Example-based Hypernetworks for Out-of-Distribution Generalization

Tomer Volk*[1], Eyal Ben-David*[1], Ohad Amosy[2], Gal Chechik[2 3], Roi Reichart[1]
[1]Faculty of Industrial Engineering and Management, Technion, IIT
[2]Bar Ilan University, Israel
[3]NVIDIA Research
{tomervolk@campus.|eyalbd12@campus.|roiri@}technion.ac.il
{amosyoh|gal.chechik}@biu.ac.il

Abstract

While Natural Language Processing (NLP) algorithms keep reaching unprecedented milestones, out-of-distribution generalization is still challenging. In this paper we address the problem of multi-source adaptation to unknown domains: Given labeled data from multiple source domains, we aim to generalize to data drawn from target domains that are unknown to the algorithm at training time. We present an algorithmic framework based on *example-based Hypernetwork adaptation*: Given an input example, a T5 encoder-decoder first generates a unique signature which embeds this example in the semantic space of the source domains, and this signature is then fed into a Hypernetwork which generates the weights of the task classifier. In an advanced version of our model, the learned signature also serves for improving the representation of the input example. In experiments with two tasks, sentiment classification and natural language inference, across 29 adaptation settings, our algorithms substantially outperform existing algorithms for this adaptation setup. To the best of our knowledge, this is the first time Hypernetworks are applied to domain adaptation or in example-based manner in NLP.[1]

1 Introduction

Deep neural networks (DNNs) have substantially improved natural language processing (NLP), reaching task performance levels that were considered beyond imagination until recently (Conneau and Lample, 2019; Brown et al., 2020). However, this unprecedented performance typically depends on the assumption that the test data is drawn from the same underlying distribution as the training data. Unfortunately, as text may stem from many origins, this assumption is often not met in practice. In such cases, the model faces an out-of-distribution (OOD) generalization scenario, which often yields significant performance degradation.

To alleviate this difficulty, several OOD generalization approaches proposed to use unlabeled data from the target distribution. For example, a prominent domain adaptation (DA, (Daumé III, 2007; Ben-David et al., 2010)) setting is unsupervised domain adaptation (UDA, (Ramponi and Plank, 2020)), where algorithms use labeled data from the source domain and unlabeled data from both the source and the target domains (Blitzer et al., 2006, 2007; Ziser and Reichart, 2017, 2018b). In many real-world scenarios, however, it is impractical to expect training-time access to target domain data. This could happen, for example, when the target domain is unknown, when collecting data from the target domain is impractical or when the data from the target domain is confidential (e.g. in healthcare applications or in applications that involve user data). In order to address this setting, three approaches were proposed.

The first approach follows the idea of *domain robustness*, generalizing to unknown domains through optimization methods which favor robustness over specification (Hu et al., 2018; Oren et al., 2019; Sagawa et al., 2020; Wald et al., 2021). Particularly, these approaches train the model to focus on domain-invariant features and overlook properties that are associated only with some specific source domains. In contrast, the second approach implements a domain expert for each source domain, hence keeping the knowledge acquired from each domain separated from the knowledge acquired from the others. In this *mixture-of-experts (MoE)* approach (Kim et al., 2017; Guo et al., 2018; Wright and Augenstein, 2020), an expert is trained for each domain sepa-

* Both authors equally contributed to this work.
[1]Our code and data are available at https://github.com/TomerVolk/Hyper-PADA

rately, and the predictions of these experts are aggregated through averaging or voting.

To bridge the gap between these opposing approaches, a third intermediate approach has been recently proposed by Ben-David et al. (2021). Their PADA algorithm, standing for a Prompt-based Autoregressive Approach for Adaptation to Unseen Domains, utilizes both domain-invariant and domain-specific features to perform *example-based* adaptation. Particularly, given a test example it generates a unique prompt that maps this example to the semantic space of the source domains of the model, and then conditions the task prediction on this prompt. In PADA, a T5-based algorithm (Raffel et al., 2020), the prompt-generation and task prediction components are jointly trained on the source domains available to the model.

Despite their promising performance, none of the previous models explicitly learns both shared and domain-specific aspects of the data, and effectively applies them together. Particularly, robustness methods focus only on shared properties, MoE methods train a separate learner for each domain, and PADA trains a single model using the training data from all the source domains, and applies the prompting mechanism in order to exploit example-specific properties. This paper hence focuses on improving generalization to unseen domains by explicitly modeling the shared and domain-specific aspects of the input.

To facilitate effective parameter sharing between domains and examples, we propose a modeling approach based on *Hypernetworks* (HNs, Ha et al. (2017)). HNs are networks that generate the weights of another target network, that performs the learning task. The input to the HN defines the way information is shared between training examples. To the best of our knowledge, we are the first to apply HNs for DA in NLP.

We propose three models of increasing complexity. Our basic model is Hyper-DN, which explicitly models the shared and domain-specific aspects of the training domains. Particularly, it trains the HN on training data from all source domains, to generate classifier weights in a domain-specific manner. The next model, Hyper-DRF, an example-based HN, performs parameter sharing at both the domain and the example levels. Particularly, it first generates an example-based signature as in PADA, and then uses this signature as input to the HN so that it can generate example-specific classifier weights.[2] Finally, our most advanced model is Hyper-PADA which, like Hyper-DRF, performs parameter sharing at both the example and domain levels, using the above signature mechanism. Hyper-PADA, however, does that at both the task classification and the input representation levels. For a detailed description see §3.

We follow Ben-David et al. (2021) and experiment in the any-domain adaptation setup (§4,5). Concretely, given access to labeled datasets from multiple domains, we perform leave-one-out experiments, training the model on all domains but one and testing it on the remaining domain. Further, while our models are designed for cross-domain (CD) generalization, we can also explore cross-language cross-domain adaptation (CLCD) setups, by utilizing a multilingual pre-trained language model. Hyper-PADA outperforms an off-the-shelf SOTA model (a fine-tuned T5-based classifier, without any domain adaptation effort) by 9.5% (accuracy), 8.4% (accuracy) and 14.8% (macro-F1) in CLCD and CD sentiment classification (12 settings each) and CD MNLI (5 settings), on average, respectively. Moreover, our HN-based methods outperform previous models from the three families described above. Finally, ablative comparisons between our HN-based algorithms shed light on the relative importance of their components.

2 Related Work

2.1 Domain Adaptation

Domain Adaptation (DA) is a fundamental challenge in NLP, with two common setups: supervised and unsupervised. In supervised DA, the algorithm utilizes a small amount of labeled data from the target domain (Daumé III and Marcu, 2006; Bollegala et al., 2011), while in unsupervised DA it has access to labeled data form the source domains and unlabeled data from both source and target domains (Blitzer et al., 2006, 2007; Reichart and Rappoport, 2007; Glorot et al., 2011). Most recent DA research addresses the more realistic UDA setup. Since the rise of DNNs, the main focus of UDA resaerch shifted to representation learning methods (Titov, 2011; Glorot et al., 2011; Ganin and Lempitsky, 2015; Ziser and Reichart, 2017, 2018a, 2019; Rotman and

[2] DRFs stand for *Domain Related Features* and DN stands for *Domain Name*. See §3.2

Reichart, 2019; Han and Eisenstein, 2019; Ben-David et al., 2020; Lekhtman et al., 2021).

The recent DA setup that we consider in this paper assumes no training-time knowledge about the target domain (denoted as *any-domain adaptation* by Ben-David et al. (2021)). As discussed in §1, some papers that addressed this setup follow the domain robustness path (Arjovsky et al., 2019), while others learn a mixture of domain experts (Wright and Augenstein, 2020) or train the model on data from multiple domains and adapt test examples from unknown domains through prompting (Ben-David et al., 2021). Unlike previous DA work in NLP, we perform adaptation through hypernetworks which are trained to generate the weights of the task classifier in a domain-based or example-based manner. This framework allows us to both explicitly model domain-invariant and domain-specific aspects of the training data, and perform example-based adaptation.

Li et al. (2021) perform example-based adaptation. They address the same setup as us, multi-source adaptation to unknown domains, but for dependency parsing. Their model integrates two designated NNs which generate domain-invariant and domain-specific representations for each input example. However, they do not apply HNs and hence cannot share parameters at the task classification level as we do. Moreover, they send the entire input example into the designated NNs, while we aim to learn a more sophisticated signature mechanism which aligns the input example with the source domains ((Ben-David et al., 2021), see §3), in order to facilitate effective parameter sharing across domains and examples, at both the classifier and the representation learning levels.

2.2 Hypernetworks

Hypernetworks (Ha et al., 2017) are (typically small) networks that learn to generate weights for other networks. Intuitively, HNs can generate diverse personalized models, conditioned on the input. HNs were applied in areas like computer vision (Klein et al., 2015; Riegler et al., 2015; Klocek et al., 2019), continual learning (von Oswald et al., 2020), federated learning (Shamsian et al., 2021), weight pruning (Liu et al., 2019), Bayesian neural networks (Krueger et al., 2017; Ukai et al., 2018; Pawlowski et al., 2017; Deutsch et al., 2019), multi-task learning (Shen et al., 2018; Klocek et al., 2019; Serrà et al., 2019; Meyerson and Miikkulainen, 2019) and block code decoding (Nachmani and Wolf, 2019).

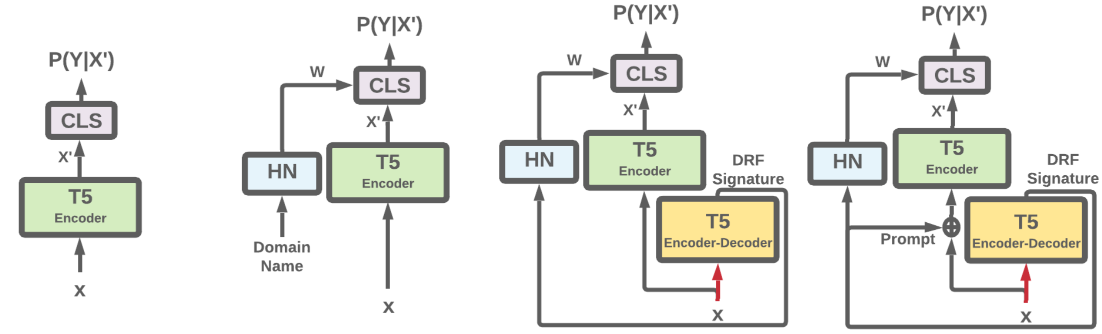
Despite being widely used in other ML branches, HNs research in NLP is limited. HNs were shown to be effective for language modeling (Suarez, 2017) and machine translation (Platanios et al., 2018). Moreover, Üstün et al. (2020) and Mahabadi et al. (2021) applied HNs to Transformer architectures (Vaswani et al., 2017) in cross-lingual parsing and multi-task learning, by generating adapter (Houlsby et al., 2019) weights and keeping the pre-trained language model weights fixed. In contrast to previous Transformer-based approaches, we apply HNs for generating the weights of a task classifier, where we train the HN jointly with the fine-tuning of a large LM. Furthermore, following Ben-David et al. (2021) we perform example-based adaptation, a novel application of HNs in NLP: To the best of our knowledge, HNs have not been applied in NLP in an example-based manner before.

3 Domain Adaptation with Hypernetworks

In this section, we present our HN-based modeling framework for domain adaptation. We present three models in increased order of complexity: We start by generating parameters only for the task classifier in a domain-based manner (Hyper-DN), proceed to example-based classifier parametrization (Hyper-DRF) and, finally, introduce example-based parametrization at both the classifier and the text representation levels (Hyper-PADA).

Throughout this section we use the running example of Table 1. This is a Natural Language Inference (NLI) example from one of our experimental MNLI (Williams et al., 2018) setups. In this task, the model is presented with two sentences, Premise and Hypothesis, and it should decide the relationship of the latter to the former: Entailment, Contradiction or Neutral (see §4).

§3.1 describes the model architectures and their training procedure. §3.2 then delves into the specific details of the DRF scheme, borrowed from Ben-David et al. (2021). The DRFs are utilised in order to embed input examples in the semantic space of the source domains, hence supporting example-based classifier parametrization and improved example representation.



(a) T5-NoDA (b) Hyper-DN (c) Hyper-DRF (d) Hyper-PADA

Figure 1: The four models representing the evolution of our HN-based domain adaptation framework. From left to right: ***T5-NoDA*** is a standard NLP model comprised of a pre-trained T5 encoder with a classifier on top of it, both are fine-tuned with the downstream task objective. ***Hyper-DN*** employs an additional hypernetwork (HN), which generates the classifier (CLS) weights given the domain name (or an "UNK" specifier for examples from unknown domains). ***Hyper-DRF*** and ***Hyper-PADA*** are multi-stage multi-task models (first-stage inputs are in red, second stage inputs in black), comprised of a T5 encoder-decoder, a separate T5 encoder, a HN and a task classifier (CLS). At the first stage, the T5 encoder-decoder is trained for example-based DRF signature generation (§3.2). At the second stage, the HN and the T5 encoder are jointly trained using the downstream task objective. In Hyper-PADA, the DRF signature of the first stage is applied both for example representation and HN-based classifier parametrization, while in Hyper-DRF it is applied only for the latter purpose. In all HN-based models, our HN is a simple two-layer feed-forward NN (§4.3).

3.1 Models

Hyper Domain Name (Hyper-DN) Our basic model (Figure 1b) integrates a pre-trained T5 language encoder, a classifier (CLS), and a hypernetwork (HN), which generates the classifier weights. *Hyper-DN* casts the domain name as the input of the HN. Since the domain name is unknown at test-time inference, we use a special "UNK" token to represent unknown domains at this stage, for all input examples. In order to make this dummy domain name familiar to the model, during training we sample an α proportion of the training examples for which we use the "UNK" token as the HN input, instead of the domain name. This architecture supports parameter sharing between the input domains, and optimizes the weights for examples from unknown domains – all at the classifier level.

In the example of Table 1, the premise and hypothesis of the test example are fed into the T5 encoder, and the "UNK" token is fed to the HN. In this model, there is no generation of either a domain-name or an example-specific signature.

Hyper-DRF Parameter sharing based on the domain of an input example may not be sufficient, especially that the boundaries between domains are not always well defined. As an example, the sentence pair of our running example is taken from the *Government* domain but is also semantically related to the *Travel* domain. Thus, we present ***Hyper-DRF*** (Figure 1c), an example-based adaptation architecture, which makes use of domain-related features (DRFs, see § 3.2) in addition to the domain name. Importantly, this model may connect the input example to semantic aspects of multiple source domains.

Hyper-DRF is a multi-stage multi-task autoregressive model, which first generates a DRF signature for the input example, and then uses this signature as an input to the HN. The HN, in turn, generates the task-classifier (CLS) weights, but, unlike in Hyper-DN, these weights are example-based rather than domain-based. The model is comprised of the following components: **(1)** a T5 encoder-decoder model which generates the DRF signature of the input example in the first stage (*travel: city, area, town, reports, modern* in our running example); **(2)** a (separate) T5 encoder to which the example is fed in the second stage; and **(3)** a HN which is fed with the DRF signature, as generated in the first stage, and generates the

Premise. *Homes not located on one of these roads must place a mail receptacle along the route traveled.* **Hypothesis.** *Other roads are far too rural to provide mail service to.* **Domain.** *Government.* **Label.** *Entailment.* **DRF Signature.** *travel: city, area, town, reports, modern*
Fiction: *jon, tommy, tuppence, daan, said, looked, man, poirot, eyes, drew, inglethorp, mrs, julius, adrin, asked, sir, knew, doro, vandemeyer, stared, nodded, cavendish, fell, walked, dave* **Slate:** *clinton, president, says, york, percent, critics, new, bush, sex, starr, political, book, story, article, bill, newsweek,* ***reports****, according, robert, press, wrote, may, show, issues, cover* **Telephone:** *yeah, know, well, really, think, like, lot, mean, huh, get, right, hum, guess, okay, going, got, things, stuff, kind, pretty, good, probably, kids, something, yes* **Travel:** *century, island, built,* ***city****, museum, temple, ancient,* ***town****, palace, located, west, visitors, beach, sea, shops, church,* ***area****, south, roman,* ***modern****, known, tourists, along, visit, river*

Table 1: An example of Hyper-DRF and Hyper-PADA application to an MNLI example. In this setup the source training domains are *Fiction, Slate, Telephone and Travel* and the unknown target domain is *Government*. The top part presents the example and the DRF signature generated by the models. The bottom-part presents the DRF set of each source domain in this setup.

weights of the task-classifier (CLS). This CLS is fed with the example representation, as generated by the T5 encoder of (2), to predict the task label.

Below we discuss the training of this model in details. The general scheme is as follows: We first train the T5 encoder-decoder of the first stage ((1) above), and then jointly train the rest of the architecture ((2) and (3) above), which is related to the second stage. For the first training stage we have to assign each input example a DRF signature. In §3.2 we provide the details of how, following Ben-David et al. (2021), the DRF sets of the source training domains are constructed based on the source domain training corpora, and how a DRF signature is comprised for each training example in order to effectively train the DRF signature generator ((1) above). For now, it is sufficient to say that the DRF set of each source domain is comprised of words that are strongly associated with this domain, and the DRF signature of each example is a sequence of DRFs (words).

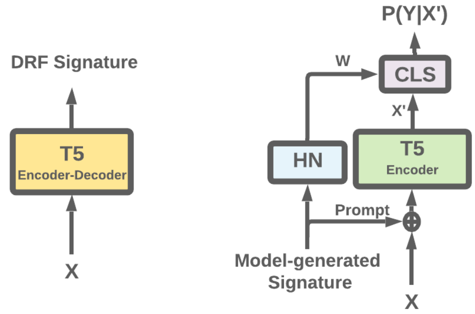
During inference, when introduced to an example from an unknown domain, *Hyper-DRF* generates its DRF signature using its trained generator (T5 encoder-decoder). This way, the signature of a test example may consist of features from the DRF sets of one or more source domains, forming a mixture of semantic properties of these domains. For example, in our running example, while the input sentence pair is from the unknown *Government* domain, the model generates a signature based on the *Travel* and *Slate* domains. Importantly, unlike in Hyper-DN, there is no need in an "UNK" token as input to the HN since the DRF signatures are example-based.

Hyper-PADA While Hyper-DRF implements example-based adaptation, parameter-sharing is modeled only at the classifier level: The language representation (with the T5 encoder) is left untouched. Our final model, ***Hyper-PADA***, casts the DRF-based signature generated at the first stage of the model, both as a prompt concatenated to the input example before it is fed to the T5 language encoder, and as an input to the HN.

Specifically, the architecture of *Hyper-PADA* (Figure 1d) is identical to that of Hyper-DRF. At its first stage, which is identical to the first stage of Hyper-DRF, it employs a generative T5 encoder-decoder which learns to generate an example-specific DRF signature for each input example. Then, at its second stage, the DRF signature is used in two ways: (A) unlike in Hyper-DRF, it is concatenated to the input example as a prompt, and the concatenated example is then fed into a T5 encoder, in order to create a new input representation (in Hyper-DRF the original example is fed into the T5 encoder); and (B) as in Hyper-DRF, it is fed to the HN which generates the task-classifier weights. Finally, the input representation constructed in (A) is fed into the classifier generated in (B) in order to yield the task label.

Training While some aspects of the selected training protocols are based on development data experiments (§4), we discuss them here in order to provide a complete picture of our framework.

For Hyper-DN, we found it most effective to jointly train the HN and fine-tune the T5 encoder using the task objective. As discussed



(a) Generator (b) Discriminator

Figure 2: *Hyper-PADA* training. The generative (T5 encoder-decoder) and discriminative (HN, T5 Ecnoder and CLS) components are trained separately, using source domains examples.

above, Hyper-DRF and Hyper-PADA are multi-stage models, where the HN (in both models) and the T5 language encoder (in hyper-PADA only) utilize the DRF signature generated in the first stage by the T5 encoder-decoder. Our development data experiments demonstrated significant improvements when using one T5 encoder-decoder for the first stage, and a separate T5 encoder for the second stage. Moreover, since the output of the first stage is discrete (a sequence of words), we cannot train all components jointly.

Hence, as illustrated in Figure 2 (for Hyper-PADA, but the same applies for Hyper-DRF), we train each stage of these models separately. First, the T5 encoder-decoder is trained to generate the example-based DRF signature (§3.2). Then, the HN and the (separate) T5 encoder are trained jointly with the task objective.

We next motivate the use of DRFs, provide their definition, and present their selection process for each source domain. We then describe the DRF-based prompt/signature annotation process, which is used for training.

3.2 Domain Related Features (DRFs)

In order to perform example-based domain adaptation, the first stage of the Hyper-DRF and Hyper-PADA models maps each input example into a sequence of Domain Related Features (DRFs). Selecting the DRF sets of the source domains is hence crucial for these models, as they should allow the models to map input examples to the semantic space of the source domains. Since a key goal of example-based adaptation is to account for soft domain boundaries, it is important that the DRF set of each source domain should reflect both the unique semantic aspects of this domain and the aspects it shares with other source domains.

To achieve these goals, we follow the definitions, selection, and annotation processes in Ben-David et al. (2021). For completeness, we briefly describe these ideas here.

DRF Set Construction Let S be the set of all source domains, and $S_j \in S$ the domain for which we construct the DRF set. We perform the following selection process, considering all the training data from the participating source domains. First, we define the domain label of a sentence to be 1 if the sentence is from S_j and 0 otherwise. We then look for the top l words with the highest *mutual information (MI)* with the 0/1 labels. Then, since MI could indicate association with each of the labels (related to the domain (1) or not (0)), and we are interested only in words associated with the domain, we select only words that meet the criterion:

$$\frac{C_{S \setminus S_j}(w)}{C_{S_j}(w)} \leq \rho, \; C_{S_j}(w) > 0$$

Where $C_{S \setminus S_j}(w)$ is the count of the word w in all of the source domains except S_j, $C_{S_j}(w)$ is the word count in S_j and ρ is a domain-specificity parameter: The smaller it is, the stronger is the association. The DRF set of S_j is denoted with R_j.

Annotating DRF-based Signatures for Training In order to train the DRF signature generator of Hyper-DRF and Hyper-PADA we have to construct a DRF signature for each training example. Our goal in this process is to match each training example with those DRFs in its domain's DRF set that are most representative of its semantics. We do this in an automatic manner.

Let $w_1, ...w_n$ be the tokens of a sentence x from the domain S_j. Each DRF $r_j \in R_j$ is assigned with the following score:

$$score(r_j, \{w_1, ...w_n\} \in x) = \min_{i=1,...n} \{s(r_j, w_i)\}$$

$$s(r_j, w_i) = \|\Phi(r_j) - \Phi(w_i)\|_2^2$$

where $\Phi(x)$ is the embedding of x in the pre-trained embedding layer of an off-the-shelf BERT model. Then, let $T_1, ..., T_k$ be the k DRFs with the lowest scores and D the domain name. We define the DRF signature of x to be the following string: "$D : T_1, ..., T_k$".

Sentence. *This documentary is poorly produced, has terrible sound quality and stereotypical "life affirming" stories. There was nothing in here to support Wal-Mart, their business practices or their philosophy.*
Domain. *DVD.*
Label. *Negative.*
DRF Signature. *music: history, rock, sound, story*

Table 2: An example of Hyper-DRF and Hyper-PADA application to a sentiment classification example. The source domains are *Books*, and *Music*. Generated DRF features from the *Books* and *Music* domains are in blue and green, respectively.

To summarize, we utilize this annotation only during training, as a training signal for the DRF signature generator (in stage 1 of both Hyper-DRF and Hyper-PADA). Tables 1 and 2 provide MNLI and sentiment classification examples and their DRF signatures, as generated by Hyper-PADA and Hyper-DRF in a specific adaptation setup.

4 Experimental Setup

4.1 Tasks, Datasets, and Setups

While our focus is on domain adaptation, the availability of multilingual pre-trained language encoders allows us to consider two setups: (1) Cross-domain transfer (CD); and (2) cross-language cross-domain transfer (CLCD). We consider multi-source adaptation and experiment in a leave-one-out fashion: In every experiment we leave one domain (CD) or one domain/language pair (CLCD) out, and train on the datasets that belong to the other domains (CD) or the datasets that belong to both other domains and other languages (CLCD; neither the target domain nor the target language are represented in the training set).

Cross-domain Transfer (CD) for Natural Language Inference We experiment with the MNLI dataset (Williams et al., 2018).[3] In this task, each example consists of a premise-hypothesis sentence pair and the relation between the the latter and the former: Entailment, contradiction, or neutral. The corpus consists of ten domains, five of which are split to train, validation, and test sets, while the other five do not have training sets. We experiment with the former five: Fiction (F), Government (G), Slate (S), Telephone (TL), and Travel (TR).

Sentiment Analysis (En, De, Fr, Jp)			
Domain	**Training (src)**	**Dev (src)**	**Test (trg)**
Books (B)	500	100	2000
DVD (D)	500	100	2000
Music (M)	500	100	2000
MNLI (En)			
Domain	**Training (src)**	**Dev (src)**	**Test (trg)**
Fiction (F)	2500	200	1,973
Government (G)	2500	200	1,945
Slate (SL)	2500	200	1,955
Telephone(TL)	2500	200	1,966
Travel (TR)	2500	200	1,976

Table 3: The number of examples in each domain (and language) of our two tasks. We denote the examples used when a domain is included as a source domain (src), and when it is the target domain (trg). For sentiment we present the number of examples in a single language, while there are four different languages - English (En), Deutsch (De), French (Fr), and Japanese (Jp), each with the same number of examples per domain.

Since the MNLI test sets are not publicly available, we use the validation sets as our test sets and split the train sets to train and validation. We downsample each domain to have 2500 train and 200 validation examples, focusing on a challenging low-resource adaptation setup (Table 3).

Cross-language Cross-domain (CLCD) and Multilingual Cross-domain (CD) Transfer for Sentiment Analysis We experiment with the task of sentiment classification, using the Websis-CLS-10 dataset (Prettenhofer and Stein, 2010),[4] which consists of Amazon reviews from 4 languages (English (En), Deutsch (De), French (Fr), and Japanese (Ja)) and 3 product domains (Books (B), DVDs (D), and Music (M)).

We perform one set of multilingual cross-domain (CD) generalization experiments and one set of cross-language cross-domain (CLCD) experiments. In the former, we keep the training language fixed and generalize across domains, while in the latter we generalize across both languages and domains. Hence, experimenting in a leave-one-out fashion, in the CLCD setting we focus each time on one domain/language pair. For instance, when the target pair is *English-Books*, we train on the training sets of the *{French, Deutsch,*

[3] https://cims.nyu.edu/~sbowman/multinli/

[4] https://zenodo.org/record/3251672#.YdQiIWhBwQ8

Japanese} languages and the *{Movies, Music}* domains (a total of 6 sets), and the test set consists of *English* examples from the *Books* domain. Likewise, in the CD setting we keep the language fixed in each experiment, and generalize from two of the domains to the third one. We hence have 12 CLCD experiments (one with each language/domain pair as target) and 12 CD experiments (for each language we perform one experiment with each domain as target). As for MNLI, we downsample each language-domain pair to include 500 train and 100 validation examples (Table 3).

4.2 Models and Baselines

We compare our hypernetwork based models (***Hyper-DN***, ***Hyper-DRF***, and ***Hyper-PADA***) to models from three families (see §1): (a) *domain expert models* that does not share information across domains: a model trained on the source domains and applied to the target domain with no adaptation effort (*T5-NoDA*); and a mixture of domain-specific experts, where a designated model is trained on each target domain, and test decisions are made through voting between the predictions of these models (*T5-MoE*, (Wright and Augenstein, 2020)); (b) *domain robustness models*, targeting generalization to unknown distributions through objectives that favor robustness over specification (*T5-DANN* (Ganin and Lempitsky, 2015) and *T5-IRM* (Arjovsky et al., 2019)); and (c) *example-based multi-source adaptation* through prompt learning (*PADA* (Ben-David et al., 2021), the SOTA model for our setup).

Below we briefly discuss each of these models. All models, except from T5-MoE are trained on a concatenation of the source domains training sets.

(a.1) T5-No-Domain-Adaptation (T5-NoDA) A model consisting of a task classifier on top of a T5 encoder. The entire architecture is fine-tuned on the downstream task (see Figure 1a).

(a.2) T5-Mixture-of-Experts (T5-MoE) We fine-tune an expert model (with an identical architecture to the one used by *T5-NoDA*) on the training data from each domain. At inference, we average the class probabilities of all experts, and the class with the maximal probability is selected.

(b.1) T5-Invariant-Risk-Minimization (T5-IRM) An expert with the same architecture as *T5-NoDA*, but with an objective term that penalizes representations that have different optimal classifiers across domains.

(b.2) T5-Domain-Adversarial-Network (T5-DAN) An expert with the same architecture as *T5-NoDA*, but with an additional adversarial domain classifier head (fed by the T5 encoder) which facilitates domain invariant representations.

(c.1) PADA A T5 encoder-decoder that is fed with each example and generates its DRF signature. The example is then appended with this signature as a prompt, fed again to the T5 encoder and the resulting representation is fed into the task classifier. We follow the implementation and training details from (Ben-David et al., 2021).

For each setup we also report an upper-bound: The performance of the model trained on the training sets from all source domains (or source language/domain pairs in CLCD) including that of the target, when applied to the target domain's (or language/domain pair in CLCD) test set.

4.3 Implementation Details

For all the pre-trained models we use the *Huggingface* Transformers library (Wolf et al., 2020).[5] For the T5 model we use the T5-base model (Raffel et al., 2020) for MNLI, and the MT5-base model (Xue et al., 2021) for sentiment classification. For contextual representation of the HN input (domain name or "UNK' in Hyper-DN, DRF signature in Hyper-DRF and Hyper-PADA), we use the BERT-base-uncased and the mBERT-based-uncased models, for MNLI and sentiment classification, respectively.

We choose $\rho = 1.5$ for the DRF set construction process. In the DRF signature annotation process, we take the $k = 5$ most associated DRFs for each input example. When generating the signature (in Hyper-DRF and Hyper-PADA) we employ the Diverse Beam Search algorithm (Vijayakumar et al., 2016) with the T5 decoder, using the following parameters: 5 sequences, with a beam size of 5, a 5 beams group and a diversity penalty of 0.1.

The HN consists of two linear layers of the same input and output dimensions (1×768), each of which is followed by a ReLU activation layer. The output of the second layer is fed into two parallel linear layers, one to predict the weights of the linear classifier (a 2×768 matrix), and one to predict its bias (a 1×2 vector). For task classification, we

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

	Deutsch			English			French			Japanese			
	B	**D**	**M**	**B**	**D**	**M**	**B**	**D**	**M**	**B**	**D**	**M**	**Avg**
T5-NoDA	77.1	75.8	63.9	78.4	78.8	64.5	83.0	82.6	75.1	61.5	79.9	79.7	75.0
T5-MoE	81.9	76.6	79.6	86.0	81.2	81.6	85.0	84.9	77.2	82.2	83.6	82.0	81.8
T5-DANN	82.1	77.8	80.8	84.6	78.8	79.0	84.2	82.6	77.2	68.7	78.8	81.6	79.7
T5-IRM	71.2	70.2	75.8	80.8	72.5	73.0	82.3	80.6	78.4	75.5	75.8	78.4	76.2
PADA	57.7	74.8	74.2	71.8	75.9	78.8	81.8	82.0	76.8	77.2	78.8	80.0	75.8
Hyper-DN	**86.2**	80.8	84.4	85.6	84.2	83.4	86.5	84.5	81.6	81.3	82.0	83.2	83.7
Hyper-DRF	85.9	81.2	84.6	**86.4**	84.0	83.9	85.7	85.5	81.4	82.2	82.0	**83.9**	83.9
Hyper-PADA	85.7$^{\ddagger\diamond+}$	**81.8**$^{\clubsuit\ddagger\diamond+}$	**85.0**$^{\clubsuit\ddagger+}$	86.0$^{\ddagger\diamond}$	**84.4**$^{\clubsuit\ddagger\diamond+}$	**85.1**$^{\clubsuit\diamond+}$	**86.6**$^{\clubsuit\ddagger\diamond+}$	**85.9**$^{\ddagger\diamond+}$	**81.8**$^{\clubsuit\ddagger\diamond+}$	**83.9**$^{\ddagger\diamond+}$	**83.9**$^{\clubsuit\diamond+}$	83.8$^{\ddagger\diamond+}$	**84.5**
Upper-bound	86.7	83.8	86.4	88.7	85.9	86.9	87.9	87.3	83.9	84.4	86.4	86.9	86.3

Table 4: CLCD sentiment classification accuracy. The statistical significance of the Hyper-PADA results (with the McNemar paired test for labeling disagreements (Gillick and Cox, 1989), $p < 0.05$) is denoted with: $\clubsuit$ (vs. the best of Hyper-DN and Hyper-DRF), $+$ (vs. the best domain expert model), $\diamond$ (vs. the best domain robustness model), and $\ddagger$ (vs. PADA (example-based adaptation)).

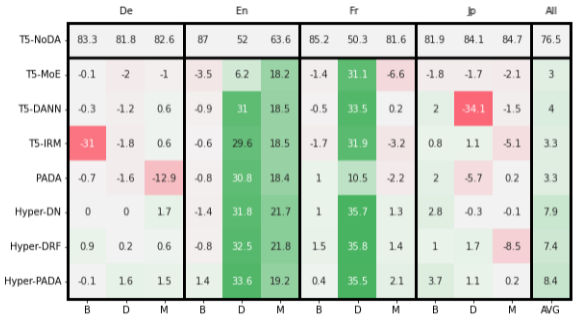


Figure 3: Accuracy improvements over T5-NoDA, in cross-domain (CD) generalization for four languages: German, English, French, and Japanese. From the 28 out of 36 settings where Hyper-PADA outperforms the best model in each of the baselines groups, in 23 cases the difference is significant (we follow the same protocol as in Table 4).

	F	G	S	TL	TR	Avg
T5-NoDA	58.2	66.0	60.2	74.3	69.1	65.6
T5-MoE	55.6	65.3	57.7	58.1	64.3	60.2
T5-DANN	72.1	76.9	65.7	74.8	76.1	73.1
T5-IRM	51.1	64.6	51.7	54.7	64.5	57.3
PADA	76.7	79.6	75.3	78.1	75.2	77.0
Hyper-DN	74.5	81.2	74.9	76.7	79.8	77.4
Hyper DRF	75.3	82.3	73.8	76.3	78.7	77.3
Hyper PADA	**79.0**$^{\clubsuit\ddagger\diamond+}$	**84.1**$^{\clubsuit\ddagger\diamond+}$	**78.2**$^{\clubsuit\ddagger\diamond+}$	**79.8**$^{\clubsuit\diamond+}$	**81.1**$^{\ddagger+}$	**80.4**
Upper-bound	80.2	85.8	77.9	81.5	83.4	81.8

Table 5: Cross-domain MNLI results (Macro-F1). The statistical significance of Hyper-PADA vs. the best baseline from each group (with the Bootstrap test, $p < 0.05$) is denoted similarly to Table 4.

feed the linear classifier (CLS) with the average of the encoder token representations.

Generative models are trained for 3 epochs and discriminative models for 5 epochs. We use the Cross Entropy loss for all models, optimized with the ADAM optimizer (Kingma and Ba, 2015), a batch size of 16, and a learning rate of $5 * 10^-6$. We limit the number of input tokens to 128.

5 Results

Table 4 and Figure 3 present sentiment classification accuracy results for CLCD and CD transfer, respectively (12 settings each), while Table 5 reports Macro-F1 results for MNLI in 5 CD settings. We report accuracy or F1 results for each setting, as well as the average performance across settings. Finally, we report statistical significance following the guidelines at Dror et al. (2018), comparing Hyper-PADA to the best performing model in each of the three baseline groups discussed in §4: (a) domain expert models (T5-NoDA and T5-MoE); (b) domain robustness models (T5-DANN and T5-IRM) and (c) example-based adaptation (PADA). We also report whether the improvement of Hyper-PADA over the simpler HN-based models, Hyper-DN and Hyper-DRF, is significant.

Our results clearly demonstrate the superiority of Hyper-PADA and the simpler HN-based models. Specifically, Hyper-PADA outperforms all baseline models (i.e. models that do not involve hypernetwork modeling, denoted bellow as non-HN models) in 11 of 12 CLCD settings, in 8 of 12 CD sentiment settings, and in all 5 CD MNLI settings, with an average improvement of 2.7%, 4.4% and 3.4% over the best performing baseline in each of the settings, respectively. Another impressive result is the gap between Hyper-PADA and the T5-NoDA model, which does not perform adaptation: Hyper-PADA outperforms this model by 9.5%, 8.4% and 14.8% in CLCD and CD sentiment classification and CD MNLI, respectively.

Hyper-DN and Hyper-DRF are also superior to all non-HN models across settings (Hyper-DRF in 10 CLCD sentiment settings, in 7 CD sentiment settings and in 2 CD MNLI settings, as well as on average in all three tasks; Hyper-DN in 8 CLCD sentiment settings, in 8 CD sentiment settings, and in 2 CD MNLI settings, as well as on average in all three tasks). It is also interesting to note that the best performing baselines (non-HN models) are different in the three tasks: While T5-MoE (group (a) of domain expert baselines) and T5-DANN (group (b) of domain robustness baselines) are strong in CLCD sentiment classification, PADA (group (c) of example-based adaptation baselines) is the strongest baseline for CD MNLI (in CD sentiment classification the average performance of all baselines is within a 1% regime). This observation is related to another finding: Using the DRF-signature as a prompt in order to improve the example representation is more effective in CD MNLI (which is indicated both by the strong performance of PADA and the 3.1 F1 gap between Hyper-PADA and Hyper-DRF) than in CLCD and CD sentiment classification (which is indicated both by the weaker PADA performance and by the 0.6% (CLCD) and 1% (CD) accuracy gaps between Hyper-PADA and Hyper-DRF).

These findings support our modeling considerations: (1) integrating HNs into OOD generalization modeling (as the HN-based models strongly outperform the baselines); and (2) integrating DRF signature learning into the modeling framework, both as input to the HN (Hyper-DRF and Hyper-PADA) and as means of improving example representation (Hyper-PADA).

Ablation Analysis To demonstrate the impact of example-based classifier parametrization, Figure 4 plots the diversity of the example-based classifier weights as generated by Hyper-PADA vs. the improvement of Hyper-PADA over PADA in the CLCD sentiment classification settings.[6] We choose to compare these models because both of them use the self-generated signature for improved example representation, but only Hyper-PADA uses it for classifier parametrization. The relatively high correlations between the two measures is an encouraging indication, suggesting the potential importance of example-based parametrization for improved task performance.

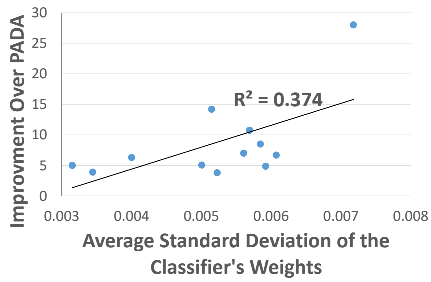


Figure 4: Correlation between the diversity of the example-based classifier weights generated by Hyper-PADA, and the improvement of this model over PADA in CLCD sentiment classification. The Spearman Correlation is 0.475. For CD sentiment classification, the corresponding numbers are 0.539 and 0.175, for Pearson and Spearman correlations respectively (not shown in the graph).

6 Discussion

We presented a Hypernetwork-based framework for example-based domain adaptation, designed for multi-source adaptation to unseen domains. Our framework provides several novelties: (a) the application of hypernetworks to domain adaptation in NLP; (b) the application of hypernetworks in example-based manner (which is novel at least in NLP, to the best of our knowledge); (c) the generation of example-based classifier weights, based on a learned signature which embeds the input example in the semantic space spanned by the source domains; and (d) the integration of all the above with an example representation mechanism that is based on the learned signature. While the idea of DRF signatures and their use for example representation in example-based adaptation is borrowed from Ben-David et al. (2021), the above novelties are unique contributions of this work. Our extensive experiments, with 2 tasks, 4 languages and 8 domains, for a total of 29 adaptation settings, clearly demonstrate the superiority of our framework over a range of previous approaches, and the positive impact of each of our modeling decisions.

In future work we would like to apply our framework to additional tasks. including sequence tagging and generation tasks, Ultimately, our goal is to shape our methodology to the level that NLP technology becomes available to as many textual domains as possible, with minimum data annotation and collection efforts.

[6]For diversity we compute the standard deviation of each classifier weight coordinate, and average the resulting values.

References

Martín Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2019. Invariant risk minimization. *CoRR*, abs/1907.02893.

Eyal Ben-David, Nadav Oved, and Roi Reichart. 2021. PADA: A prompt-based autoregressive approach for adaptation to unseen domains. *CoRR*, abs/2102.12206.

Eyal Ben-David, Carmel Rabinovitz, and Roi Reichart. 2020. PERL: pivot-based domain adaptation for pre-trained deep contextualized embedding models. *Trans. Assoc. Comput. Linguistics*, 8:504–521.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Mach. Learn.*, 79(1-2):151–175.

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, 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.

Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. 2011. Relation adaptation: Learning to extract novel relations with minimum supervision. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 2205–2210. IJCAI/AAAI.

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. *CoRR*, abs/2005.14165.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 7057–7067.

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.

Lior Deutsch, Erik Nijkamp, and Yu Yang. 2019. A generative model for sampling high-performance and diverse weights for neural networks. *CoRR*, abs/1905.02898.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1383–1392. 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.

L. Gillick and Stephen J. Cox. 1989. Some statistical issues in the comparison of speech recognition algorithms. In *IEEE International Conference on Acoustics, Speech, and Signal Pro-*

cessing, ICASSP '89, Glasgow, Scotland, May 23-26, 1989, pages 532–535. IEEE.

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 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*, pages 4694–4703.

David Ha, Andrew M. Dai, and Quoc V. Le. 2017. Hypernetworks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

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.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Weihua Hu, Gang Niu, Issei Sato, and Masashi Sugiyama. 2018. Does distributionally robust supervised learning give robust classifiers? In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2034–2042. PMLR.

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*.

Benjamin Klein, Lior Wolf, and Yehuda Afek. 2015. A dynamic convolutional layer for short rangeweather prediction. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 4840–4848. IEEE Computer Society.

Sylwester Klocek, Lukasz Maziarka, Maciej Wolczyk, Jacek Tabor, Jakub Nowak, and Marek Smieja. 2019. Hypernetwork functional image representation. In *Artificial Neural Networks and Machine Learning - ICANN 2019 - 28th International Conference on Artificial Neural Networks, Munich, Germany, September 17-19, 2019, Proceedings - Workshop and Special Sessions*, volume 11731 of *Lecture Notes in Computer Science*, pages 496–510. Springer.

David Krueger, Chin-Wei Huang, Riashat Islam, Ryan Turner, Alexandre Lacoste, and Aaron C. Courville. 2017. Bayesian hypernetworks. *CoRR*, abs/1710.04759.

Entony Lekhtman, Yftah Ziser, and Roi Reichart. 2021. Dilbert: Customized pre-training for domain adaptation with category shift, with an application to aspect extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 219–230.

Ying Li, Meishan Zhang, Zhenghua Li, Min Zhang, Zhefeng Wang, Baoxing Huai, and Nicholas Jing Yuan. 2021. Apgn: Adversarial and parameter generation networks for multi-source cross-domain dependency parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1724–1733.

Zechun Liu, Haoyuan Mu, Xiangyu Zhang, Zichao Guo, Xin Yang, Kwang-Ting Cheng, and Jian Sun. 2019. Metapruning: Meta learning for automatic neural network channel pruning. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 3295–3304. IEEE.

Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. 2021. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 565–576. Association for Computational Linguistics.

Elliot Meyerson and Risto Miikkulainen. 2019. Modular universal reparameterization: Deep multi-task learning across diverse domains. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 7901–7912.

Eliya Nachmani and Lior Wolf. 2019. Hypergraph-network decoders for block codes. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 2326–2336.

Yonatan Oren, Shiori Sagawa, Tatsunori B. Hashimoto, and Percy Liang. 2019. Distributionally robust language modeling. 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 4226–4236. Association for Computational Linguistics.

Johannes von Oswald, Christian Henning, João Sacramento, and Benjamin F. Grewe. 2020. Continual learning with hypernetworks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Nick Pawlowski, Martin Rajchl, and Ben Glocker. 2017. Implicit weight uncertainty in neural networks. *CoRR*, abs/1711.01297.

Emmanouil Antonios Platanios, Mrinmaya Sachan, Graham Neubig, and Tom M. Mitchell. 2018. Contextual parameter generation for universal neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 425–435. Association for Computational Linguistics.

Peter Prettenhofer and Benno Stein. 2010. Cross-language text classification using structural correspondence learning. In *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*, pages 1118–1127. The Association for Computer Linguistics.

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. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Alan Ramponi and Barbara Plank. 2020. Neural unsupervised domain adaptation in NLP - A survey. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 6838–6855. International Committee on Computational Linguistics.

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.

Gernot Riegler, Samuel Schulter, Matthias Rüther, and Horst Bischof. 2015. Conditioned regression models for non-blind single image super-resolution. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, San-*

tiago, Chile, December 7-13, 2015, pages 522–530. IEEE Computer Society.

Guy Rotman and Roi Reichart. 2019. Deep contextualized self-training for low resource dependency parsing. *Transactions of the Association for Computational Linguistics*, 7:695–713.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. 2020. Distributionally robust neural networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Joan Serrà, Santiago Pascual, and Carlos Segura. 2019. Blow: a single-scale hyperconditioned flow for non-parallel raw-audio voice conversion. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 6790–6800.

Aviv Shamsian, Aviv Navon, Ethan Fetaya, and Gal Chechik. 2021. Personalized federated learning using hypernetworks. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 9489–9502. PMLR.

Falong Shen, Shuicheng Yan, and Gang Zeng. 2018. Neural style transfer via meta networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8061–8069.

Joseph Suarez. 2017. Language modeling with recurrent highway hypernetworks. 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 3267–3276.

Ivan Titov. 2011. Domain adaptation by constraining inter-domain variability of latent feature representation. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 62–71. The Association for Computer Linguistics.

Kenya Ukai, Takashi Matsubara, and Kuniaki Uehara. 2018. Hypernetwork-based implicit posterior estimation and model averaging of CNN. In *Proceedings of The 10th Asian Conference on Machine Learning, ACML 2018, Beijing, China, November 14-16, 2018*, volume 95 of *Proceedings of Machine Learning Research*, pages 176–191. PMLR.

Ahmet Üstün, Arianna Bisazza, Gosse Bouma, and Gertjan van Noord. 2020. Udapter: Language adaptation for truly universal dependency parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 2302–2315. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, 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.

Yoav Wald, Amir Feder, Daniel Greenfeld, and Uri Shalit. 2021. On calibration and out-of-domain generalization. In *Thirty-Fifth Conference on Neural Information Processing Systems*.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi 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 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.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 483–498. Association for Computational Linguistics.

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. Deep pivot-based modeling for cross-language cross-domain transfer with minimal guidance. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 238–249. Association for Computational Linguistics.

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.

Yftah Ziser and Roi Reichart. 2019. Task refinement learning for improved accuracy and stability of unsupervised domain adaptation. In *proceedings of the 57th annual meeting of the Association for Computational Linguistics*, pages 5895–5906.