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 recently, but they still struggle when applied to out-of-distribution examples. In this paper, we address a very challenging and previously underexplored version of this domain adaptation problem. In our setup 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 two tasks: Rumour Detection and Multi-Genre Natural Language Inference (MNLI), for a total of 10 multi-source adaptation scenarios, *PADA* strongly outperforms state-of-the-art approaches and additional strong baselines.[1]

1 Introduction

Natural Language Processing (NLP) algorithms are gradually achieving milestones that were beyond imagination only a few years ago (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 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.

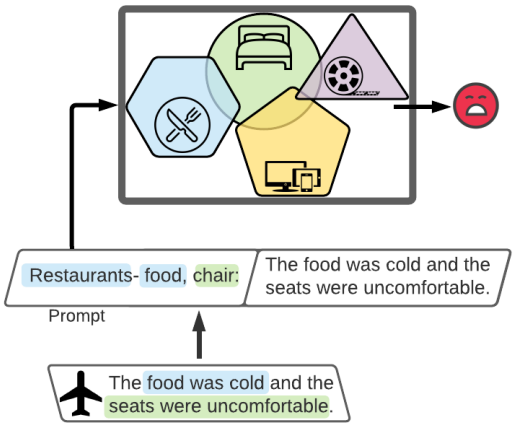


Figure 1: Text classification with PADA. Colored texts signify relation to a specific source domain. *PADA* first generates the domain name, followed by a set of DRFs related to the input example. Then it uses the prompt to predict the task label.

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; Jiang and Zhai, 2007; McClosky et al., 2010; Rush et al., 2012; Schn-

* Both authors equally contributed to this work.

[1] Our code and data are available at https://github.com/eyalbd2/PADA.

abel and Schütze, 2014). Still, the challenge of adaptation to *any possible target domain*, that is unknown at training time, is under-explored.

In this work, we focus on adaptation to any target domain (§3). We consider this to be the "Holy Grail" for DA systems. 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 utilize 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 domain, the model first generates properties that belong to familiar 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 highly correlated with one of the source domains, encoding domain-specific semantics. We leverage the DRFs of the various source domains in order to generate a joint feature space which spans 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. We compare it to three types of models: state-of-the-art benchmark models for multi-source DA (§5.2.1), T5-based baselines corresponding with ideas presented in multi-source DA work (§5.2.2), and variants of *PADA* that utilize different subsets of its components (§5.2.3). We consider two classification tasks (§5.1): Rumour detection and Multi-Genre Natural Language Inference (MNLI) and 10 DA setups. *PADA* outperforms all other models in most settings, exhibiting and impressive ability in reducing the performance drop between source and target domains.

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. Afterwards, we describe autoregressive language models and, finally, provide a summary of our main contributions with respect to these branches of literature.

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. Several representation-learning methods have shown highly effective for unsupervised DA. 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. Empirically, the resulting representations proved to transfer better across domains despite not using any text-based knowledge (Glorot et al., 2011; Chen et al., 2012; Yang and Eisenstein, 2014; Ganin et al., 2016). Another line of work 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

[2] Since we use a language model, pre-trained on massive unlabeled data, it is possible that this model was exposed to text from our source or target domains. However, in the downstream task training process, the model receives examples only from the source domains and is unaware of its future target domains.

abundant in the source and target domains. A recent line of work married the two approaches and achieved impressive performance gains (*AE-SCL* and *PBLM*, (Ziser and Reichart, 2017, 2018b)). Later on, Han and Eisenstein (2019) presented a pre-training method, followed by Ben-David et al. (2020) who introduced *PERL*, a pivot-based variant for pre-training contextual word embeddings.

Despite the impressive progress of UDA models, they all 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 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 target-domain data. In particular, good pivots are ones that are correlative with the task label. Instead, we define task-invariant DRFs, features which are highly correlated with the identity of the domain. Note, that domains are highly correlated with words, and hence our DRFs are lexical in nature.

We next discuss multi-source DA, a setting that considers more than one source domain. In this work we utilize multiple source domains to achieve better generalization to unknown target domains, although our method can also be applied when a single source is available.

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 Transfomers-based MoE. They found a naive expert-averaging 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 settings 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 (LM). The first implements the classic markovian language modeling approach, by training only a Transformer decoder model to generate the next word given its previous context, in an autoregressive manner (Radford and Narasimhan, 2018). This approach exhibited impressive performance mainly for generative tasks, possibly due to its autoregressive nature (Radford et al., 2019). The second casts the classic autoregressive language modeling task, as a masked tokens prediction task, by training a Transformer's encoder to derive contextualized word embeddings (CWE) (Devlin et al., 2019; Liu et al., 2019; Sanh et al., 2019). When fine-tuning the model for a new discriminative task, a custom decoder is implemented and jointly trained with the model's pre-trained encoder. This optimizes the language model's CWE representations for predicting the downstream task. This approach exhibited impressive performance mainly for discriminative tasks, due to the emphasis on downstream task-specific

fine-tuned representations (Devlin et al., 2019; Liu et al., 2019).

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 language model (encoder-decoder), 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. With far more pre-trained parameters than its predecessors 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 being performed, is prepended as a prefix to all task-related input examples. Other works have also utilized such a prompting mechanism for slightly different purposes, yet all use it as a way for priming the model for *in-context learning* (Brown et al., 2020), to elicit context-specific information from the language model (Jiang et al., 2019; Sun and Lai, 2020; Shin et al., 2020), or as a method for tuning the model (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.

Summary of Contributions In light of the related work, we argue to make four main contributions: First, we stress the importance of any-domain adaptation, i.e. establishing a DA system for unknown target domains. Compared to previous settings in the field, any-domain adaptation better reflects a long standing DA goal: generalization beyond the training distribution. Moreover, it is a realistic setup that seeks to improve the applicability of DA algorithms and overcome their engineering limitations. Second, we define DRFs, features which correlate with the domain label. DRFs provide a semantic signature of the domain, by this separating different domains but also bridging the gaps between domains with similar semantics. In contrast to pivot features, DRFs are task-independent, thus can be integrated into any task.

Third, we apply a prompting mechanism for DA, where the prompt is formed by DRFs which describe semantic relations between the domain and the example. The prompt is autoregressively generated for a given input example, and employed during task prediction. Fourth, we propose training a single model on multiple source domains for the purpose of any-domain adaptation. This comes in contrast to the MoE approach, where a separate expert model is trained for each source domain, which results in a much larger overall model and lacks the ability to share parameters and semantic features overlapping between different domains.

3 Problem Statement: Any-Domain Adaptation

We first provide a formal definition for DA. We then proceed to define the problem setting addressed within this work. Finally, we position our task-definition in relation to other commonly studied problem settings for DA.

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 Definition In this work, we are focused 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 for unsupervised multi-source domain adaptation, by

[3] On the contrary, in *inductive transfer learning* $\mathcal{T}_S$ differs from $\mathcal{T}_T$.

assuming we have no knowledge or access to labeled or unlabeled data from the target domains. We only assume to have access to labeled training data from K source domains $\{\mathcal{D}_{S_i}\}_{i=1}^{K}$, where $\mathcal{D}_{S_i} \triangleq \{(x_t^{S_i}, y_t^{S_i})\}_{t=1}^{K}$. 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 the proposed problem "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. Under such terminology, our problem can be placed somewhere in between the works above, without properly identifying the differences. In our view these differences should be clarified, and thus we present a designated terminology for our setup.

4 Prompt-based Autoregressive Domain Adaptation

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 seperate 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 (Raffel et al., 2020; Shin et al., 2020; Li and Liang, 2021), 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 *restau-*

rants 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 string of features derived from the DRF set of S_i, and training examples are automatically annotated with their R_i in a process that is 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 training protocol of the T5 model for multi-task cases, 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.

Our *PADA* model conditions the classification on the generated prompt. To consider the effect of this conditioning, we also consider a simpler variant of *PADA*, 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 emphasize the fact 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.

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

4.2 Domain Related Features

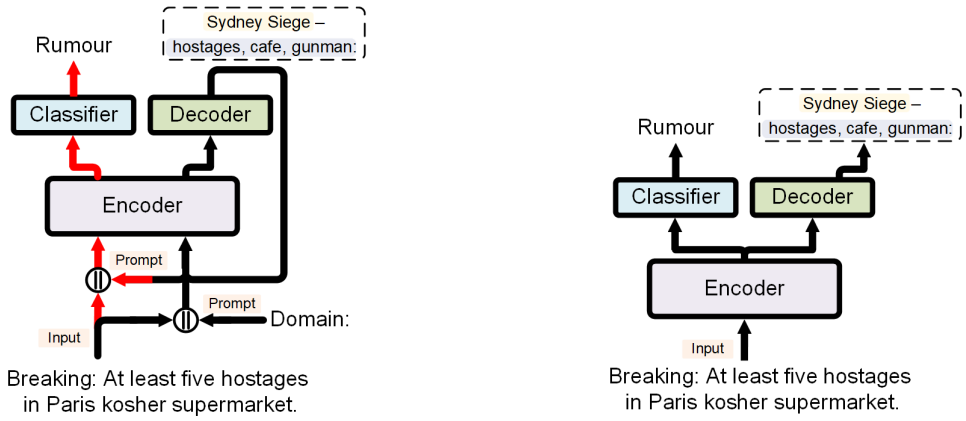
For each domain we define the DRF set such that these features provide a semantic signature for the domain. Importantly, if two domains have shared semantics, for example the *restaurants* and the *cooking* domains, we expect their DRFs to semantically overlap. Since the prompt of each training example 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 domains. 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 phrase 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 the each source domain as follows. Let examples (texts) from the jth source domain ($\mathcal{S}_j$) be labeled with 1, and examples from all other domains ($\mathcal{S}\backslash\mathcal{S}_j$) be labeled with 0. We first calculate the *mutual-information* (MI) between all tokens and the domain label space, and sort the tokens in descending order according to their MI score. Note, that the MI criterion might also promote tokens which are highly disassociated with $\mathcal{S}_j$. Thus, we only select the top tokens which meet the following condition:

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

where $C_{\mathcal{S}_j}(n)$ is the count of the n-gram n in $\mathcal{S}_j$, $C_{\mathcal{S}\backslash\mathcal{S}_j}(n)$ 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 $\mathcal{S}\backslash\mathcal{S}_j$, 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 jth domain with R_j.



(a) *PADA*. (b) *PADA-NC*.

Figure 2: A multi-task model with a generative head trained for DRF generation (§4.1) and a discriminative head for rumour classification. Text marked with blue signifies the DRFs and text marked with yellow signifies the domain name. Black arrows (→) mark the first inference step and red arrows (→) 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 *Sydney-siege*, and associates it with 3 DRFs: *hostages*, *cafe*, and *gunman*.

Given a training example i from domain j, we select the m features from R_j which are most associated to this example to form its prompt. To do that, we compute the L2 distance between the T5 embeddings of the DRF features with 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]

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.

[4]In this computation we consider the non-contextual embeddings learned by T5 during its pre-training. In our experiments we consider only unigrams (words) as DRFs.

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

Table 1: The number of examples in our domains for the Rumour Detection and MNLI datasets. We denote the examples used when a domain is included as a source domain (src), and when it is the target domain (trg).

5 Experimental Setup

5.1 Task and Datasets

As described in §3, we are interested in the settings of any-domain adaptation. Thus, we experiment with multi-source DA tasks, where a model is trained on several domains and applied to a new one. We experiment with two datasets (tasks): Rumour Detection and the Multi-Genre Natural Language Inference (MNLI). 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, 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 Detection[5] The rumour detection dataset is based on 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.

5.2 Evaluated Models

Our main model is ***PADA***: The multi-task model that first generates 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 three types of models: (a) state-of-the-art rumour detection models for our multi-source DA setup;[8] (b) T5-based baselines corresponding with ideas presented in multi-source DA work (§2); and (c) Ablation models that use specific parts of our proposed model, *PADA*, to highlight the importance of its various components. We next describe these models.

5.2.1 Benchmark 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 is demonstrated to achieve state-of-the-art performance for rumour detection (the paper reports the CCRF results as its previous state-of-the-art).[9]

5.2.2 Baseline Models

T5-MoE A T5-based MoE ensemble model. For each source domain, a separate pre-trained T5-base model is fine-tuned on the domain's training

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

[6] This 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/

[8] Unfortunately, we are not familiar with existing multi-source DA models for MNLI. We implement our T5-based baselines for both rumour detection and MNLI, and these are very strong baselines that for rumour detection outperform the previous state-of-the-art models reported in the literature.

[9] In this preliminary version, we compare to the results reported in the original CCRF and Tr-MoE papers, which used the same test sets and generated the training an development data following the same sampling procedure we applied. Note, that due to the random sampling of the training and development sets, their training and development sets might differ from ours, although they are of the same size like ours.

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 T5-based pre-trained language model's encoder, 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 available source domains, in a straightforward manner.

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

5.2.3 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 and Hyper-parameters

For all implemented models we use the *HuggingFace* Transformers library (Wolf et al., 2020).[10] We train all models for 5 epochs with an early stopping criterion according to performance on the development data. All parameters are optimized using the *ADAM* optimizer (Kingma and Ba, 2015), with a batch size of 32, warmup ratio of 0.1, and a learning rate of $5 \cdot 10^{-5}$. The maximum input length of the T5-based models is set to 128 tokens, and the maximum output length is 40 tokens. We pad shorter sequences and truncate longer ones to the maximum input length.

The T5-based models do not follow the same task classification 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.[11]

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: $\alpha_{rumour} = 0.75$ and $\alpha_{mnli} = 0.1$ for Rumour Detection and MNLI, respectively. 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 hyper-parameters: We generate 5 candidates, using a beam size of 10, with 5 beam groups, and a diversity penalty value of 1.5. The ρ parameter in the DRF extraction procedure (§4.2) was tuned to 1.5, for all domains.

6 Results

Tables 2 and 3 present our results. Each table reports results for its task on 5 settings, which correspond to 5 different target domains, and the average score across all settings. We report the binary-F1 score for rumour detection, and the macro-F1 score for MNLI.[12] As described in §5.2, we distinguish between benchmark models proposed by previous works for rumour detection (*CCRF* and *Tr-MoE*), T5-based baseline models which follow ideas proposed in previous works (*T5-NoDA* and *T5-MoE*), and our own ablation models (*PADA-DN* and *PADA-NC*), which are simplified variants of *PADA*.

PADA 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*

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

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

[12] Binary-F1 measures the F1 score on the positive class. It is useful in cases of unbalanced datasets where the positive class is of interest. In the rumour detection dataset, 34% of the examples belong to the positive class.

Rumour Detection

	All → C	All → FR	All → GW	All → OS	All → S	AVG
CCRF	63.6	46.5	70.4	69.0	61.2	62.1
Tr-MoE	67.9	45.4	74.5	62.6	64.7	63.0
T5-NoDA	64.1	46.9	**75.1**	72.0	71.0	65.8
T5-MoE	68.1	46.0	73.6	65.3	66.3	63.9
PADA-DN	66.4	53.7	72.4	71.4	70.1	66.8
PADA-NC	65.8	**54.8**	71.6	72.2	74.0	67.7
PADA	**68.6**	54.4	73.0	**75.2**	**75.1**	**69.3**

Table 2: Binary-F1 scores for the rumour detection tasks.

MNLI

	All → F	All → G	All → SL	All → TE	All → TR	AVG
T5-NoDA	76.4	83.5	75.5	74.9	81.3	78.3
T5-MoE	74.0	82.0	73.4	74.6	78.3	76.5
PADA-DN	**77.0**	**84.4**	75.6	76.3	80.5	78.8
PADA-NC	76.2	83.6	75.4	77.2	81.4	78.8
PADA	76.4	83.4	**76.9**	**78.9**	**82.5**	**79.6**

Table 3: Macro-F1 scores for the MNLI tasks.

models that is the best performing model. Interestingly, it is *T5-NoDA*, a model that does not perform any DA, that outperforms all models that do not belong to the *PADA* family, including the MoE models. This highlights the limitations of previous approaches to our problem.

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 in our setup. 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. These results reveal that the improvements gained by *PADA* are in fact substantial in both tasks.

The higher quality predictions are not the only advantage of *PADA* over MoE (both *T5-MoE* and *Tr-MoE*). Particularly, for *PADA* we train a single model while for MoE we train a unique model for each source domain, hence 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, and this number is kept fixed regardless of the number of source domains.

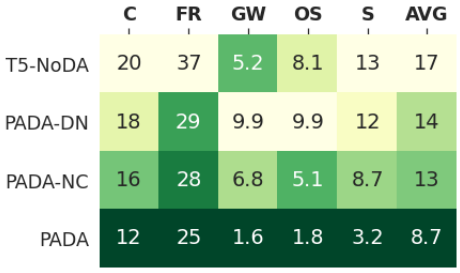
Our results exhibit the superiority of *PADA* and its variants, *PADA-DN* and *PADA-NC*, over all other models in 9 of 10 settings. Particularly, *PADA* outperforms the non-*PADA* models in 7 of our 10 settings, *PADA-NC* outperforms these models in 6 settings, and *PADA-DN* in 5 settings. Moreover, *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*).

7 Ablation Analysis

This paper is a preliminary version that will soon be extended and submitted to a leading NLP publication venue. Hence we provide here a single ablation study. More studies will be provided in future versions.

Performance shifts between source and target As mentioned in §1, DA addresses a key challenge in machine learning: Generalization beyond the

(a) **Rumour Detection**



(b) **MNLI**

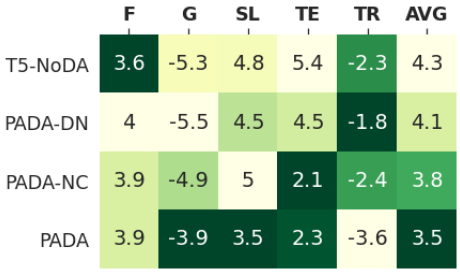


Figure 3: Heatmaps presenting performance shifts between source domains and target domains, for Rumour Detection (3a) and MNLI (3b), in all multi-source DA settings. Negative values represent improved target compared to source results. We normalize the colors per column, such that darker colors represent smaller absolute value of performance shifts.

training distribution. Mostly, DA research focuses on improving performance in target domains. For some DA settings, a performance gain on the target domain may simply be attributed to an improvement on the source domain. Other times, an improvement on the source domain actually widens the performance gap between the two 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 two heatmaps depicting the performance shifts of each model between the source domains and the target domains, in our multi-source DA settings. 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: *PADA*, *PADA-NC*, *PADA-DN*, and *T5-NoDA* models.[13] The general trend is clear, *PADA* not only performs better on the target domain, but it also substantially reduces performance drops between source and target domains. Furthermore, it is not surprising that *T5-NoDA* triggers the largest average absolute performance shifts (17% for rumour detection and 4.3% for MNLI), since it is not a DA model. In contrast, the average of *PADA*'s absolute performance shifts are 8.7% for rumour detection and 3.5% for MNLI.

8 Discussion

We addressed a very challenging variant of the domain adaptation problem in NLP, which has received limited attention in previous literature. In our setup an algorithm is trained on several source domains and then applied to examples from an unseen domain. Importantly, the target domain is unknown at training time, and the algorithm is not exposed to data (annotated or not) from this domain or to any other information about it. Apart from the intellectual contribution, a model that effectively operates in this setup provides an efficient solution: It is a single model that 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 a function of the number of source or target domains.

PADA, our proposed algorithm, takes advantage of the prompting mechanism of the T5 autoregressive language model. In order to map the test example into the semantic space spanned by the source domains, it generates a unique example-based signature (prompt): a sequence of unrestricted length, consisting of source domain names as well as predefined Domain Related Features (DRFs) that characterize each of the source domains.

Our experimental results with two tasks and ten multi-source adaptation settings demonstrate the effectiveness of our approach compared to state-of-the-art methods and strong baselines, 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 uni-

[13] These are the best performing models according to Table 2 and Table 3.

fied 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, dedicated model per domain.

When considering the prompt generation mechanism proposed in *PADA*, our approach is still limited by the set of source domains *PADA* is trained on. This affects the type of DRFs which are generated for a given test example, and might yield sub-optimal features 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. These limitations are non-trivial to solve, yet might affect the model's overall ability to generalize to any possible domain, and might harm its overall performance. In future work, we plan to explore new ideas for tackling these limitations, in hopes of improving overall model performance and generalization in the any-domain adaptation setup. Furthermore, our framework can naturally be extended to accommodate multiple tasks and domains in a single model, which is another direction for future research.

References

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. *J. Artif. Intell. Res.*, 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, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. 2016. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 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.

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.

Jing Jiang and ChengXiang Zhai. 2007. Instance weighting for domain adaptation in NLP. 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.

Zhengbao Jiang, F. F. Xu, J. Araki, and Graham Neubig. 2019. 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.

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, Ziyan 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.

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.

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.

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, Guanhang 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 domain 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.