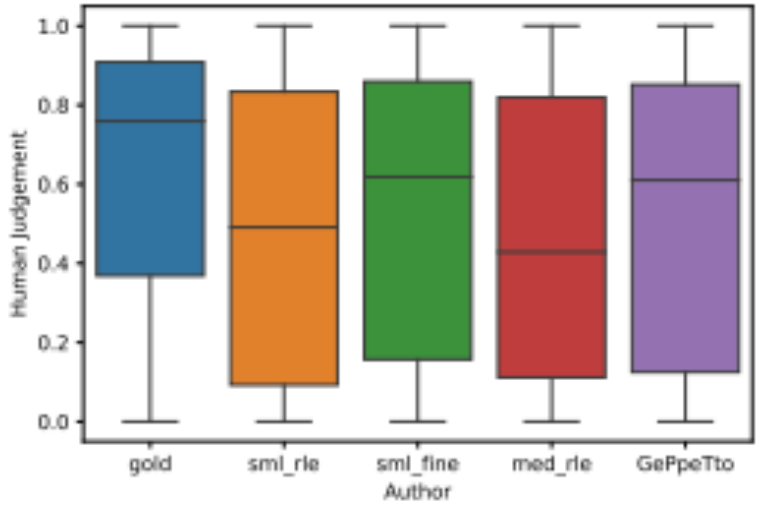

Figure 1: Human judgement scores for Italian texts.

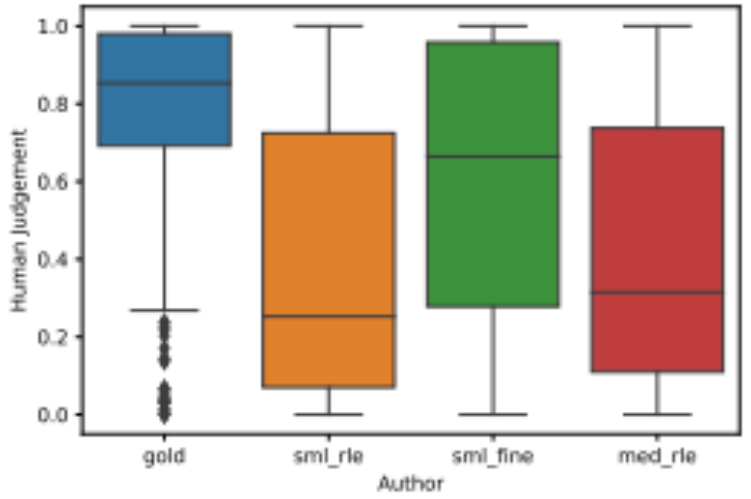

Figure 2: Human judgement scores for Dutch texts.

Don’t Stop Pretraining: Adapt Language Models to Domains and Tasks (Gururangan et al., ACL’20)

- Problem
  - Do latest pretrained LM work universally or is it still helpful to build separate pretrained models for specific domains?
  - How benefits of continued pretraining varies with factors like amount of labeled task data, proximity of target domain to original pretrain corpus?
- Results
  - RoBERTa + pretrain (domain unlabeled) improves for both low/high resource settings (Domain adaptive pretraining - DAPT)
  - RoBERTa + pretrain (task unlabeled) improves with or without DAPT (Task adaptive pretraining)
  - RoBERTa + pretrain (automatically selected task specific unlabeled) cost effective approach

Don’t Stop Pretraining: Adapt Language Models to Domains and Tasks (Gururangan et al., ACL’20)

- Domains
  - Biomedical (BIOMED)
  - Computer science (CS)
  - Newstext (RealNews)
  - Amazon reviews (Amazon)
- Vocab overlap (top 10K most freq. unigrams)
- Expectation: More dissimilar the domain from the pretraining domain, higher the potential for DAPT

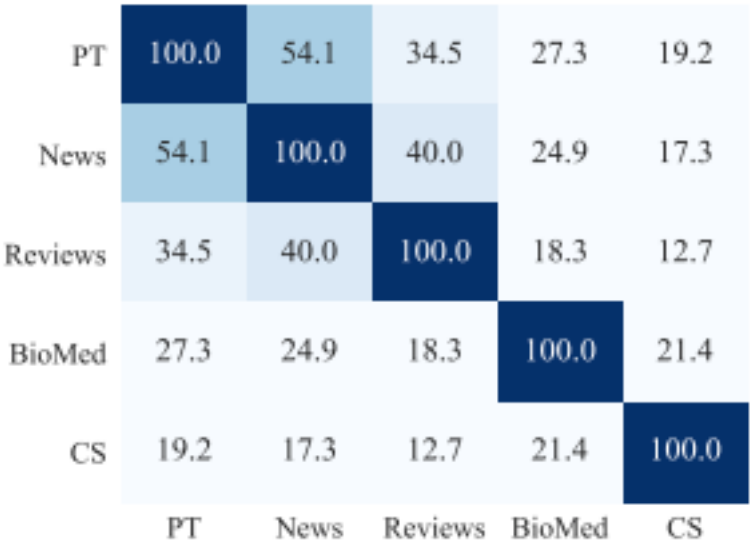

Figure 2: Vocabulary overlap (%) between domains. PT denotes a sample from sources similar to RoBERTa’s pretraining corpus. Vocabularies for each domain are created by considering the top 10K most frequent words (excluding stopwords) in documents sampled from each domain.

Don’t Stop Pretraining: Adapt Language Models to Domains and Tasks (Gururangan et al., ACL’20)

- DAPT MLM loss

Domain	Pretraining Corpus	# Tokens	Size	$\mathcal{L}_{\text{RoB.}}$	$\mathcal{L}_{\text{DAPT}}$
BIOMED	2.68M full-text papers from S2ORC (Lo et al., 2020)	7.55B	47GB	1.32	0.99
CS	2.22M full-text papers from S2ORC (Lo et al., 2020)	8.10B	48GB	1.63	1.34
NEWS	11.90M articles from REALNEWS (Zellers et al., 2019)	6.66B	39GB	1.08	1.16
REVIEWS	24.75M AMAZON reviews (He and McAuley, 2016)	2.11B	11GB	2.10	1.93
ROBERTA (baseline)	see Appendix §A.1	N/A	160GB	‡1.19	-

Table 1: List of the domain-specific unlabeled datasets. In columns 5 and 6, we report ROBERTA’s masked LM loss on 50K randomly sampled held-out documents from each domain before ($\mathcal{L}_{\text{RoB.}}$) and after ($\mathcal{L}_{\text{DAPT}}$) DAPT (lower implies a better fit on the sample). ‡ indicates that the masked LM loss is estimated on data sampled from sources *similar* to ROBERTA’s pretraining corpus.

Don’t Stop Pretraining: Adapt Language Models to Domains and Tasks (Gururangan et al., ACL’20)

- Downstream classification tasks

Domain	Task	Label Type	Train (Lab.)	Train (Unl.)	Dev.	Test	Classes
BIOMED	CHEMPROT	relation classification	4169	-	2427	3469	13
	†RCT	abstract sent. roles	18040	-	30212	30135	5
CS	ACL-ARC	citation intent	1688	-	114	139	6
	SCIERC	relation classification	3219	-	455	974	7
NEWS	HYPERPARTISAN	partisanship	515	5000	65	65	2
	†AGNEWS	topic	115000	-	5000	7600	4
REVIEWS	†HELPFULNESS	review helpfulness	115251	-	5000	25000	2
	†IMDB	review sentiment	20000	50000	5000	25000	2

Table 2: Specifications of the various target task datasets. † indicates high-resource settings. Sources: CHEMPROT (Kringelum et al., 2016), RCT (Dernoncourt and Lee, 2017), ACL-ARC (Jurgens et al., 2018), SCIERC (Luan et al., 2018), HYPERPARTISAN (Kiesel et al., 2019), AGNEWS (Zhang et al., 2015), HELPFULNESS (McAuley et al., 2015), IMDB (Maas et al., 2011).

Dom.	Task	ROBA.	DAPT	¬DAPT
BM	CHEMPROT	$81.9_{1.0}$	$\mathbf{84.2}_{0.2}$	$79.4_{1.3}$
	†RCT	$87.2_{0.1}$	$\mathbf{87.6}_{0.1}$	$86.9_{0.1}$
CS	ACL-ARC	$63.0_{5.8}$	$\mathbf{75.4}_{2.5}$	$66.4_{4.1}$
	SCIERC	$77.3_{1.9}$	$\mathbf{80.8}_{1.5}$	$79.2_{0.9}$
NEWS	HYP.	$86.6_{0.9}$	$\mathbf{88.2}_{5.9}$	$76.4_{4.9}$
	†AGNEWS	$\mathbf{93.9}_{0.2}$	$\mathbf{93.9}_{0.2}$	$93.5_{0.2}$
REV.	†HELPFUL.	$65.1_{3.4}$	$\mathbf{66.5}_{1.4}$	$65.1_{2.8}$
	†IMDB	$95.0_{0.2}$	$\mathbf{95.4}_{0.2}$	$94.1_{0.4}$

Table 3: Comparison of ROBERTA (ROBA.) and DAPT to adaptation to an *irrelevant* domain (¬DAPT). Reported results are test macro-F_1, except for CHEMPROT and RCT, for which we report micro-F_1, following Beltagy et al. (2019). We report averages across five random seeds, with standard deviations as subscripts. † indicates high-resource settings. Best task performance is boldfaced. See §3.3 for our choice of irrelevant domains.

Don’t Stop Pretraining: Adapt Language Models to Domains and Tasks (Gururangan et al., ACL’20)

- Task adaptive pretraining
  - Datasets curated to capture specific tasks of interest tend to cover only a subset of text available within the broader domain
  - CHEMPROT – extracting relations between chemicals and proteins focusing on abstracts from hand selected PubMed articles
  - Hypothesis – Cases where task data is narrowly defined subset of broader domain, pretraining on task dataset itself or data relevant to the task may be helpful
  - TAPT is faster than DAPT

Domain	Task	RoBERTa	Additional Pretraining Phases		
			DAPT	TAPT	DAPT + TAPT
BioMed	ChemProt	$81.9_{1.0}$	$84.2_{0.2}$	$82.6_{0.4}$	$\mathbf{84.4}_{0.4}$
	†RCT	$87.2_{0.1}$	$87.6_{0.1}$	$87.7_{0.1}$	$\mathbf{87.8}_{0.1}$
CS	ACL-ARC	$63.0_{5.8}$	$75.4_{2.5}$	$67.4_{1.8}$	$\mathbf{75.6}_{3.8}$
	SciERC	$77.3_{1.9}$	$80.8_{1.5}$	$79.3_{1.5}$	$\mathbf{81.3}_{1.8}$
News	HyperPartisan	$86.6_{0.9}$	$88.2_{5.9}$	$\mathbf{90.4}_{5.2}$	$90.0_{6.6}$
	†AGNews	$93.9_{0.2}$	$93.9_{0.2}$	$94.5_{0.1}$	$\mathbf{94.6}_{0.1}$
Reviews	†Helpfulness	$65.1_{3.4}$	$66.5_{1.4}$	$68.5_{1.9}$	$\mathbf{68.7}_{1.8}$
	†IMDB	$95.0_{0.2}$	$95.4_{0.1}$	$95.5_{0.1}$	$\mathbf{95.6}_{0.1}$

Table 5: Results on different phases of adaptive pretraining compared to the baseline RoBERTa (col. 1). Our approaches are DAPT (col. 2, §3), TAPT (col. 3, §4), and a combination of both (col. 4). Reported results follow the same format as Table 3. State-of-the-art results we can compare to: ChemProt (84.6), RCT (92.9), ACL-ARC (71.0), SciERC (81.8), HyperPartisan (94.8), AGNews (95.5), IMDB (96.2); references in §A.2.

Don’t Stop Pretraining: Adapt Language Models to Domains and Tasks (Gururangan et al., ACL’20)

- Task adaptive pretraining
  - TAPT optimizes for single task performance, to the detriment of cross-task transfer

BIOMED	RCT	CHEMPROT
TAPT	$87.7_{0.1}$	$82.6_{0.5}$
Transfer-TAPT	$87.1_{0.4}$ (↓0.6)	$80.4_{0.6}$ (↓2.2)

CS	ACL-ARC	SCIERC
TAPT	$67.4_{1.8}$	$79.3_{1.5}$
Transfer-TAPT	$64.1_{2.7}$ (↓3.3)	$79.1_{2.5}$ (↓0.2)

NEWS	HYPERPARTISAN	AGNEWS
TAPT	$89.9_{9.5}$	$94.5_{0.1}$
Transfer-TAPT	$82.2_{7.7}$ (↓7.7)	$93.9_{0.2}$ (↓0.6)

REVIEWS	HELPFULNESS	IMDB
TAPT	$68.5_{1.9}$	$95.7_{0.1}$
Transfer-TAPT	$65.0_{2.6}$ (↓3.5)	$95.0_{0.1}$ (↓0.7)

Table 6: Though TAPT is effective (Table 5), it is harmful when applied *across* tasks. These findings illustrate differences in task distributions within a domain.

Don’t Stop Pretraining: Adapt Language Models to Domains and Tasks (Gururangan et al., ACL’20)

- Curated TAPT
  - Large unlabeled corpus + downsampled to collect task annotations
  - Large unlabeled corpus should have similar distribution to task’s training data
  - Data
    - Downsample RCT to 500 (180K unlabeled)
  - DAPT + Curated TAPT is best
  - Curating large amounts of data from task distribution is extremely beneficial

Pretraining	BIOMED RCT-500	NEWS HYP.	REVIEWS IMDB †
TAPT	$79.8_{1.4}$	$90.4_{5.2}$	$95.5_{0.1}$
DAPT + TAPT	$83.0_{0.3}$	$90.0_{6.6}$	$95.6_{0.1}$
Curated-TAPT	$83.4_{0.3}$	$89.9_{9.5}$	$95.7_{0.1}$
DAPT + Curated-TAPT	$\mathbf{83.8}_{0.5}$	$\mathbf{92.1}_{3.6}$	$\mathbf{95.8}_{0.1}$

Table 7: Mean test set macro-F_1 (for HYP. and IMDB) and micro-F_1 (for RCT-500), with Curated-TAPT across five random seeds, with standard deviations as subscripts. † indicates high-resource settings.

Don’t Stop Pretraining: Adapt Language Models to Domains and Tasks (Gururangan et al., ACL’20)

- Automated Data Selection for TAPT
  - Embed task-relevant data from domain and from task in shared space
  - Select candidates from the domain based on queries using the task data
  - VAMPIRE, BoW LM
  - kNN-TAPT
  - RAND-TAPT

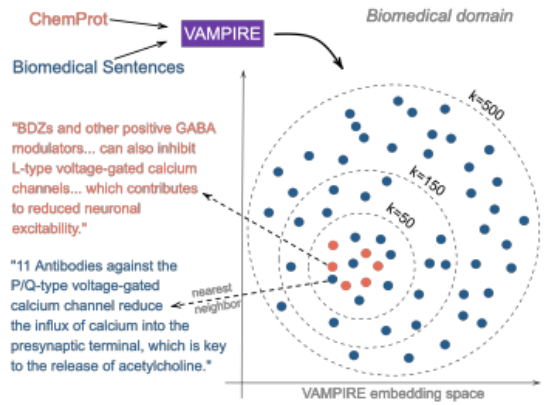

Figure 3: An illustration of automated data selection (§5.2). We map unlabeled CHEMPROT and 1M BIOMED sentences to a shared vector space using the VAMPIRE model trained on these sentences. Then, for each CHEMPROT sentence, we identify k nearest neighbors, from the BIOMED domain.

Pretraining	BIOMED		CS
	CHEMPROT	RCT-500	ACL-ARC
ROBERTA	$81.9_{1.0}$	$79.3_{0.6}$	$63.0_{5.8}$
TAPT	$82.6_{0.4}$	$79.8_{1.4}$	$67.4_{1.8}$
RAND-TAPT	$81.9_{0.6}$	$80.6_{0.4}$	$69.7_{3.4}$
50NN-TAPT	$83.3_{0.7}$	$80.8_{0.6}$	$70.7_{2.8}$
150NN-TAPT	$83.2_{0.6}$	$81.2_{0.8}$	$73.3_{2.7}$
500NN-TAPT	$83.3_{0.7}$	$81.7_{0.4}$	$\mathbf{75.5}_{1.9}$
DAPT	$\mathbf{84.2}_{0.2}$	$\mathbf{82.5}_{0.5}$	$75.4_{2.5}$

Table 8: Mean test set micro-F_1 (for CHEMPROT and RCT) and macro-F_1 (for ACL-ARC), across five random seeds, with standard deviations as subscripts, comparing RAND-TAPT (with 50 candidates) and kNN-TAPT selection. Neighbors of the task data are selected from the domain data.