

PADA: A Prompt-based Autoregressive Approach for Adaptation to Unseen Domains

Eyal Ben-David *

Nadav Oved *

Roi Reichart

Technion, Israel Institute of Technology

{eyalbd12@campus.|nadavo@campus.|roiri@}technion.ac.il

Abstract

Natural Language Processing algorithms have made incredible progress, but they still struggle when applied to out-of-distribution examples. We address a challenging and underexplored version of this domain adaptation problem, where an algorithm is trained on several source domains, and then applied to examples from an unseen domain that is unknown at training time. Particularly, no examples, labeled or unlabeled, or any other knowledge about the target domain are available to the algorithm at training time. We present *PADA*: A Prompt-based Autoregressive Domain Adaptation algorithm, based on the T5 model. Given a test example, *PADA* first generates a unique prompt and then, conditioned on this prompt, labels the example with respect to the NLP task. The prompt is a sequence of unrestricted length, consisting of pre-defined Domain Related Features (DRFs) that characterize each of the source domains. Intuitively, the prompt is a unique signature that maps the test example to the semantic space spanned by the source domains. In experiments with 3 tasks (text classification and sequence tagging), for a total of 14 multi-source adaptation scenarios, *PADA* substantially outperforms strong baselines.¹

1 Introduction

Natural Language Processing (NLP) algorithms are gradually achieving remarkable milestones (Devlin et al., 2019; Lewis et al., 2020a; Brown et al., 2020). However, such algorithms often rely on the seminal assumption that the training set and the test set come from the same underlying distribution. Unfortunately, this assumption often does

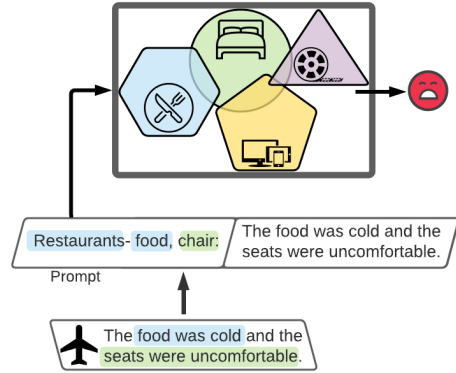


Figure 1: Text classification with PADA. Colored texts signify relation to a specific source domain. *PADA* first generates the domain name, followed by a set of DRFs related to the input example. Then it uses the prompt to predict the task label.

not hold since text may emanate from many different sources, each with unique distributional properties. As generalization beyond the training distribution is still a fundamental challenge, NLP algorithms suffer a significant degradation when applied to out-of-distribution examples.

Domain Adaptation (DA) explicitly addresses the above challenge, striving to improve out-of-distribution generalization of NLP algorithms. DA algorithms are trained on annotated data from source domains, to be effectively applied in a variety of target domains. Over the years, considerable efforts have been devoted to the DA challenge, focusing on various scenarios where the target domain is known at training time (e.g. through labeled or unlabeled data) but is yet under-represented (Roark and Bacchiani, 2003; Daumé III and Marcu, 2006; Reichart and Rappoport, 2007; McClosky et al., 2010; Rush et al., 2012; Schnabel and Schütze, 2014). Still, the challenge of adaptation to *any possible target domain*, that is unknown at training time, is underexplored.

In this work, we focus on adaptation to any tar-

* Both authors equally contributed to this work.

¹Our code and data will be released upon acceptance.

get domain, which we consider a “Holy Grail” of DA (§3). Apart from the pronounced intellectual challenge, it also presents unique modeling advantages as target-aware algorithms typically require training a separate model for each target domain, leading to an inefficient overall solution.

Intuitively, better generalization to unseen domains can be achieved by integrating knowledge from several source domains. We present *PADA: a Prompt-based Autoregressive Domain Adaptation* (§4) algorithm, which utilizes an autoregressive language model (T5, (Raffel et al., 2020)), and includes a novel prompting mechanism that is adapted to multiple source domains. Given a new example, from any unknown domain, the model first generates properties that belong to familiar (source) domains and relate to the given example. Then, the generated properties are used as prompts while the model performs the downstream task.²

In order to generate effective prompts, we draw inspiration from previous work on pivot features (Blitzer et al., 2006; Ziser and Reichart, 2018a; Ben-David et al., 2020) to define sets of Domain Related Features (DRFs, §4.2). DRFs are tokens which are strongly associated with one of the source domains, encoding domain-specific semantics. We leverage the DRFs of the various source domains in order to span their shared semantic space. Together, these DRFs reflect the similarities and differences between the source domains, in addition to domain-specific knowledge.

Consider the task of review sentiment classification (Figure 1). The model is familiar with four source domains: *restaurants*, *home-furniture*, *electronic-devices*, and *movies*. When the model encounters a review, this time from the *airlines* domain, it uses DRFs to project the example into the shared semantic space, via the prompting mechanism. In the given example the DRFs marked in blue and green relate to the *restaurants* and the *home-furniture* domains, respectively. The DRF-based prompt is then used in classification.

We evaluate *PADA* in the multi-source DA setting, where the target domain is unknown during training (§5, 6). We consider two text classification tasks (Rumour Detection and Multi-Genre Natural Language Inference (MNLI)), and a se-

²We use a language model, pre-trained on massive unlabeled data, and it is possible that this model was exposed to text from the source or target domains. Yet, the downstream task training is based only on examples from the source domains without any knowledge of future target domains.

quence tagging task (Aspect Prediction), for a total of 14 DA setups. *PADA* outperforms strong baselines, yielding substantial error reductions.

2 Related Work

There are two prominent setups in DA research: *supervised* and *unsupervised*. *Supervised* algorithms utilize scarce labeled examples from the target domain (Daumé III, 2007), whereas *unsupervised* methods assume only source labeled data and unlabeled source and target data in-hand (Blitzer et al., 2006). We first describe research in the setting of unsupervised DA with a focus on pivot-based methods. We then continue with the study of DA methods with multiple sources, focusing on mixture of experts models. Finally, we describe autoregressive language models and the unique manner in which we employ T5 for DA.

Unsupervised Domain Adaptation (UDA)

With the breakthrough of deep neural network (DNN) modeling, attention from the DA community has been directed to representation learning approaches. One line of work employs DNN-based autoencoders to learn latent representations. These models are trained on unlabeled source and target data with an input reconstruction loss (Glorot et al., 2011; Chen et al., 2012; Yang and Eisenstein, 2014; Ganin et al., 2016). Another branch employs pivot features to bridge the gap between a source domain and a target domain (Blitzer et al., 2006, 2007; Pan et al., 2010). Pivot features are prominent to the task of interest and are abundant in the source and target domains. Recently, Ziser and Reichart (2017, 2018b) married the two approaches. Later on, Han and Eisenstein (2019) presented a pre-training method, followed by Ben-David et al. (2020) who introduced a pivot-based variant for pre-training contextual word embeddings.

Crucially, UDA models assume access to unlabeled data from the target domain in-hand during training. We see this as a slight relaxation to the goal of generalization beyond the training distribution. Moreover, this definition has engineering disadvantages, as a new model is required for each target domain. To this end, we pursue the any-domain adaptation setting, where unlabeled target data is unavailable at training time.

We draw inspiration from pivot-based modeling. The pivot definition relies on labeled source domain data and unlabeled source and target do-

main data. Particularly, good pivots are ones that are correlated with the task label. Instead, we define task-invariant DRFs, features which are highly correlated with the identity of the domain. Since domains are highly correlated with words, our DRFs are lexical in nature.

While our method can operate with a single source domain, we utilize multiple source domains to facilitate generalization to unknown target domains. We next discuss multi-source DA.

Multi-Source Domain Adaptation Most existing *multi-source DA* methods follow the setup definitions of unsupervised DA, while considering more than one source domain. A prominent approach is to fuse models from several sources. Early work trained a classifier for each domain and assumed all source domains are equally important for a test example (Li and Zong, 2008; Luo et al., 2008). More recently, adversarial-based methods used unlabeled data to align the source domains to the target domains (Zhao et al., 2018; Chen and Cardie, 2018). Meanwhile, Kim et al. (2017) and Guo et al. (2018) explicitly weighted a Mixture of Experts (MoE) model based on the relationship between a target example and each source domain. However, Wright and Augenstein (2020) followed this work and tested a variety of weighting approaches on a Transformers-based MoE and found a naive weighting approach to be very effective.

We recognize two limitations in the proposed MoE solution. First, it is unscalable, as it requires an expert for each source domain, resulting in an (typically linear) increase in model parameters with the number of source domains. Second, domain experts are tuned for domain-specific knowledge. However, test examples may arrive from unknown domains, and may reflect a complicated combination of the sources. To cope with this, MoE ensembles the predictions of the experts using heuristic methods, such as a simple average or a weighted average based on the predictions of a domain-classifier. Our results indicate that this approach is sub-optimal.

In this work, we train one model which shares its parameters across all domains. Furthermore, we are interested in adapting to any target domain, such that no information about potential target domains is known at training time. Some of the above works (Wright and Augenstein, 2020) in fact avoid utilizing target data, thus they fit the any-domain setting and form two of our baselines.

Yet, in contrast to these works, we see this definition as a core part of this study.

Autoregressive Language Modeling Previous works consider two main approaches when training a Transformer-based (Vaswani et al., 2017) language model. The first implements the classic markovian language modeling approach, by training a Transformer decoder model to autoregressively generate the next word given its previous context (Radford and Narasimhan, 2018; Radford et al., 2019). The second casts the autoregressive language modeling approach as a masked tokens prediction task, by training a Transformer’s encoder to derive contextualized word embeddings (Devlin et al., 2019; Liu et al., 2019; Sanh et al., 2019). When fine-tuning the model for a new task, a custom decoder is implemented and jointly trained with the model’s pre-trained encoder.

A third approach has been recently proposed, in an attempt to combine the best of both worlds from previous approaches. It proposes training a full Transformer (encoder-decoder) language model, to autoregressively generate masked, missing or perturbed token spans from the input sequence, as a sequence-to-sequence task (Raffel et al., 2020; Lewis et al., 2020b). Recently, Raffel et al. (2020) presented T5, a transformer-based model which proposes a text-to-text transfer learning approach. T5 essentially treats all tasks as generative, while utilizing a prompt phrase which denotes the specific task being performed. Combined with its text-to-text approach, T5 demonstrated its superiority in many discriminative and generative tasks, while eliminating the need for task-specific network architecture.

A particularly interesting and useful characteristic of T5 is its prompting mechanism. The prompt phrase, which is typically used to instruct the model about the task to be performed, is prepended as a prefix to all task-related input examples. Recent works have also explored such prompting mechanisms in order to adapt a language model for different purposes (Brown et al., 2020), to elicit sentiment or topic-related information (Jiang et al., 2020; Sun and Lai, 2020; Shin et al., 2020), or as an efficient fine-tuning method (Li and Liang, 2021). In this work, we make use of T5’s prompting mechanism as a way of priming the model to encode domain-specific characteristics relating to each test example.

3 Any-Domain Adaptation

DA and Transfer Learning A prediction task (e.g., Rumour Detection) is defined as $\mathcal{T} = \{\mathcal{Y}\}$, where \mathcal{Y} is the task’s label space. We denote \mathcal{X} to be a feature space, $P(X)$ to be the marginal distribution over \mathcal{X} , and $P(Y)$ the prior distribution over \mathcal{Y} . The domain is then defined by $\mathcal{D}^{\mathcal{T}} = \{\mathcal{X}, P(X), P(Y), P(Y|X)\}$. DA is a particular case of transfer learning, namely *transductive transfer learning*, in which \mathcal{T}_S and \mathcal{T}_T , the source and target tasks, are the same. However, $\mathcal{D}_S^{\mathcal{T}}$ and $\mathcal{D}_T^{\mathcal{T}}$, the source and target domains, differ in at least one of their underlying probability distributions, $P(X)$, $P(Y)$, or $P(Y|X)$.³ The goal in DA is to learn a function f from a set of source domains $\{\mathcal{D}_{S_i}\}_{i=1}^K$ that generalizes well to a set of target domains $\{\mathcal{D}_{T_i}\}_{i=1}^M$.

The Any-Domain Setting We focus on building an algorithm for a given task, that is able to adapt to *any domain*. To this end, we assume zero knowledge about the target domain, \mathcal{D}_T , at training time. Hence, we slightly modify the classic setting of unsupervised multi-source domain adaptation, by assuming we have no knowledge or access to labeled or unlabeled data from the target domains. We only assume access to labeled training data from K source domains $\{\mathcal{D}_{S_i}\}_{i=1}^K$, where $\mathcal{D}_{S_i} \triangleq \{(x_t^{S_i}, y_t^{S_i})\}_{t=1}^{n_i}$. The goal is to learn a model using only the source domains data, that generalizes well to an unknown target domain.

Any-Domain Adaptation and Zero-Shot Learning We avoid naming our setting “zero-shot DA” since we consider zero-shot learning to be an overloaded term, and its usage differs across different works. On the one hand, the authors of GPT-3 (Brown et al., 2020) used this term to indicate a shift to an unknown target task \mathcal{T}_T and unknown domain \mathcal{D}_T . On the other hand, Kodirov et al. (2015) assume a task/label-space shift while the target domain is known during training, Blitzer et al. (2009) assume access to unlabeled data from various domains including the target domain, and Peng et al. (2018) use data of a different task from the target domain. Our problem setting differs from those presented in the works above, yet all could be described as “zero-shot” to a certain extent. In our view these differences should be clarified, and thus we present a designated terminology for our setup.

³In *inductive transfer learning* \mathcal{T}_S differs from \mathcal{T}_T .

4 Prompt-based Autoregressive DA

As previously discussed in §2, we recognize two major limitations with the solution proposed by the MoE-based approach: (1) It is not scalable. The total number of trained parameters, which is already large for a single model, linearly grows as the number of source domains increases, since an expert model needs to be trained separately for each domain. Naturally, this also increases the overall training time; and (2) In this approach a separate model is trained for each domain. Intuitively, domain-specific experts are tuned towards domain-specific knowledge, at times at the expense of cross-domain knowledge which highlights the relationship between different domains. Moreover, since domain partitioning is often somewhat arbitrary (consider for example the differences between the *dvd* and *movie* domains), we do not want to strictly confine our model to a specific partitioning and rather encourage a more lenient approach towards domain boundaries.

We hence propose a single model that encodes information from multiple domains. Our model is designed such that test examples from new unknown domains can trigger the most relevant parameters in the model. This way we allow our model to share information between domains and use the most relevant information at test time. Our model is inspired by recent research on prompting mechanisms for autoregressive language models. Recent work has shown the effectiveness of the prompting mechanism in priming these models (§2), albeit not in the context of DA.

We start (§4.1) by describing the general architecture of our model, and continue (§4.2) with the domain related features that form our prompts.

4.1 The Model

We present our *Prompt-based Autoregressive algorithm for Domain Adaptation (PADA)*, Figure 2a). *PADA* employs a pre-trained T5 language model and learns to generate example-specific Domain Related Features (DRFs) in order to facilitate accurate task predictions. This is implemented through a two-step multi-task mechanism, where first a DRF set is generated to form a prompt, and then the task label is predicted.

Formally, assume an input example $(x_i, y_i) \sim S_i$, such that x_i is the input text, y_i is the task label and S_i is the domain of this example. For the input x_i , *PADA* is trained to first generate N_i , the

domain name, followed by R_i , the DRF signature of x_i , and given this prompt to predict the label y_i . At test time, when the model encounters an example from an unknown domain, it generates a prompt that may consist of one or more domain names as well as features from the DRF sets of one or more source domains, and based on this prompt it predicts the task label.

Consider the example in Figure 1, which describes a sentiment classification model, trained on the *restaurants*, *home-furniture*, *electronic-devices*, and *movies* source domains. The model observes a test example from the *airlines* domain, a previously unseen domain whose name is not known to the model. The model first generates the name of the training domain which is most appropriate for this example, *restaurants* in this case. Then, it continues to generate the words “food” and “chair”, features related to the *restaurants* and *home-furniture* domains, respectively. Finally, given this prompt, the model predicts the example’s (negative) sentiment.

In order to separate the prompt generation task from the discriminative classification task, we train our model within a multi-task framework. *PADA* is trained to perform two tasks, depending on the example’s prompt: One for generating a prompt, consisting of features from the DRF set of the example’s domain, and another for predicting the example’s label. For the first, generative task, the model receives examples with the special prompt ‘Domain:’, which primes the model to generate N_i and R_i . Note, that R_i is a set of features derived from the DRF set of S_i , and training examples are automatically annotated with their R_i , as described in §4.2. For the second, discriminative task, the model receives a prompt, consisting of N_i and R_i , and its task is to predict y_i .

Following the multi-task training protocol of T5, we mix examples from each task. To this end, we define a task proportion mixture parameter α . Each example from the training set forms an example for the generative task with probability α , and an example for the discriminative task with probability $1 - \alpha$. The greater the value of α , the more the model will train for the generative task.

PADA conditions the classification on the generated prompt. To evaluate the effect of this conditioning, we also consider a simpler *PADA* variant, which jointly performs the classification and generation tasks, but does not use the output of

the generation task as a prompt for the classification task. We name this variant *PADA-NC* to stress that the discriminative task does not condition on the output of the generative component. The differences between *PADA* and *PADA-NC* are highlighted in Figure 2.

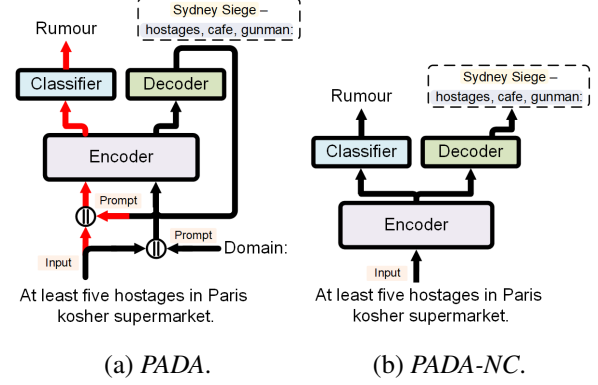


Figure 2: A multi-task model with a generative head trained for DRF generation and a discriminative head for Rumour Detection. Text marked with blue signifies the DRFs and text marked with yellow signifies the domain name. Black arrows (\rightarrow) mark the first inference step and red arrows (\rightarrow) mark the second inference step. The presented models are trained on 4 different source domains: *Ferguson*, *Germanwings-crash*, *Ottawa-shooting*, and *Sidney-siege*, while the test example arrives from the *Charlie-Hebdo* domain. The models identify the origin of the example to be *Sidney-siege*, and associates it with 3 DRFs: *hostages*, *cafe*, and *gunman*.

At the heart of our method is the clever selection of the DRF set of each domain. We next discuss these features and their selection process.

4.2 Domain Related Features

For each domain we define the DRF set such that these features provide a semantic signature for the domain. Importantly, if two domains have shared semantics, for example the *restaurants* and the *cooking* domains, we expect their DRFs to semantically overlap. Since the prompt of each training example consists of a subset of features from the DRF set of its domain, we should also decide on a prompt generation rule that can annotate these training examples with their relevant features.

In order to reflect the semantics of the domain, DRFs should occur frequently in this domain. Moreover, they should be substantially more common in that specific domain relative to all other do-

mains. Despite their prominence in a specific domain, DRFs can also relate to other domains. For instance, consider the example presented in Figure 2. The word “hostages” is highly associated with the “Charlie-Hebdo” domain and is indeed one of its DRFs. However, this word is also associated with the “Sydney-Siege” domain, which is another domain in the Rumour Detection dataset (Zubiaga et al., 2016). Moreover, since both domains are related to similar events, it is not surprising that the DRF set of the former contains the feature *terrorists* and the DRF set of the latter contains the feature *gunman*. The similarity of these features facilitates parameter sharing in our model.

We define the DRF set for each source domain as follows. Let examples (texts) from the j th source domain (\mathcal{S}_j) be labeled with 1, and examples from all other domains ($\mathcal{S} \setminus \mathcal{S}_j$) be labeled with 0. We first calculate the *mutual-information* (MI) between all tokens and this binary variable, and choose the l tokens with the highest MI score. Note, that the MI criterion might promote tokens which are highly associated with ($\mathcal{S} \setminus \mathcal{S}_j$) rather than with \mathcal{S}_j . Thus, we filter the l tokens according to the following condition:

$$\frac{C_{\mathcal{S} \setminus \mathcal{S}_j}(n)}{C_{\mathcal{S}_j}(n)} \leq \rho, \quad C_{\mathcal{S}_j}(n) > 0$$

where $C_{\mathcal{S}_j}(n)$ is the count of the n-gram n in \mathcal{S}_j , $C_{\mathcal{S} \setminus \mathcal{S}_j}(n)$ is the count of this n-gram in all source domains except for \mathcal{S}_j , and ρ is an n-gram frequency ratio parameter. Intuitively, the smaller ρ is, the more certain we are that the n-gram is especially associated with \mathcal{S}_j , compared to other domains. Since the number of examples in \mathcal{S}_j is much smaller than the number of examples in $\mathcal{S} \setminus \mathcal{S}_j$, we choose $\rho \geq 1$ but do not allow it to be too large. As a result, this criterion allows for features which are associated with \mathcal{S}_j but also related to other source domains, to be part of the DRF set of \mathcal{S}_j . We denote the DRF set of the j th domain with R_j .

Given a training example i from domain j , we select the m features from R_j which are most associated with this example to form its prompt. To do that, we compute the Euclidean distance between the T5 embeddings of the DRF features and the T5 embeddings of each of the example’s tokens. We then rank this list of pairs by their scores and select the top m features.⁴

⁴In this computation we consider the non-contextual em-

Rumour Detection			
Domain	Training (src)	Dev (src)	Test (trg)
Charlie-Hebdo (C)	1,663	416	2,079
Ferguson (FR)	914	229	1,143
Germanwings-crash (GW)	375	94	469
Ottawa-shooting (OS)	712	178	890
Sydney-siege (S)	976	245	1,221

MNLI			
Domain	Training (src)	Dev (src)	Test (trg)
Fiction (F)	2,547	1,972	1,972
Government (G)	2,541	1,944	1,944
Slate (SL)	2,605	1,954	1,954
Telephone(TL)	2,754	1,965	1,965
Travel (TR)	2,541	1,975	1,975

Aspect			
Domain	Training (src)	Dev (src)	Test (trg)
Device (D)	2,302	255	1,279
Laptops (L)	2,726	303	800
Restaurants (R)	3,487	388	800
Service(SE)	1,343	149	747

Table 1: The number of examples in each domain of our three tasks. We denote the examples used when a domain is included as a source domain (src), and when it is the target domain (trg).

To conclude, our methods for domain-specific DRF set extraction and for prompt annotation of training examples, demonstrate three attractive properties. First, every example has its own unique prompt. Second, our prompts map each training example to the semantic space of its domain. Lastly, the domain-specific DRF sets may overlap in their semantics, either by including the same tokens or by including tokens with similar meanings. This way they provide a more nuanced domain signature compared to the domain name alone. This is later used during the inference phase when the model can generate an example-specific prompt that consists of features from the DRF sets of the various source domains.

5 Experimental Setup

5.1 Task and Datasets

We experiment with three multi-source DA tasks, where a model is trained on several domains and applied to a new one. We consider two text classification tasks, Rumour Detection and Multi-Genre Natural Language Inference (MNLI), and one sequence tagging task – Aspect Prediction. The details of the training, development and test sets of each domain are provided in Table 1. Our experiments are performed in a leave-one-out fashion: We train the model on all domains but one, and keep the held-out domain for testing. Particularly,

beddings learned by T5 during its pre-training. In our experiments we consider only unigrams (words) as DRFs.

training is done on the training data of the source domains and development on their development data, while the test data is taken from the target domain, which is unknown at training time. We repeat the experiments in each task such that each domain is used as a target domain.

Rumour Detection⁵ We experiment with the PHEME dataset of rumourous tweets (Zubiaga et al., 2016). It contains 5,802 tweets, which followed 5 different real-world events, and are labelled as rumourous or non-rumourous. We treat each event as a separate domain: Charlie-Hebdo (C), Ferguson (FR), Germanwings-crash (GW), Ottawa-shooting (OS), and Sidney-siege (S).

We follow the data processing procedure of Wright and Augenstein (2020) and split each domain (event) corpus by a 4:1 ratio, establishing training and development sets. Since the corpora are relatively small, we want to avoid further shrinking the size of the test set. Hence, we include all examples available from the target domain to form the test set.⁶

MNLI (Williams et al., 2018)⁷ This corpus is an extension of the SNLI dataset (Bowman et al., 2015). Each example consists of a pair of sentences, a premise and a hypothesis. The relationship between the two may be entailment, contradiction, or neutral. The corpus includes data from 10 domains: 5 are matched, with training, development and test sets, and 5 are mismatched, without a training set. We experiment only with the five matched domains: Fiction (F), Government (G), Slate (SL), Telephone (TL) and Travel (TR).

Since the test sets of the MNLI dataset are not publicly available, we use the original development sets as our test sets for each target domain, while source domains use these sets for development. We explore a lightly supervised scenario, which emphasizes the need for a DA algorithm. Thus, we randomly downsample each of the training sets by a factor of 30, resulting in 2,000 – 3,000 examples per set.

⁵https://figshare.com/articles/dataset/PHEME_dataset_of_rumours_and_non-rumours/4010619

⁶This does not harm the integrity of our experiments, since the training and development sets are sampled from the source domains and the test set is sampled only from the target domain.

⁷<https://cims.nyu.edu/~sbowman/multinli/>

Aspect Prediction The Aspect Prediction dataset is based on aspect-based sentiment analysis (ABSA) corpora from four domains: Device (D), Laptops (L), Restaurant (R), and Service (SE). The D data consists of reviews from Toprak et al. (2010), the SE data includes web service reviews (Hu and Liu, 2004), and the L and R domains consist of reviews from the SemEval-2014 ABSA challenge (Pontiki et al., 2014).

We follow the training and test splits defined by Gong et al. (2020) for the D and SE domains, while the splits for the L and R domains are taken from the SemEval-2014 ABSA challenge. To establish our development set, we randomly sample 10% out of the training data.

5.2 Evaluated Models

Our main model is **PADA**: The multi-task model that first generates the domain name and domain related features to form a prompt, and then uses this prompt to predict the task label (§4.1, Figure 2a). We compare it to two types of models: (a) T5-based baselines corresponding to ideas presented in multi-source DA work, as well as other recent state-of-the-art models (§2); and (b) Ablation models that use specific parts of **PADA**, to highlight the importance of its components.

5.2.1 Baseline Models

Content-CRF (CCRF, Zubiaga et al. (2017)) A CRF model trained for Rumour Detection. Each prediction is conditioned on the tweet in question as well as tweets which preceded it, alongside a combination of content-based and social features.

Transformer-based Mixture of Experts (Tr-MoE, Wright and Augenstein (2020)) For each source domain, a separate transformer-based DistilBERT expert model (Sanh et al., 2019) is trained on the domain’s training set, and an additional model is trained on data from all source domains. At test time, the average of the class probabilities of these models is calculated and the highest probability class is selected. This model is named *MoE-avg* by Wright and Augenstein (2020) and has demonstrated to achieve state-of-the-art performance for Rumour Detection (the paper reports the CCRF results as its previous state-of-the-art).

T5-MoE A T5-based MoE ensemble model. For each source domain, a separate pre-trained T5 model is fine-tuned on the domain’s training set (i.e. a domain expert model). During inference,

the final predictions of the model are decided using the same averaging procedure as in *Tr-MoE*.

T5-No-Domain-Adaptation (T5-NoDA) A pre-trained T5 model, which feeds the same task classifier used in *PADA* (see below) to predict the task label. In each DA setting, the model is trained on the training data from all source domains.

T5-Domain-Adversarial-Network (T5-DAN) A model that integrates T5-NoDA with an adversarial domain classifier to learn domain invariant representations (Ganin and Lempitsky, 2015).

T5-Invariant-Risk-Minimization (T5-IRM) A T5-based model which penalizes feature distributions that have different optimal linear classifiers for each domain. IRM (Arjovsky et al., 2019) is an established machine learning baseline for generalization beyond multiple-source distributions (Koh et al., 2020). The model is trained on the training data from all source domains.

T5-UpperBound (T5-UB) An in-domain model with identical architecture as *T5-NoDA*. It is trained on the training data from all domains and tested on the development data of each domain. We treat its performance as an upper bound for the average target performance across all DA settings, for any T5-based model in our setup.

5.2.2 Ablation Models

PADA-DN A simplified variant of our *PADA* model, which assigns only a domain name as a prompt to the input text. Since the domain name is unknown at test time, we create multiple variants of each test example, each with one of the training domain names as a prompt. For the final predictions of the model we follow the same averaging procedure as in *Tr-MoE* and *T5-MoE*.

PADA-NC A multi-task model similar to *PADA*, except that it simultaneously generates the example-specific domain name and DRF prompt, and predicts the task label. This model, which does not condition the task prediction on the prompt, is discussed in §4.1 (Figure 2b).

5.3 Implementation Details

For all implemented models we use the *Hugging-Face* Transformers library (Wolf et al., 2020).⁸

⁸<https://github.com/huggingface/transformers>

The T5-based text classification models do not follow the same procedure originally described in Raffel et al. (2020). Instead, we add a simple *1D-CNN* classifier on top of the T5 encoder to predict the task label (Figure 2). The number of filters in this classifier is 32 with a filter size of 9. The generative component of the T5-based models is identical to that of the original T5.⁹ Our T5-based models for Aspect Prediction cast sequence tagging as a sequence-to-sequence task, employing the text-to-text approach of Raffel et al. (2020) to generate a 'B' (begin), 'I' (in) or 'O' (out) token for each input token. Other than this change, these models are identical to the T5-based models for text classification.

We train all text classification models for 5 epochs and all sequence tagging models for 60 epochs, with an early stopping criterion according to performance on the development data. We use the cross-entropy loss function for all models, optimizing their parameters with the *ADAM* optimizer (Kingma and Ba, 2015). We employ a batch size of 32 for text classification and 24 for sequence tagging, warmup ratio of 0.1, and a learning rate of $5 \cdot 10^{-5}$. The maximum input and output lengths of all T5-based models is set to 128 tokens. We pad shorter sequences and truncate longer ones to the maximum input length.

For *PADA*, we tune the α (example proportion-mixture, see §4.1) parameter considering the value range of $\{0.1, 0.25, 0.5, 0.75, 0.9\}$. The chosen values are: $\alpha_{rumour} = 0.75$, $\alpha_{mnli} = 0.1$ and $\alpha_{absa} = 0.1$. For each example, we select the top $m = 5$ DRFs most associated with it, for its prompt. For the generative component of the T5-based models, we perform inference with the Diverse Beam Search algorithm (Vijayakumar et al., 2016), considering the following hyperparameters: We generate 5 candidates, using a beam size of 10, with 5 beam groups, and a diversity penalty value of 1.5. The l and ρ parameters of the DRF extraction procedure (§4.2) were tuned to 1000 and 1.5, respectively, for all domains.

6 Results

Text Classification Table 2 presents our results. We report the binary-F1 score for Rumour Detection, and the macro-F1 score for MNLI.¹⁰ *PADA*

⁹We experimented with the original T5 classification method as well, but *PADA* consistently outperformed it.

¹⁰Binary-F1 measures the F1 score on the positive class. It is useful in cases of unbalanced datasets where the positive

	Rumour Detection						MNLI					
	All → C	All → FR	All → GW	All → OS	All → S	AVG	All → F	All → G	All → SL	All → TE	All → TR	AVG
<i>CCRF</i>	63.6	46.5	70.4	69.0	61.2	62.1	-	-	-	-	-	-
<i>Tr-MoE</i>	68.0	46.1	74.8	58.2	64.9	62.4	64.3	73.9	65.3	62.4	69.8	67.1
<i>T5-NoDA</i>	64.1	46.9	75.1	72.0	71.0	65.8	76.4	83.5	75.5	74.9	81.3	78.3
<i>T5-DAN</i>	64.9	52.4	69.1	72.7	64.4	64.7	74.4	76.3	61.0	72.4	77.7	72.4
<i>T5-IRM</i>	63.5	39.4	70.1	44.2	65.7	56.6	72.0	81.5	73.2	69.3	78.9	75.0
<i>T5-MoE</i>	68.1	46.0	73.6	65.3	66.3	63.9	74.0	82.0	73.4	74.6	78.3	76.5
<i>PADA-DN</i>	66.4	53.7	72.4	71.4	70.1	66.8	77.0	84.4	75.6	76.3	80.5	78.8
<i>PADA-NC</i>	65.8	54.8	71.6	72.2	74.0	67.7	76.2	83.6	75.4	77.2	81.4	78.8
<i>PADA</i>	68.6	54.4	73.0	75.2	75.1	69.3	76.4	83.4	76.9	78.9	82.5	79.6

Table 2: Binary-F1 scores for the Rumour Detection task and macro-F1 scores for the MNLI task.

	Aspect Prediction				
	All → D	All → L	All → R	All → SE	AVG
<i>T5-NoDA</i>	31.1	45.6	40.2	37.9	38.7
<i>T5-DAN</i>	28.4	38.0	49.1	33.4	33.2
<i>T5-IRM</i>	37.1	44.6	47.4	41.5	42.7
<i>T5-MoE</i>	39.5	31.4	31.4	30.9	33.3
<i>PADA-DN</i>	41.1	42.6	29.0	30.8	35.9
<i>PADA-NC</i>	41.7	48.2	50.1	40.1	45.0
<i>PADA</i>	41.9	50.9	50.8	41.3	46.2

Table 3: Binary-F1 scores for Aspect Prediction.

outperforms all models in 6 of 10 settings, exhibiting average performance gains of 3.5% and 1.3% in Rumour Detection and MNLI, respectively, over *T5-NoDA*, the best model that does not belong to our *PADA* framework. Moreover, in 9 of 10 settings it is one of the *PADA* models that is the best performing model. Interestingly, it is *T5-NoDA*, which does not perform any DA, that outperforms (on average and in most model-to-model comparisons) all models that do not belong to the *PADA* family, including the MoE models.

While the performance gains differ between the tasks, they partly stem from the different performance gaps between source and target domains in each of these tasks. Recall, that we consider the *T5-UB* performance on its development sets for Rumour Detection (82.8%) and MNLI (80.8%), to be the upper bound for the average target performance across all DA settings, for any T5-based model. When considering the gaps between this upper bound and *T5-NoDA* (65.8% for Rumour Detection and 78.3% for MNLI), *PADA* reduces the error rate by 21% for Rumour Detection and 52% for MNLI. The improvements gained by *PADA* are in fact substantial in both tasks.

The advantage of *PADA* over MoE goes beyond improved predictions. Particularly, for *PADA* we train a single model while for MoE we train a unique model for each source domain, hence

class is of interest. In the Rumour Detection dataset, 34% of the examples belong to the positive class.

the number of parameters in the MoE framework linearly increases with the number of source domains. For example, in our setups, *Tr-MoE* trains five DistilBERT models (one for each source domain and one for all source domains together), resulting in $5 \cdot 66M = 330M$ parameters. In contrast, the *PADA* models keep the 220M parameters of T5, regardless of the number of source domains.

Our results exhibit the superiority of *PADA* and its variants, *PADA-DN* and *PADA-NC*, over all other models in 9 of 10 settings. Particularly, *PADA* outperforms the non-*PADA* models in 7 of 10 settings, *PADA-NC* outperforms these models in 6 settings, and *PADA-DN* in 5 settings. Moreover, *PADA* outperforms the *PADA-DN* variant in all Rumour Detection settings and in 3 out of 5 MNLI settings, and its *PADA-NC* variant in 8 of 10 settings overall. These results highlight the importance of our design choices: (a) including DRFs in the example-specific prompts, tailoring them to express the relation between the source domains and the test example (*PADA* vs *PADA-DN*); and (b) utilizing an autoregressive component, where the generated DRF prompts are used by the task classification component (*PADA* vs *PADA-NC*).

Sequence Tagging In order to demonstrate the wide applicability of our approach, we go beyond text classification (with 2 (Rumour Detection) or 3 (MNLI) classes) and also consider Aspect Prediction: A sequence tagging task. We are particularly curious to see if the aforementioned patterns replicate in this qualitatively different task. Our results are presented in Table 3, where we report the binary-F1 score (the F1 score of the aspect class). Crucially, the patterns we observe for text classification can also be detected for sequence tagging. Particularly, *PADA* is the best performing model in 3 of 4 settings and on average, outperforming the second-best model, *T5-IRM*, by 3.5% on average. Given the average results of *T5-UB* (69.4%) and

T5-NoDA (38.7%), the error reduction is 24%.

Moreover, *PADA* outperforms its simpler variants, *PADA-DN* and *PADA-NC*, in all 4 setups. *PADA-NC* outperforms all non-*PADA* models in 3 of 4 setups, and in the other setup it is outperformed only by *T5-IRM* and *PADA*. A noticeable difference from text classification is the weakness of *PADA-DN*, which is outperformed by all non-*PADA* models in 2 setups, and by 2 of these models in a third setup, as well as on average across all setups. This is an indication of the importance of the DRFs in the prompt generated by *PADA*.

Performance Shifts between Source and Target

When a DA method improves model performance on the target domain, this can result in either increasing or decreasing the performance gap between the source and target domains. If a model performs similarly on its source training domains and on unseen target domains, its source domain performance can also provide an important indication for its future performance in such unseen domains. We hence consider such stability in performance as a desired property in our setup where future target domains are unknown.

Figure 3 presents three heatmaps, depicting the performance shifts of each model between the source domains and the target domains. We measure each model’s in-domain performance by calculating an F1 score across all development examples from its source domains, and its performance on the target domain test set, as described in §6. We then calculate the difference between source and target performance measures, and report results for the best performing models in our experiments. The general trend is clear: *PADA* not only performs better on the target domain, but it also substantially reduces the source-target performance gap. While *T5-NoDA*, which is not a DA model, triggers the largest average absolute performance shifts – 17% for Rumour Detection, 4.3% for MNLI, and 34% of Aspect Prediction – the averages of *PADA*’s absolute performance shift are 8.7%, 3.5%, and 26%, respectively.

7 Discussion

We addressed the problem of multi-source domain adaptation when the target domain is not known at training time. Effective models for this setup can be applied to any target domain with no data requirements about the target domains and without an increase in the number of model parameters as

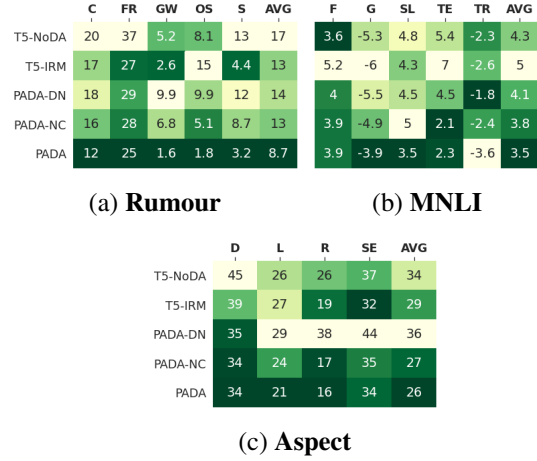


Figure 3: Heatmaps presenting performance shifts between source domains and target domains (columns), for our three tasks. Negative values correspond to improved target compared to source results. We normalize the colors per column, such that darker colors represent smaller absolute value of performance shifts.

a function of the number of source or target domains. *PADA*, our algorithm, takes advantage of the prompting mechanism of the T5 autoregressive language model and maps test examples into the semantic space spanned by the source domains.

Our experimental results with three tasks and fourteen multi-source adaptation settings demonstrate the effectiveness of our approach compared to strong alternatives, as well as the importance of the model components and of our design choices. Moreover, as opposed to the MoE paradigm, where a model is trained for each source domain, *PADA* provides a single unified model. Intuitively, this approach also seems more cognitively plausible – a single model attempts to adapt itself to new incoming domains, rather than employing an independent model per domain.

The prompt generation mechanism of *PADA* is naturally limited by the set of source domains it is trained on. This might yield sub-optimal DRFs for test examples stemming from domains which are semantically extremely unrelated to any of the source domains. Furthermore, we do not directly optimize the prompt generation process with the main prediction task, which might also contribute to sub-optimal DRF generation. In future work, we hope to improve these aspects of our approach and to explore natural extensions that accommodate multiple tasks and domains in a single model.

References

- Martín Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2019. [Invariant risk minimization](#). *CoRR*, abs/1907.02893.
- Eyal Ben-David, Carmel Rabinovitz, and Roi Reichart. 2020. [Perl: Pivot-based domain adaptation for pre-trained deep contextualized embedding models](#). *Transactions of the Association for Computational Linguistics*, 8:504–521.
- John Blitzer, Mark Dredze, and Fernando Pereira. 2007. [Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification](#). In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*. The Association for Computational Linguistics.
- John Blitzer, Dean P Foster, and Sham M Kakade. 2009. [Zero-shot domain adaptation: A multi-view approach](#).
- John Blitzer, Ryan T. McDonald, and Fernando Pereira. 2006. [Domain adaptation with structural correspondence learning](#). In *EMNLP 2006, Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, 22-23 July 2006, Sydney, Australia*, pages 120–128. ACL.
- Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. [A large annotated corpus for learning natural language inference](#). In *EMNLP*.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Minmin Chen, Zhixiang Eddie Xu, Kilian Q. Weinberger, and Fei Sha. 2012. [Marginalized denoising autoencoders for domain adaptation](#). In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. icml.cc / Omnipress.
- Xilun Chen and Claire Cardie. 2018. [Multinomial adversarial networks for multi-domain text classification](#). 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 1226–1240. Association for Computational Linguistics.
- Hal Daumé III. 2007. [Frustratingly easy domain adaptation](#). In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*. The Association for Computational Linguistics.
- Hal Daumé III and Daniel Marcu. 2006. [Domain adaptation for statistical classifiers](#). *Journal of artificial Intelligence research*, 26:101–126.
- 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.
- Yaroslav Ganin and Victor S. Lempitsky. 2015. [Unsupervised domain adaptation by backpropagation](#). In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1180–1189. JMLR.org.
- Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. 2016. [Domain-adversarial training](#)

- of neural networks. *The journal of machine learning research*, 17:59:1–59:35.
- Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. [Domain adaptation for large-scale sentiment classification: A deep learning approach](#). In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 513–520. Omnipress.
- Chenggong Gong, Jianfei Yu, and Rui Xia. 2020. [Unified feature and instance based domain adaptation for aspect-based sentiment analysis](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 7035–7045. Association for Computational Linguistics.
- Jiang Guo, Darsh J. Shah, and Regina Barzilay. 2018. [Multi-source domain adaptation with mixture of experts](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4694–4703. Association for Computational Linguistics.
- Xiaochuang Han and Jacob Eisenstein. 2019. [Unsupervised domain adaptation of contextualized embeddings for sequence labeling](#). 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 4237–4247. Association for Computational Linguistics.
- Minqing Hu and Bing Liu. 2004. [Mining and summarizing customer reviews](#). In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22-25, 2004*, pages 168–177. ACM.
- Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. [How can we know what language models know](#). *Transactions of the Association for Computational Linguistics*, 8:423–438.
- Young-Bum Kim, Karl Stratos, and Dongchan Kim. 2017. [Domain attention with an ensemble of experts](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 643–653. Association for Computational Linguistics.
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization](#). In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Elyor Kodirov, Tao Xiang, Zhen-Yong Fu, and Shaogang Gong. 2015. [Unsupervised domain adaptation for zero-shot learning](#). In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 2452–2460. IEEE Computer Society.
- Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanus Phillips, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. 2020. [WILDS: A benchmark of in-the-wild distribution shifts](#). *CoRR*, abs/2012.07421.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. [BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020b. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Shoushan Li and Chengqing Zong. 2008. [Multi-domain sentiment classification](#). In *ACL 2008, Proceedings of the 46th Annual Meeting of*

- the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA, Short Papers*, pages 257–260. The Association for Computer Linguistics.
- Xiang Lisa Li and Percy Liang. 2021. [Prefix-tuning: Optimizing continuous prompts for generation](#). *CoRR*, abs/2101.00190.
- 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](#). *CoRR*, abs/1907.11692.
- Ping Luo, Fuzhen Zhuang, Hui Xiong, Yuhong Xiong, and Qing He. 2008. [Transfer learning from multiple source domains via consensus regularization](#). In *Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, Napa Valley, California, USA, October 26-30, 2008*, pages 103–112. ACM.
- David McClosky, Eugene Charniak, and Mark Johnson. 2010. [Automatic domain adaptation for parsing](#). In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 2-4, 2010, Los Angeles, California, USA*, pages 28–36. The Association for Computational Linguistics.
- Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. 2010. [Cross-domain sentiment classification via spectral feature alignment](#). In *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, pages 751–760. ACM.
- Kuan-Chuan Peng, Ziyang Wu, and Jan Ernst. 2018. [Zero-shot deep domain adaptation](#). In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XI*, volume 11215 of *Lecture Notes in Computer Science*, pages 793–810. Springer.
- Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. [Semeval-2014 task 4: Aspect based sentiment analysis](#). In *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014*, pages 27–35. The Association for Computer Linguistics.
- A. Radford and Karthik Narasimhan. 2018. [Improving language understanding by generative pre-training](#).
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. [Language models are unsupervised multitask learners](#). *OpenAI Blog*, 1(8).
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21:1–67.
- Roi Reichart and Ari Rappoport. 2007. Self-training for enhancement and domain adaptation of statistical parsers trained on small datasets. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 616–623.
- Brian Roark and Michiel Bacchiani. 2003. [Supervised and unsupervised PCFG adaptation to novel domains](#). In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2003, Edmonton, Canada, May 27 - June 1, 2003*. The Association for Computational Linguistics.
- Alexander M. Rush, Roi Reichart, Michael Collins, and Amir Globerson. 2012. [Improved parsing and POS tagging using inter-sentence consistency constraints](#). In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea*, pages 1434–1444. ACL.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. [Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter](#). *CoRR*, abs/1910.01108.
- Tobias Schnabel and Hinrich Schütze. 2014. [FLORS: fast and simple domain adaptation for](#)

- part-of-speech tagging. *Transactions of the Association for Computational Linguistics*, 2:15–26.
- Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. [Autoprompt: Eliciting knowledge from language models with automatically generated prompts](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4222–4235. Association for Computational Linguistics.
- Fan-Keng Sun and Cheng-I Lai. 2020. [Conditioned natural language generation using only unconditioned language model: An exploration](#). *ArXiv*, abs/2011.07347.
- Cigdem Toprak, Niklas Jakob, and Iryna Gurevych. 2010. [Sentence and expression level annotation of opinions in user-generated discourse](#). In *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*, pages 575–584. The Association for Computer Linguistics.
- 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.
- Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David J. Crandall, and Dhruv Batra. 2016. [Diverse beam search: Decoding diverse solutions from neural sequence models](#). *CoRR*, abs/1610.02424.
- Adina Williams, Nikita Nangia, and Samuel R Bowman. 2018. [A broad-coverage challenge corpus for sentence understanding through inference](#). In *2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2018*, pages 1112–1122.
- 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*, pages 38–45, Online. Association for Computational Linguistics.
- Dustin Wright and Isabelle Augenstein. 2020. [Transformer based multi-source domain adaptation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7963–7974.
- Yi Yang and Jacob Eisenstein. 2014. [Fast easy unsupervised domain adaptation with marginalized structured dropout](#). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers*, pages 538–544. The Association for Computer Linguistics.
- Han Zhao, Shanghang Zhang, Guanhua Wu, João Paulo Costeira, José M. F. Moura, and Geoffrey J. Gordon. 2018. [Multiple source domain adaptation with adversarial learning](#). In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*. OpenReview.net.
- Yftah Ziser and Roi Reichart. 2017. [Neural structural correspondence learning for domain adaptation](#). In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, Canada, August 3-4, 2017*, pages 400–410. Association for Computational Linguistics.
- Yftah Ziser and Roi Reichart. 2018a. [Pivot based language modeling for improved neural domain adaptation](#). 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 1241–1251.
- Yftah Ziser and Roi Reichart. 2018b. [Pivot based language modeling for improved neural do-](#)

[main adaptation](#). 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 1241–1251. Association for Computational Linguistics.

Arkaitz Zubiaga, Maria Liakata, and Rob Procter. 2017. [Exploiting context for rumour detection in social media](#). In *International Conference on Social Informatics*, pages 109–123. Springer.

Arkaitz Zubiaga, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Peter Tolmie. 2016. [Analysing how people orient to and spread rumours in social media by looking at conversational threads](#). *PloS one*, 11(3):e0150989.