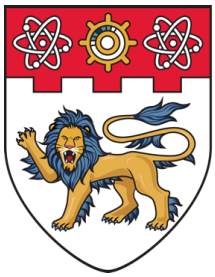


NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

DATA-EFFICIENT DOMAIN ADAPTATION FOR PRETRAINED LANGUAGE MODELS

XU GUO

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

2023

DATA-EFFICIENT DOMAIN ADAPTATION FOR PRETRAINED LANGUAGE MODELS

XU GUO

School of Computer Science and Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2023

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

December 15, 2022
.....
Date

Xu Guo
.....
XU GUO

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

December 15, 2022

Date

HAN YU

Authorship Attribution Statement

This thesis contains material from 3 papers published in the following peer-reviewed conferences or journals as well as 1 paper pending to be submitted to a peer-reviewed journal in which I was the first author.

Chapter 2 is preprinted as Xu Guo and Han Yu. Domain Adaptation and Generalization of Large-scale Pretrained Language Models: A Survey. (2022). arXiv Preprint:2210.02952. The contributions of the authors are as follows:

- Prof Han Yu and I discussed the initial research direction.
- I reviewed the related literature.
- I prepared the manuscript draft, which was revised by Prof Han Yu.

Chapter 3 is published as Xu Guo, Boyang Li and Han Yu. Improving the Sample Efficiency of Prompt Tuning with Domain Adaptation. In Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi (To appear). arXiv Preprint:2210.02952.

The contributions of the co-authors are as follows:

- Prof Boyang Li and Prof Han Yu provided the initial research direction.
- I proposed the solution, which was improved under discussions with Prof Boyang Li and Prof Han Yu.
- I prepared the manuscript draft, which was revised by Prof Boyang Li and Prof Han Yu.
- Prof Boyang Li and I designed the experiments and studies. I prepared the data, wrote the code and analyzed the experimental results.

Chapter 4 is published as Xu Guo, Boyang Li, Han Yu, and Chunyan Miao. 2021. Latent-Optimized Adversarial Neural Transfer for Sarcasm Detection. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 5394–5407, Online. Association for Computational Linguistics.

The contributions of the co-authors are as follows:

- Prof Boyang Li and Prof Han Yu provided the initial research direction.
- I proposed the solution, which was improved under discussions with Prof Boyang Li and Prof Han Yu. Prof Chunyan Miao provided insightful comments.
- I prepared the manuscript draft, which was revised by Prof Boyang Li and Prof Han Yu.

- Prof Boyang Li and I designed the experiments and studies. I prepared the data, wrote the code and analyzed the experimental results.

Chapter 5 is published as Xu Guo, Han Yu, Boyang Li, Hao Wang, Pengwei Xing, Siwei Feng, Zaiqing Nie, and Chunyan Miao. 2022. Federated Learning for Personalized Humor Recognition. ACM Trans. Intell. Syst. Technol. 13, 4, Article 68 (August 2022), 18 pages. https://doi.org/10.1145/3511710.

The contributions of the co-authors are as follows:

- Prof Han Yu and I discussed the initial research direction.
- Prof Han Yu proposed the initial solution, which was improved under discussions with me and Hao Wang.
- I prepared the manuscript draft, which was revised by Prof Boyang Li and Prof Han Yu. Siwei Feng, Zaiqing Nie and Chunyan Miao provided useful comments.
- Prof Han Yu and I designed the experiments and studies. I prepared the data, wrote the code and analyzed the experimental results. Pengwei Xing contributed to the preliminary experiments.

December 15, 2022

Date

Xu Guo

XU GUO

Acknowledgements

First and foremost, my greatest thanks go to my supervisor, Professor Han Yu, for his comprehensive and prompt guidance. Professor Yu always encourages me to explore my own research interests, which helped me develop an independent thinking habit. His support and encouragement have made my Ph.D. journey full of warmth and joy.

I wish to thank Professor Albert Boyang Li for constantly providing me with research advice. I appreciate his patience in teaching me to build up critical thinking and presentation skills. Thanks to him for helping me grow up and develop research expertise.

My sincere thanks also go to Professor Chunyan Miao for her help with both my study and life during my Ph.D., Professor Cyril Leung for his research advice, and Dr. Zhiqi Shen for his guidance on my teaching assistantship. Thanks to my thesis committee members, Professor Shijian Lu, Luu Anh Tuan, and Hong Xu for their valuable comments. Special thanks to Professor Yiqiang Chen and all the great group members with whom I received my very first research experience at ICT in Beijing.

My Ph.D. study is fully supported by NTU research scholarship, to which I would like to express my deep gratitude. During my Ph.D., I joined many great research projects. Thanks to Di Wang for his guidance in AISG projects, Qiong Wu and Yong Liu for their guidance in Alibaba projects, as well as collaborators from TTSH and Alibaba. I also would like to thank the recommender system group at Lazada with whom I translated my research work to make real-world impacts. Thanks to all my lovely friends both inside and outside NTU who have helped me in many other important ways.

I am full of gratitude to my dear parents and boyfriend for their unconditional love and support throughout the years. They have witnessed my growth and are always with me through all the happy and difficult times. This thesis is dedicated to them.

XU GUO, December 2022

“We can only see a short distance ahead, but we can see plenty there that needs to be done.”

—Turing

To my dear family

Abstract

Recent advances in Natural Language Processing (NLP) are built on a range of large-scale pretrained language models (PLMs), which are based on deep transformer neural networks. These PLMs simultaneously learn contextualized word representations and language modeling by training the entire model on massive unlabeled corpora using self-supervised learning techniques, bringing about a paradigm shift that moves our focus from customizing different models for different tasks to adapting one PLM to all tasks.

Studying how to adapt a general-purpose PLM to a specific domain of interest is of great significance to the deployment of PLMs. The mainstream practice is to finetune a PLM with a task-specific head on a labeled dataset from the target domain. However, for most target applications, labeled data is limited and even scarce in many low-resource scenarios. The huge number of parameters in a PLM often leaves those small datasets struggling to harness the power of the language priors. As a result, even under the same task, when a PLM finetuned on one dataset is applied to another dataset with some domain gap, it sometimes encounters performance degradation due to overfitting the previous training set. This phenomenon hinders the wide adoption of PLMs in practice, particularly in the face of new domains, calling for approaches to enhance the generalization performance of PLMs during adaptation without requesting more labeled data.

Early domain adaptation methods, which leverage similar source domains to boost model performance on the target domains, are developed based on customized models using traditional neural networks such as LSTMs. These models are shallow, require longer training time to converge, and have no prior knowledge compared to PLMs. Studies show that some popular domain adaptation methods can even harm the generalization performance of PLMs on the target domains. The unique characteristics of PLMs such as unprecedented scales, rich language priors, and many hitherto underexplored skills could be uncontrollable factors that make them exhibit different learning behaviors compared to traditional models. To this end, there is a need to develop algorithms for PLMs to enhance their domain adaptation performance, thereby accelerating their wide adoption in real-world scenarios.

This thesis aims to explore techniques that can efficiently make use of the target domain labeled data and better adapt a given PLM to the target domains of interest by effectively transferring knowledge from similar source domains to the target domains. To achieve this goal, I conduct research from three perspectives throughout a machine learning pipeline, each assuming only specified locations can be updated with available computing resources. That is, we keep all other conditions fixed and only make updates to the input data, model representations, and output predictions respectively. We show how to achieve better generalization performance with limited labeled data from the target domains under each scenario. To sum up, we propose a new algorithm to generate adversarial perturbations using the domain adaptation objective to enhance the transferability of soft prompt tuning in low-resource scenarios, a new model optimization algorithm that takes into account the next-step gradients of adversarial domain discriminator when optimizing the task classifiers to accommodate competing losses and a new federated learning framework that calibrates the conditional probability distribution to adapt the same PLM to multiple domains under different label distributions. We present the specific problems, related works, detailed methods, extensive experiments, and thorough discussions in the following chapters, and shed light on how to base on traditional machine learning methods while catering to newly emerging learning paradigms.

Contents

List of Figures

List of Tables

Symbols and Acronyms

Symbols

$\mathcal{R}^n$	the n-dimensional Euclidean space
$\|\cdot\|$	the 2-norm of a vector or matrix in Euclidean space
$\|\cdot\|_G$	the induced norm of a vector in G-space
$\|\cdot\|_E$	the induced norm of a vector or matrix in probabilistic space
$\odot$	the Hadamard (component-wise) product
$\otimes$	the Kronecker product
$\langle \cdot, \cdot \rangle$	the inner product of two vectors
$\circ$	the composition of functions
∇f	the gradient vector
$\mathcal{C}^k$	the function with continuous partial derivatives up to k orders
$\mathbf{1}$	all-ones column vector with proper dimension
$O(\cdot)$	order of magnitude or ergodic convergence rate (running average)
$o(\cdot)$	non-ergodic convergence rate

Acronyms

NLP	Natural Language Processing
LM	Language Model
PLM	Pretrained Language Model
RNN	Reccurent Neural Network
LSTM	Long Short-Term Memory
BERT	Bidirectional Encoder Representations from Transformers
T5	Text-To-Text Transfer Transformer
FT	Fine Tuning
PT	Prompt Tuning
OPTIMA	bOosting Prompt Tuning wIth doMain Adaptation
DA	Domain Adaptation
DG	Domain Generalization
SDA	Supervised Domain Adaptation
UDA	Unsupervised Domain Adaptation
DANN	Domain Adversarial Neural Newtwork
ANT	Adversarial Neural Transfer
LOANT	Latent-Optimized Adversarial Neural Transfer
AT	Adversarial Training
VAT	Virtual Adversarial Training
MTL	Multitask Learning
MAML	Model-Agnostic Meta Learning
FL	Federated Learning
PFL	Personalized Federated Learning
i.i.d.	independent and identically distributed
$s.t.$	subject to

Chapter 1

Introduction

1.1 Motivation

Recent advances in Natural Language Processing (NLP) are built on a range of large-scale Pre-trained Language Models (PLMs), such as GPT [Radford et al., 2018a, 2019], BERT [Devlin et al., 2019], ALBERT [Lan et al., 2020], RoBERTa [Liu et al., 2020], BART [Lewis et al., 2020], and T5 [Raffel et al., 2020a]. They play a central role as the *foundation model* of AI [Bommasani et al., 2021] for their knowledgeable yet incomplete character. Having Transformer [Vaswani et al., 2017a] as their basic neural architecture, these PLMs often have millions or billions of parameters that are capable of learning sophisticated language skills from widely available huge corpora using self-supervised learning techniques, such as masked language modeling [Mao and Liu, 2019]. They are often deemed to have rich language priors for their superior language understanding performance and the capability to solve downstream tasks given relatively fewer labeled training data than traditional models. PLMs simultaneously learn contextualized word representations and language modeling by training the model in an end-to-end manner on the massive unlabeled corpora, bringing about a paradigm shift that moves our focus from customizing different models for different tasks to adapting one PLM for all tasks.

The necessity of domain adaptation for PLMs. Adapting a general-purpose PLM to the target domain of interest is crucial for real-world deployment and is therefore at the heart of current NLP research. However, PLMs are trained on the universal language corpora, leaving domain challenges unresolved. Despite that the recently growing scale of PLMs allows them to demonstrate better generalization performance on zero-shot

and few-shot settings, their application scope is still limited. First, they may fail to work in unseen domains, and we have to adapt them. This is not surprising given that some domains are quite confidential such as healthcare domains which were never seen during pretraining. However, most of the NLP benchmarks do not diverge a lot from the pretraining data and therefore we need more challenging benchmarks. The conventional finetuning paradigm can quickly adapt a PLM to a new domain by finetuning a given PLM with a task-specific head on a labeled task-specific dataset. It has been a de-facto standard in NLP since 2018. However, finetuning the entire model requires more GPUs and storage, and the training can be easily trapped in overfitting. Instead, we can tune a specific part of the PLM to adapt the model as what I have studied in Chapter 3. Second, those very huge models may not always be affordable for users, and we need to have alternative choices. Models such as PaLM [Chowdhery et al., 2022] and GPT-3.5 can have hundreds of billions of parameters. They offer APIs for which users are charged a fee because running such huge models consumes a lot of computing resources. On the other hand, we can choose a relatively smaller but still powerful model as a starting point and finetune them on our domain-specific data as what I have studied in Chapters 4 and 5. However, no matter which setting we adopted, PLMs still require sufficient labeled data from the target domains to make effective adaptations.

Challenges for domain adaptation with PLMs. Despite the growing volumes of online data, labeled data is still limited for most target applications and even scarce in many low-resource scenarios. Given the huge number of parameters in a PLM, small-scale supervised datasets often fail to exploit the language priors of the PLM effectively. As a result, despite working on the same task, when a PLM finetuned on one domain-specific dataset is applied to another dataset with some domain gap, it can sometimes incur performance degradation due to overfitting the previous small training set. A bunch of evidence has surfaced showing that their performance can degrade when they are applied to a narrower domain where data varies substantially from the pretraining corpus [Thompson et al., 2019, Araci, 2019, Chalkidis et al., 2020, Miller et al., 2021]. The mismatch between the pretraining and adaptation data distributions makes PLMs struggle to be widely adopted in practice. To solve this issue, finding a related source domain where labeled data are abundant and leveraging domain adaptation is an alternative.

Limitations of existing domain adaptation methods for PLM. A large number of early domain adaptation studies [Pan and Yang, 2009, Weiss et al., 2016, Ramponi and Plank, 2020] have shown that using similar source-domain datasets can boost model

performance on the target domains. This is especially meaningful for those data-hungry tasks such as abstractive summarization [Yu et al., 2021a]. However, traditional domain adaptation methods are developed based on shallow neural networks without pretraining, which can be unfavorable to PLMs [Ryu et al., 2022] due to their very large scale, rich language priors, and many hitherto underexplored skills [Radford et al., 2019]. These sophisticated characteristics make PLMs exhibit different learning behavior and generalization abilities from traditional models. For example, Wright and Augenstein [2020] and Karouzos et al. [2021] find that domain adversarial training on top of BERT is unstable and has little effect on cross-domain performance. Nevertheless, directly fitting a single PLM on non-identical domains is suboptimal [Bilen and Vedaldi, 2017] and may even incur negative transfer due to the domain shifts [Lekhtman et al., 2021]. Our studies [Guo et al., 2021a] corroborate these findings, and we further reveal that this can be attributed to the loss competition problem between adversarial and task-specific losses in the shared feature space, as well as the dominance of large source domain data. Moreover, performance gains on the target domain can come at the expense of general-domain performance due to the semantic gap between the embedding spaces of different domains [Wang et al., 2021a], which is recognized as the catastrophic forgetting problem in continual learning [McCloskey and Cohen, 1989]. All of these pioneering studies and findings suggest that there is a need to develop algorithms that can enhance the domain adaptation capability of PLMs, thereby accelerating their wide adoption in real-world tasks.

In this thesis, I will explore *how to enhance the domain adaptation capability of PLMs without requesting more labeled data from the target domain*. I will conduct the research by making ablations on a standard machine learning pipeline. In particular, given a PLM, a small amount of target-domain data, and a potential source-domain dataset, I will explore three scenarios from the input to the output of a model:

- With the model parameters and target labels fixed, how to adapt a PLM to the target domain by only updating the input data?
- With the input data and the target labels fixed, how to adapt a PLM to the target domain by optimizing the model representations?
- With the input data and the model fixed while the target label distributions vary across different users, how to adapt the same PLM to multiple users?

1.2 Contributions

This thesis provides new learning techniques that can achieve data-efficient domain adaptation which brings us better target-domain performance with the same amount of labeled data or the same performance with far fewer labeled data compared with existing possible solutions. Our technical contributions can be summarized as follows:

- *A new soft prompt tuning technique that produces transferable soft prompts with adversarial training by generating adversarial perturbations with a domain adaptation objective (S3)*: This is the first technique that generates domain-adaptive perturbations to enhance the transferability of soft prompts, which are prepended to the input text embeddings without updating the PLM. Specifically, to solve the problem that soft prompt tuning heavily relies on a large labeled training set and enables its applicability to few-shot and even zero-shot settings, we propose bOosting Prompt TunIng with doMain Adaptation (OPTIMA). It is the first domain adaptation technique for soft prompt tuning, which does not require any labeled data from the target domain. Empirical results show that using unlabeled target-domain data boost performance significantly. At the heart of OPTIMA is a targeted regularization technique that encourages smooth decision boundaries only in the areas where the distributions of two domains are similar. Through empirical evaluation, we show that OPTIMA outperforms state-of-the-art baselines, improves data efficiency significantly, and effectively addresses domain shifts.

- *A new model optimization technique that boosts learning performance on small-scale domains under domain adversarial training by considering the gradients of the opponent's next step via look-ahead learning (S5)*: This is the first technique that accommodates competing losses by enforcing task classifiers to look at the future gradients from the adversarial discriminator. Inspired by the existence of multiple small sarcasm datasets, we propose to use transfer learning to bridge dataset differences. We find that training a shared-private neural network on top of PLMs under domain adversarial training on a larger-scale source-domain dataset and a smaller-scale target-domain dataset simultaneously can enforce the model to be dominated by the source domain data. We propose a Latent-Optimized Adversarial Neural Transfer (LOANT) model for cross-domain sarcasm detection. By conducting stochastic gradient descent (SGD) with one-step look-ahead, LOANT

outperforms traditional adversarial neural transfer, multi-task learning, and meta-learning baselines, and establishes a new state-of-the-art F-score of 46.41%. It is the first study of transfer learning between different sarcasm detection datasets.

- *A new model personalization training strategy that enables adapting the same PLM to different domains with each having a different label distribution (S7)*: This is the first federated learning framework that adapts a federated PLM to different domains to achieve personalized humor recognition without making multiple copies of the PLM and updating each of them separately. We relaxed the common assumption in the humor recognition literature that users have a consensus about whether or not a given text is humorous. We proposed an FL-based solution with a personalized adaptation strategy to enable personalized humor recognition with good generalization, while not exposing private humor preference data. We conducted extensive experiments to evaluate our FedHumor model. Results show that our approach is significantly superior to existing humor recognition methods and alternative training strategies in terms of personalized humor recognition.

1.3 Outline of the Thesis

Chapter 1 introduces the background of the research topic and particularly the motivations of *developing enable data-efficient domain adaptation techniques for PLMs*.

Chapter 2 presents an extensive literature review under a proposed taxonomy. We categorize the related works in Table 2.1 and explain each research direction accordingly.

Chapter 3 introduces the technique - bOosting Prompt TunIng with doMain Adaptation (OPTIMA), which is the first technique for enabling soft prompt tuning in low-resource settings with domain adaptation. We show that soft prompts can be trained to encourage PLMs to produce smooth decision boundaries against domain gaps thereby enhancing the portability of soft prompts in the face of new domains.

Chapter 4 introduces the technique - Latent-Optimized Adversarial Neural Transfer (LOANT) - for enhancing the transferability of PLMs from a data-abundant domain to a data-scare domain, demonstrated with the sarcasm detection task. We reveal a loss competition problem in the shared feature space and show how LOANT accommodates these competing losses.

Chapter 5 introduces a federated learning framework - FedHumor - for personalized humor recognition where different people have different label distributions which is kept private. We show how to adapt the same PLM to different people.

Chapter 6 summarizes the thesis and discusses some future research directions.

Chapter 2

Literature Review

This chapter includes two parts, one focuses on introducing the basics and history of pretrained language models (Section 2.1), and the other part focuses on reviewing and discussing related works on adapting large-scale PLMs to target domains (Section 2.2).

2.1 The History of Language Models

Why do we need a (pretrained) language model?

Unlike digital images and signals, languages are composed of discrete text symbols and need to be transformed into continuous representations for computers to describe their meanings. The challenge in this process is that human languages can involve an infinite variety of linguistic sentences and the variety continues to expand. It is impossible for a computer program to calculate over an infinite space.

To avoid the dilemma, a language model is developed to assign probabilities to sequences of words and pick up the *most likely expression* as its estimation. A unique continuous representation is assigned to a basic *word token* and the combinatorial representation for a sentence is usually different from another. The language model is built on text corpora, which could be documents, books, or web pages. The set of all the unique basic tokens is termed as *vocabulary*, which is shipped along with the language model. The corpora contain rich information about how frequently a word occurs, the context around each word, and more. Thus, how to exploit the potential of the corpora is tricky.

Training a language model from scratch is resource-intensive. The rule of thumb is to pretrain a strong and robust language model that can generalize to a range of NLP problems. The first of this kind is the n-gram model, where n-gram probabilities are stored. It highly depends on how representative the training corpora are. Therefore, the trend at that time is to open-source a range of text corpora of either general or domain-specific genres. We introduce this kind of LMs in Section 2.1.1. Then comes the pre-trained word embeddings, where the word representations are stored and their probabilities are described by a shallow neural network. They allow word-word relationships to be quantitatively compared in the latent space. We introduce the details in Section 2.1.2. The last milestone of PLMs is transformer-based large-scale language models, which encapsulate both the learning of word representations and language modeling. PLMs in the current literature generally refer to this kind of LMs. They have revolutionized the practice in the NLP community with their strikingly superior performance dominating a variety of benchmarks. We introduce them in Section 2.1.3.

2.1.1 Statistical Language Models

The goal of statistical language modeling is to learn the joint probability distribution for a sequence of words. Statistical language models represent a word with its probability distribution over the whole vocabulary, which is approximated by its frequency of occurrence in the corpus. Therefore, a sentence can be represented as a sequence of probability values. n-gram model is a typical statistical language model. The term can trace back as early as 1948 in Shannon's paper [Shannon, 2001] where he used this term to describe the joint probability of n consecutive letters in communication. The first n-gram language model appear as early as in 1976 by Jelinek [1976] where they applied n-gram to speech recognition. Since then until late 1990s, a range of techniques were proposed to make n-gram models more useful for different problems.

Unigram Model holds the assumption that a word occurs independently from another, e.g.:

$$P(\text{a lovely dog}) = P(\text{a}) \cdot P(\text{lovely}) \cdot P(\text{dog}) \tag{2.1}$$

The following example illustrates how a unigram model represents a sentence:

a	lovely	dog
0.2	0.05	0.1

N-gram Model holds the Markov assumption that the probability of a word to occur depends on its preceding $N-1$ words (a.k.a., a Markov chain), e.g., a bigram ($N=2$) approximates the probability of a word, w_m, comes after the sequence $w_0, \ldots, w_{m-1}$ as:

$$P(w_m|w_{0:m-1}) \approx P(w_m|w_{m-1}) \tag{2.2}$$

Therefore, the joint probability of a sequence of $m+1$ words is decomposed as the product of all the bigram conditional probabilities:

$$P(w_{0:m}) \approx P(w_0) \cdot P(w_1|w_0) \cdot \ldots \cdot P(w_m|w_{m-1}) = \prod_{i=1}^{m} P(w_i|w_{i-1}) \tag{2.3}$$

An intuitive approach to estimate each of the bigram probabilities is using their relative occurrence frequencies:

$$P(w_i|w_{i-1}) = \frac{P(w_{i-1:i})}{P(w_{i-1})} \approx \frac{C(w_{i-1:i})}{C(w_{i-1})} \tag{2.4}$$

Here is a concrete example of how to use such a language model in practice. Suppose we are translating a part of the speech word by word and predicting whether the word comes after *a lovely* is *dog* or *door*. We then calculate the frequency of the bigrams, *lovely dog* and *lovely door*, from our corpus and obtain $C(\textit{a lovely dog}) = 3$ while $C(\textit{a lovely door}) = 0$. Then the language model will choose *dog* rather than *door* as the candidate. The insight behind forcing language models to maximize the joint probability of a sequence of words is that we are more likely to find sentences that match the habitual usage of languages.

Evaluation. To compare an n-gram language model with another, we need an unseen test corpus separate from the training corpus. Whichever model predicts significantly higher joint probabilities for the sentences in the test corpus is of higher quality. In practice, we don't directly use the joint probability as the performance metric but normalize them over the entire vocabulary, which is named *perplexity*. Suppose the vocabulary for a language model contains V words, then the perplexity on a sentence, $w_0, \ldots, w_m$, is computed as follows:

$$PP(w_{0:m}) = P(w_{0:m})^{-\frac{1}{V}} \tag{2.5}$$

Thus, the lower the perplexity, the better the language model. Comparing perplexity is equivalent to comparing *entropy*, which computes the log of the perplexity:

$$E(w_{0:m}) = -\frac{1}{V} \cdot \log P(w_{0:m}) = -\frac{1}{V} \sum_{i=1}^{m} \log P(w_i|w_{i-1}) \tag{2.6}$$

Entropy measures how much information is carried if the sequence $w_{0:m}$ has a probability of $P(w_{0:m})$. Higher probability means lower entropy, meaning that the model is very confident about predicting the sequence to be $w_{0:m}$ and that learning on the fact contributes no new information. Training a language model to minimize the perplexity or the entropy will engage the model to put more effort on learning to predict hard sequences.

Impact of Corpus. Obviously, the quality of a statistical language model depends on its training corpus. The language model can be a biased one if the corpus contains documents of specific interests, e.g., a collection of sports news. It will have a tendency to generate text of a specific genre. Therefore, we need to find a training corpus that has a similar genre to our specific task. In addition, the larger the corpus, the closer the approximated probability will be to its true probability distribution. Unlike today's trends on releasing pretrained weights of large-scale language models, the trend in those days was releasing high-quality corpus. For example, the Web 1 Trillion 5-gram Corpus [Franz and Brants, 2006] from Google covers 5-gram sequences that appear in over 40 books.

Deployment. Storage is the main consideration in applying n-gram language models because retrieving a large set of n-grams can incur a long computation time. In those days, the computation facility is not adequate. Only frequent n-grams are kept in storage [Gao and Lee, 2000]. The language model needs to store each n-gram and its probability in the database. A range of retrieval techniques were explored for efficient use of such language models such as Bloom Filter based on hashing [Talbot and Osborne, 2007]. It is interesting to compare with today's fashion of efficient methods for adapting large-scale pretrained language models in practice.

Smoothing. The probability of a test sentence can be zero when any of the n-grams from the sentence never occur in the training corpus. This will be problematic when comparing the perplexities of two language models. To circumvent this problem, smoothing techniques are often adopted when computing the probabilities. For example, Laplace

smoothing, a.k.a. additive smoothing, add 1 to all the n-gram counts. A comparison between different smoothing techniques can be found in an early empirical study in 1998 on how they affect the performance of n-gram language models for speech recognition [Chen and Goodman, 1999].

Out of Vocabulary (OOV) Problem. To enable our language model to adapt to unseen domains, we need an open vocabulary that can incorporate unknown words, or OOV words, instead of a closed one. A common practice is to add a pseudo word, denoted as $<$ UNK¿, to the vocabulary and treat the unknown words as $<$ UNK $>$. To know the joint probabilities of n-grams that contain $<$ UNK $>$, we need our training corpus to have those n-grams. To achieve this, we replace words that occur at the tail of the distribution over the entire vocabulary as $<$ UNK $>$. The assumption behind this practice is that given a large enough training corpus, words that are unknown to the vocabulary are more likely to be rare words.

Limitations. First, building a more powerful language model requires a larger training corpus for modeling more n-grams, and eventually leads to a curse of dimensionality. Second, they are difficult to adapt to new domains. The model itself heavily relies on statistical knowledge in the training corpora. For domains that cannot provide large enough training corpora, statistical models cannot perform well. Moreover, there are many cases when scaling up the training corpora does not help much. Real-word test corpora often contain words that lay outside of our vocabulary, it is difficult for $n$grams to generalize to a variety of test domains. Traditional adaptation approaches were to concatenate very short n-grams seen in the training corpus, which can be vulnerable to context drift.

2.1.2 Neural Language Models

The resurgence of deep learning has advanced NLP by using neural networks to learn more complicated tasks from larger datasets where previous simple models barely surpass. The very first neural language model comes in 2000 with Bengio et al. [2000, 2003] who propose to use neural networks for language modeling. Different from n-gram language models in which a word w_i in the vocabulary is associated with a n-gram conditional probability, $p(w_i|w_{i-n+1:i-1})$, they associate each word to a d-dimensional vector, $z_i \in \mathbb{R}^d$. The idea came from the concept of "distributed representation" [Hinton, 1984] and is commonly known as *word embeddings* today. In their experiments, they

found that a feed-forward neural network works better than Recurrent Neural Networks (RNNs). With a neural network f parameterized by θ, we can estimate the conditional probability of a word w_i to occur in a context of n-grams by mapping their corresponding high-dimensional vectors to a probability distribution over the vocabulary V:

$$h = f_\theta(z_{i-n+1}, \ldots, z_{i-1}), h \in \mathbb{R}^{|V|}, \tag{2.7}$$

where the i-th element of h represents the estimation of the probability for the word w_i to be the i-th word v_i in the vocabulary, which is obtained by applying softmax:

$$\hat{P}(w_i = v_i | w_{i-n+1}, \ldots, w_{i-1}; \theta) = \frac{e^{h_i}}{\sum_j e^{h_j}}. \tag{2.8}$$

θ and the word representations z are determined by maximizing the estimated joint probabilities on a training corpus with backpropagation (a.k.a. gradient descent) upon convergence.

The use of language models began to take place in more complicated NLP tasks beyond speech translation in which they are simply treated as priors for correcting the generated text. Instead, language modeling became one of the self-supervised learning tasks for pretraining a general-purpose model, such as the pretrained word embeddings and today's large-scale pretrained language models. Generally speaking, pretraining aims to equip a model with knowledge learned from largely available datasets and transfer that knowledge to tasks that have limited in-domain data for training.

Word2Vec. Mikolov et al. [2013a] from Google open-sourced the first large set of pretrained word vectors for a vocabulary of 3 million words, named word2vec[1]. It is trained with a 2-layer neural networks on a Google New datasets which has roughly 100 million words. word2vec breaks the isolation of words in their representations by demonstrating the synonym between words with cosine similarity:

$$\text{cosine}(z_i, z_j) = \frac{z_i \cdot z_j}{\|z_i\| \cdot \|z_j\|}. \tag{2.9}$$

This can be achieved by two *local context window* methods, namely Continuous Bags of Words (CBOW) and Skip-gram. With CBOW, a feedforward neural network is given the word vectors of a bag of n history words $w_{i-n:i-1}$ and a bag of n future words $w_{i+1:i+n}$

[1] https://github.com/tmikolov/word2vec

and is required to predict the current word w_i. This bears the similar idea as n-gram language models. With Skip-gram, the neural network receives the word vector of the current word w_i and tries to correctly predict all the n words before and after the current word by minimizing the average negative log-likelihood (recall Eq. 2.5 and Eq. (2.6)) :

$$-\log P(w_{i-n}, \ldots, w_{i+n}) = -\frac{1}{2n} \sum_{j \neq 0, j=-n}^{n} \log P(w_{i+j}|w_i). \tag{2.10}$$

Word vectors produced by CBOW and Skip-gram all show higher performance in both semantic and syntactic evaluation. Subsequent research found that Skip-gram augmented with negative sampling, which also requires the model to predict a noisy word w^O wrongly, can yield higher-quality word vectors [Mikolov et al., 2013b]:

$$-\log P(w_{i-n}, \ldots, w_{i+n}) + \frac{1}{K} \sum_{k=1}^{K} \log P(w_k^O|w_i). \tag{2.11}$$

GloVe. Pennington et al. [2014] from Standford University open-sourced another new set of pretrained word embeddings called GloVe[2]. It is trained on a combination of the Wikipedia dump and Gigaword5. GloVe makes use of word co-occurrence statistics in the whole corpus as supervision which is ignored by local context window methods. Instead of maximizing the conditional probability for every word, they formulate it as a regression problem and train the model to approximate the statistical priors.

$$P(w_i|w_k) = f_\theta(z_i^T z_k) \approx \frac{C_{ik}}{C_i}, \tag{2.12}$$

where C_{ik} represents the count for the word w_i to occur with a context word w_k and C_i represents the count for any word to occur with the word w_i. Such supervision results in a word vector space that excels word2vec by a large margin in the word analogy tasks.

Both word2vec and GloVe have been used in the research community as the commonly used pretrained word embeddings for years. However, their shallow neural architecture and simple training signals limit them from fully exploiting the potential in the corpora. Since then, more unsupervised learning techniques were designed to pretrain language models, attempting to encode rich information from large amounts of unlabeled data into word representations. These pretrained word embeddings, when integrated with supervised learning, can boost performance on a range of downstream tasks.

[2] `https://nlp.stanford.edu/projects/glove/`

Skip-Thought Vectors. Kiros et al. [2015] are probably the first to adopt an encoder-decoder neural architecture to pretrain word vectors, where the encoder encodes a sentence $w_{1:i}$ to a fixed-length vector $\boldsymbol{h}_i$ from which the decoder generates its subsequent sentence $w_{i+1:n}$. The training objective is to minimize the negative log-likelihood over the decoded sentence:

$$-\sum_{k=i+1}^{n} \log P(w_k|\boldsymbol{h}_i) \tag{2.13}$$

After pretraining, they use the pretrained encoder as a generic feature extractor for downstream tasks. Skip-Thought Vectors are often used for sentence representations and have demonstrated superior performance on text classification tasks than using word2vec.

ELMo. To capture the *polysemy* of word meanings in different linguistic contexts, Peters et al. [2018] proposed to pretrain a *deep* bidirectional LSTM language model and use the internal states of LSTM as the representatations of words, which is called ELMo (Embeddings from Language Models). For a given supervised task, we can either use the final internal state or a weighted average of the internal states from all the L LSTM layers and the input embedding layer to represent a word w_i:

$$\text{ELMo}(w_i) = \sum_{l=0}^{L} s_l \boldsymbol{h}_l, \tag{2.14}$$

where the weights can be trained together with downstream tasks and $\sum_{l=0}^{L} s_l = 1$. Deep neural architecture can capture underlying context-dependent relationships between words. There were also other similar research on learning contextualized word representations such as CoVE [McCann et al., 2017] and TagLM [Peters et al., 2017].

Contextualized word representations have demonstrated better performance than previous de-contextualized ones in more complex NLP problems such as word sense disambiguation. Their presence marks the transition from pretraining the shallow word embeddings to pretraining the deep language models.

Generalization. The key of developing language models is to generalize them to unknown domains. Neural language models demonstrate their superiority in this goal over n-gram language models. This is achieved by introducing a pretraining stage to derive high-quality word embeddings and even the whole language models to warm-start the task-specific models. Pre-trained models can be used by all downstream tasks, laying the foundation for artificial general intelligence.

2.1.3 An Emergent Trend: Large-Scale PLMs

There has been a recent push to try to further the capabilities of language models by training large-scale deep neural networks, namely transformers [Vaswani et al., 2017b], on large amounts of unlabeled corpora, which give birth to a range of PLMs for natural language understanding and generation. The transformers do not use any RNNs or LSTMs, but are purely based on feedforward neural networks and the attention mechanism. This makes it possible to train transformers in parallel, allowing deeper and wider neural networks to be trained under the same training budget. In contrast to previous practice of learning word embeddings and task-specific neural models separately, models that are built on PLMs only needs to add a simple task prediction layer on top of PLMs output, and finetune them together for a few epochs can achieve superior performance than previous state-of-the-art models.

Attention Mechanism. It was first proposed in [Bahdanau et al., 2015] to cope with the problem in which the performance of neural machine translation models deteriorates on longer input sentences, which is caused by forcing the RNN encoder to compress the entire input sentence into a vector. Specifically, for an input sentence $w_{1:m}$ encoded by a bidirectional RNN as a sequences of hidden vectors $\boldsymbol{h}_{1:m}$, they propose to average attention-weighted vectors using weights $s_{1:m}$ learned by a feedforward neural network with softmax:

$$\boldsymbol{c} = \sum_{i=1}^{m} s_i \boldsymbol{h}_i. \tag{2.15}$$

The decoder generates words as usual but conditioned on $\boldsymbol{c}$ instead of $\boldsymbol{h}_i$. As a result, the attention mechanism encourages the decoder to focus on input words that largely contribute to the translation by assigning them with higher weights.

Transformer Network. A fundamental limitation with RNNs and LSTMs lies in their inherent sequential computation nature where every hidden state $\boldsymbol{h}_i$ has to wait for the computation of the previous and future hidden states. Budget for computation time and storage stretches as the input length increases. Vaswani et al. [2017b] from Google proposed a new network architecture for sequence modeling, named Transformer, which is solely based on *feedforward neural networks with attention mechanism*. Instead of recurrently modeling the word dependency, they use attention to capture the relationships between words at arbitrary distances. For sequence modeling problems, the Transformer

utilizes two kinds of neural architectures, the encoder and decoder networks. The encoder network focuses on modeling word dependency while the decoder network focuses on language modeling to successfully generate the expected sequence.

Open-source Pretrained Language Models. Based on the architecture, current open-sourced PLMs can be divided into three categories:

- 1) Encoder-only PLMs consist of solely stacked encoders of the Transformer, such as BERT [Mao and Liu, 2019], ALBERT [Lan et al., 2020], RoBERTa [Liu et al., 2020], DPR [Karpukhin et al., 2020], and so on. They focus on a deep understanding of natural languages and therefore can capture contextualized word representations for a given piece of text.

- 2) Decoder-only PLMs consist of solely stacked decoders of the Transformer, such as GPT [Radford et al., 2018a], GPT-2 [Radford et al., 2019], CTRL [Keskar et al., 2019], CPM [Zhang et al., 2021], LLaMA [Touvron et al., 2023], OPT Zhang et al. [2022], and so on. They are good at generating high-quality natural languages. The recently popular GPT-3.5 and ChatGPT demonstrate pretty strong generation quality simply based on human-written prompts.

- 3) Encoder-decoder PLMs adopt both a stack of encoders and a stack of decoders, a seq2seq machine translation architecture, with a goal to excel in both comprehension and generation tasks. PLMs under this category include T5 [Raffel et al., 2020a], BART [Lewis et al., 2020], CodeGen Nijkamp et al. [2022], Tk-Instruct Wang et al. [2022] and so on. While they aim to encapsulate all kinds of tasks, they are not specialized for either language understanding or language generation.

Some popular PLMs, though not publicly available, provided APIs that allow users to specify input text and receive a charged response as output. For example, GPT-3 [Brown et al., 2020], InstructGPT [Ouyang et al., 2022], FLAN [Wei et al., 2021a], PaLM [Chowdhery et al., 2022], Flan-T5 Chung et al. [2022], and so on. These PLMs are trained on different large-scale general-purpose corpora. Some of them are trained using different augmentation techniques. Therefore, different PLMs are likely to exhibit different semantic understanding or generation biases towards certain text genres or domains. Depending on the number of transformer layers, each type of PLM can have different scales ranging from millions to billions of parameters. It is usually the case that the bigger the model, the better the zero-shot and few-shot performance. These PLMs

can also be further trained towards a specific text genre or domain, which is discussed in Section 2.2.2. A common practice of using PLMs is to finetune it together with a task-specific head on much smaller supervised tasks. Fintuning transformer-based PLMs generally excels finetuning LSTM-based PLMs on a broader range of tasks [Radford et al., 2018b].

2.2 Domain Adaptation for PLMs

Adapting PLMs to specific domains is crucial to the practical deployment of PLMs. In this section, we systematically examine all the existing possible solutions for adapting PLMs to target domains of interest. We revisit traditional domain adaptation (DA), robust training, and other techniques such as personalization techniques to deal with domain shift problems. We cater to the recent popular learning paradigms such as prompt learning and federated learning and aim to develop methods to enhance domain adaptation of PLMs under these trends. To better overview what solutions hold the potential to achieve this goal under different constraints, we propose a taxonomy (Table 2.1) following a standard machine learning pipeline, covering data augmentation, model optimization, and personalization techniques. The mainstream approaches focus on data augmentation and model optimization or a combination of them. Personalization has been less explored and most of the methods we surveyed are either recently emerging methods or a resurgence of traditional machine learning approaches. This category considers practical scenarios such as personalized healthcare and customized machine translator which are characterized as either small-data problems or highly imbalanced problems (i.e., some classes are scarce). We deem this category promising for the practical adoption of PLMs at scale. The following content is presented following the taxonomy. At the end of the chapter, we summarize related works by visualizing their assumptions, techniques, and PLMs in Table 2.2 to provide a big picture of the current state of research.

Theories Earlier theoretical [Ben-David et al., 2006] and experimental [Blitzer et al., 2007, Saenko et al., 2010] analysis for domain adaptation demonstrate that the test error of supervised machine learning methods generally increases in proportion to the distribution differences between the training and test sets [Ben-David et al., 2010]. A theoretical analysis for domain adaptation of language models indicates that pretraining on a larger out-of-domain dataset before fine-tuning on a small in-domain dataset can achieve better generalization than only in-domain training [Grangier and Iter, 2022]. They also show

TABLE 2.1: A taxonomy for domain adaptation and generalization of PLMs.

Category	Definition	Subcategory
Data Augmentation	Methods that expand or shrink the source domain training set, or prompt input data with prior information.	S1: Importance Weighting S2: Pseudo Labeling S3: Data Synthesis S4: Prompt Learning
Model Optimization	Methods that optimize the model parameters using different learning objectives to induce better data representations.	S5: Continual Learning S6: Adversarial Learning S7: Metric Learning
Model Personalization	Methods that adapt the same PLM to multiple different domains together where each domain distribution is relatively stable.	S8: Posterior Adaptation S9: Specification S10: Reparameterization

that larger size of pretraining sets does not necessarily bring performance gains to target domains. When their underlying distributions are similar, pretraining can benefit the target-domain tasks.

Settings and Assumptions. The rapid development of transfer learning has given birth to a number of transfer learning settings with each holding a different assumption on the given data [Pan and Yang, 2009]. We review two adaptation settings for PLMs, namely adaptation from pretraining to downstream tasks, which is usually achieved by continual learning (Section 2.2.2), and adaptation from related source domains to target domains of our interest. Regarding the feature space and the amount of labels available, recent literature commonly adopt the following assumptions:

A1. Both domains share the whole feature space. Only the source domain has labels;

A2. The source and target domains share a part of the feature space. Only the source domain has labels;

A3. Both domains share the whole feature space. Both domains have labels;

A4. The source and target domains share a part of the feature space. Both domains have labels.

We follow Ramponi and Plank [2020] and make a difference between supervised domain adaptation (SDA) and unsupervised domain adaptation (UDA) depending on whether the target domain has labels. Note that in existing literature, UDA often assumes a large amount of unlabeled data which is dense while SDA often assumes a small amount of

labeled data which is sparse. Therefore, SDA does not always present an easier setting than UDA. A few research papers adopts a semi-supervised domain adaptation setting where a small amount of target-domain data are labeled, which is classified into SDA in our thesis.

Other surveys. Comprehensive surveys for domain adaptation or pretrained language models exist, each revisits related works from a different perspective: transfer learning surveys [Pan and Yang, 2009, Weiss et al., 2016] provide a holistic view including but not limited to DA; DA for visual applications [Patel et al., 2015, Csurka, 2017, Wang and Deng, 2018]; multiple-source domain adaptation (MDA) [Mansour et al., 2008, Sun et al., 2015]; neural UDA for NLP applications based on shallow and non-pretrained language models [Ramponi and Plank, 2020]; DA and MDA for machine translation [Saunders, 2022]; taxonomy of PLMs [Qiu et al., 2020] and comprehensive guide to use PLMs for NLP tasks [Min et al., 2021] and particularly for text generation tasks [Li et al., 2021]; parameter-efficient adaptation methods for PLMs [Ding et al., 2022].

2.2.1 Data Augmentation

The section of data augmentation includes all the techniques that achieve domain adaptation on the input level. The goal is to enhance the training data with more quality information that can adapt PLMs to target domains. There are three ways to achieve this. Importance weighting methods learn to weight the source domain such that only samples that are related to the target domain will be used to update the model. Pseudo-labeling methods expand the source domain training set by pseudo-labeling the unlabeled data from the target domain and adding the pseudo-labeled data into the training set to continually train the PLM. Prompting methods enrich the input data with additional information such as task descriptions about the target domain to prompt the PLMs to perform target-domain tasks.

S1. Importance Sampling

Importance sampling methods [Owen, 2013] identify and select relevant data and try to reduce the negative impact of irrelevant data from the source domain during domain adaptation. Earlier research focus on designing a metric or criterion to measure the relevance of a source-domain instance to a target domain with language models as knowledge priors, e.g., the difference between the cross entropy of the sentences from two

domains:

$$\Delta\mathcal{H}(\mathcal{D}_s, \mathcal{D}_t) = \sum_{y\in\mathcal{D}_s} p(y)\log q(y|\theta) - \sum_{y\in\mathcal{D}_t} p(y)\log q(y|\theta), \tag{2.16}$$

where p is the empirical distribution over the domain corpus while q is the distribution predicted by the language model. The top k sentence pairs will be selected to improve model performance by minimizing the importance weighted cross entropy over the source domain dataset $\mathcal{D}_s$:

$$\mathcal{L}(\theta, \hat{w}) = -\frac{1}{|\mathcal{D}_s|}\sum_{y\in\mathcal{D}_s} \hat{w}(y; \mathcal{D}_s, \mathcal{D}_t)\log p(y|\theta), \tag{2.17}$$

where $\hat{w}$ estimates the importance weights using metrics such as $\Delta\mathcal{H}$. The quality of sampled training set from the source domain depends on the relative size of the source and target datasets and the quality of the estimators [Grangier and Iter, 2022]. They have been applied to enhance machine translation performance [Axelrod et al., 2011, Wang et al., 2018].

Dynamic data selection methods [van der Wees et al., 2017] relax the hard selection procedure by assigning the normalized scores to source-domain samples and retain all the source-domain vocabulary while lowering the importance of irrelevant data during training. Influence function [Koh and Liang, 2017] traces a model's prediction through backpropagated gradients over its training data to identify those training points that are important for making the prediction. It has been applied to the pretrained ResNet for image processing tasks [Pruthi et al., 2020], vanilla Transformers for neural machine tranlation [Wang et al., 2021b, Mohiuddin et al., 2022, Iter and Grangier, 2021] and so on. Other approaches may include training a domain classifier to select source-domain data based on the domain probability [Ma et al., 2019], which involves a multi-source setting. The lower the probability, the more similar the sample is to the target domain. They successfully enhanced domain adaptation of BERT on classification tasks. Importance weighting techniques have also been studied for partial domain adaptation (PDA) where the target-domain classes are only a subset of source-domain classes [Zhang et al., 2018, Cao et al., 2019]. Zhang et al. [2018] propose a two domain classifier strategy to identify the importance score of source samples. Cao et al. [2019] propose a progressive weighting scheme to quantify the transferability of source examples to achieve PDA. However, PDA has not been studied for PLMs adaptation yet.

S2. Pseudo labeling

Pseudo labeling is a straightforward name for methods that use a source-domain classifier to generate pseudo labels for unlabeled data from the target domain. Compared with importance sampling, pseudo labeling focuses on utilizing target-domain unlabeled data. It is also known as self-training [McClosky et al., 2006] which utilizes the most confident labeled data from target domain to augment the source-domain labeled dataset to continuously train the source-domain model. The resulted model gains an improved discriminative ability on target-domain features. Pseudo labeling can be used in both SDA and UDA settings.

However, pseudo-labels are generally noisy. Since the capacity of PLMs is large enough, simply finetuning PLMs can easily overfit the corrupted labels and therefore hurt the generalization performance. Chen et al. [2020] propose to combine domain-adversarial learning with pseudo labeling where a trainable confusion matrix is optimized against a domain discriminator to reduce the gap between the pseudo-labels and the ground truth. El Mekki et al. [2021] applies this approach [Chen et al., 2020] to enhance BERT in Arabic cross-domain sentiment analysis. Ye et al. [2020] enhances the quality of pseudo labels by combining self training with knowledge distillation, which distills feature discriminative ability from PLMs to a smaller feature extractor. Liu et al. [2021] propose to reduce the domain shift through cycle self-training where a target classifier is trained on the pseudo labeled target-domain dataset and is required to perform well on the labeled source-domain dataset. Self-training can be extended to gradual domain adaptation in which intermediate domains are treated as target domains step by step [Kumar et al., 2020].

Apart from self-training, pseudo labeling can be used to induce domain invariance for domain adaptation. Wang et al. [2019a] uses a LSTM-based model to generate pseudo questions for target-domain passages and train a domain classifier to discriminate a given passage-question pair as coming from which domain. The answer generator is trained on the induced domain-invariant representations to adapt to the target domain task. Pseudo labeling methods can be applied to solve other machine learning problems such as learning from label proportions [Ardehaly and Culotta, 2016].

S3. Data Synthesis. Data synthesis methods refer to those generating new data points for the target domain using PLMs. It has been a new trend in the realm of domain adaptation recently. The purpose is to leverage the strong generation capability of generative PLMs such as BART to generate labeled pseudo-data for the target domain using labeled

information from the source domain. This neatly bypasses the domain discrepancy between different domains. For example, Yu et al. [2021b] proposed a review generation approach for cross-domain aspect-based sentiment analysis (ABSA). The synthesized review data is obtained by converting the domain-specific attributes (e.g., aspects, opinions, and collocations) of a source-domain example to those of the target domain using BERT. Similarly, Li et al. [2022] improved the quality of the synthesized data using a more powerful generative PLM, BART, to generate the masked attributes for cross-domain aspect and opinion extraction. Different from the direct generation approach, Chen et al. [2021] and Calderon et al. [2022] proposed to augment the original task by generating counterfactual data points that cross the domain but preserve the labels. The task-specific classifier is trained to be robust on the enhanced datasets, thereby disentangling the resilience on domain-specific features. In particular, Chen et al. [2021] adopted a reinforcement learning framework to gradually enhance the data generator and the sentiment classifier. In [Yang et al., 2022], authors set up a few-shot learning setting in which only a few in-domain examples are provided and the model is required to perform cross-domain named entity recognition. To overcome the small data challenge, they propose to generate pseudo examples using BERT to generate the masked entities in the context. The synthesized examples are used to enlarge the few-shot dataset to improve training stability. Overall, generative data augmentation methods generate synthesized data, which is domain-agnostic, to augment the original dataset. Training models on such augmented dataset is expected to be independent on domain-specific attributes and thereby improving cross-domain generalization performance.

S4. Prompt Learning

Recently, prompt methods arise as a new paradigm for adapting PLMs to downstream tasks. Prompting refers to methods that prompt the PLMs with additional information about the data such as task descriptions that are used to augment the input data. The resulted input data is usually in a cloze-question format where a template with prompt w_p encloses the original instance w_{in} and a mask token is left for PLMs to predict the label words w_l:

$$P(w_l|w_{in}, w_p) = \arg\max_{w_p} f_\theta(w_{in}, w_p). \tag{2.18}$$

With a properly designed prompt for the task at hand, a PLM can correctly generate the label words based on its inner language modeling knowledge priors, which has demonstrated excellent few-shot performance on a range of datasets [Brown et al., 2020]. The effect of prompt may be derived from the fact that it leverages the language modeling

objective to activate some parameters inside the PLM, making the relationship between the original input data and the label stronger. Recent studies show that the PLMs can behave differently with different kinds of prompts input to GPT [Meng et al., 2022] and other PLMs [Creswell and Shanahan, 2022].

The additional information that carried by prompts is restricted by the length of manually written prompts. Instead, soft prompt tuning methods [Lester et al., 2021, Li and Liang, 2021, Liu et al., 2022, Hambardzumyan et al., 2021] learn prompts through backpropagation on training data. They have demonstrated comparable performance with full-model tuning when the PLMs are large enough. SPOT [Vu et al., 2022] proposes to pretrain soft prompts on a set of source-domain datasets and then use the trained soft prompts to boost prompt tuning for target domains. PPT [Gu et al., 2022] introduces unsupervised tasks such as next sentence prediction as the pre-text task for prompt pretraining. After that, the soft prompts are finetuned on the few-shot target-domain data. OPTIMA [Guo et al., 2022a] improves over SPOT and PPT by directly performing domain adaptation. Prompts are also shown to boost full-model fine-tuning in LM-BFF [Gao et al., 2021], PET [Schick and Schütze, 2021a,b], and PERFECT [Rabeeh et al., 2022].

2.2.2 Model Optimization

This section describes methods that achieve domain adaptation by designing loss functions and regularization techniques to optimize models. The goal is to effectively train the model to learn to transfer the knowledge from related source domains to the target domains. There are three ways to achieve this goal. Continual learning methods continuously train the PLM on the target domain using pretraining tasks, aiming to adapt the contextualized representations to the target-domain text genre to reduce the gap between pretraining and downstream tasks. Adversarial learning methods refer to domain adversarial training, which extracts domain-invariant representations in a GAN-like setup, and adversarial training, which enhances model generalization performance by smoothing its decision boundary against adversarial perturbations. Metric learning methods train the model to optimize certain metric such that the source domain data representations look like those of the target domain. Models that perform well on those source domain samples can generalize to similar target domain samples as well.

S5. Continual Learning

Continual learning aims to specialize a PLM to a particular domain by continuing the pretraining task on the abundant unlabeled corpora, such as BioBERT [Lee et al., 2020] trained on biological documents, SciBERT [Beltagy et al., 2019] trained on scientific papers, ClinicalBERT[Alsentzer et al., 2019, Huang et al., 2019a] and ClinicalCLNet[Huang et al., 2019b] trained on clinical notes, FinBERT [Araci, 2019] trained on financial new, LegalBERT [Chalkidis et al., 2020] trained on legal documents, BERTweet [Nguyen et al., 2020] trained on English tweets, and BERTweetFR [Guo et al., 2021b] trained on french tweets. During this stage, we continue to train the PLM using the same pretraining task, which is usually a language modeling objective, on the target domain datasets. When data from the target domain are far from enough for the pretraining tasks, then the pretraining objective often plays a role of regularization in the loss functions for the downstream tasks. Based on the purpose of continual pretraining, we categorize methods into vocabulary adaptation, pretrain-finetune and continual adaptation.

1. Vocabulary Adaptation. Texts in specialized fields may contain numerous domain-specific terms which are not found in open domain corpora and are therefore not well captured by the vocabulary of PLMs. Domain-specific vocabularies can help domain adaptation of PLMs [Gu et al., 2021]. For example, training the special token embeddings of GPT-2 can enable it in task-oriented dialogue use cases without the need of training new dialogue submodules [Budzianowski and Vulić, 2019]. Zhang et al. [2020] propose to extend the vocabulary of RoBERTa with frequent words from target domains and continue to finetune RoBERTa using a self-constructed reading comprehension task based on coarse annotations. The post-trained RoBERTa was shown to improve low-resource QA tasks from the target domain. Yao et al. [2021] propose to expand the vocabulary of BERT with domain-specific corpus and continue to train BERT on the target domain using masked language modeling and knowledge distillation objectives to distill BERT to a small-scale LM which is supposed to be trained as an expert for domain-specific tasks. Sachidananda et al. [2021] show that the common words from target domains can be represented as the mean of their subword embeddings without further pretraining, which can also effectively adapt BERT to new domains. Poerner et al. [2020] propose to adapt BioBERT to the target domains by aligning its word embeddings to the embeddings trained with Word2Vec [Mikolov et al., 2013c] on the target-domain corpus. To alleviate the semantic shift problem of tokens embeddings during continual pretraining in target domains, Vu et al. [2020] propose a mask learning strategy to adversarially mask out tokens that are hard to reconstruct by the BERT. Lekhtman et al. [2021] propose to use aspect category information to selectively mask tokens for masked

language modeling and continue to pretrain BERT to induce both domain-invariant and category-invariant representations for cross-domain aspect extraction.

2. Pretrain-Finetune. Studies in [Gururangan et al., 2020] show that it is helpful to tailor a pretrained model to the domain of a target task through a second phase of pretraining for both high- and low-resource settings. The cross-domain adapted BERT [Rietzler et al., 2019] demonstrates that domain-specific language modeling followed by supervised task-specific finetuning can significantly boost aspect-based sentiment classification. Karouzos et al. [2021] propose to add MLM loss on the target-domain data as regularization during finetuning BERT for source-domain sentiment classification. Du et al. [2020] propose to enable BERT with domain awareness by introducing adversarial domain discrimination into the continual pretraning stage where BERT is further pretrained on the sentiment classification datasets using masked language modeling. Unsupervised domain adaptation of BERT to languages of special genres, such as Early Modern English and Tweets, can be achieved by only finetuning the contextualized embeddings using masked language modeling on unlabeled text from the target domain [Han and Eisenstein, 2019]. Continual pretraining on unlabeled data from target domain using language modeling has shown to be effective to adapt AraBERT [Antoun et al., 2020] to tweets data for arabic dialect identification [Beltagy et al., 2020]. Nishida et al. [2020] shows that finetuning BERT with language modeling on the target-domain datasets while performing reading comprehension on the source-domain QA datasets can better adapt BERT to the target-domain QA datasets.

3. Continual Adaptation. Continual pretraining is closely related to lifelong learning or continual adaptation [Parisi et al., 2019] in which a general model is continuously adapted to new domains. Xu et al. [2021] shows that gradually finetuning BERT-based dialogue models in a multi-stage process is better than one-stage finetuning. Thompson et al. [2019] adopt the lifelong learning setting and train the BART across different domains for text generation. Despite simple and easy to deploy, this learning setting typically incur a catastrophic forgetting problem [McCloskey and Cohen, 1989, Kirkpatrick et al., 2017]. Blindly continue pretraining a given PLM on the target domain can be trapped in the forgetting problem. Studies in [Yu et al., 2021a] demonstrate that the dissimilarity between the pretraining data and target domain task can degrade the effectiveness of BART in abstractive summarization in which case seeking a relate source domain to perform domain adaptation can be helpful. The performance degradation on the target domain can be reduced by inventing more advanced techniques such

as look-ahead learning on the domain discriminator under adversarial neural transfer [Guo et al., 2021a], where BERT representations can be better adapted to the target domain. To enable temporal domain adaptation of PLMs to emering data, Jin et al. [2022] studied different continual learning algorithms to continue pretraining RoBERTa in new domains. Experiments show that distillation-based continual learning achieves better temporal generalization performance than other possible solutions include tuning domain-specific adapters [Houlsby et al., 2019] and memory replay methods [Chaudhry et al., 2019].

S6. Adversarial Learning

Adversarial learning methods generally employ a GAN-like setup [Goodfellow et al., 2014] where a domain discriminator is optimized against the task-specific learning objectives.

1. Domain-adversarial Training. Instead of directly fitting a single PLM on non-identical domains, the leading solution to this problem is to reconfigure the network into domain-agnostic and domain-specific layers [Rebuffi et al., 2017, Wang et al., 2019b]. The mainstream domain adaptation approaches in the literature are developed based on domain-adversarial neural networks (DANN) [Ganin et al., 2016] or adversarial discriminative domain adaptation (ADDA) framework [Tzeng et al., 2017]. The goal is to induce domain-invariant representations via the domain-agnostic layers and map the source and target data into a common feature space by solving a min-max game between

$$\arg\min_{\theta_G} \mathcal{L}_C(x_s, y_s) - \mathcal{L}_{AD}(x_s, x_t), \tag{2.19}$$

and

$$\arg\min_{\theta_D} \mathcal{L}_{AD}(x_s, x_t), \tag{2.20}$$

where θ_D denotes the parameters of domain discriminator and θ_G denotes the reset parameters of the model including the task classifier. $\mathcal{L}_{AD}$ computes the cross entropy for domain classification over source and target domains:

$$\mathcal{L}_{AD}(x_s, x_t) = -\log(P(x_s = 1)) - \log(P(x_t = 0)). \tag{2.21}$$

Lee et al. [2019] employs a domain discriminator and applies domain-adversarial training to achieve domain generalization of BERT for QA tasks. Wang et al. [2019a] generates pseudo questions for unlabeled target-domain passages and a domain classifier is

applied on top of BERT to discriminate which domain a passage-question pair comes from. Zou et al. [2021] use the domain discriminator to deceive an autoencoder to enforce RoBERTa to produce domain-invariant representations. Ghosal et al. [2020] propose to improve DANN with an external knowledge base, ConceptNet, to enhance both domain-specific and general knowledge extraction for cross-domain sentiment analysis. Tang et al. [2020] propose to exploit structural domain similarity to enhance the discriminability of domain-invariant representations for the target-domain data.

Studies in [Ryu et al., 2022, Guo et al., 2021a] found that a catastrophic forgetting problem occurs when the ADDA framework is applied to the BERT model. Guo et al. [2021a] propose a look-ahead optimization strategy to accommodate the adversarial domain discrimination loss and the task-specific classification loss when optimizing BERT representations. Ryu et al. [2022] propose to use knowledge distillation [Hinton et al., 2015] to distill knowledge from source encoder to target encoder, thereby regularizing ADDA for unsupervised domain adaptation of BERT.

2. Adversarial Robustness and Consistency Training. Adversarial robustness refers to ensuring models to be robust against adversarially generated perturbations. Consistency training [Sajjadi et al., 2016, Xie et al., 2020] forces the model to make consistent predictions against small perturbations which are not necessarily to be adversarial noise. Both techniques try to smooth the decision boundary to improve the generalization performance of a model in the face of a small distribution deviation within a tolerance bound:

$$x'_s = x_s + \epsilon \bigtriangledown_{x_s} \mathcal{L}_C(x_s, y_s). \tag{2.22}$$

The perturbed samples x'_s will be added to the training set to train the model to minimize the original classification loss over them:

$$\mathcal{L}_{AT} = \mathcal{L}_C(x_s, x'_s, y_s) \tag{2.23}$$

This kind of regularization technique has been widely adopted in NLP. For example, Park et al. [2022] produce discrete virtual adversarial noise to the token embeddings. Yoon et al. [2021] apply mixup to perturb the spans of the input texts for text classification for consistency training. Kim et al. [2021] propose a consistency training framework to enhance the conversational dependency of question answering. They have shown to be able to boost the generalization performance of a model. Recent studies show that AT can also help domain adaptation by focusing on smoothing the decision boundary where

source and target domain are similar [Guo et al., 2022a, Liu et al., 2019, Jiang et al., 2020]:

$$\mathcal{L}_{all} = \mathcal{L}_C(x_s, y_s) + \lambda_1 \cdot \mathcal{L}_{AT}(x_s, y_s) + \lambda_2 \cdot \mathcal{L}_{AD}(x_s, x_t). \tag{2.24}$$

Using λ_{AD} to generate perturbations can reduce the domain gap thereby enhancing domain adaptation [Jiang et al., 2020]. In [Guo et al., 2022a], authors found that optimizing the domain discrimination loss and task classification loss for T5-based prompt tuning across domains suffer from low capacity of the soft prompts while applying λ_{AD} to generate transferable perturbations can avoid the loss competition problem [Guo et al., 2021a]. Moreover, the problem of tail classes alignment across domains can also be alleviated by training against adversarial perturbations for semantic segmentation [Yang et al., 2020]. However, this topic has not been studied in NLP yet.

S7. Metric Learning

Metric learning techniques have also been explored for the purpose of domain adaptation. The goal of applying metric learning is to train the neural networks to optimize a designed metric such that the resulted representations can have certain property. Earlier research focus on aligning the output distributions of the source and the target domains by minimizing the discrepancy between them. Tzeng et al. [2014] was the first to adopt Maximum Mean Discrepancy (MMD) [Gretton et al., 2012] metric for both SDA and UDA settings. MMD is computed on the CNN representations of source and target images as a measurement of distribution discrepancy:

$$\mathcal{L}_{\mathrm{M}} = \| \frac{1}{|X_s|} \sum_{x_s \in X_s} f_\theta(x_s) - \frac{1}{|X_t|} \sum_{x_t \in X_t} f_\theta(x_t) \|. \tag{2.25}$$

The model tries to learn representations that are invariant to source and target domains by minimizing the squared MMD loss together with the task-specific loss:

$$\mathcal{L} = \mathcal{L}_C(X_l, Y) + \lambda \mathcal{L}^2_{\mathrm{M}}(X_s, X_t), \tag{2.26}$$

where X_l contain all the labeled data from source and target domains. Another commonly used metric is correlation analysis. Sun and Saenko [2016] and Sun et al. [2016] are the first to use correlation analysis to reduce the domain shift in UDA. Rahman et al. [2020] combine the correlation analysis and adversarial learning to achieve domain adaptation and generalization. The goal of correlation-based domain alignment is

to minimize the difference between the covariance of the source features and the covariance of the target features:

$$\mathcal{L}_{cor}(x_s, x_t) = \|\text{cov}(f_\theta(x_s)) - \text{cov}(f_\theta(x_t))\|_F^2, \tag{2.27}$$

which is used to regularize the overall training objective:

$$\mathcal{L} = \mathcal{L}_C(X_l, Y) + \lambda \mathcal{L}_{cor}(X_s, X_t). \tag{2.28}$$

Mutual information (MI) has also been exploited for domain adaptation [Li et al., 2020] in which the MI between the representations from two domains are maximized to extract domain-invariant features on top of XLM [Conneau and Lample, 2019]. This kind of approach stems from Informax optimization, which refers to the principle that when a set of input values is mapped to a set of output values through a function, the Shannon mutual information between them should be maximized. Enforcing neural network representations of data to match a specific statistical prior came with adversarial autoencoders [Makhzani et al., 2015]. Deep Informax [Hjelm et al., 2019] extends this idea to Informax Optimization problems to constrain representation learning. The quality of the learned representations can be measured by the mutual information between them and the corresponding input data [Belghazi et al., 2018]. They can be used for independent component analysis.

2.2.3 Model Personalization

This section introduce techniques that can achieve multi-target domain adaptation with a single PLM. The goal is to adapt the same PLMs to different domains manifested as personal preferences. We divide the possible solutions into three categories. Posterior adaptation methods adapt the conditional predictions $P(X|Y)$ using statistical priors such as label proportions to adjust the predicted distributions towards the target domain distributions. Specification methods specify a small set of the parameters in the PLM to be adapted to the target domain without changing the remaining parameters. Reparameterization methods add external parameters beside the PLM to map the PLM representations to a lower space where the goal is to learn from the target domain data without changing any parameter of the PLM.

Motivation. Understanding personal habits of language usage in terms of named entities [Li et al., 2018], part-of-speech [Sennrich and Haddow, 2016], and syntactic structure [Aharoni and Goldberg, 2017], is important to personalize a system to different users. These information are contained in domain-specific data. The challenge is that finetuning every copy of the same PLM on a different domain could be prohibitive as the model size and the number of domains grow. The task of personalizing the same PLMs to different domains at scale is at the intersection between domain adaptation and personalized federated learning (PFL) [Tan et al., 2022]. The latter is proposed to solve the problem in which FL-trained models incur performance drop across different data distributions from different clients. In view of the practical significance of providing customized NLP service for applications such as personalized response generation [Yang et al., 2017, Zhang et al., 2019], we aggregate those methods that are promising to adapt the same PLMs to different domains into this category. There are three kinds of ways to achieve this. Posterior adaptation methods study how to adapt a fixed pretrained model to a domain where the label distributions shift from the training data. Specification methods specifies a small amount of the inner parameters of PLMs to be tuned using the domain-specific datasets. Reparameterization methods inject new parameters to PLMs without changing any of the pretrained parameters.

S8. Posterior Adaptation

Empirical rish minimization trains a neural model to estimate the *posterior* probability $\hat{p}(Y|X)$ to describe how likely the observed training data x happen to be the label y. The model tries to approximate the true class priors $p(y|x)$ by learning from more representative training data or using different optimization techniques. However, in practical evaluation scenarios, the prior probabilities may differ from that of the training set and may even change from one domain to another, which is often called prior shift or label shift [Šipka et al., 2022]. Coping with prior shift is important for personalizing PLMs to multiple different domains since re-training a PLM is quite expensive and can easily overfit an imbalanced and small dataset. Based on Bayes rules, we can derive the following:

$$p(y|x) = \frac{p(y)}{p(x)} \cdot \frac{\hat{p}(y|x)\hat{p}(x)}{\hat{p}(y)} \propto \hat{p}(y|x) \cdot \frac{p(y)}{\hat{p}(y)}, \tag{2.29}$$

where the ratio $\frac{p(y)}{\hat{p}(y)}$ implies the prior shift. In [Šipka et al., 2022], authors propose the test-time adaptation of a fixed pretrained classifier after a prior shift happens by re-weighting its predictions based on confusion matrices. To avoid over-confident predictions due to overfitting to some classes, Alexandari et al. [2020] propose to calibrate

the confidence of classifier predictions by adding class-specific bias terms:

$$\hat{p}(y|x) = \frac{\exp(z_i(x)/\beta + b_i)}{\sum_j \exp(z_j(x)/\beta + b_j)}, \tag{2.30}$$

where $z_i(x)$ represents the output logits of the input x and β is a temperature scaling factor. These methods focus on solving the label shift problem of one target domain. Extending this problem to multiple domains calls for another line of machine learning research called Learning from label proportions (LLP) in which the training data is provided in groups and only the label distribution for each group is given [Quadrianto et al., 2008, Rueping, 2010]. Given a model parameterized by θ, the task is to predict individual labels $y \in \{-1, 1\}$ for each group. The key is how to utilize the given label proportion to optimize model's predictions. Yu et al. [2013] propose the $\propto$SVM regularization approach which minimizes a penalization term $\mathcal{L}_{pn}$ to reduce the difference between the true label proportion p_k and the estimated label proportion $\hat{p}_k$ of group k :

$$\mathcal{L}_{pn}(\hat{p}_k(y|x), p_k(y); \theta) = |\hat{p}_k(y|x) - p_k(y)| \tag{2.31}$$

Solving LLP problems enables interesting applications such as modeling voting behaviors across different demographic groups [Yu et al., 2013]. However, the use of LLP methods may raise concerns about privacy leakage resulted from observing label proportions. To mitigate this issue, Guo et al. [2022b] propose to adopt federated learning framework in which training data and label proportions are kept on local devices while only the model parameters are communicated between devices. On each client the true label proportion is used to penalize the estimated one with a temperature scaling factor β tuned on validation set:

$$\mathcal{L}_{pk}(\hat{p}_k(y|x), p_k(y); \theta) = \frac{\hat{p}_k(y|x)}{p_k(y)^\beta} \tag{2.32}$$

S9. Specification

Specification methods specify a part of the parameters of a PLM to be tuned for domain adaptation. A cross-lingual study [Lee et al., 2021] found that selectively post-train parameters of RoBERTa which is pretrained on high-resource languages can better adapt it to low-resource languages. Michel and Neubig [2018] propose personalized model adaptation solely performed on the output vocabulary bias vector. Wuebker et al. [2018]

propose a parameter-efficient domain adaptation setting for training personalized machine translation models. Most of the model parameters are frozen during training while a set of offset tensors are personalized to each user and need to be trained. Structured sparsity is encouraged on the offset tensors via group lasso regularization [Scardapane et al., 2017] to reduce parameters consumption from each user. Similar as the design of residual adapters for visual domains [Rebuffi et al., 2017], modular domain adaptation for text understanding [Chen et al., 2022] also successfully applied domain-specific bias and normalization terms in customizing models to different users. Sun et al. [2021] proposes partial PFL which loads a subset of the global model's parameters as initialization on each client and shows improved generalization under cross-domain evaluation. BitFit [Ben Zaken et al., 2022] updates the bias of PLMs while freezing the rest parameters. Their ablation studies also show that finetuning only a subset of all the bias terms in the PLM can achieve similar performance as finetuning the whole bias set, indicating a specialization on the bias set is promising to adapt the PLM to multiple domains together. However, when equipped with large enough PLMs, different parameter-efficient adaptation methods result in similar performance [Ding et al., 2022], indicating an upper bound may exist for the adaptation performance of PLMs.

It is possible to localize knowledge in a PLM in order to make targeted parameter updates without forgetting most of the already-learned knowledge [Dai et al., 2022, De Cao et al., 2021, Mitchell et al., 2022]. However, repeated editing the PLM can still exhibit forgetting problems [Hase et al., 2021]. Elucidating the scenarios in which personalization does or does not benefit performance is an important direction for future work.

S10. Re-parameterization

Re-parameterization methods adapt large-scale PLMs by optimizing a low-dimensional subspace of the model or transformed from the model. Adapter modules [Rebuffi et al., 2017, Wang et al., 2019b] come to compress many visual domains together to adapt a single model to multiple domains together without ignoring domain-specific features. The first approach to adapt a frozen PLMs to domain-specific datasets came with Adapter tuning [Houlsby et al., 2019] which inserts an adapter module, simply two linear layers with activation and skip-connection, between transformer blocks of PLMs. Cooper Stickland et al. [2021] injects domain-specific and language-specific adapters to a vanilla Transformer which is pretrained on multilingual data and adapt it to new domains and new languages together for machine translation. Chronopoulou et al. [2022] specializes

GPT-2 in a number of domains by constructing a hierarchical tree with each node associated with an adapter module. The GPT-2 is finetuned together with those adapters with the task of language modeling. The hierarchical adapters allows the partially sharing of similar domains, which generalizes better than assigning each domain with a domain-specific adapter and enforcing all domains to share the same adapter. AdapterDrop [Rücklé et al., 2021] enhances the generalization performance of Adapter tuning by learning to drop out some adapter layers. Compacter [mahabadi et al., 2021] reduces trainable parameters of adapter by decomposing the linear layers into low-rank matrices while maintaining the same performance. Parallel Adapter [He et al., 2022] inserts an adapter to every transformer layer in parallel which allows adaptation to be faster than the sequential Adapter [Houlsby et al., 2019]. LoRA [Hu et al., 2022] injects a trainable low-rank matrix aside each dense layer in the transformers to enforce the layer parameters to be decomposed into low-rank matrices.

TABLE 2.2: A visualization of the assumptions, approaches and PLMs adopted in related works.

Assumptions				**Approaches**										**PLMs**	**Related Work**
A1	A2	A3	A4	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10		
✓				✓						✓				BERT	[Ma et al., 2019]
✓					✓				✓					ELMo,BERT	[El Mekki et al., 2021, Wang et al., 2019a]
✓									✓	✓				BERT,RoBERTa	[Ryu et al., 2022, Zou et al., 2021]
✓					✓					✓				BERT	[Chen et al., 2020]
✓							✓		✓					T5	[Ye et al., 2020]
✓							✓	✓						T5,CPM	[Gu et al., 2022]
			✓							✓				BERT	[Guo et al., 2021a]
			✓				✓							T5	[Vu et al., 2022]
✓								✓						BERT	[Karouzos et al., 2021]
✓					✓									BERT	[Liu et al., 2021]
✓							✓	✓						T5	[Gu et al., 2022]
✓							✓		✓					T5	[Guo et al., 2022a]
		✓						✓						BERT,BART,RoBERTa	[Thompson et al., 2019, Lekhtman et al., 2021, Rietzler et al., 2019, Xu et al., 2021, Jin et al., 2022]
✓								✓	✓					BERT	[Du et al., 2020]
✓								✓						BERT,BART	[Yu et al., 2021a, Han and Eisenstein, 2019, Nishida et al., 2020]
✓									✓					BERT,RoBERTa	[Wang et al., 2019a, Lee et al., 2019, Zou et al., 2021, Xie et al., 2020]
		✓							✓					BERT	[Park et al., 2022]
		✓								✓				XLM	[Li et al., 2020]
✓											✓			BERT	[Guo et al., 2022b]
		✓										✓		RoBERTa	[Lee et al., 2021, Chen et al., 2022]
			✓										✓	GPT-2	[Chronopoulou et al., 2022]
	✓					✓								BERT	[Yu et al., 2021b, Chen et al., 2021, Yang et al., 2022]
	✓					✓								BART	[Li et al., 2022]
	✓					✓								T5	[Calderon et al., 2022]

Chapter 3

Sample-Efficient Prompt Tuning with Domain Adaptation

3.1 Motivation

Prompt tuning [Lester et al., 2021, Li and Liang, 2021, Liu et al., 2022, Hambardzumyan et al., 2021] is an effective method for adapting large-scale pretrained language models for downstream tasks. While keeping the PLM weights unchanged, prompt tuning trains input vectors, called soft prompts, that are input to the PLM alongside the text. The success of prompt tuning has inspired subsequent studies on parameter-efficient adaptation of PLMs [Ding et al., 2022, Su et al., 2021, Wei et al., 2021b, Zhu et al., 2022].

However, training effective soft prompts usually requires sufficient labeled training data [Su et al., 2021]. Studies have shown that prompt tuning significantly underperforms full-model tuning on many few-shot classification tasks [Gu et al., 2022]. Our experiments corroborate this finding. In addition, we find that, in few-shot learning, prompt tuning is equally, if not more, sensitive to random seed choices compared to full-model tuning, despite having far fewer trainable parameters (§3.3.4). Gu et al. [2022] address this by transferring prompts learned from a source domain to the target domain with few training data.

We investigate a related but different application scenario, domain adaptation, where we have many unlabeled examples but no labeled examples from the target domain. Such situations are common when data are abundant but the labeling cost, including annotator

recruitment, annotator training, and quality assurance, is high. As a result, utilizing the unlabeled examples efficiently is crucial to performance.

We propose bOosting Prompt TunIng with doMain Adaptation (OPTIMA). Employing regularization from adversarial perturbation, OPTIMA learns a smooth decision boundary that passes through regions of low data density. In addition, recognizing that the feature distributions in the two domains may overlap only partially, we propose to focus the regularization on regions where the target-domain and source-domain data exhibit high similarity. We illustrate the intuition in Figure 3.1.

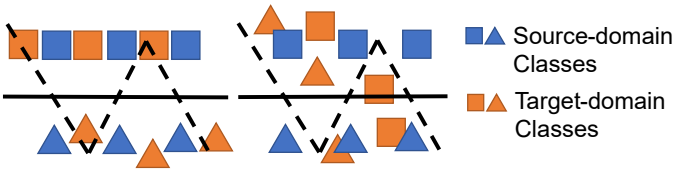


FIGURE 3.1: Smooth vs. zigzag decision boundaries. Left: When the distribution of the target-domain data (orange) is similar to the source domain (blue), the smooth decision boundary (solid line) generalizes better than the zigzag boundary. Right: When the distributions are different, smoothness is of dubious benefit.

3.2 Domain Adaptation for Prompt Tuning

In this section, we first introduce prompt tuning for text classification. Then, we introduce how to enhance the in-domain generalization performance of soft prompts by augmenting the input with virtual perturbations. Next, we propose how to optimize the perturbations to reduce the domain gap and obtain soft prompts with domain-invariant knowledge. Finally, we show how to use soft prompts to boost few-shot learning in the target domain.

3.2.1 Preliminaries: Prompt Tuning

We start by introducing some notations. The input $\boldsymbol{x}$ is a sequence of n token embeddings, $\boldsymbol{x} = \langle x_1, \ldots, x_n \rangle$. The trainable soft prompt sequence $\boldsymbol{p}$ has m embeddings, $\boldsymbol{p} = \langle p_1, \ldots, p_m \rangle$. The manually designed hard prompt sequence $\boldsymbol{h}$ has k token embeddings $\boldsymbol{h} = \langle h_1, \ldots, h_k \rangle$. All embedding vectors have d dimensions. The soft prompt and the hard prompt are both task-specific. The hard prompt text is usually a natural

language description of the task, whereas the soft prompts do not correspond to any text and are trained directly using gradient descent.

For classification problems, we adopt the masked language modeling formulation, which aims to predict a predefined verbalizer token $y \in \mathcal{V}$ at a masked position in the input. For example, for binary classification, the words "yes" and "no" may be used as verbalizers that indicate positive and negative predictions, where we may define the label space as $\mathcal{Y} = \{\text{yes}, \text{no}\}$. In encoder-only networks such as BERT [Devlin et al., 2019], the output of the encoder is mapped to the label space $\mathcal{Y}$ via a projection head. In encoder-decoder networks like T5 [Raffel et al., 2020b], the decoder is responsible for generating the verbalizer token.

We concatenate all sequences and the embedding of the [MASK] token, $e([\text{MASK}])$, to form the final input to the PLM: $\langle \boldsymbol{p}; \boldsymbol{h}; \boldsymbol{x}; e[\text{MASK}] \rangle$. For simplicity, we use the function $f(\boldsymbol{x}, \boldsymbol{p})$ to denote the PLM prediction at the masked position, which is a multinomial distribution over $\mathcal{Y}$. We adopt the cross-entropy classification loss ℓ_{xe} with the ground-truth label $y \in \mathcal{Y}$.

$$\ell_{\text{xe}}(\boldsymbol{x}, y, \boldsymbol{p}) = -\log P(f(\boldsymbol{x}, \boldsymbol{p}) = y). \tag{3.1}$$

We optimize the soft prompt by minimizing the expected loss over the labeled training set, $\mathcal{D}$:

$$\boldsymbol{p}^* = \arg\min_{\boldsymbol{p}} \mathbb{E}_{(\boldsymbol{x},y)\in\mathcal{D}} \big[\ell_{\text{xe}}(\boldsymbol{x}, y, \boldsymbol{p})\big]. \tag{3.2}$$

3.2.2 The OPTIMA Approach

We build OPTIMA off two intuitions regarding domain adaptation. First, as the target domain provides no direct supervision, it is easy to overfit the source domain. Therefore, it is important to mitigate overfitting by regularizing the network to maintain a smooth decision boundary.

Under an adversarial learning framework, we seek a small perturbation $\boldsymbol{\delta}$ that, when added to the input, results in maximum change in the model prediction. After that, we optimize the model parameters to minimize the prediction change under the adversarially perturbed input. The overall result is a network whose output $f(\boldsymbol{x})$ changes little where a small change is added to the input $\boldsymbol{x}$. In the sense of Lipschitz continuity, such a decision boundary is smooth. Smooth decision boundaries can be understood as passing

through regions of low data density and are shown to improve generalization [Huang et al., 2020, Cicek and Soatto, 2019, Kim et al., 2019].

The second intuition is that we do not have to regularize the entire decision boundary. As the source and target domains may have different data distributions, all that matters is the decision boundary segment close to the target-domain data. Therefore, we target the regularization and the perturbation $\boldsymbol{\delta}$ to areas on the data manifold where the source domain and target domain are similar.

Specifically, we have a labeled dataset from the source domain, $\mathcal{D}_s = \{(\boldsymbol{x}_s^{(i)}, y_s^{(i)})\}_{i=1}^{N_s}$, drawn i.i.d. from the distribution P_s and an unlabeled dataset from the target domain, $\mathcal{D}_t = \{\boldsymbol{x}_t^{(j)}\}_{j=1}^{N_t}$, drawn i.i.d. from the distribution P_t. We define ℓ_{KL} as the KL divergence between the prediction of the original input and that of the perturbed input,

$$\ell_{\text{KL}}(\boldsymbol{\delta}, \boldsymbol{p}, \boldsymbol{x}_s) = \text{KL}(f(\boldsymbol{x}_s, \boldsymbol{p}) \parallel f(\boldsymbol{x}_s + \boldsymbol{\delta}, \boldsymbol{p})). \tag{3.3}$$

ℓ_{KL} measures how much the model prediction changes when the perturbation $\boldsymbol{\delta}$ is applied to $\boldsymbol{x}_s$ and captures the smoothness of the decision boundary. We illustrate the intuition in Figure 3.2.

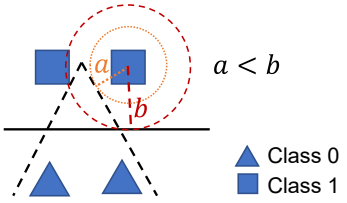


FIGURE 3.2: Intuition about perturbation and smoothness. Under the zigzag (non-smooth) decision boundary, a small perturbation with a well-chosen direction is sufficient to flip the predicted class. The smooth boundary requires a larger perturbation.

Further, we introduce a domain discriminator network parameterized by $\boldsymbol{\theta}_d$, which attempts to distinguish data instances from the two domains. This network is trained to reduce the domain discrimination loss $\mathcal{L}_{\text{disc}}$,

$$\begin{aligned}\mathcal{L}_{\text{disc}}(\boldsymbol{\delta}, \boldsymbol{x}_s, \boldsymbol{x}_t) =& \mathbb{E}_{\boldsymbol{x}_s, \boldsymbol{x}_t}\big[-\log P(z=1|\boldsymbol{x}_s + \boldsymbol{\delta}) \\ & -\log P(z=1|\boldsymbol{x}_s) \\ & -\log P(z=0|\boldsymbol{x}_t)\big],\end{aligned} \tag{3.4}$$

where z is the network output. This loss is a variation of the cross-entropy with an additional term where $\boldsymbol{x}_s$ is perturbed by $\boldsymbol{\delta}$. In addition, we define an adversarial loss,

$$\ell_{\text{adv}}(\boldsymbol{\delta}, \boldsymbol{x}_s) = -\log P(z = 1|\boldsymbol{x}_s + \boldsymbol{\delta}), \tag{3.5}$$

which, when maximized, causes the domain discriminator to mistake the perturbed source example $\boldsymbol{x}_s + \boldsymbol{\delta}$ as coming from the target domain.

For a given source-domain input, $\boldsymbol{x}_s$, we find the perturbation $\boldsymbol{\delta}^*$ within a ϵ-radius ball that maximizes the following objective,

$$\boldsymbol{\delta}^* = \underset{\boldsymbol{\delta}, \|\boldsymbol{\delta}\|_2 \leq \epsilon}{\arg\max}\, \ell_{\text{KL}}(\boldsymbol{\delta}, \boldsymbol{p}, \boldsymbol{x}_s) + \ell_{\text{adv}}(\boldsymbol{\delta}, \boldsymbol{x}_s). \tag{3.6}$$

Here, $\ell_{\text{adv}}(\boldsymbol{\delta}, \boldsymbol{x}_s)$ can be understood as a regularization term for $\boldsymbol{\delta}$. By maximizing ℓ_{KL}, we seek a disturbance to the input that causes the most change in the model prediction. At the same time, the disturbed input $\boldsymbol{x}_s + \boldsymbol{\delta}^*$ from the source domain should resemble data in the target domain, in order to maximize $\ell_{\text{adv}}(\boldsymbol{\delta}, \boldsymbol{x}_s)$; ℓ_{adv} constrains $\boldsymbol{\delta}^*$ to the region where the data from the two domains are similar.

We optimize the above loss w.r.t. $\boldsymbol{\delta}$ using projected gradient ascent (PGA). After every gradient descent step, $\boldsymbol{\delta}$ is projected back to the ϵ-radius ball $\mathcal{Q}_\epsilon = \{\boldsymbol{\delta} | \|\boldsymbol{\delta}\|_2 \leq \epsilon\}$. We write the projection operation as

$$\prod_{\|\cdot\|_2 \leq \epsilon} (\boldsymbol{\phi}) = \underset{\boldsymbol{\delta} \in \mathcal{Q}_\epsilon}{\arg\min} \|\boldsymbol{\delta} - \boldsymbol{\phi}\|_2 = \frac{\epsilon \boldsymbol{\phi}}{\max(\epsilon, \|\boldsymbol{\phi}\|_2)}. \tag{3.7}$$

The update to $\boldsymbol{\delta}$ can be written as

$$\boldsymbol{\delta} \leftarrow \prod_{\|\cdot\|_2 \leq \epsilon} \left(\boldsymbol{\delta} + \eta_\delta \frac{\boldsymbol{g}}{\|\boldsymbol{g}\|_2} \right), \tag{3.8}$$

$$\boldsymbol{g} = \bigtriangledown_{\boldsymbol{\delta}} (\ell_{\text{KL}}(\boldsymbol{\delta}, \boldsymbol{p}, \boldsymbol{x}_s) + \ell_{\text{adv}}(\boldsymbol{\delta}, \boldsymbol{x}_s)), \tag{3.9}$$

where η_δ is the learning rate. We normalize $\boldsymbol{g}$ to make sure the updates have the same magnitude.

During the training session, we alternately optimize the perturbation $\boldsymbol{\delta}$ and the soft prompt $\boldsymbol{p}$. With $\boldsymbol{\delta}^*$ found by PGA, we optimize the following loss function over $\boldsymbol{p}$ using standard gradient-based optimization.

$$\mathcal{L}_R = \mathbb{E}_{(\boldsymbol{x}_s, y_s) \in \mathcal{D}_s} \left[\ell_{\text{xe}}(\boldsymbol{x}_s, y_s, \boldsymbol{p}) + \ell_{\text{KL}}(\boldsymbol{\delta}^*, \boldsymbol{p}, \boldsymbol{x}_s) \right]$$
$$\boldsymbol{p}^* = \arg\min_{\boldsymbol{p}} \mathcal{L}_R \tag{3.10}$$

$\mathcal{L}_R$ is the empirical expectation computed over the current mini-batch. With the same $\boldsymbol{\delta}^*$, we also minimize the domain discrimination loss over the discriminator network parameter $\boldsymbol{\theta}_d$.

3.2.3 The OPTIMA Algorithm

We show the complete OPTIMA algorithm as Algorithm 1. With lines 5 and 6, we create an initial perturbation $\boldsymbol{\delta}_0^{(i)}$ for every source data point $\boldsymbol{x}_s^{(i)}$. From line 7 to line 13, we iteratively update the perturbation $\boldsymbol{\delta}^{(i)}$ associated with every source-domain data point $\boldsymbol{x}_s^{(i)}$ using projected gradient ascent on $\ell_{\text{KL}} + \ell_{\text{adv}}$. After K iterations, we find $\boldsymbol{\delta}^{(i)*} = \boldsymbol{\delta}_{K-1}^{(i)}$, compute $\bigtriangledown_{\boldsymbol{p}} \mathcal{L}_R$ accordingly, and update $\boldsymbol{p}$ with stochastic gradient descent (SGD) and learning rate η_p (line 16). In line 17, we update the domain discriminator parameters $\boldsymbol{\theta}_d$ using SGD with the current mini-batches. Though we show the vanilla SGD updates in lines 16-17, we can easily switch to other optimizers such as SGD with momentum or Adam [Kingma and Ba, 2015].

3.2.4 Comparison with Virtual Adversarial Training

Virtual Adversarial Training (VAT) [Miyato et al., 2018, 2016] is a pioneering work that applies adversarial perturbation to unlabeled examples in a semi-supervised learning (SSL). The SSL assumption is that we have labeled data $(\boldsymbol{x}, y) \overset{\text{i.i.d.}}{\sim} P$ and unlabeled data $\boldsymbol{x} \overset{\text{i.i.d.}}{\sim} P$. Notice that $\boldsymbol{x}$ is drawn from the same distribution P regardless of the existence of the label y. VAT finds disturbance $\boldsymbol{\delta} \in \mathcal{Q}_\epsilon$ that maximizes the change in the model prediction $\text{KL}(f(\boldsymbol{x}) \parallel f(\boldsymbol{x} + \boldsymbol{\delta}))$. After that, the neural network minimizes cross-entropy on labeled data and the KL-divergence under disturbance on all data. Similar ideas have been explored by Park et al. [2022], Cicek and Soatto [2019], Kim et al. [2019].

Algorithm 1: OPTIMA

Input: A labeled source-domain dataset $\mathcal{D}_s = \{(\boldsymbol{x}_s^{(i)} y_s^{(i)})\}_{i=1}^{N_s}$ and an unlabeled target-domain dataset $\mathcal{D}_t = \{\boldsymbol{x}_t^{(j)}\}_{j=1}^{N_t}$, perturbation ball radius ϵ, ascent steps K and step size η_δ.

Initialize: Soft prompts embeddings $\boldsymbol{p}$ and domain discriminator θ_d, learning rates η_p, η_d.

repeat

  Sample a pair of batches, each of B data points, from $\mathcal{D}_s$ and $\mathcal{D}_t$;

  for $i = 0, ..., B$ **do**

    Forward computation: $f(\boldsymbol{x}_s^{(i)}, \boldsymbol{p}),\ \forall \boldsymbol{x}_s^{(i)}$

    Sample a $\boldsymbol{\delta}_0^{(i)} \sim \mathcal{U}(-1, 1), \forall \boldsymbol{x}_s^{(i)}$

    $\boldsymbol{\delta}_0^{(i)} \leftarrow \prod_{\|\cdot\|_2 \leq \epsilon}(\boldsymbol{\delta}_0^{(i)})$

    for $t = 0, ..., K-1$ **do**

      Forward with $\boldsymbol{\delta}_t^{(i)}$: $f(\boldsymbol{x}_s^{(i)} + \boldsymbol{\delta}_t, \boldsymbol{p})$

      Compute $\ell_{\text{KL}}(\boldsymbol{\delta}_t^{(i)}, \boldsymbol{p})$ (Eq. 3.3)

      Compute $\ell_{\text{adv}}(\boldsymbol{\delta}_t^{(i)})$ (Eq. 3.4)

      Perform PGA on $\boldsymbol{\delta}_t^{(i)}$:

        $\boldsymbol{g} \leftarrow \bigtriangledown_{\boldsymbol{\delta}_t^{(i)}}(\ell_{\text{KL}}(\boldsymbol{\delta}_t^{(i)}, \boldsymbol{p}) + \ell_{\text{disc}}(\boldsymbol{\delta}_t^{(i)}))$

        $\boldsymbol{\delta}_{t+1}^{(i)} \leftarrow \prod_{\|\cdot\|_2 \leq \epsilon}(\boldsymbol{\delta}_t^{(i)} + \eta_\delta \cdot \frac{\boldsymbol{g}}{\|\boldsymbol{g}\|_2})$

    end

  end

  Compute $\mathcal{L}_R$ (Eq. 3.10) and $\mathcal{L}_{\text{disc}}$ with $\boldsymbol{\delta}_{K-1}$

  $\boldsymbol{p} \leftarrow \boldsymbol{p} - \eta_p \bigtriangledown_{\boldsymbol{p}} \mathcal{L}_R(\boldsymbol{x}_s, y_s, \boldsymbol{p})$

  $\theta_d \leftarrow \theta_d - \eta_d \bigtriangledown_{\theta_d} \mathcal{L}_{\text{disc}}(\boldsymbol{\delta}_{K-1}, \boldsymbol{x}_s, \boldsymbol{x}_t; \theta_d)$

until the maximum training epoch is reached;

Output: Learned soft prompt $\boldsymbol{p}$

A critical difference between SSL and domain adaptation is that the unlabeled data are drawn from a different distribution (P_t) than the labeled data (P_s). As the two distributions may overlap in some regions and diverge in others, regularizing over the entire source dataset may be ineffective. Thus, we propose to focus the smoothness constraint on the regions of the data manifold where the source-domain and target-domain data are similar.

3.3 Experimental Evaluation

We evaluate the representations learned by OPTIMA under zero-shot and few-shot settings.

3.3.1 Datasets

We investigate domain adaptation on six text classification datasets in two tasks. In the task of paraphrase detection, we employ MRPC and QQP. In the task of natural language inference, we employ four datasets, including MNLI [Williams et al., 2018], SNLI [Bowman et al., 2015], CB [de Marneffe et al., 2019] and SICK [Marelli et al., 2014]. The statistics and the label space $\mathcal{Y}$ of each dataset can be found in Table 3.1.

Dataset	Train	Test	n_{class}	Verbalizers
MRPC	4,076	408	2	Yes/No
QQP	363,847	40,430	2	Yes/No
MNLI	392,702	9,815	3	Yes/Neutral/No
SNLI	549,367	9,842	3	Yes/Neutral/No
SICK	4,439	4,906	3	Yes/Neutral/No
CB	250	56	3	Yes/Neutral/No

TABLE 3.1: Dataset characteristics.

We prepare 8 groups of cross-domain experiments, two for paraphrase detection and 6 for natural language inference (NLI), as shown in Table 3.2.

Paraphrase	NLI from MNLI	NLI from SNLI
MRPC → QQP	MNLI → SNLI	SNLI → MNLI
QQP→ MRPC	MNLI → SICK	SNLI → SICK
	MNLI → CB	SNLI → CB

TABLE 3.2: The set of domain adaptation experiments.

3.3.2 Baselines

We include eight competitive single-domain and cross-domain baselines. Out of the eight, baselines #2-#4 do not use any transfer learning from the source domain. Baselines #5-#8 utilize transfer learning and data from the source domain.

1) Frozen PLM. Large PLMs have demonstrated non-trivial zero-shot performance [Brown et al., 2020]. Here, we directly apply T5-large [Raffel et al., 2020b] with the manually written hard prompt and take the verbalizer with the highest probability as the prediction.

2) Prompt Tuning (PT). We feed the input data with both soft and hard prompts to a frozen T5-large model and finetune the soft prompt embeddings on the few-shot training set from the target domain.

3) Fine Tuning (FT). We feed the input data with the hard prompt to T5-large and finetune the entire network on the few-shot target-domain data. Notice that we use the verbalizer rather than training a separate task-specific prediction head.

4) Prompt-based Fine Tuning (PFT). A representative method on exploiting soft prompts for fine-tuning, e.g., PERFECT [Rabeeh et al., 2022]. For a fair comparison, we wrap the input with both soft and hard prompts and finetune both the PLM and the soft prompts on target-domain data. The predictions are mapped via verbalizers.

5) Pre-trained Prompt Tuning (PPT). We follow Gu et al. [2022], who proposes to transfer to sentence-pair classification tasks by pretraining on the next sentence prediction task with 10GB text from OpenWebText [Gokaslan and Cohen, 2019]. We download the pretrained checkpoint and finetune the soft prompt on the target domain directly.

6) Soft Prompt Transfer (SPOT). Vu et al. [2022] propose to pretrain soft prompts on source-domain datasets and finetune the learned soft prompt on the target-domain datasets. We apply this approach to different source-target pairs.

7) Prompt Tuning with FreeLB. FreeLB [Zhu et al., 2020] is an adversarial training approach, which generates the adversarial perturbation from the supervised classification loss,

$$\boldsymbol{\delta}^* = \underset{\boldsymbol{\delta}, \|\boldsymbol{\delta}\| \leq \epsilon}{\arg\max}\, \ell_{\text{xe}}(\boldsymbol{x}_s + \boldsymbol{\delta}, y_s, \boldsymbol{p}). \tag{3.11}$$

After that, we find the optimal $\boldsymbol{p}$ by minimizing $\ell_{\text{xe}}(\boldsymbol{x}_s, y_s, \boldsymbol{p}) + \ell_{\text{xe}}(\boldsymbol{x}_s + \boldsymbol{\delta}, y_s, \boldsymbol{p})$. The adversarial training may be understood as another type of smoothness constraint, as the network attempts to maintain the same prediction despite the strongest possible perturbation.

8) Prompt Tuning with VAT. We apply the original VAT [Miyato et al., 2018] to generate the perturbations that maximally alter model predictions on the source domain,

$$\boldsymbol{\delta}^* = \underset{\boldsymbol{\delta}, \|\boldsymbol{\delta}\| \leq \epsilon}{\arg\max}\, \ell_{\text{KL}}(\boldsymbol{\delta}, \boldsymbol{p}, \boldsymbol{x}_s), \tag{3.12}$$

and optimize $\boldsymbol{p}$ as in Equation 3.10. This can be seen as an ablation of OPTIMA, as Equation 3.12 omits the ℓ_{adv} term from Equation 3.6.

3.3.3 Experiment Settings

Pretraining. For all methods that utilize source domain data, we train the soft prompts using the whole source-domain training set and perform model selection using the source-domain validation set. When domain adaptation is applied, we additionally use the entire target-domain training set for training with all labels removed. To mitigate variance, we train each method using 3 different random seeds, yielding three different models. For zero-shot evaluation, we report the mean score of the three models.

Few-shot Evaluation Protocol. In PET [Schick and Schütze, 2021b], authors evaluated its few-shot performance using a fixed training set. In LM-BFF [Gao et al., 2021], authors conducted more studies on the configuration of few-shot settings and proposed to average 5 randomly sampled few-shot sets. We determine the sample size, 16, based on a statistical analysis[1] on the sample size required for investigating an unknown population mean under student t-test. Here we adopt a significance level $\alpha = 0.05$, the risk of rejecting a true hypothesis that the performance of one method is better than the other. Following Gao et al. [2021], we sample the few-shot training set and validation set from the original target training set. Each set contains 8 data points per class. We evaluate the trained model on the original target validation set. To mitigate the high variance of few-shot learning, we repeat the sampling 16 times and report the average of 48 runs (16 samples $\times$ 3 models).

For all the cross-domain few-shot learning methods, the few-shot test performance of 3 differently pre-trained soft prompts are averaged for each given $\mathcal{D}_{train}$ and $\mathcal{D}_{dev}$ splits, and we obtain 16 averaged few-shot performance. Then we compute the mean and standard deviation for the 16 test results.

Model Settings. For all the experiments, unless specified, we use the LM-adapted version of T5-large as the PLM. Results in Lester et al. [2021] (Figure 3) shows that T5 further trained for LM Adaptation works the best for prompt tuning, which is also adopted by Gu et al. [2022] and Vu et al. [2022]. For the domain discriminator, we use a linear

[1] `https://www.itl.nist.gov/div898/handbook/prc/section2/prc222.htm`

classification layer with parameters $\theta_d = [\boldsymbol{w}, \boldsymbol{b}], \boldsymbol{w} \in \mathbb{R}^{1024\times 2}, \boldsymbol{b} \in \mathbb{R}^2$, where 1024 is the dimension of the output hidden states from the decoder of T5-large model.

Training Settings. Following [Lester et al., 2021], we use Adafactor [Shazeer and Stern, 2018] as the optimizer and set the learning rate to 0.3 for all the pre-training tasks on the entire source domain dataset. We use the cosine learning rate scheduler for all methods. For the pre-training stage, we set the maximum number of training steps to $30,000$ and evaluate the models on the validation set every $1,000$ steps. We set the batch size to 8 for MRPC and QQP, and 18 for the NLI datasets. For the few-shot learning setting, we set the maximum number of training steps to $1,000$ and evaluate models on $|\mathcal{D}_{dev}|$ every 4 steps. we set batch size to 4 for MRPC and QQP, and 6 for the NLI datasets. All the training are done on NVIDIA V-100 with 32 GB.

Soft and Hard Prompts. Following Lester et al. [2021] and Gu et al. [2022], for all methods other than PPT, we set the soft prompt length to 100, initialized to the first 100 token embeddings of T5 in alphabetic order. We combine soft prompts with hard prompts with details in the Appendix.

	Hybrid Template
T1	$\boldsymbol{P} < S_1 >$ and $< S_2 >$ are equivalent? [MASK]
T2	$\boldsymbol{P}$ hypothesis: $< S_1 >$ premise: $< S_2 >$ answer: [MASK]

TABLE 3.3: The hybrid templates where $\boldsymbol{P}$ represents learnable soft prompts. $< S_1 >$ and $< S_2 >$ are sentence pairs. [MASK] represents the labels to be predicted. T1 is the template adopted by the paraphrase detection and question pair classification tasks. T2 is the template adopted by four natural language inference tasks.

Evaluation Metrics. Following Lester et al. [2021], we use accuracy and F1 score to evaluate the performance on the MRPC and QQP datasets. Following Gu et al. [2022], we use accuracy for NLI. For zero-shot model selection, we use the source-domain validation set. For few-shot model selection, we use the target-domain validation set.

3.3.4 Few-shot Performance

We adopt few-shot classification to evaluate the representations learned by different models and pretraining methods. We show the few-shot performance in Table 3.4 and make the following observations. First, OPTIMA significantly outperforms all baseline models across all the few-shot test cases, including the state-of-the-art SPOT baseline.

Method	Params	PLM	Source	QQP Acc.	QQP F1	MRPC Acc.	MRPC F1	MNLI Acc.
Frozen	0		✗	45.5	54.9	33.8	11.8	41.7
PT	102K		✗	48.4 ± 4.9	52.5 ± 5.5	53.1 ± 11.4	55.9 ± 23.4	33.4 ± 1.6
FT	770M	T5-Large	✗	55.1 ± 6.7	52.0 ± 6.0	59.5 ± 7.8	67.9 ± 12.6	35.6 ± 2.4
PFT	770M		✗	55.1 ± 5.1	57.8 ± 3.1	58.9 ± 11.0	65.3 ± 11.8	35.6 ± 3.6
PPT	410K	T5-XXL	✓	52.1 ± 11.1	56.2 ± 21.1	52.1 ± 11.1	56.2 ± 21.1	34.4 ± 1.4
				MRPC → QQP Acc.	MRPC → QQP F1	QQP → MRPC Acc.	QQP → MRPC F1	SNLI → MNLI Acc.
SPOT	102K	T5-Large	✓	64.5 ± 2.7	64.5 ± 0.8	68.7 ± 2.5	77.1 ± 2.9	74.3 ± 0.9
FreeLB	102K		✓	65.0 ± 2.4	64.5 ± 1.5	68.5 ± 2.2	77.6 ± 2.2	75.0 ± 1.0
VAT	102K		✓	66.2 ± 2.0	64.9 ± 0.7	69.6 ± 1.9	79.0 ± 2.1	74.9 ± 1.1
DANN	102K		✓	63.4 ± 2.5	62.5 ± 2.7	68.0 ± 3.5	76.2 ± 5.1	73.1 ± 1.4
OPTIMA	102K		✓	**69.1*** ± 1.7	**65.8*** ± 1.9	**71.2*** ± 1.7	**79.9*** ± 1.7	**78.4*** ± 0.6

Method	Params	PLM	Source	SNLI Acc.	SICK Acc.	CB Acc.
Frozen	0		✗	35.9	37.1	55.4
PT	102K		✗	34.6 ± 2.4	61.5 ± 7.8	38.3 ± 13.6
FT	770M	T5-Large	✗	41.6 ± 3.8	67.6 ± 6.3	51.2 ± 7.8
PFT	770M		✗	38.6 ± 5.1	71.3 ± 6.4	57.3 ± 9.2
PPT	410K	T5-XXL	✓	34.7 ± 2.8	54.6 ± 14.0	43.0 ± 14.6

				MNLI → SNLI Acc.	SNLI → SICK Acc.	MNLI → SICK Acc.	SNLI → CB Acc.	MNLI → CB Acc.
SPOT	102K	T5-Large	✓	78.8 ± 1.1	69.9 ± 5.3	72.9 ± 5.9	61.7 ± 5.0	65.3 ± 3.4
FreeLB	102K		✓	81.5 ± 0.7	69.5 ± 6.8	73.1 ± 4.8	61.6 ± 4.2	66.1 ± 3.3
VAT	102K		✓	80.9 ± 0.9	68.6 ± 6.4	72.7 ± 6.3	59.0 ± 5.5	68.7 ± 4.8
DANN	102K		✓	71.1 ± 3.2	69.0 ± 6.7	73.4 ± 3.7	55.7 ± 5.5	66.9 ± 4.6
OPTIMA	102K		✓	**82.1*** ± 0.8	**73.3** ± 6.8	**74.8** ± 4.4	**64.8*** ± 1.1	**71.2*** ± 3.1

TABLE 3.4: Few-shot test performance. Results in bold are the best and results underlined are the best in the single-domain group. Results marked with * are significantly better than all the others under the student t-test ($p < 0.05$).

We perform statistical significance tests that compare OPTIMA to all baselines in a pairwise manner. In all but the SICK experiments, the differences between OPTIMA and all baselines are statistically significant. We attribute the performance to the high-quality representation of OPTIMA, resulting from domain adaptation.

Second, DANN performs much worse than perturbation-based methods. As discussed earlier, we suspect the poor performance of DANN is partially due to the limited capacity of prompts (102K parameters in our case). In OPTIMA, the perturbation optimizes for domain invariance (Eq. 3.6), whereas the prompt optimizes for only task-specific losses (Eq. 3.10), which simplifies optimization for soft prompts.

Third, OPTIMA outperforms the VAT baseline, especially in the NLI tasks, where the performance difference ranges from 1.2% in MNLI→SNLI to 5.8% in SNLI→CB. The VAT baseline is an ablation of OPTIMA and omits the targeted regularization term when finding the perturbation. This comparison demonstrates the effectiveness of the proposed targeted smoothness constraint.

Finally, our experiments are consistent with earlier results of Gu et al. [2022], which show that prompt tuning (PT) suffers from high variance in the results. In the single-domain experiments, finetuning the entire T5-Large (FT) exhibits comparable, if not lower, variances than PT, even though FT updates about 7500× more parameters. This underscores the importance of using pretrained prompts from a source domain. Indeed, all transfer learning methods utilizing a source domain similar to the target (SPOT, FreeLB, VAT, and OPTIMA) yield sizable performance gains than single-domain methods. Notably, FreeLB, VAT and OPTIMA are obviously better than SPOT across the benchmarks, which underscores the importance of alleviating overfitting to source-domain datasets.

Sample Efficiency. We perform an additional experiment where we increase the number of available samples per class from the target domain, and show the results in Figure 3.3. We observe that 4-shot OPTIMA achieves comparable performance as full-model finetuning on 128-shot dataset. Similarly, 8-shot OPTIMA achieves an accuracy comparable to 64-shot SPOT. These results clearly demonstrate the superior sample efficiency of OPTIMA.

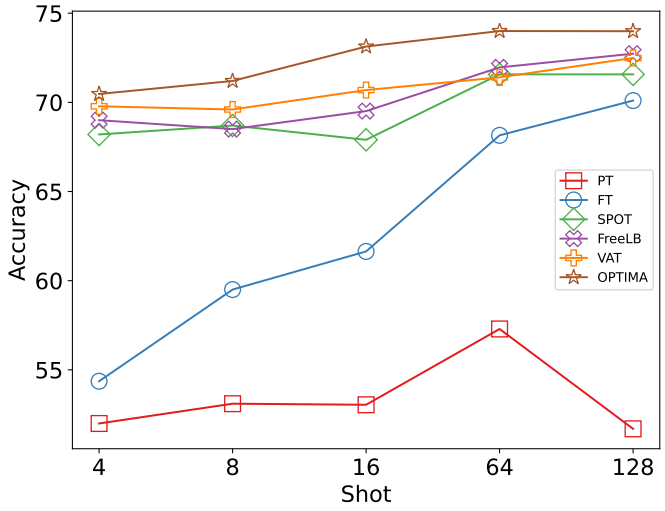


FIGURE 3.3: Average test performance on the QQP-to-MRPC test case. PT and FT are trained on MRPC directly.

3.3.5 Zero-shot Performance

Zero-shot performance on the target domain is also an effective way to evaluate the learned representations. We show the zero-shot performance in Table 3.5 and make the

following observations.

Method	MRPC Acc.	MRPC → QQP Acc.	MRPC → QQP F1	QQP Acc.	QQP → MRPC Acc.	QQP → MRPC F1	MNLI → CB Acc.
SPOT	82.5 ± 1.5	60.9 ± 4.6	63.6 ± 2.0	80.9 ± 2.2	65.7 ± 3.4	73.2 ± 5.7	63.2 ± 5.7
FreeLB	85.5 ± 0.3	63.1 ± 3.7	63.9 ± 1.0	82.2 ± 2.7	69.4 ± 1.1	78.7 ± 1.3	67.8 ± 3.9
VAT	84.7 ± 0.8	64.8 ± 4.6	64.1 ± 1.7	81.9 ± 0.7	68.9 ± 1.5	78.5 ± 1.5	67.8 ± 5.8
OPTIMA	**85.7** ± 0.7	**68.9** ± 0.8	**66.3** ± 0.6	**82.7** ± 1.3	**71.2** ± 0.4	**80.0** ± 0.6	**68.3** ± 2.6
Method	MNLI Acc.	MNLI → SNLI Acc.	MNLI → SICK Acc.	SNLI Acc.	SNLI → MNLI Acc.	SNLI → SICK Acc.	SNLI → CB Acc.
SPOT	83.4 ± 0.8	79.2 ± 1.0	51.8 ± 0.7	88.9 ± 0.1	75.6 ± 0.4	52.7 ± 1.9	47.6 ± 3.7
FreeLB	**84.8** ± 0.8	81.8 ± 0.7	52.2 ± 0.2	**89.9** ± 0.1	77.5 ± 0.5	52.9 ± 1.9	47.5 ± 4.7
VAT	83.7 ± 0.3	81.0 ± 0.2	51.4 ± 1.4	88.7 ± 0.1	77.1 ± 1.3	51.8 ± 2.1	45.8 ± 0.8
OPTIMA	84.6 ± 0.3	**82.1** ± 0.8	**53.2** ± 1.1	89.2 ± 0.1	**79.1** ± 0.1	**53.1** ± 1.0	**49.4** ± 4.2

TABLE 3.5: Source-domain and zero-shot target-domain test performance.

First, OPTIMA still takes the highest spot in performance in all target domains, outperforming the second best baseline by up to 4.1%. In the source domain, OPTIMA is comparable with the baselines. Second, the ablation baseline, VAT, is consistently surpassed by OPTIMA, which again confirms the utility of our proposal. Third, the state-of-the-art method, SPOT, in the majority of cases produces results with higher variance than the three perturbation-based methods. This suggests that adversarial perturbation is effective against overfitting. Lastly, except in the MNLI → SICK task, DANN performs rather poorly across the benchmarks, indicating that DANN is not suitable for prompt tuning.

3.3.6 Class Similarity and Transfer Learning

We investigate the relationship between domain similarity and transfer learning performance using CB as the target domain as an example. CB is a difficult target. On SNLI, all models in Table 3.5 achieve in-domain test accuracies greater than 88%, but few-shot SNLI-to-CB transfer obtains accuracies around 61%. This is disappointing given that even Frozen achieves 55.4% on CB.

To investigate the underlying cause, we plot the TF-IDF textual similarities between different domains of NLI in Figure 3.4 and 3.5, and between MRPC and QQP in Figure 3.6. We compare SPOT, which performs direct transfer without any smoothness regularization, and OPTIMA in the form of confusion matrices in Figure 3.7 and F1 scores in Figure 3.8 and 3.9.

Figure 3.4(a) shows irregular similarities between classes of SNLI and CB, which explains the difficulty in transfer learning. For example, the SNLI Neutral class is more

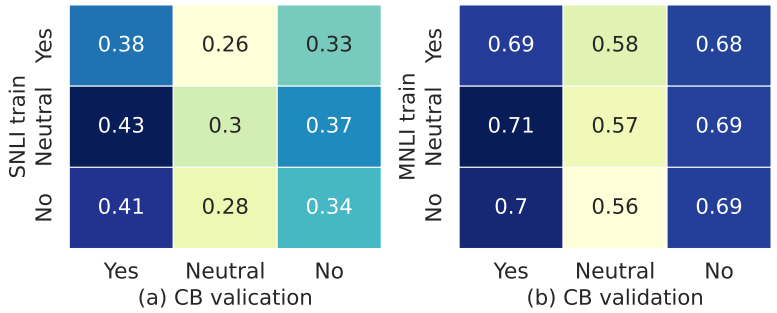


FIGURE 3.4: TF-IDF similarity for SNLI, MNLI, and CB, where we treat all text in one class as a document.

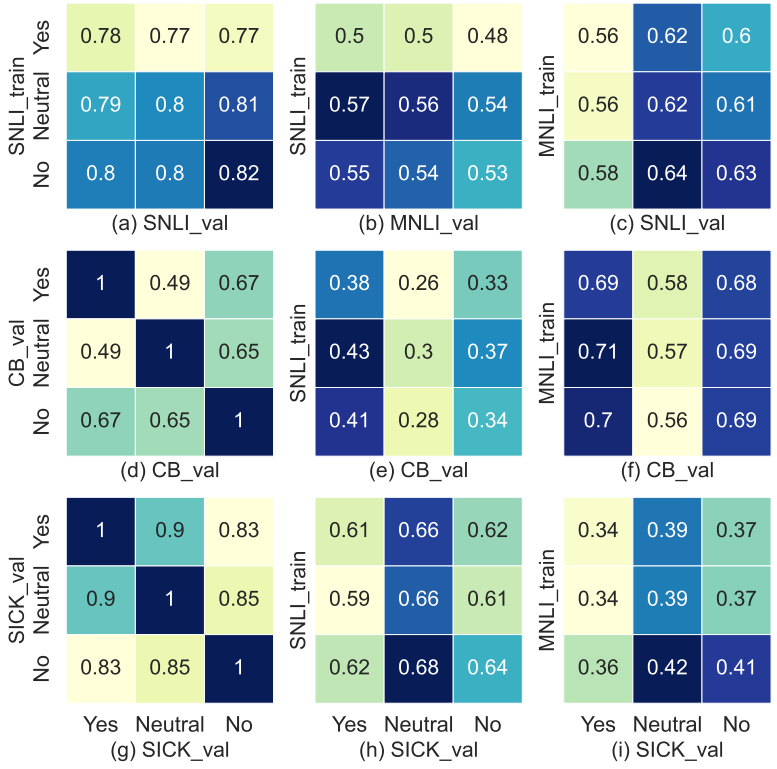


FIGURE 3.5: Document similarity using TF-IDF for each pair of NLI datasets.

similar to the CB Yes class than the CB Neural class. The CB Neutral class has low similarity to all SNLI classes. This leads to significant confusion for the few-shot SPOT classifier in the SNLI-to-CB transfer and especially low accuracy for the CB Neutral class (Figure 3.8). The situation is similar for the MNLI-to-CB transfer. Interestingly, the regularization of OPTIMA is able to alleviate the domain shift and obtain accuracy improvements for the CB No and Neutral classes.

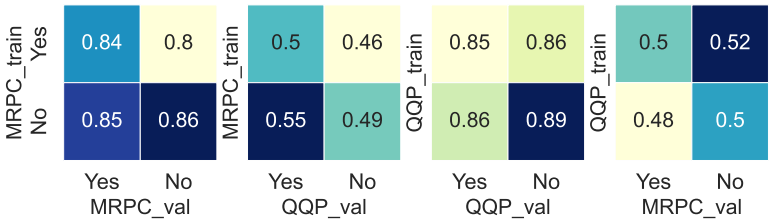


FIGURE 3.6: Document similarity for MRPC and QQP datasets between their classes.

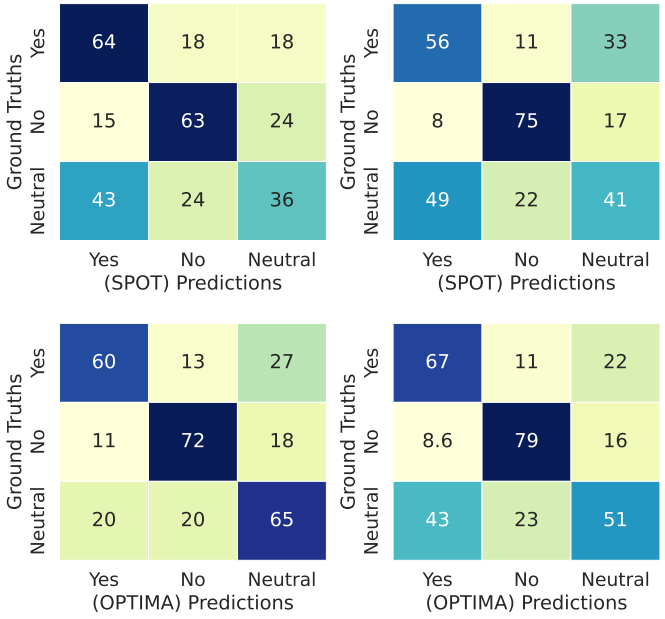


FIGURE 3.7: Confusion matrices for 8-shot transfer learning to CB. Each result is the average across 48 runs.

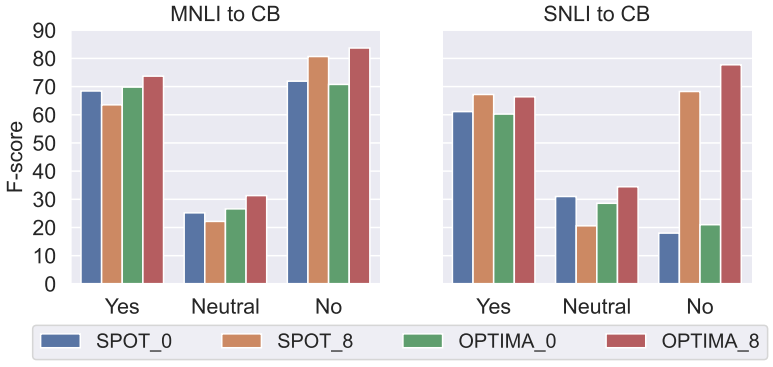


FIGURE 3.8: F1-score on the three classes of the CB datasets. SPOT_0 and OPTIMA_0 denote zero-shot performance. SPOT_8 and OPTIMA_8 denote 8-shot performance.

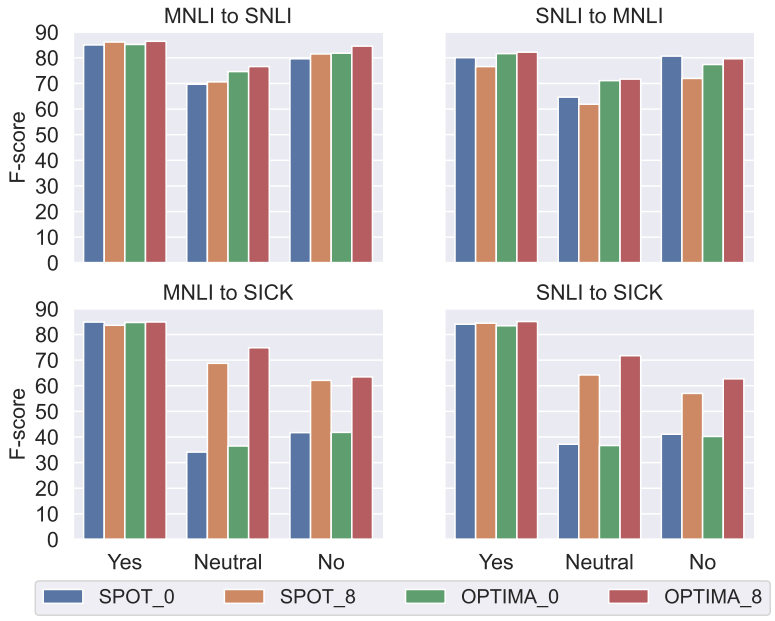


FIGURE 3.9: F-score on three classes for NLI datasets. SPOT_0 and OPTIMA_0 are compared for their zero-shot performance. SPOT_8 and OPTIMA_8 are compared for their 8-shot performance.

3.4 Summary

In this chapter, we propose OPTIMA to enhance soft prompt transfer performance by regularizing the training on the source domain under perturbations generated with domain adaptation. Extensive experiments demonstrate that OPTIMA significantly enhances the transferability and sample efficiency of prompt tuning. Compared to competitive baselines, soft prompt trained with OPTIMA generalizes better to the source domain and significantly boosts zero-shot and few-shot learning in the target domain. We observe that pre-training soft prompts on a similar dataset confer more benefits than pre-training on a dissimilar dataset. We expect the current work to contribute to the wide deployment of PLMs.

We identify a few limitations of the current work.

- The domain adaptation problem formulation requires unlabeled data from the target domain. Although unlabeled data are easy to obtain in most cases, doing so might be difficult for some data-scarce domains.
- The proposed regularization technique addresses the situation where the source and target domains have different data distributions. When the two distributions

are exactly the same, the technique degenerates to simply adversarial training. When the two distributions are extremely dissimilar, the transfer is unlikely to yield performance improvements. A unified framework that automatically detects domain distances and applies the correct method may be desirable.

- The power of perturbations has the most effect in the few-shot / zero-shot settings. When the target domain has abundant labeled data, the gap between soft prompt tuning and our method will diminish.

Chapter 4

Optimizing Domain Adversarial Training for Data-scarce Domains

4.1 Motivation

Sarcastic language is commonly found in social media posts [González-Ibáñez et al., 2011, Maynard and Greenwood, 2014], forum discussions [Khodak et al., 2018], product reviews [Davidov et al., 2010, Filatova, 2012] and everyday conversations [Gibbs, 2000]. Detecting sarcasm is an integral part of creative language understanding [Veale et al., 2019] and online opinion mining [Kannangara, 2018]. Due to highly contextualized expressions, detecting sarcasm is a challenging task, even for humans [Fox Tree et al., 2020].

A challenge specific to sarcasm detection is the difficulty in acquiring ground-truth annotations. Human-annotated datasets [Filatova, 2012, Riloff et al., 2013, Van Hee et al., 2018, Oprea and Magdy, 2020] usually contain only a few thousand texts, resulting in many small datasets. In comparison, automatic data collection using distant supervision signals like hashtags [Ptáček et al., 2014, Bamman and Smith, 2015, Joshi et al., 2015] yielded substantially larger datasets. Nevertheless, the automatic approach also led to label noise. For example, Oprea and Magdy [2020] found nearly half of the tweets with sarcasm hashtags in one dataset are not sarcastic. The existence of diverse datasets and data collection methods prompts us to exploit their commonality through transfer learning. Specifically, we transfer knowledge learned from large and noisy datasets to

improve sarcasm detection on small human-annotated datasets that serve as effective performance benchmarks.

Adversarial neural transfer (ANT) [Ganin and Lempitsky, 2015, Liu et al., 2017, Kim et al., 2017, Kamath et al., 2019] employs an adversarial setup where the network learns to make the shared feature distributions of the source domain and the target domain as similar as possible, while simultaneously optimizing for domain-specific performance. However, as the domain-specific losses promote the use of domain-specific features, these training objectives may compete with each other implicitly. This leads to optimization difficulties and potentially degenerate cases where the domain-specific classifiers ignore the shared features and no meaningful transfer occurs between domains.

To deal with this, we propose Latent-Optimized Adversarial Neural Transfer (LOANT). The latent optimization strategy can be understood with analogies to to one-step look-ahead during gradient descent and Model-Agnostic Meta Learning [Finn et al., 2017]. By forcing domain-specific losses to accommodate the negative domain discrimination loss, it improves training dynamics [Balduzzi et al., 2018].

4.2 The LOANT Method

In supervised transfer learning, we assume labeled data for both the source domain and the target domain are available. The source domain dataset D_s comprises of data points in the format of (x_s, y_s) and the target domain dataset D_t comprises of data points in the format of (x_t, y_t). The labels y_s and y_t are one-hot vectors. The task of supervised cross-domain sarcasm detection can be formulated as learning a target-domain function $f_t(x_t)$ that predict correct labels for unseen x_t.

4.2.1 Model Architecture

Fig. 4.1 shows the model architecture for adversarial neural transfer (ANT) [Liu et al., 2017, Kim et al., 2017, Kamath et al., 2019]. We use a large pretrained neural network, BERT [Devlin et al., 2019], as the sentence encoder, though the architecture is not tied to BERT and can use other pretrained encoders. We denote the parameters of the BERT encoder as w_b, and its output for data in the source domain and the target domain as

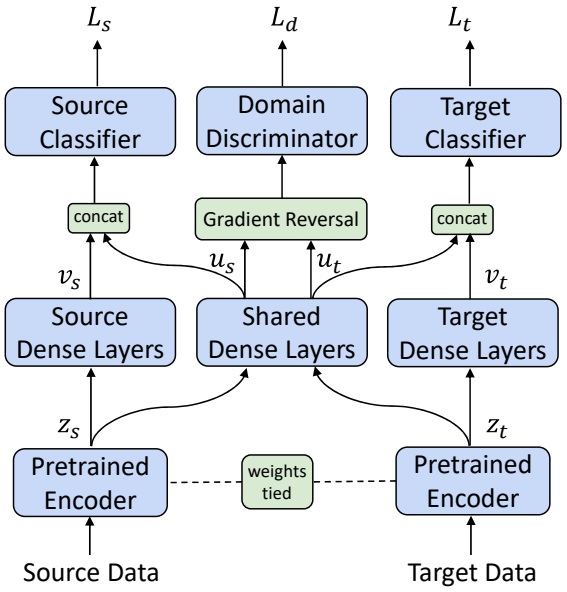


FIGURE 4.1: Network architecture of the Adversarial Neural Transfer model.

$z_s \in \mathbb{R}^D$ and $z_t \in \mathbb{R}^D$ respectively. We denote this encoder operation as

$$z_s = E(x_s, w_b),\ z_t = E(x_t, w_b) \tag{4.1}$$

On top of these outputs, we apply domain-specific dense layers to create domain-specific features v_s, v_t and shared dense layers to create shared features u_s, u_t. We use w_s, w_t, and w_{sh} to denote the parameters for the source dense layers, the target dense layers, and the shared dense layers.

The concatenation of features $[v_s, u_s]$ is fed to the source-domain classifier, parameterized by θ_s; $[v_t, u_t]$ is fed to the target-domain classifier, parameterized by θ_t. The two classifiers categorize the tweets into sarcastic and non-sarcastic and are trained using cross-entropy. For reasons that will become apparent later, we make explicit the reliance on z_s and z_t:

$$\begin{aligned} \mathcal{L}_s(z_s) &= -\sum_i y_{s,i} \log p(\hat{y}_{s,i}|z_s), \\ \mathcal{L}_t(z_t) &= -\sum_i y_{t,i} \log p(\hat{y}_{t,i}|z_t), \end{aligned} \tag{4.2}$$

where $\hat{y}_s$ and $\hat{y}_t$ are the predicted labels and i is the index of the vector components.

Simultaneously, the domain discriminator learns to distinguish the features u_s and u_t as coming from different domains. The domain discriminator is parameterized by θ_d. It is

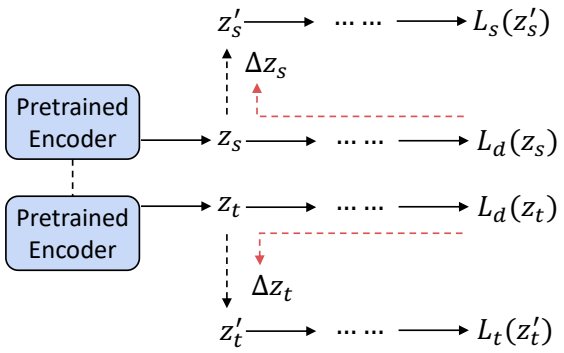


FIGURE 4.2: Schematic of the latent optimization strategy. The solid black arrows indicate the forward pass and the dotted red arrows indicate the backward pass.

trained to minimize the domain classification loss,

$$\mathcal{L}_d(z_t, z_s) = -\log p(0|u_s) - \log p(1|u_t). \tag{4.3}$$

Through the use of the gradient reversal layer, the shared dense layers and the feature encoder maximizes the domain classification loss, so that the shared features u_s and u_t become indistinguishable and conducive to transfer learning. In summary, the network weights $w_b, w_s, w_t, w_{sh}, \theta_s, \theta_t$ are trained to minimize the following joint loss,

$$\mathcal{L}^{\text{ANT}} = \mathcal{L}_s(z_s) + \mathcal{L}_t(z_t) - \mathcal{L}_d(z_t, z_s), \tag{4.4}$$

whereas θ_d is trained to minimize $\mathcal{L}_d(z_t, z_s)$.

It is worth noting that the effects of three loss terms in Eq. 4.4 on the shared parameters w_{sh} and w_b may be competing with each other. This is because optimizing sarcasm detection in one domain will encourage the network to extract domain-specific features, whereas the domain discrimination loss constrains the network to avoid such features. It is possible for the competition to result in degenerate scenarios. For example, the shared features u_s and u_t may become indistinguishable but also do not correlate with the labels y_s and y_t. The domain classifiers may ignore the shared features u_s and u_t and hence no transfer happens. To cope with this issue, we introduce a latent optimization strategy that forces domain-specific losses to accommodate the domain discrimination loss.

4.2.2 Latent Representation Optimization

We now introduce the latent representation optimization strategy. First, we perform one step of stochastic gradient descent on $-\mathcal{L}_d$ on the encoded features z_s and z_t with learning rate γ,

$$z'_s = z_s + \gamma \frac{\partial \mathcal{L}_d(z_s, z_t)}{\partial z_s}, \tag{4.5}$$

$$z'_t = z_t + \gamma \frac{\partial \mathcal{L}_d(z_s, z_t)}{\partial z_t}. \tag{4.6}$$

We emphasize that this is a *descent* step because we are minimizing $-\mathcal{L}_d$.

After that, we use the updated z'_s and z'_t in the computation of the losses

$$\mathcal{L}_s^{\text{LO}}(z_s, z'_s) = \mathcal{L}_s(z_s) + \mathcal{L}_s(z'_s), \tag{4.7}$$

$$\mathcal{L}_t^{\text{LO}}(z_t, z'_t) = \mathcal{L}_t(z_t) + \mathcal{L}_t(z'_t). \tag{4.8}$$

The new joint objective hence becomes

$$\begin{aligned} \mathcal{L}^{\text{LO}} = \mathcal{L}_s^{\text{LO}}(z_s, z'_s) + \mathcal{L}_t^{\text{LO}}(z_t, z'_t) \\ - \mathcal{L}_d(z_s, z_t), \end{aligned} \tag{4.9}$$

which is optimized using regular stochastic gradient descent (SGD) on $w_b, w_s, w_t, w_{sh}, \theta_s$, and θ_t.

Here we show the general case of gradient computation. Consider any weight vector w in the neural network. Equations 4.5 and 4.6 introduce two intermediate variables z'_s and z'_t, which are a function of the model parameter w. Therefore, we perform SGD using the following total derivative

$$\begin{aligned} \frac{\mathrm{d}\mathcal{L}^{\text{LO}}}{\mathrm{d}w} = \frac{\partial \mathcal{L}^{\text{LO}}}{\partial w} + \frac{\partial \mathcal{L}_s^{\text{LO}}(z'_s)}{\partial z'_s} \frac{\partial z'_s}{\partial w} \\ + \frac{\partial \mathcal{L}_t^{\text{LO}}(z'_t)}{\partial z'_t} \frac{\partial z'_t}{\partial w}. \end{aligned} \tag{4.10}$$

where

$$\begin{aligned}\frac{\partial z'_s}{\partial w} &= \frac{\partial z_s}{\partial w} + \gamma \frac{\partial^2 \mathcal{L}_d(z_s)}{\partial z_s \, \partial w} \\ \frac{\partial z'_t}{\partial w} &= \frac{\partial z_t}{\partial w} + \gamma \frac{\partial^2 \mathcal{L}_d(z_t)}{\partial z_t \, \partial w}\end{aligned} \tag{4.11}$$

For every network parameter other than the encoder weight w_b, $\partial z / \partial w$ is zero. The second-order derivative $\partial^2 \mathcal{L}_d(z) / \partial z \, \partial w$ is difficult to compute due to the high dimensionality of w. Since γ is usually very small, we adopt a first-order approximation and directly set the second-order derivative to zero. Letting $\phi_s = [w_s, \theta_s]$ and $\phi_t = [w_t, \theta_t]$, we now show the total derivatives for all network parameters:

$$\begin{aligned}\frac{\mathrm{d}\mathcal{L}^{\mathrm{LO}}}{\mathrm{d}w_b} &= \frac{\partial \mathcal{L}^{\mathrm{ANT}}}{\partial w_b} + \frac{\partial \mathcal{L}_s(z'_s)}{\partial w_b} + \frac{\partial \mathcal{L}_t(z'_t)}{\partial w_b} \\ &\quad + \frac{\partial \mathcal{L}_s(z'_s)}{\partial z'_s} \frac{\partial z_s}{\partial w_b} + \frac{\partial \mathcal{L}_t(z'_t)}{\partial z'_t} \frac{\partial z_t}{\partial w_b}\end{aligned} \tag{4.12}$$

$$\frac{\mathrm{d}\mathcal{L}^{\mathrm{LO}}}{\mathrm{d}w_{sh}} = \frac{\partial \mathcal{L}^{\mathrm{ANT}}}{\partial w_{sh}} + \frac{\partial \mathcal{L}_s(z'_s)}{\partial w_{sh}} + \frac{\partial \mathcal{L}_t(z'_t)}{\partial w_{sh}} \tag{4.13}$$

$$\begin{aligned}\frac{\mathrm{d}\mathcal{L}^{\mathrm{LO}}}{\mathrm{d}\phi_s} &= \frac{\partial \mathcal{L}_s(z_s)}{\partial \phi_s} + \frac{\partial \mathcal{L}_s(z'_s)}{\partial \phi_s} \\ \frac{\mathrm{d}\mathcal{L}^{\mathrm{LO}}}{\mathrm{d}\phi_t} &= \frac{\partial \mathcal{L}_t(z_t)}{\partial \phi_t} + \frac{\partial \mathcal{L}_t(z'_t)}{\partial \phi_t} \\ \frac{\mathrm{d}\mathcal{L}^{\mathrm{LO}}}{\mathrm{d}\theta_d} &= \frac{\partial \mathcal{L}_d(z_s, z_t)}{\partial \theta_d}\end{aligned} \tag{4.14}$$

First-order Approximation. We explain the gradients for the model parameters $w_b, w_{sh}, \phi_s, \phi_t$ and θ_d. Generically, we apply the first-order approximation by substituting Eq.4.11 into Eq. 4.10 and setting the Hessian to zero, which gives

$$\begin{aligned}\frac{\mathrm{d}\mathcal{L}^{\mathrm{LO}}}{\mathrm{d}w} &= \frac{\partial \mathcal{L}^{\mathrm{LO}}}{\partial w} + \frac{\partial \mathcal{L}_s^{\mathrm{LO}}(z'_s)}{\partial z'_s} \frac{\partial z_s}{\partial w} \\ &\quad + \frac{\partial \mathcal{L}_t^{\mathrm{LO}}(z'_t)}{\partial z'_t} \frac{\partial z_t}{\partial w}.\end{aligned} \tag{4.15}$$

Note that z_s and z_t depend on only the parameter w_b. For the rest of the parameters, w_{sh}, ϕ_s, ϕ_t and θ_d, the partial derivatives $\frac{\partial z_s}{\partial w}$ and $\frac{\partial z_t}{\partial w}$ are zero.

Now we consider the joint objective (Eq. 4.9), which contains domain-specific classification losses produced by both the old latent vector z and the new latent vector z'. Thus,

we derive at the generic formula

$$\begin{aligned}\frac{\partial \mathcal{L}^{\text{LO}}}{\partial w} &= \frac{\partial \mathcal{L}_s^{\text{LO}}}{\partial w} + \frac{\partial \mathcal{L}_t^{\text{LO}}}{\partial w} - \frac{\partial \mathcal{L}_d}{\partial w} \\ &= \frac{\partial \mathcal{L}_s(z_s)}{\partial w} + \frac{\partial \mathcal{L}_t(z_t)}{\partial w} - \frac{\partial \mathcal{L}_d(z_s, z_t)}{\partial w} \\ &\quad + \frac{\partial \mathcal{L}_s(z'_s)}{\partial w} + \frac{\partial \mathcal{L}_t(z'_t)}{\partial w}\end{aligned} \tag{4.16}$$

By the same reasoning above, the total derivative of $\mathcal{L}^{\text{LO}}$ against w_b is

$$\begin{aligned}\frac{\mathrm{d}\mathcal{L}^{\text{LO}}}{\mathrm{d}w_b} &= \frac{\partial \mathcal{L}^{\text{LO}}}{\partial w_b} + \frac{\partial \mathcal{L}_s^{\text{LO}}(z'_s)}{\partial z'_s}\frac{\partial z_s}{\partial w_b} \\ &\quad + \frac{\partial \mathcal{L}_t^{\text{LO}}(z'_t)}{\partial z'_t}\frac{\partial z_t}{\partial w_b}\end{aligned} \tag{4.17}$$

$$\begin{aligned}\frac{\partial \mathcal{L}^{\text{LO}}}{\partial w_b} &= \frac{\partial \mathcal{L}_s(z_s)}{\partial w_b} + \frac{\partial \mathcal{L}_t(z_t)}{\partial w_b} - \frac{\partial \mathcal{L}_d(z_s, z_t)}{\partial w_b} \\ &\quad + \frac{\partial \mathcal{L}_s(z'_s)}{\partial w_b} + \frac{\partial \mathcal{L}_t(z'_t)}{\partial w_b}\end{aligned} \tag{4.18}$$

For the rest of the parameters, the computation is slightly different as they do not contribute to z_s and z_t.

$$\begin{aligned}\frac{\partial \mathcal{L}^{\text{LO}}}{\partial w_{sh}} &= \frac{\partial \mathcal{L}_s(z_s)}{\partial w_{sh}} + \frac{\partial \mathcal{L}_t(z_t)}{\partial w_{sh}} - \frac{\partial \mathcal{L}_d(z_s, z_t)}{\partial w_{sh}} \\ &\quad + \frac{\partial \mathcal{L}_s(z'_s)}{\partial w_{sh}} + \frac{\partial \mathcal{L}_t(z'_t)}{\partial w_{sh}}\end{aligned} \tag{4.19}$$

$$\frac{\partial \mathcal{L}^{\text{LO}}}{\partial \phi_s} = \frac{\partial \mathcal{L}_s(z_s)}{\partial \phi_s} + \frac{\partial \mathcal{L}_s(z'_s)}{\partial \phi_s} \tag{4.20}$$

$$\frac{\partial \mathcal{L}^{\text{LO}}}{\partial \phi_t} = \frac{\partial \mathcal{L}_t(z_t)}{\partial \phi_t} + \frac{\partial \mathcal{L}_t(z'_t)}{\partial \phi_t} \tag{4.21}$$

The parameter of the domain discriminator θ_d is updated to minimize $\mathcal{L}_d(z_s, z_t)$. This is in contrast to the rest of the model, which minimizes $-\mathcal{L}_d(z_s, z_t)$. The update rule for θ_d is

$$\theta_d \leftarrow \theta_d - \eta \frac{\partial \mathcal{L}_d(z_s, z_t)}{\partial \theta_d} \tag{4.22}$$

Fig. 4.2 illustrates the latent optimization process. Algorithm 2 shows the LOANT algorithm.

Algorithm 2: Training of LOANT

Input: source data (x_s, y_s), target data (x_t, y_t), learning rate γ
Initialize model parameters w
repeat
  Sample N batches of data pairs
  for i = 1 **to** N **do**
    Compute forward loss $\mathcal{L}_s$, $\mathcal{L}_t$, $\mathcal{L}_d$;
    Compute $\triangle z_s = \frac{\partial \mathcal{L}_d(z_s)}{\partial z_s}$ and $\triangle z_t = \frac{\partial \mathcal{L}_d(z_t)}{\partial z_t}$;
    Update the latent representations $z'_s = z_s + \gamma \triangle z_s$ and $z'_t = z_t + \gamma \triangle z_t$;
    Compute the new joint loss $\mathcal{L}^{\text{LO}} = \mathcal{L}_s^{\text{LO}} + \mathcal{L}_t^{\text{LO}} - \mathcal{L}_d$;
    Update w using gradient descent.
until the maximum training epoch

4.2.3 Understanding LOANT

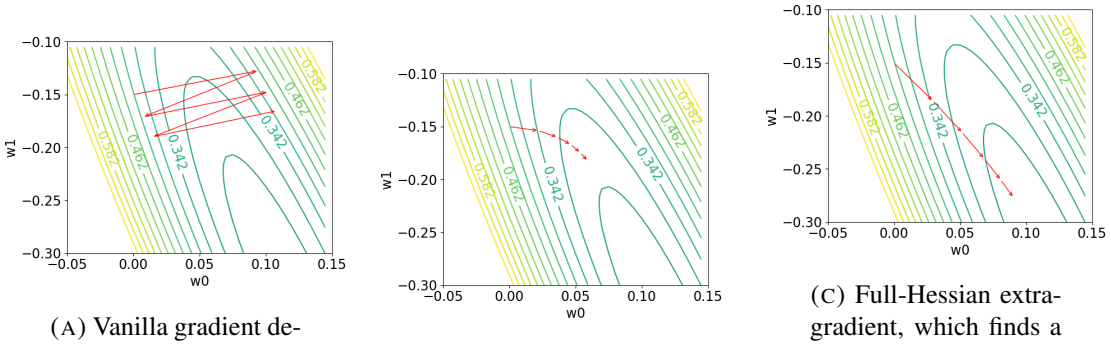
To better understand the LOANT algorithm, we relate LOANT to the extragradient technique and Model-Agnostic Meta Learning [Finn et al., 2017].

The vanilla gradient descent (GD) algorithm follows the direction along which the function value decreases the fastest. However, when facing an ill-conditioned problem like the one in Fig. 4.3, GD is known to exhibit slow convergence because the local gradients are close to being orthogonal to the direction of the local optimum.

For comparison with LOANT, we consider the extragradient (EG) method [Korpelevich, 1976, Azizian et al., 2020] that uses the following update rule when optimizing the function $f(w)$ with respect to w,

$$w \leftarrow w - \eta \frac{\mathrm{d} f(w - \gamma \frac{\partial f(w)}{\partial w})}{\mathrm{d} w}. \tag{4.23}$$

Similar to LOANT, we can adopt a first-order approximation to EG if we set the Hessian term to zero in the total derivative. Instead of optimizing the immediate function value $f(w)$, this method optimizes $f(w - \gamma \frac{\partial f}{\partial w})$, which is the function value after one more GD step. This can be understood as looking one step ahead along the optimization trajectory. In the contour diagrams of Fig. 4.3, we show the optimization of a 2-dimensional quadratic function. This simple example showcases how the ability to look one step ahead can improve optimization in pathological loss landscapes. We motivate the nested optimization of LOANT by drawing an analogy between EG and LOANT.



(A) Vanilla gradient descent, which exhibits a zigzag trajectory. $\eta = 0.025$.

(B) First-order extragradient, which sets the Hessian term to zero. $\eta = 0.025$. $\gamma = 0.01$.

(C) Full-Hessian extragradient, which finds a direct path to the local minimum, enabling a large learning rate $\eta = 0.1$.

FIGURE 4.3: Minimization of a 2D function $f(w) = w^\top Aw + b^\top w + c$. A is positive definite and has a condition number of 40. The initial point is $(0, -0.15)$. The red arrows show the trajectory of w. The look-ahead capability of extragradient finds a much more direct path to the local minimum than vanilla gradient descent.

It is worth noting that LOANT differs from the EG update rule in important ways. Specifically, in EG the inner GD step and the outer GD step are performed on the same function $f(\cdot)$, whereas LOANT performs the inner step on $\mathcal{L}_d$ and the outer step on $\mathcal{L}_s$ or $\mathcal{L}_t$.

For a similar idea with multiple losses, we turn to MAML [Finn et al., 2017]. In MAML, there are K tasks with losses $\mathcal{L}_1, \ldots, \mathcal{L}_k, \ldots, \mathcal{L}_K$. On every task, we perform a one-step SGD update to the model parameter $w \in \mathbb{R}^L$,

$$w_{T_k} = w - \gamma \frac{\partial \mathcal{L}_k(w)}{\partial w}. \tag{4.24}$$

After going through K tasks, the actual update to w is calculated using the parameters w_{T_k},

$$w \leftarrow w - \eta \frac{1}{K} \sum_k \frac{\mathrm{d}\mathcal{L}_k(w_{T_k})}{\mathrm{d}w}. \tag{4.25}$$

Utilizing the idea of look ahead, in MAML we update w so that subsequent optimization on any single task or combination of tasks would achieve good results.

Adversarial neural transfer has three tasks, the source-domain and target-domain classifications and the negative discriminator loss. The updates performed by LOANT in Eq. 4.5 and 4.6 are similar to MAML's look-ahead update in Eq. 4.24. Specifically, when we update model parameters using the gradient from the total loss $\mathcal{L}^{\text{LO}}$, we prepare for the next descent step on $-\mathcal{L}_d$. Therefore, LOANT can be understood as forcing

domain-specific losses to accommodate the domain discrimination loss and mitigating their competition.

LOANT differs from MAML since, in the inner update, LOANT updates the sentence-level features z_s and z_t instead of the model parameters w. As z_s and z_t are usually of much smaller dimensions than w, this leads to accelerated training and reduced memory footprint. For example, in the BERT-base model [Devlin et al., 2019], L is 110 million and D is 768. Within the regular range of batch size B, $BD \ll L$. In the experiments, we verify the benefits of LOANT in terms of accuracy and time and space complexity.

4.3 Experimental Evaluation

4.3.1 Datasets

We conduct four cross-domain sarcasm detection experiments by transferring from an automatically collected dataset to a manually annotated dataset. The two automatically collected datasets include Ptáček [Ptáček et al., 2014] and Ghosh[1] [Ghosh and Veale, 2016], which treat tweets having particular hastags such as `#sarcastic`, `#sarcasm` or `#not` as sarcastic and others as not sarcastic. We crawled the Ptáček dataset using the NLTK API[2] according to the tweet ids published online[3].

The two manually annotated datasets include SemEval-18[4] [Van Hee et al., 2018] and iSarcasm [Oprea and Magdy, 2020]. SemEval-18 consists of both sarcastic and ironic tweets supervised by third-party annotators and thus is used for *perceived* sarcasm detection. The iSarcasm dataset contains tweets written by participants of an online survey and thus is an example of *intended* sarcasm detection.

Table 4.1 summarizes the statistics of the four datasets. The SemEval-18 dataset is balanced while the iSarcasm dataset is imbalanced. The two source datasets are more than ten times the size of the target datasets. For all datasets, we use the predefined test set and use a random 10% split of the training set as the development set.

[1] `https://github.com/AniSkywalker/SarcasmDetection/tree/master/resource`
[2] `http://www.nltk.org/howto/twitter.html`
[3] `http://liks.fav.zcu.cz/sarcasm/`
[4] `https://github.com/Cyvhee/SemEval2018-Task3/tree/master/datasets`

Dataset	Train	Val	Test	% Sarcasm
Ptáček	51009	5668	6298	49.50%
Ghosh	33373	3709	4121	44.84%
SemEval-18	3398	378	780	49.12%
iSarcasm	3116	347	887	17.62%

TABLE 4.1: Dataset statistics, including number of samples in each split and the proportion of sarcastic texts.

We preprocessed all datasets using the lexical normalization tool for tweets from Baziotis et al. [2017]. We cleaned the four datasets by dropping all the duplicate tweets within and across datasets, and trimmed the texts to a maximum length of 100. To deal with class imbalance, we performed upsampling on the target-domain datasets, so that both the sarcastic and non-sarcastic classes have the same size as source domain datasets.

4.3.2 Baselines

We compare LOANT with several competitive single-task and multi-task baselines.

MIARN [Tay et al., 2018]: A state-of-the-art short text sarcasm detection model ranked top-1 on the iSarcam dataset. The model is a co-attention based LSTM model which uses the word embeddings pretrained on Twitter data[5].

Dense-LSTM [Wu et al., 2018]: A state-of-the-art single-task sarcasm detection model ranked top-1 on the SemEval-18 dataset. The model is a densely connected LSTM network consisting of four Bi-LSTM layers and the word embeddings pretrained on two Twitter datasets.

BERT: We finetune the BERT model [Devlin et al., 2019] with an additional simple classifier directly on the target dataset.

S-BERT is a two-stage finetuning of the BERT model. We first finetune BERT on the source dataset and the best model is selected for further fine-tuning on the target dataset.

MTL: We implemented a multi-task learning (MTL) model, which has the same architecture as LOANT except that the domain discriminator is removed. We use BERT as the shared text encoding network.

[5] `https://nlp.stanford.edu/projects/glove/`

MTL+LO: In this baseline, we applied latent optimization to MTL. As MTL does not have the adversarial discriminator, we use the domain-specific losses to optimize latent representations:

$$z'_s = z_s - \gamma \frac{\partial \mathcal{L}_s(z_s)}{\partial z_s} \tag{4.26}$$

$$z'_t = z_t - \gamma \frac{\partial \mathcal{L}_t(z_t)}{\partial z_t} \tag{4.27}$$

We use the above to replace Equations 4.5 and 4.6 and keep the rest training steps unchanged. This model is compared against MTL to study the effects of LO in non-adversarial training for cross-domain sarcasm detection.

ANT: This is the conventional adversarial neural transfer model with the same architecture as LOANT. The only difference is that we do not apply latent optimization. For fair comparisons, we use BERT as the text encoder.

ANT+MAML: In Section 4.2.3, we discussed the similarity between LO and MAML. Therefore, we create a baseline that uses a MAML-like strategy for encouraging the collaboration of different loss terms. Instead of optimizing the latent representation z_s and z_t, we first take a SGD step in the parameter space of w_b,

$$w'_b = w_b + \gamma \frac{\partial \mathcal{L}_d(z_s, z_t)}{\partial w_b}. \tag{4.28}$$

After that, we use w'_b to compute the gradients used in the actual updates to all model parameters, including w_b.

4.3.3 Experimental Settings

Model Settings. For all models using the BERT text encoder, we use the uncased version of the BERT-base model and take the 768-dimensional output from the last layer corresponding to the `[CLS]` token to represent a sentence. The BERT parameters are always shared between domains. For other network components, we randomly initialize the dense layers and classifiers. To minimize the effect of different random initializations, we generate the same set of initial parameters for each network component and use them across all baselines wherever possible.

The source dense layer, the shared dense layer, and the target dense layer are single linear layers with input size of 768 and output size of 768 followed by the tanh activation. The

classifier in all models consists of two linear layers. The first linear layer has input size of 768×2 (taking both shared and domain-specific features) and output size of 768 followed by the ReLU activation. The second linear layer has input size 768 and output size 2 for binary classification. After that we apply the softmax operation. We follow the released code[6] to implement the Gradient Reversal Layer. It is controlled by a schedule which gradually increases the weight of the gradients from the domain discrimination loss.

Hyperparameters. We set the batch size to 128 for all models and search for the optimal learning rate (LR) from 2e-5 to 1e-4 in increments of 2e-5 using the F-score on the development set. All the learning rates chosen by each setting can be found in Table 4.2. The best learning rate for fine-tuning BERT on SemEval-18 and iSarcasm is 4e-5. S-BERT model is finetuned twice, first on the source domain and then on the target domain. Thus, we search for one best learning rate for each finetuning using the source and target development sets respectively. The best first-round LR is 6e-05 for Ptáče and 8e-5 for Ghosh.

Models	Ptáče → SemEval	Ghosh → SemEval	Ptáče → iSarcasm	Ghosh → iSarcasm
S-BERT	1e-4	1e-4	4e-5	2e-5
MTL	6e-4	8e-5	4e-5	1e-4
MTL+LO	6e-4	8e-5	4e-5	1e-4
ANT	2e-5	4e-5	2e-5	2e-5
ANT+MAML	2e-5	4e-5	2e-5	2e-5
LOANT	2e-5	4e-5	2e-5	2e-5

TABLE 4.2: Learning rate chosen by each model on the given search grid.

Other models, MTL, ANT and the LO-adpated versions are selected using the target development set. For a rigorous comparison, we use the best LR for ANT when training LOANT and the best LR for MTL when training MTL+LO.

Training Setting. We optimize all models using Adam [Kingma and Ba, 2014] with batch size of 128. We tune the learning rate (LR) on the development set from 1e-5 to 1e-4 in increments of 2e-5. To objectively assess the effects of latent optimization (LO), we first find the best LR for the base models such as ANT and MTL. After that, with the best LR unchanged, we apply LO to ANT and MTL. We use the cosine learning rate schedule for all models. All models are trained for 5 epochs on Nvidia V100 GPUs

[6] https://github.com/fungtion/DANN

with 32GB of memory in mixed precision. Due to the large model size and pretrained weights of BERT, 5 epochs are sufficient for convergence.

Evaluation Metrics. Following [Van Hee et al., 2018, Oprea and Magdy, 2020, Wu et al., 2018], we select and compare models using the F-score on the sarcastic class in each dataset. We additionally report the corresponding Recall and Precision. In all our experiments, we use the development set for model selection and report their performance on the test set. To evaluate the efficiency of LOANT versus MAML-based training, we also compare their required GPU memory and average training time in each epoch. We compare models on the target domain datasets.

4.3.4 Comparison with the States of the Art

We compare LOANT with state-of-the-art methods on the SemEval-18 dataset [Van Hee et al., 2018] and the iSarcasm datast [Oprea and Magdy, 2020]. Table 4.4 and 4.4 presents the test performance of LOANT and all baseline models. Our LOANT model consistently outperforms all single-task baselines by large margins. In particular, LOANT outperforms MIARN by 10.02% on iSarcasm [Oprea and Magdy, 2020] whereas the fine-tuned BERT achieved 1.48% lower than MIARN.

On SemEval-18, the fine-tuned BERT achieves better test performance than other four single-task baselines. The results indicate that fine-tuning BERT, a popular baseline, does not always outperform the traditional LSTM networks specifically designed for the task. We hypothesize that the large BERT model can easily overfit the small datasets used, which highlights the challenge of sarcasm detection.

4.3.5 Transfer Learning Performance

The middle and bottom sections of Table 4.3 and 4.4 present the test performance of six transfer learning models (S-BERT, MTL, ANT, MTL+LO, ANT+MAML, and LOANT) under four groups of transfer learning experiments. These models generally outperform the single-task models, demonstrating the importance of transfer learning. Among these, we have the following observations.

Effects of the Domain Discriminator. The performance differences between MTL and ANT can be explained by the addition of the domain discriminator, which encourages

	Model	F-score	Recall	Precision
Single-task	Random†	0.3730	0.3730	0.3730
	Unigram SVM†	0.5890	0.6590	0.5320
	LSTM†	0.5260	0.4440	0.6450
	DenseLSTM *	<u>0.6510</u>	0.7106	0.6005
	BERT	0.6626	0.7055	0.6246
Source: Ptáče	S-BERT	0.6676	0.7055	0.6337
	MTL	0.6404	0.7896	0.5386
	ANT	0.6348	0.8187	0.5184
	MTL+LO	0.6598	0.7346	0.5989
	ANT+MAML	0.6454	0.7540	0.5641
	LOANT (ours)	**0.6702**	0.8025	0.5754
Source: Ghosh	S-BERT	0.6512	0.7766	0.5607
	MTL	0.6525	0.7475	0.5789
	ANT	0.6626	0.8899	0.5278
	MTL+LO	0.6622	0.8058	0.5620
	ANT+MAML	0.6338	0.7281	0.5610
	LOANT (ours)	**0.6818**	0.7734	0.6096

Results marked with † are reported in [Van Hee et al., 2018], * are in [Wu et al., 2018] and ‡ are in [Oprea and Magdy, 2020].

.

TABLE 4.3: Single-task and multi-task Performance on SemEval-18. The best performed F-score on the four groups of transfer learning are in bold. The best single task learning results are underlined.

Model	F-score	Recall	Precision
SIARN‡	0.3420	0.7820	0.2190
MIARN‡	<u>0.3640</u>	0.7930	0.2360
LSTM‡	0.3360	0.7470	0.2170
DenseLSTM‡	0.3180	0.2760	0.3750
BERT	0.3492	0.4904	0.2711
S-BERT	0.3710	0.5541	0.2788
MTL	0.3767	0.3503	0.4074
ANT	0.3857	0.5159	0.3079
MTL+LO	0.4379	0.4267	0.4496
ANT+MAML	0.3951	0.5605	0.2923
LOANT (ours)	**0.4642**	0.4968	0.4357
S-BERT	0.3383	0.5732	0.2400
MTL	0.3838	0.5159	0.3056
ANT	0.4063	0.4904	0.3468
MTL+LO	0.3987	0.4012	0.3962
ANT+MAML	0.3589	0.4904	0.2830
LOANT (ours)	**0.4101**	0.4649	0.3668

TABLE 4.4: Single-task and multi-task Performance on iSarcasm.

the shared features under the source domain and the target domain to have the same

distributions. In the four pairs of experiments, ANT marginally outperforms MTL by an average of 0.9% F-score. In the Ptáček → SemEval-18 experiment, the domain discriminator causes F-score to decrease by 0.56%. Overall, the benefits of the adversarial discriminator to transfer learning appear to be limited. As discussed earlier, the competition between the domain-specific losses and the negative domain discrimination loss may have contributed to the ineffectiveness of ANT.

Effects of Latent Optimization. We can observe the effects of LO by comparing ANT with LOANT and comparing MTL with MTL+LO. Note that in these experiments we adopted the best learning rates for the baseline models ANT and MTL rather than the latent-optimized models. On average, LOANT outperforms ANT by 3.42% in F-score and MTL+LO outperforms MTL by 2.63%, which clearly demonstrates the benefits provided by latent optimization.

Latent Space vs. Model Parameter Space. In the ANT+MAML baseline, we adopt a MAML-like optimization strategy, which performs the look-ahead in the BERT parameter space instead of the latent representation space. Interestingly, this strategy does not provide much improvements and on average performs 1.40% worse than ANT. LOANT clearly outperforms ANT+MAML.

In addition, optimization in the latent space also provides savings in computational time and space requirements. Table 4.5 shows the time and memory consumption for different transfer learning methods. Adding LO to ANT has minimal effects on the memory usage, but adding MAML nearly doubles the memory consumption. On average, ANT+MAML increases the running time of LOANT by 3.1 fold.

		SemEval-18	iSarcasm
	Model	RAM/Time	RAM/Time
Source: Ptáče	LOANT	1.01x/2.41x	1.01x/2.55x
	MTL+LO	1.01x/1.92x	1.01x/1.91x
	ANT	1.00x/1.00x	1.00x/1.00x
	ANT + MAML	1.99x/8.31x	1.93x/10.2x
Source: Ghosh	LOANT	1.01x/2.44x	1.01x/1.94x
	MTL+LO	1.01x/1.94x	1.01x/1.89x
	ANT	1.00x/1.00x	1.00x/1.00x
	ANT + MAML	1.99x/8.41x	1.93x/10.7x

TABLE 4.5: Running time and maximum memory footprint for different transfer learning methods.

The Influence of Domain Divergence. In transfer learning, the test performance depends on the similarity between the domains. We thus investigate the dissimilarity between datasets using the Kullback–Leibler (KL) divergence between the unigram probability distributions,

$$d_{KL} = \sum_{g \in V} P_t(g) \log \frac{P_t(g)}{P_s(g)}. \tag{4.29}$$

where $P_s(g)$ and $P_t(g)$ are the probabilities of unigram g for the source domain and target domain respectively. V is the vocabulary. Table 4.6 shows the results. Ptáček is more similar to the two target datasets than Ghosh. Among the two target datasets, iSarcasm is more similar to Ptáček than SemEval-18.

Comparing LOANT and ANT, we observe that the largest improvement, 7.85%, happens in the Ptáček → iSarcasm transfer where domain divergence is the smallest. The Ptáček → SemEval-18 transfer comes in second with 3.54%. Transferring from Ghosh yields smaller improvements. Further, we observe the same trend in the comparison between MTL+LO and MTL. The largest improvement brought by LO is 6.12% in the Ptáček → iSarcasm transfer. As one may expect, applying LO leads to greater performance gains when the two domains are more similar.

	SemEval-18	iSarcasm
Ptáček	0.1631	0.0521
Ghosh	0.2300	0.2217

TABLE 4.6: The KL divergence of word probability over the overlapped vocabulary for each pair of domains.

4.3.6 Source Domain Performance

The original goal is to use automatically collected sarcasm datasets, which are large but *noisy*, to improve performance on human-annotated datasets, which are *clean* and provide good performance measure. That is why we provided only the target domain performance. Upon close inspection, LOANT also improves the performance on the source domain, even though model selection was performed on the target domain. Table 4.7 shows the results.

In Table 4.8, we also show the results after model selection on both domains. Naturally, this might lead to slightly lowered target-domain performance than achieved by model selection on target domain only. Comparing LOANT with ANT, and MTL+LO with

Domain	ANT	LOANT	MTL	MTL+LO
Ptacek	0.8307	**0.8484**	**0.8640**	0.8629
iSarcasm	0.3857	**0.4642**	0.3767	**0.4379**
Average	0.6082	**0.6563**	0.62035	**0.6504**
Ghosh	**0.7345**	0.6596	0.6609	**0.6688**
iSarcasm	0.4063	**0.4101**	0.3838	**0.3953**
Average	**0.5704**	0.5349	0.5224	**0.5321**
Ptacek	**0.8626**	0.8612	**0.8722**	0.8666
SemEval18	0.6348	**0.6702**	0.6404	**0.6598**
Average	0.7487	**0.7657**	0.7563	**0.7632**
Ghosh	0.7161	**0.7752**	**0.7700**	0.7579
SemEval18	0.6626	**0.6818**	0.6525	**0.6622**
Average	0.6894	**0.7285**	**0.7113**	0.7101

TABLE 4.7: Test F1 score. Models selected using the target domain only.

MTL, our results show that, in most cases, LO-based models improve both source and target domain F1. In particular, target domain F1 obtains more improvement than source domain F1. This suggests that LO provides benefits to knowledge transfer.

Domain	ANT	LOANT	MTL	MTL+LO
Ptacek	0.8307	**0.8484**	**0.8640**	0.8629
iSarcasm	0.3857	**0.4642**	0.3767	**0.4379**
Average	0.6082	**0.6563**	0.6204	**0.6504**
Ghosh	0.7787	**0.7826**	**0.7859**	0.7807
iSarcasm	**0.3965**	0.3215	0.3764	**0.3953**
Average	**0.5876**	0.5521	**0.5812**	0.5880
Ptacek	0.8567	**0.8612**	**0.8720**	0.8632
SemEval18	0.6463	**0.6702**	0.6594	**0.6666**
Average	0.7515	**0.7657**	**0.7657**	0.7649
Ghosh	0.7919	**0.7962**	0.7672	**0.7884**
SemEval18	0.6427	**0.6490**	0.6357	**0.6442**
Average	0.7173	**0.7226**	0.7015	**0.7163**

TABLE 4.8: Test F1 score. Models selected with the average F1 on the two domains.

4.4 Summary

Transfer learning holds promise for the effective utilization of multiple datasets for sarcasm detection. In this chapter, we propose a latent optimization (LO) strategy for adversarial transfer learning for sarcasm detection. By providing a look-ahead in the gradient updates, the LO technique allows multiple losses to accommodate each other. This

proves to be particularly effective in adversarial transfer learning where the domain-specific losses and the adversarial loss potentially conflict with one another. With the proposed LOANT method, we set a new state-of-the-art for the iSarcasm dataset. We hope the joint utilization of multiple datasets will contribute to the creation of contextualized semantic understanding that is necessary for successful sarcasm detection.

We identify a few limitations of this work:

- First, apart from the cross-domain sarcasm detection task, we should find more similar settings from other tasks for evaluation. While it is impossible to find all applicable scenarios for evaluation, we encourage future research to validate the proposed method and expand its insights drawn from current experiments.
- Second, our studies are conducted on top of BERT. There are also many other pre-trained encoders introduced in Sub-Chapter 2.1.3. It is not known whether those different pre-trained models may benefit from the proposed method or exhibit different behaviors. We encourage future research to take a comparative study among different pre-trained models.
- Third, despite the proposed look-ahead learning strategy explicitly introducing the gradients of the opponent classifier, how much is left to be done for annealing the competition between competing objectives is implicit. Understanding the fundamental issues of multi-objective optimization is difficult but rewarding.

Chapter 5

Personalizing Federated Language Model for Diverse Domains

5.1 Motivation

In the past decade, we have witnessed the broad and far-reaching success of artificial intelligence (AI) in novel application areas such as conversational characters, which provide increasingly human-like interactions to users. One particularly challenging issue in the understanding of human language is the modeling of humor. Humor is a prominent example of linguistic creativity [Veale, 2012] and plays a vital role in human communication. Unlike AI tasks with objective ground truth, the task of humor recognition is characterized by subjectivity in humor understanding. Due to individual differences in cognitive processes, the same joke can be perceived differently by its audience [Aykan and Nalçacı, 2018, Martin et al., 2003, Heintz and Ruch, 2019]. This is empirically proved via data analysis on a real-world dataset [Hossain et al., 2019] in which each joke is rated by five persons in terms of its funniness, and the results show that the variance of human's perceived funniness on the same joke is nontrivial (Figure 5.1).

Computational approaches for recognizing humor generally regard the task as a binary classification problem [Yang et al., 2015, Chen and Soo, 2018]. Traditional research mainly focuses on extracting expressive features as inputs to a classifier in order to improve classification performance. Neural networks such as Convolutional Neural Networks (CNNs) [Chen and Soo, 2018] and transformer networks [Mao and Liu, 2019]

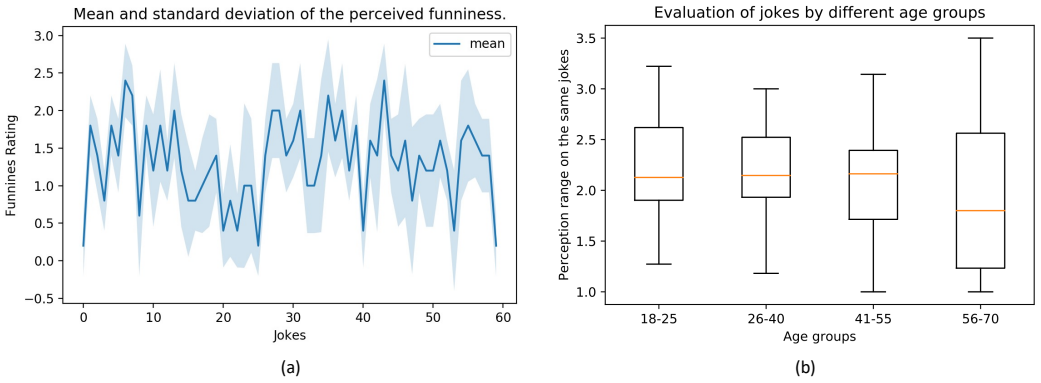


FIGURE 5.1: Empirical analysis of a random set of 60 jokes from a real-world humor rating dataset reflecting non-trivial subjectivity in human perception. (a) shows that users' perceived funniness on the same jokes vary from person to person and the variance differs from joke to joke (shaded area). (b) shows that the effect in (a) is consistent across different age groups, albeit at different levels of variance.

have shifted the focus from feature engineering to automatic feature extraction. Existing humor recognition models are generally based on the assumption that users have a consensus about whether or not a given text is humorous (as illustrated in Figure 5.2(a)) [Mao and Liu, 2019, Yang et al., 2015, Chen and Soo, 2018, Liu et al., 2018, Zhang and Liu, 2014]. However, in reality, humor preference is highly subjective. As a result, these approaches perform well in recommending humorous content with popular appeal, but cannot achieve personalized humor recognition.

To ensure that a general language model can be adapted to a personal domain, one has to collect sufficient *labeled data* to obtain the desired performance. However, it is impossible to obtain enough labels for *tail classes* from people who have extreme preferences for humorous content. That is, the long-tailed, imbalanced datasets are inevitable in subjective tasks and can undermine the reliability of a general language model. Moreover, these facts cannot be known in advance in a privacy-preserving setting.

We bridge this important gap in the humor recognition literature by proposing a personalized humor recognition approach - FedHumor. We adopt the assumption that humor labels from different users regarding the same text contents can be diverse, and formalize the problem as a conditional binary classification task. The federated learning (FL) architecture is used as the basis for our proposed personalized humor recognition model. We deem that the *FL training strategy can create an ensemble model which prefers the*

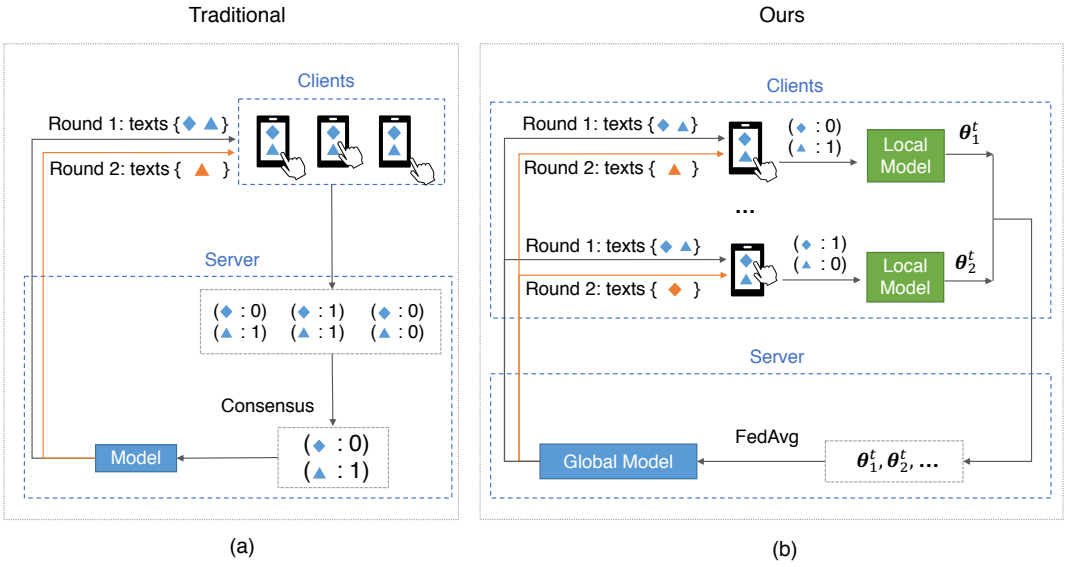


FIGURE 5.2: Traditional setting versus personalized federated learning setting for training and applying a humor recognition model (*Best viewed in color*). In (a), a humor recognition model is trained on a centralized dataset whose labels are determined by majority voting (consensus) from round 1; the trained model is used to recommend funny texts to all clients in round 2, without distinction. In (b), a humor recognition model is trained on distributed datasets where the individual labels and distributions are preserved on local devices at round 1; the trained model will recommend texts to each client at round 2, possibly in different sequences.

hypothesis function that accounts for more than one possible label distribution. Allowing different domains to tie weights under the same framework can alleviate overfitting the local imbalanced data to some extent, thereby reducing the need to obtain more labeled data for tail classes.

FedHumor extends the popular Federated Averaging (FedAvg) algorithm [McMahan et al., 2017] with a diversity adaptation strategy to enhance the handling of disparate user preferences in humor recognition. As a result, it can adjust the order of humor contents (e.g., jokes) recommended to different users according to their individual preferences (as illustrated in Figure 5.2(b)). The federated training process allows multiple views over diverse label distributions, thus enhancing the generalization of the humor recognition model. Meanwhile, the distributed learning process allows personalized adaptations to different users' preferences to be learned locally, thus enhancing the personalization performance of the model. Moreover, this approach does not require sensitive data concerning each user's personal humor preference to be exposed, complying with the General Data Protection Regulation (GDPR) requirement on data privacy preservation.

5.2 Preliminaries

5.2.1 Federated Gradient-based Optimization

We find the optimal parameters $\boldsymbol{\theta}^*$ for the function $f_{\boldsymbol{\theta}}$ by optimizing a differentiable loss function $\mathcal{L}(\boldsymbol{\theta})$ on a training set $\mathcal{D}$, where $\boldsymbol{x} \sim p(\boldsymbol{x}), \forall \boldsymbol{x} \in \mathcal{D}$.

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\mathcal{D}; \boldsymbol{\theta}) \tag{5.1}$$

The optimal $\boldsymbol{\theta}^*$ is found by iterated gradient descent. At each step t, we update $\boldsymbol{\theta}$ with learning rate η controlling the step size in the direction of the gradient $\bigtriangledown_{\boldsymbol{\theta}^t} \mathcal{L}(\mathcal{D}; \boldsymbol{\theta}^t)$:

$$\boldsymbol{\theta}^{t+1} \leftarrow \boldsymbol{\theta}^t - \eta \bigtriangledown_{\boldsymbol{\theta}^t} \mathcal{L}(\mathcal{D}; \boldsymbol{\theta}^t) \tag{5.2}$$

The above update is repeated until convergence or a predefined number of iterations is reached.

Under the Federated Learning setting, m participants in the federation wish to jointly optimize a model without sharing data with each other. Federated Averaging (FedAvg) [McMahan et al., 2017] provides a general method to perform this optimization across m data silos, $\mathcal{D} = \bigcup_{i=1}^{m} \mathcal{D}_i$ and $\boldsymbol{x} \sim p^{(i)}(\boldsymbol{x})$, $\forall \boldsymbol{x} \in \mathcal{D}_i$. The local model parameters are adapted by gradient descent from global parameters $\boldsymbol{\theta}^t$ at each step. To distinguish local parameters and global parameters, we let $\boldsymbol{\theta}_i^{t,k}$ denote the local parameters at the i^{th} participant after applying k local updates to the global parameter $\boldsymbol{\theta}^t$.

$$\begin{aligned} \boldsymbol{\theta}_i^{t,0} &\leftarrow \boldsymbol{\theta}^t \\ \boldsymbol{\theta}_i^{t,k+1} &\leftarrow \boldsymbol{\theta}_i^{t,k} - \eta_t \bigtriangledown_{\boldsymbol{\theta}_i^{t,k}} \mathcal{L}(\mathcal{D}_i; \boldsymbol{\theta}_i^{t,k}) \end{aligned} \tag{5.3}$$

After every K local iterations, we synchronize the parameters across participants by averaging local updates. The global parameter update can be written as

$$\boldsymbol{\theta}^{t+1} \leftarrow \frac{1}{m} \sum_{i=1}^{m} \boldsymbol{\theta}_i^{t,K} \tag{5.4}$$

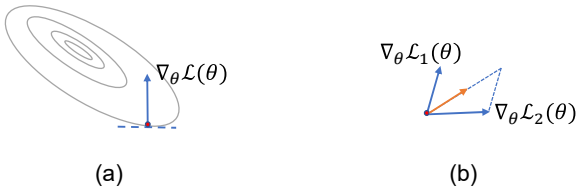


FIGURE 5.3: Model parameters updated following standard gradient descent (a), and averaged gradient descent in federated learning (b). *Best viewed in color.*

This implies that the model parameters are updated following the direction of the sum of the gradient vectors as illustrated in Figure 5.3. Traditionally, the parameters are updated iteratively on batches of labeled samples randomly sampled from the training set. In this case, the gradient descent direction is guided by the loss over a batch of observed samples (Figure 5.3(a)). In federated learning, multiple batches of samples are observed by the global model during every training iteration, while only a compromised step (i.e. $\frac{1}{2}(\bigtriangledown_{\boldsymbol{\theta}}\mathcal{L}_1(\boldsymbol{\theta}) + \bigtriangledown_{\boldsymbol{\theta}}\mathcal{L}_2(\boldsymbol{\theta}))$) is taken to update parameters $\boldsymbol{\theta}$ (Figure 5.3(b)).

5.2.2 Problem Formulation

Consider a practical scenario in which a conversational character wants to tell a user a joke and hence needs to predict whether the user will perceive the joke as humorous or not. The conversational character has a set of historical texts that have binary feedback from a set of users, who may disagree with each other on if a joke is funny. Instead of attempting to reconcile the differences of the user-generated labels, we study a personalized humor recognition problem in which the predicted classes of jokes are conditioned on the personal preferences of users.

Formally, $\boldsymbol{X} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_n)$ denotes a total of n input jokes, where $\boldsymbol{x}_j \sim p(\boldsymbol{x})$ is determined by a content publisher. $Y_i = \{y_1^{(i)}, ..., y_n^{(i)}\}$, where $y_j^{(i)} \in \{0, 1\}$, denotes the target labels produced by user i on inputs $\boldsymbol{X}$. We use α, a quantified funniness threshold lying between the liked and disliked jokes in the historical data of a user, to denote personal humor preference. For example, the distribution $p(y|\boldsymbol{x}, \alpha_i)$ over the binary labels of user i is determined by α_i. Given m users' historical data, the task here is to predict Y for a user having $\alpha \sim p(\alpha)$. We treat this task as a conditional binary classification task.

In a standard supervised learning setting, the probability of input $p(\boldsymbol{x})$ is assumed to be unchanged. A traditional binary classification task tries to train a model $f_{\boldsymbol{\theta}}$ to estimate

the true $p(y|\boldsymbol{x})$ based on sufficient observations in order to construct the relationship between data and labels $p(\boldsymbol{x}, y)$:

$$p(\boldsymbol{x}, y) = p(y|\boldsymbol{x})p(\boldsymbol{x}). \tag{5.5}$$

Note that the standard learning task assumes that the input data $\boldsymbol{x} \sim p(\boldsymbol{x})$ while the label $y \sim p(y|\boldsymbol{x})$ only depends on the input data $\boldsymbol{x}$. When applied to existing humor recognition datasets, it involves an underlying assumption that different users follow the same label distribution $p(y|\boldsymbol{x})$. However, this assumption may not hold for humor perception in practice.

Here, we assume that $p(y|\boldsymbol{x})$ relates to users' preference α. The problem thus becomes training the model $f_{\boldsymbol{\theta}}$ to estimate the conditional probability $p(y|\boldsymbol{x}, \alpha)$ based on given $\alpha \sim p(\alpha)$ and observed $\boldsymbol{x} \sim p(\boldsymbol{x})$:

$$p(\boldsymbol{x}, y, \alpha) = p(y|\boldsymbol{x}, \alpha)p(\boldsymbol{x}, \alpha) = p(y|\boldsymbol{x}, \alpha)p(\boldsymbol{x})p(\alpha). \tag{5.6}$$

Here, $p(\boldsymbol{x})$ is determined by the content publisher and $p(\alpha)$ is determined by users. So they are independent and satisfy: $p(\boldsymbol{x}, \alpha) = p(\boldsymbol{x})p(\alpha)$.

5.3 The Proposed FedHumor Model

In this section, we describe the detailed design of the proposed FedHumor model for the task of personalized humor recognition. We first introduce the contextualized text encoder model, then we present the federated learning-based model training, followed by the diversity adaptation design for different users, and finally, we introduce the model validation under the FL setting. The pseudo-code is given in Algorithm 3.

5.3.1 Model Architecture

We employed the pretrained language model - BERT [Devlin et al., 2019] - to capture contextualized sentence representations as the input features to a classification layer, and fine-tune the pretrained weights together with classifier parameters on our task. Briefly,

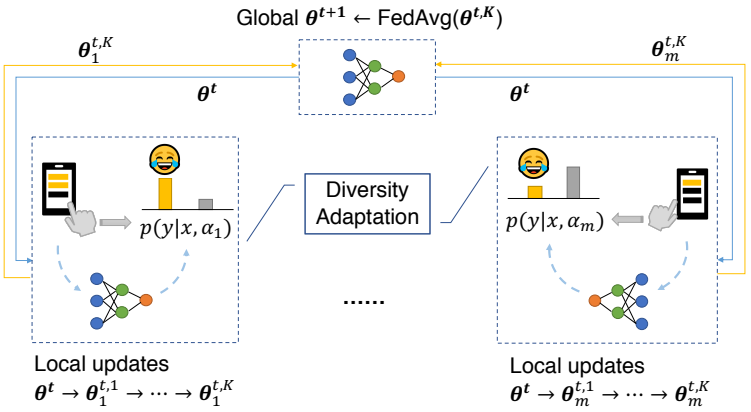


FIGURE 5.4: The training of FedHumor involves three steps: 1) the server sends the global model to clients; 2) the clients train the model locally based on their own labels, and send their updated parameters to the server; 3) the server aggregates local updates to produce a new global model.

BERT tokenizes a piece of text into a sequence of word IDs indexed by its stationary vocabulary. The IDs are used to fetch their corresponding word embeddings from an embedding table: $\boldsymbol{W} = (\boldsymbol{W}_1, ..., \boldsymbol{W}_l)$, $\boldsymbol{W} \in \mathbb{R}^{l \times d}$. l is the fixed maximum length of a sentence and d is the word embedding size. The interactions between words and order information are captured through the self-attention modules in twelve transformer layers, denoted as $\boldsymbol{T}$. The first vector ([CLS]) of the final hidden states at the last transformer layer is used to represent the final contextualized sentence representation, $\boldsymbol{x}$, for the given sentence. A non-linear classification layer, denoted as $\boldsymbol{C}$, followed by a $tanh$ activation function, accepts the sentence representations and makes predictions. The trainable parameters in the FedHumor model are $\boldsymbol{\theta} = [\boldsymbol{W}; \boldsymbol{T}; \boldsymbol{C}]$ and time complexity for each round is $\mathcal{O}(|\boldsymbol{W}||\boldsymbol{T}||\boldsymbol{C}|K)$. K is the local training iterations on each user's device while all users perform local adaptation synchronously.

5.3.2 Weight-tying Federated Training

We follow the FedAvg algorithm [McMahan et al., 2017] to aggregate the model parameters updated on each user's data silo $\mathcal{D}_i = \{\boldsymbol{X}, Y_i\}$ after each round of federated training to produce a global model, as illustrated in Figure 5.4. In every training round, the current global FL model with parameters $\boldsymbol{\theta}$ is sent to m users randomly selected from the device population, and the adapted local parameters are denoted as $\boldsymbol{\theta}_i$. For

each user whose humor preference is α_i, we use cross entropy as our loss function, and the instance loss for user i is computed as follows:

$$\ell(\boldsymbol{\theta}_i; \alpha_i) = -y\log(\tilde{p}(y|\boldsymbol{x}, \alpha_i)) - (1-y)\log(1-\tilde{p}(y|\boldsymbol{x}, \alpha_i)) + \lambda||\boldsymbol{\theta}_i||_2^2. \tag{5.7}$$

Here, $\tilde{p}(y|\boldsymbol{x}, \alpha_i))$ is the adapted estimated probability computed using Equations (5.10) and (5.11). λ is the hyperparameter controlling L2 norm regularization over local model parameters $\boldsymbol{\theta}_i$ (a.k.a. weight decay). $\boldsymbol{\theta}_i$ is determined by gradient descent to minimize the following training loss on user i's data silo:

$$\mathcal{L}(\boldsymbol{\theta}_i; \alpha_i) = \frac{1}{|\mathcal{D}_i|} \sum_{\mathcal{D}_i} \ell(\boldsymbol{\theta}_i; \alpha_i) \tag{5.8}$$

In each federated training round t, the local model parameters are adapted from $\boldsymbol{\theta}^t$ to $\boldsymbol{\theta}_i^{t,K}$ for K local iterations. The global model parameters are adapted from $\boldsymbol{\theta}^t$ to $\boldsymbol{\theta}^{t+1}$ by averaging the aggregated parameters:

$$\boldsymbol{\theta}^{t+1} \leftarrow \boldsymbol{\theta}^t - \eta_t \cdot \frac{1}{m} \sum_{i=1}^{m} \bigtriangledown_{\boldsymbol{\theta}_i^{t,K}} \mathcal{L}(\boldsymbol{\theta}_i^{t,K}; \alpha_i) \tag{5.9}$$

where m is the number of users joining the federated learning and η_t is the learning rate at each round and its value depends on the learning rate scheduler. Throughout the process, users' personal humor preference information (i.e. $p(y|\boldsymbol{x}, \alpha)$) is kept on their own devices.

5.3.3 Diversity Adaptation

The purpose of applying federated learning to humor recognition is to enhance the generalizability of traditional humor recognition models when dealing with diverse user preferences while preserving data privacy. FedHumor is designed similarly to Google's GBoard scenario [Ramaswamy et al., 2019] based on the FL paradigm. Inspired by the mechanism employed in GBoard to prevent the federated model from being dominated by highly frequent emojis, we adapt the model predictions, $\widehat{p}(y|\boldsymbol{x})$, on each class $c = 1, .., C$, for user i as follows:

$$v_c = \frac{\widehat{P}(y = c|\boldsymbol{x})}{P(y = c|\boldsymbol{x}, \alpha \in [\alpha_i, \alpha_i + \tau))^{\beta_i}} \tag{5.10}$$

where $P(y|\boldsymbol{x}, \alpha \in [\alpha_i, \alpha_i + \tau))$ is the empirical marginal label distribution from users whose threshold α falls in the range $[\alpha_i, \alpha_i + \tau)$. τ is a small real number interval that reflects a significant change in label distribution when the preference varies. For example, $\forall \alpha \in [1.0, 1.1)$ is deemed the same preference and differs from $\forall \alpha \in [1.1, 1.2)$. For brevity, we use $\alpha = \alpha_i$ to denote an established humor preference. β_i is a user-specific scaling factor tuned on the validation set. Although this introduces one tunable hyperparameter for every user, the sensitivity analysis in Section 5.4.6 shows that it is necessary to tune β_i only when α_i takes on extreme values, which results in very imbalanced label distributions. The new probability is computed using the softmax function:

$$\tilde{P}(y = c|\boldsymbol{x}, \alpha = \alpha_i) = \frac{\exp(v_c)}{\sum_{j=1}^{C} \exp(v_j)}. \tag{5.11}$$

Intuitively, this approach can penalize predictions that are too confident about a certain class due to its dominant frequency in the observed samples. It is based on the heuristic that users may possess different levels of arousal in response to the same humorous content. Those having *easy-to-amuse* or *hard-to-amuse* personalities can result in a drastically unbalanced dataset. For example, given $\alpha = \alpha_i$, if the predicted probability is $\widehat{P}(y = 1|\boldsymbol{x}) = 0.9$ while the empirical probability is $P(y = 1|\boldsymbol{x}) = 0.7$, the denominator serves as a punishment to adjust the prediction to be $\tilde{P}(y = 1|\boldsymbol{x}) \approx 0.72$ for $\beta = 1$ (computed using Equations (5.10) and (5.11)), which is closer to the real probability.

5.3.4 Federated Model Selection

After completing each round of FL training, the global FL model will be tested on the validation set on each user's device, $\mathcal{D}^{val} = \{\boldsymbol{X}^{val}, Y^{val}|\alpha\}$. In this stage, $\widehat{p}(y^{val}|\boldsymbol{x}^{val}, \alpha)$ is the estimated probability distribution over the implicit labels in the validation set by the model. The overall validation loss for each user is calculated as follows:

$$\mathcal{L}(\boldsymbol{\theta}; \alpha_i) = -\frac{1}{|\mathcal{D}^{val}|} \sum_{(\boldsymbol{x},y)\in\mathcal{D}^{val}} [y\log(\widehat{p}(y|\boldsymbol{x}, \alpha_i)) + (1 - y)\log(1 - \widehat{p}(y|\boldsymbol{x}, \alpha_i))]. \tag{5.12}$$

The validation performance across all users is calculated as follows:

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{m} \sum_{i=1}^{m} \mathcal{L}(\boldsymbol{\theta}; \alpha = \alpha_i) \tag{5.13}$$

The best global model $\boldsymbol{\theta}$ is selected based on the lowest validation loss, $\mathcal{L}(\boldsymbol{\theta})$, among all federated rounds.

Algorithm 3: FedHumor. The m users are indexed by i; D_{train} denotes training set and D_{val} denotes validation set; α_i is the humor preference of user i; β_i indicates diversity adaptation for user i; $\mathcal{B}$ denotes a mini-batch of D_{train}; E is local training epochs and T is total federated training rounds; η_t is the learning rate at each round t.

Initialize global model parameters $\boldsymbol{\theta}$;
for $t = 1, 2, ..., T$ **do**
  for each user $i = 1, 2, ..., m$ **in parallel do**
    $\boldsymbol{\theta}_i^{t+1} \leftarrow \text{ClientUpdate}\ (i, \boldsymbol{\theta}^t, \beta_i)$;
  $\boldsymbol{\theta}^{t+1} \leftarrow \frac{1}{m} \sum_{i=1}^{m} \boldsymbol{\theta}_i^{t+1}$: global parameters update (Eq. (5.9));
  for each user $i = 1, 2, ..., m$ **in parallel do**
    $\mathcal{L}(\boldsymbol{\theta}^{t+1}; \alpha_i, t) \leftarrow \text{Inference}(i, \boldsymbol{\theta}^{t+1})$;
  $\mathcal{L}(\boldsymbol{\theta}^{t+1}, t) = \frac{1}{m} \sum_{i=1}^{m} \mathcal{L}(\boldsymbol{\theta}^{t+1}; \alpha_i, t)$ (Eq. (5.13));
Output: $\boldsymbol{\theta}^* \leftarrow \min(\mathcal{L}(\boldsymbol{\theta}^{t+1}, t))$;

ClientUpdate *(i,θ,β)* :
  Prepare $D_{train} = \{\boldsymbol{X}, Y_i | \alpha_i\}$;
  for $j = 1, 2, ..., E$ **do**
    for $\mathcal{B}$ in D_{train} **do**
      $\widehat{p} = f_{\boldsymbol{\theta}}(\boldsymbol{x}; \alpha_i)$: make predictions;
      $\widetilde{p} = \text{softmax}(\widehat{p}/p(y|\alpha_i)^{\beta_i})$: adapt predictions (Eq. (5.10), (5.11));
      $\mathcal{L}(\boldsymbol{\theta}; \alpha_i) = \frac{1}{|\mathcal{B}|} \sum_{\mathcal{B}} \ell(\widetilde{p}, y; \boldsymbol{\theta})$: compute training loss (Eq. (5.8));
      $\boldsymbol{\theta}_i \leftarrow \boldsymbol{\theta} - \eta_t \bigtriangledown_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}; \alpha_i)$: local parameters update;
  return $\boldsymbol{\theta}_i$

Inference *(i, θ)*
  Prepare $D_{val} = \{\boldsymbol{X}, Y_i | \alpha_i\}$;
  $\widehat{p} = f_{\boldsymbol{\theta}}(\boldsymbol{x}; \alpha_i)$: make predictions;
  $\mathcal{L}(\boldsymbol{\theta}; \alpha_i) = \frac{1}{|\mathcal{D}_{val}|} \sum_{\mathcal{D}_{val}} \ell(\widehat{p}, y; \boldsymbol{\theta})$: validation loss (Eq. (5.12));
  return $\mathcal{L}(\boldsymbol{\theta}; \alpha_i)$

5.4 Experimental Evaluation

In this section, we present the details of our experimental setting. We first introduce a real-world humor recognition dataset and the preparation for personalized humor recognition. Then, we introduce the evaluation metrics used for performance comparison. Finally, we describe our model setting.

5.4.1 Dataset Description

Most of the existing labeled datasets for humor recognition provided either 0/1 labels or only one rating (averaged across annotators) for each joke. That is, they removed the personal preference effect during data annotation. However, the diversity of personal preferences is especially important in distributed training settings. In our experiments, we use a newly published dataset from SemEval-2020 shared Task 7[1] - assessing the funniness of edited news headlines [Hossain et al., 2019], which provided different ratings from different users for each humorous headline. Each created headline was sent to five crowdsourced annotators through Amazon Mechanical Turk and each annotator was required to rate its funniness using a score from the integer interval [0, 1, 2, 3]. For example, a news headline *"Royal wedding: Meghan's dress in detail"* was micro-edited by replacing $dress$ with $elbow$ to produce a funny version *"Royal wedding: Meghan's elbow in detail"*, which received five ratings: 0, 1, 3, 3 and 3. The original dataset made a consensus assumption on the perceived funniness of the edited headline. They reassigned an average of the five ratings, 2.0, to be the funniness rating for this joke. This example shows a common assumption of an average user preference does not reflect the diversity among different users. The statistics of the dataset are summarized in Table 5.1.

TABLE 5.1: Statistics of the public dataset

	Train	Validation	Test
Number of samples	9,652	2,419	3,024
Average Rating	0.936	0.935	0.940
Minimum Rating	0.000	0.000	0.000
Maximum Rating	3.000	3.000	2.800

[1]SemEval 2020 Task 7 (Sub-task 1) dataset downloaded from: `https://github.com/n-hossain/semeval-2020-task-7-humicroedit`

5.4.2 Implicit Label Generation

Jokes can be perceived differently and thus, receive different funniness ratings from users. Unfortunately, there are no public datasets for training an NLP model under the problem of label distribution shift across people. It is also difficult to collect true feedback from a diverse population in a privacy protection setting. To simulate the diversity in the human perception of jokes, we generate a synthetic binary humor recognition dataset to study this problem from the SemEval ratings dataset.

- First, sort the jokes by their original average funniness ratings. Specifically, given a set of n jokes rated by the content publisher with funniness ratings drawn from a non-negative ordinal interval $I = [s_{min}, s_{max}]$, e.g., $I = [0, 3]$. $\mathcal{D} = \{(\boldsymbol{x}_j, s_j))\}_{j=1}^{n}$, where $s_j \in I$ is the original dataset with ratings sorted in ascending order (Figure 5.5(a)). Users receiving this dataset would agree on the jokes with funniness ratings that are close to the boundaries of the rating interval (i.e. not funny (s_{min}) and very funny (s_{max})) and may disagree on the jokes with ratings lying in between (i.e. slightly funny and moderately funny ($s_{min} < s_j < s_{max}$)).
- Second, assume that every user can have only one unique humor preference, denoted as α_i. Based on a user's historical implicit feedback, the user's humor preference, α_i, is reflected as a specific threshold within the funniness rating interval, i.e., $s_{min} < \alpha_i < s_{max}$. The user dislikes jokes below this threshold rating, $s_j < \alpha_i$, and likes jokes above it, $s_j \geq \alpha_i$ (Figure 5.5(b)). In effect, due to the limited number of training samples observed for every funniness rating, $\forall \alpha_i \in [\alpha_i, \tau)$ is treated as the same humor preference when τ is small enough.

Based on the first assumption, we create a set of implicit labels, $Y_i = \{y^j\}_{j=1}^{n}$ for users with α_i on the same jokes $\boldsymbol{X}$. Based on the second assumption, we can create multiple sets of implicit binary labels of diverse distributions $p(y|\boldsymbol{x}, \alpha_i)$, where $y_j = 0$ for $s_j < \alpha_i$ and $y_j = 1$ for $s_j \geq \alpha_i$.

To simulate the real-world diversity of perceived funniness, we generate a diverse population with humor preferences α ranging from $\alpha = 0.2$ to $\alpha = 2.0$ with a step size of 0.1 (i.e., $\tau = 0.1$), as illustrated in Figure 5.6. The higher the α value, the more non-humorous labels will be generated. In our experiments, a valid user is regarded as having $\alpha_i \in [0.2, 2.0]$. Specifically, the funniness ratings of the dataset are transformed into implicit binary labels generated by a given user with the preference α on his own

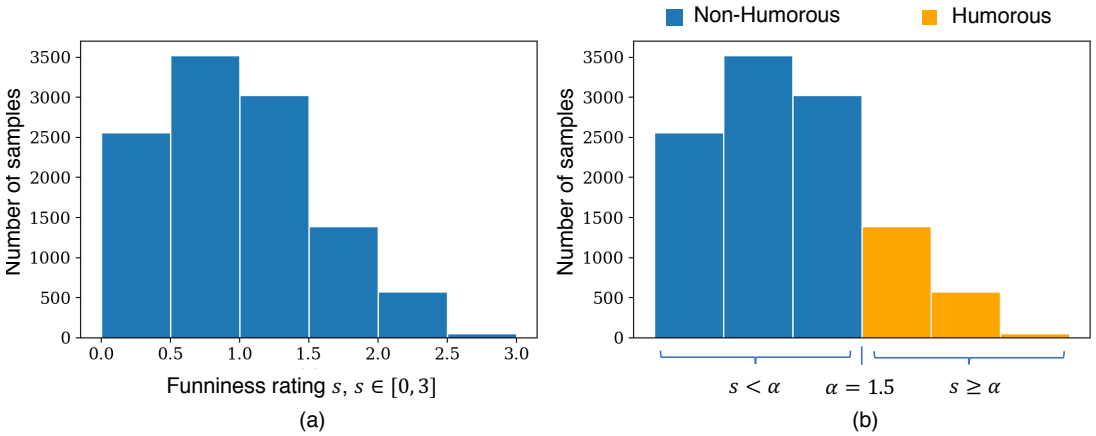


FIGURE 5.5: Transform explicit ratings into binary labels. The distribution of explicit funniness ratings on a set of jokes rated by the content publisher is shown in discrete intervals (a). An example in which a user's humor preference is quantified by $\alpha = 1.5$ (b). *Best viewed in color.*

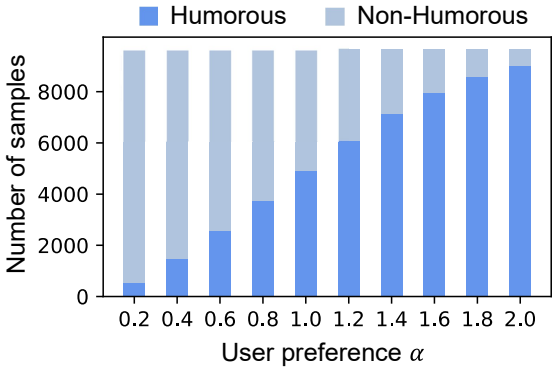


FIGURE 5.6: Generated binary labels with different distributions from funniness ratings when diverse humor preferences are considered. *Best viewed in color.*

device. The distribution over the generated implicit labels is $p(y|\boldsymbol{x}, \alpha)$. The boundary values (e.g., $\alpha = 0$ and $\alpha = 3$) that result in a one-class dataset are not considered in our experiments.

5.4.3 Evaluation Metrics

To evaluate the classification performance of FedHumor and baseline approaches, we use the *macro-averaged* F_1 score as our main metric, which is the average of the F_1 scores of two classes. We additionally report the *macro-averaged* precision and recall scores. Note that a *macro-averaged* metric implies that the generalization performance on all

classes is equally important. This is essential in personalized humor recognition, because some users with a very low or very high threshold of humor preference may produce very imbalanced labels. In this case, a model can achieve artificially high accuracy by predicting all the samples to be the dominant class. Thus we do not use the accuracy metric for evaluation. All the methods are compared on the hold-out test set. We use *macro-averaged* metrics for evaluation.

5.4.4 Model Setting

FedHumor is built based on top of the base version of BERT [Devlin et al., 2019]. The pretrained weights we utilize include its word embedding layer and twelve transformer layers[2]. A pooling layer nonlinearized by $tanh$ activation function[3] is applied on the first ([CLS]) token representation from the last transformer layer to be the final contextualized sentence representation. A dropout of 0.1 is applied on the pooled output and is further sent to a classifier, which is a fully connected layer randomly initialized from a uniform distribution, $\mathcal{U}(-\sqrt{1/d}, \sqrt{1/d})$, where d is the size of the input features[4]. In our model, $d = 768$, the length of the hidden vector corresponding to the $[\mathrm{CLS}]$ token. We follow the recommended hyperparameter setting introduced in the paper [Devlin et al., 2019] to tune learning rate, batch size, and weight decay for fine-tuning BERT on downstream tasks based on Federated Averaging setting[5].

In this section, we conduct experiments to evaluate the performance of FedHumor. We compare it against nine state-of-the-art humor recognition methods on the hold-out test set. We also study the properties of FedHumor through experiments and compare it with other learning approaches to show the advantages of following the federated learning paradigm.

[2]PyTorch implementation of the pretrained BERT model: `https://huggingface.co/transformers/pretrained_models.html`

[3]PyTorch implementation of the tanh activation function: `https://pytorch.org/docs/stable/generated/torch.nn.Tanh.html`

[4]PyTorch initialization of linear layer: `https://pytorch.org/docs/stable/generated/torch.nn.Linear.html`

[5]Code downloaded from: `https://github.com/shaoxiongji/federated-learning`

5.4.5 Comparison of Different Training Strategies

Given that the property of our data is Differently and independently distributed, we first compare different training strategies against the federated weight-tying method.

- **INDV**: A simple approach to this problem is to fine-tune a BERT model for each *individual* user based on their local ground truth labels. This method represents the traditional way of providing personalized humor recognition, which creates a unique model for each user. Yet, it's computationally expensive and might easily overfit to small imbalanced datasets in practice. Moreover, when new users registered, the server has to establish a new model to be trained for them.
- **AGG**: The second approach is to first *aggregate* all the labeled data from each user into a central database, and treat all of them as ground truth labels to train a BERT-based humor recognition model. This follows the traditional way of training a single-task machine learning model. However, this approach needs to have access to users' data, which may cause privacy breaches.
- **FED**: The third approach is to tie the weights of each individual BERT model through *federated learning*. In this setting, users can share learned BERT parameters while keeping their personal adaptation module.

To test these approaches under different levels of preference diversity, we prepare two groups of user preferences as shown below. Due to the limit of dataset size and the funniness score range, the maximum number of different user preferences we can generate from the given dataset is 18.

- Group 1: a group of three users with different preferences: $\alpha = 0.3$ represents the easy-to-amuse personality, $\alpha = 0.9$ represents the neutral personality, and $\alpha = 1.8$ represents the hard-to-amuse personality;
- Group 2: a group of 18 users with different preferences: α ranging from 0.2 to 1.9 in increments of 0.1, representing a more diverse population.

We compare the average test performance of a group between the three models. Comparison Results are shown in Table 5.2. Take the column of F score as an example. Aggregated training approach works the worst, meaning that training data with conflict

labels can degrade the performance. Individual training comes after, meaning that such issues can be alleviated by separating them. Federated training further improves performance, meaning that allowing weight-tying between different users in the federated setting can bring performance gains. Comparing the two groups, the more diverse the users, the better the Federated Learning approach performs compared to the rest.

TABLE 5.2: Average test performance (in %) of three learning strategies on two groups of users. Values in bold denote the best results. Underlined values indicate the second-best results.

		Precision	Recall	F_1 score
Group 1	AGG	<u>58.59</u>	54.89	41.66
	INDV	56.30	<u>55.32</u>	<u>53.52</u>
	FED	**60.03**	**65.57**	**55.61**
Group 2	AGG	57.40	51.25	33.05
	INDV	<u>58.14</u>	<u>55.61</u>	<u>53.03</u>
	FED	**61.67**	**66.62**	**57.48**

5.4.6 Hyperparameter Sensitivity Analysis

As introduced in Section 5.3.3, we allow a hyperparameter β on each user to control how much we want to scale the model's estimated probabilities $P(y|x, \alpha_i)$ w.r.t. the marginal local distribution $P(y|x, \alpha_i)$. As such, we conduct a sensitivity analysis through grid search to reveal whether it affects the performance of FedHumor and how to set β for each user. To do so, we vary the values of α from low ($\alpha = 0.2$) to high ($\alpha = 2.0$) and at the same time, increase the scale factor β from small ($\beta = 0$) to large ($\beta = 2.0$), all in increments of 0.1. This results in 399 experiments in total. The implicit binary labels are generated each time the α value changes. We train the BERT-based classification model for each combination of α and β and the best corresponding model is selected on the validation set. We report the macro-averaged F_1 score on the hold-out test set.

The results on the test set are shown in Figure 5.7. In general, a model can achieve better generalization performance on the preference range of $[0.5, 1.5]$. From Figure 5.6, we can see that this preference interval leads to the range of empirical probability from $P(y = 1|\boldsymbol{x}, \alpha = 0.5) \approx 0.2$ to $P(y = 1|\boldsymbol{x}, \alpha = 1.5) \approx 0.75$. In this case, for every α value, there are multiple β values that can achieve the best performance. This means that our model is not very sensitive to β and a reasonable value would suffice. For users who have very low ($\alpha < 0.5$) or very high ($\alpha > 1.5$) humor preference values, their

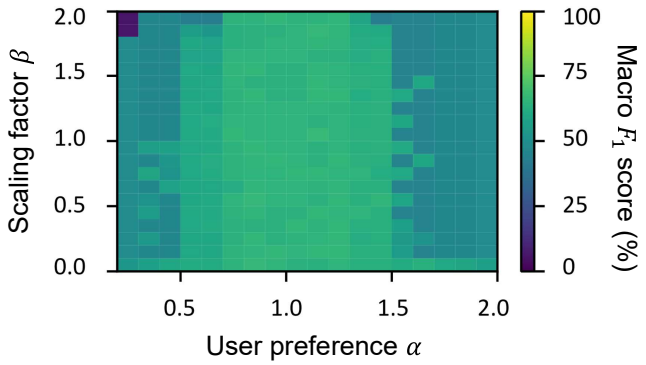


FIGURE 5.7: Tune hyper-parameter β w.r.t. α for single user. *Best viewed in color.*

labels are too imbalanced for the model to achieve good generalization performance. In this case, β is recommended to be set to a low value (e.g., around 0.1).

5.4.7 Comparison of Different Humor Recognition Models

Finally, we compare our model with three kinds of baselines. The first three models are the representative humor recognition models in literature, which do not use pretrained language model. The second six models are different versions of pretrained language models with fine-tuning. Our model use both a pretrained language model and federated training on diverse humor preferences.

- **DV-LR**: We trained Doc2Vec using the distributed bag of words approach [Mikolov et al., 2013b] for sentence representation and applied logistic regression classifier on the features for classification.
- **WV-RF**: We reproduced the humor recognition model in [Yang et al., 2015], which used a pretrained word2vec[6] model for sentence representation and a Random Forest classifier for classification.
- **WV-CNN-HN**: We reproduced the deep learning-based humor recognition model in [Chen and Soo, 2018], which augmented a CNN with a Highway Network for end-to-end humor recognition[7].

[6]Google word2vec downloaded from: `https://code.google.com/p/word2vec/`
[7]PyTorch implementation of Highway Network: `https://gist.github.com/dpressel/3b4780bafcef14377085544f44183353`

- **BERT-FZ**: We use the BERT model pretrained on lower-cased English text with 768 hidden size and 12 transformer layers. We *freeze* the pretrained model parameters and only update the parameters of a fully connected classification layer.
- **BERT-FT**: Different from the previous method, we *fine-tuned* the pretrained parameters together with the classifier parameters. This is a strong baseline as many downstream NLP tasks have shown improved performance through fine-tuning BERT on the task-specific datasets [Devlin et al., 2019]. It is also the same base model adopted by FedHumor.
- **BERT-L/C/M**: Advanced by huge model capacity trained on large corpora, pretrained language models can provide generally effective representations across domains. As such, we adopt other versions of BERT. BERT-L is pretrained with a much *larger* hidden size (1024) and deeper (24) layers. BERT-C means the true *case* and accent markers are preserved when training BERT. BERT-M means a *multilingual* BERT model that was pretrained on lower-cased text in the top 102 languages with the largest Wikipedia. The three models are all *fine-tuned* together with the classifier to form three baseline approaches.
- **ALBERT**: There are also many strong pretrained language models after BERT. We adopt ALBERT [Lan et al., 2019], which had been evaluated to have better scalability on downstream tasks than BERT. We fine-tune its pretrained parameters together with a classification layer on the humor recognition task for comparison.

Test Results are shown in Table 5.3. The metrics are the macro average values over all classes. Generally, pretrained language models with domain adaptive fine-tuning outperform all traditional models. Our model trained based on the relaxed assumption, exceeds all baselines across metrics.

The first three models (i.e., DV-LR, WV-RF, WV-CNN-HN) use static word representations which were widely adopted in humor recognition tasks before the advent of large pretrained language models. Their performance is generally not good on this dataset compared to pretrained language model-based methods. The following six models (i.e., BERT-FZ, BERT-FT, BERT-L/C/M, ALBERT) are BERT variants. The results achieved by BERT-FZ are much worse than the rest. This shows that the pretrained language models, whose learning phase does not take advantage of labeled data, can only provide general representations. They should be fine-tuned towards a particular task by further learning from domain-specific label information. BERT-L is close to but lower than

TABLE 5.3: Test performance (in %) on the user with $\alpha = 1.0$ achieved by FedHumor and all baseline approaches. Values in bold indicate the best results. Underlined values indicate the second-best results.

	Precision	Recall	F_1 score
DV-LR	53.69	53.67	53.64
WV-RF	56.70	56.10	55.20
WV-CNN-HN	56.20	54.70	51.90
BERT-FZ	54.15	53.71	52.53
BERT-FT	<u>64.91</u>	<u>64.88</u>	<u>64.87</u>
BERT-L	64.48	64.48	64.47
BERT-C	62.69	62.65	62.62
BERT-M	62.11	62.08	62.06
ALBERT	61.06	61.05	61.04
FedHumor	**66.60**	**66.56**	**66.53**

BERT-FZ showing that larger model parameters do not bring performance gain. BERT-C and BERT-M are lower than BERT-FZ indicating the preservation of the true case and knowledge from other languages did not benefit this task. ALBERT, though tested to be more scalable to downstream tasks than BERT on some datasets, also failed to outperform BERT on this task. FedHumor achieved the best results across all three evaluation metrics and 2.5% relative improvement on F_1 score than the second best result. This reveals that it is not necessary to reconcile the label difference between users. By leveraging the diversity and training the model in a distributed manner, we can improve the model's performance in personalized humor recognition.

5.5 Summary

In this chapter, we propose FedHumor - a humorous text recognition model following the federated learning paradigm - to perform personalized humor recognition based on labels from distributed data sources. To the best of our knowledge, FedHumor is the first federated learning-based humor recognition model that explicitly considers the diversity of humor preferences. We conducted the personalized humor recognition by transforming existing humor recognition datasets labeled with explicit ratings into the dataset labeled with implicit binary labels reflecting different humor preferences. Extensive experiments against nine state-of-the-art approaches show that FedHumor achieves the best performance, surpassing the best-performing existing approach by 2.5% in terms of

F_1 score. FedHumor represents a promising direction for personalized recommendation of humorous content under tightened data privacy protection regulations [Yang et al., 2019], thereby enabling innovative forms of human-AI interaction to emerge.

We identify a few limitations of current works:

- First, the studies are limited by the availability of suitable real-world datasets. The evaluation setting is designed to cover all kinds of user preferences. However, in practice, we may only be able to collect datasets labeled according to a partial set of user preferences.
- Second, the disparity in labels not only comes from people with different humor preferences but can also result from time-delayed awareness. For example, [Mottini and Chowdhury, 2019] observed that the same joke could be labeled differently even by the same person after some time. This problem relates to the research field of concept shift [Lu et al., 2018], which has made a lot of progress in adapting the model to changing target functions over time [Frías-Blanco et al., 2016, Cano et al., 2019].

Chapter 6

Conclusions and Future Works

6.1 Conclusions

In this thesis, we explored the approaches to enhance the domain adaptation of PLMs with the goal of using less labeled training data to achieve better performance. We conduct research under three scenarios from the input to the output of a model:

1) With the model parameters and target labels fixed, how to adapt a PLM to the target domain by only updating the input data?

2) With the input data and the target labels fixed, how to adapt a PLM to the target domain by optimizing the model representations?

3) With the input data and the model fixed while the target label distributions vary across different users, how to adapt the same PLM to multiple users?

We propose three new techniques which obtained nontrivial data efficiency: 1) OPTIMA can adapt frozen PLMs to new domains with zero-shot and few-shot training samples via soft prompts that are trained under adversarial perturbations with a domain adaptation objective; 2) LOANT can adapt PLMs to small datasets with better performance by effectively transferring knowledge from data-abundant domains via better optimization among competing losses, and 3) FedHumor enables adaptation of the same PLM to different domains which naturally have different label distributions in a distributed setting. To summarize, we make the following conclusions:

- We find that the effectiveness of the prompt tuning technique, which adapts frozen PLMs to downstream tasks, heavily depends on the availability of sufficient labeled training data. Initializing soft prompts with checkpoints that are pretrained on a related source domain can usually boost zero-shot and few-shot learning performance, however, is prone to negative transfer under irregular domain shift. By utilizing unlabeled data from the target domain and pretraining soft prompt with OPTIMA, we achieve superior performance than all potential and competitive baselines under zero-shot and few-shot settings.

- We find that training a shared-private neural network on top of PLMs under domain adversarial training between a larger-scale source-domain dataset and a smaller-scale target-domain dataset can enforce the model to be dominated by the source domain data. LOANT introduces a look-ahead learning strategy that accommodates competing losses under adversarial neural transfer, thereby improving learning performance on the target small dataset. We show that LOANT outperforms traditional adversarial neural transfer, multi-task learning, and meta-learning baselines, and establishes new state-of-the-art results on sarcasm detection datasets.

- We find that different people tend to have different preferences on the same subjective tasks, which can undermine the performance of the PLMs when applied to different domains such as users' mobile devices where labels are kept private. We relax the common assumption about humor recognition that users have a consensus about whether or not a given text is humorous and allow users to keep their own humor preferences. By doing so, FedHumor can learn better to adapt the same PLM to different distributed domains, especially those that naturally have highly imbalanced classes. We conducted extensive experiments to evaluate our proposed approach and show that our approach is significantly superior to existing approaches in terms of personalizing humor recognition.

6.2 Future Works

6.2.1 Pushing the Limit of Domain Adaptation

Although unlabeled data are easy to obtain in most cases, doing so might be difficult for some data-scarce domains and languages. Therefore, reducing the use of even unlabeled

data is necessary for those low-resource domains and languages. OPTIMA addresses the situation where the source and target domains have similar data distributions. When the two distributions are exactly the same, the technique degenerates to simply adversarial training. When the two distributions are extremely dissimilar, the transfer is unlikely to yield performance improvements. Therefore, a unified framework that automatically detects domain distances and applies the correct method may be desirable. Moreover, the power of perturbations has the most effect in the few-shot and zero-shot settings. When the target domain has abundant labeled data, the gap between soft prompt tuning and our method will diminish. LOANT enhances transferability across domains, however, at a sacrifice of extra forward and backward computations on the task and domain classifiers. It is only cheap when these classifiers are shallow. More look-ahead steps may offer higher performance gains, but the trade-off between the gains and the computational cost needs to be studied. FedHumor enables adapting the same PLM to distributed domains while keeping their labels private. At the heart of FedHumor is the diversification mechanism which we found to be most effective when the classes are not too extremely imbalanced. It still remains a big problem for deep neural networks to learn effectively under the challenges of imbalance, long tail, and class scarcity. Introducing approaches that can address these machine learning problems to the problem of personalizing PLMs to different domains is promising.

6.2.2 Low-resource Learning

A range of parameter-efficient adaptation methods has shown to be an alternative to finetuning for PLMs. With a few parameters tuned, the performance on downstream tasks can be comparable with finetuning [Devlin et al., 2019, Lester et al., 2021, Houlsby et al., 2019]. The rising research on parameter-efficient methods for PLMs adaptations and some pioneering transfer learning studies under this sector [Gu et al., 2022, Guo et al., 2022a] show the promise of low-recourse learning with PLMs. However, despite exciting results, little has been explored about the critical ingredients and mechanisms that made these methods work. On one hand, studies in [He et al., 2021] show that Adapter-based tuning can better mitigate the forgetting problems than finetuning on low-resource and cross-lingual tasks. On the other hand, studies in [Gu et al., 2022, Guo et al., 2022a] show that prompt tuning underperforms finetuning in few-shot settings. These shreds of evidence suggest that more exploration studies need to be designed to understand

the underlying mechanisms of different parameter-efficient adaptation methods in low-resource learning.

A new paradigm: PLM-based data generation for low-resource domains. Previous research has shown that developing complicated algorithms to adapt a PLM to a low-resource domain depends on the problem setting. One algorithm does not fit all problem settings. However, if we have enough data for every task, we can apply general machine learning algorithms. To achieve this, I propose a new paradigm for low-resource learning using publicly available Pretrained Language Models to generate datasets. To move forward, we need to address key research challenges. First, the generated data by PLMs can be noisy, resulting in wrongly associated class labels. To solve this, I propose combining contrastive learning and prompting research to identify and correct mislabelled data using nearest neighbors. Second, the generated data can be irrelevant to the given task, which stems from a lack of sufficient task information in the prompts. E.g., it may generate a political statement for a movie review sentiment analysis task. To solve this, I propose using InstructGPT [Ouyang et al., 2022] to generate enhanced prompts, which removes the limitation of human-written prompts. Finally, when the generated dataset is very huge, it is inevitable to contain some low-quality samples. To effectively use the generated dataset, I'll develop a robust training algorithm based on explainable sample importance. The main idea is to encourage the downstream model to learn from useful samples while being robust against low-quality samples. The system can be used to empower a wide range of real-world applications, as shown in Figure 6.1.

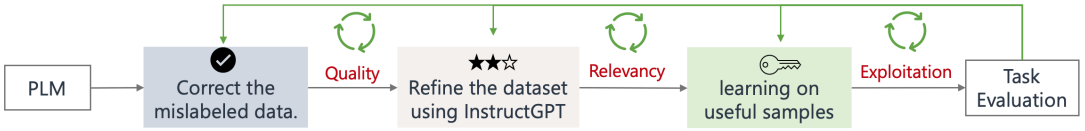


FIGURE 6.1: First, we write a prompt for a given task, to steer a Pretrained Language Model to generate a set of data candidates. The system will first identify the mislabeled data and correct labels using the first solution. Then, the system will examine out irrelevant data and use InstructGPT to automatically refine the dataset without external supervision from humans. Finally, the system will exploit the generated dataset by training the downstream model with our proposed training algorithm. The system is not only efficient because we don't train any model parameters in the generation process, but also flexible as we can turn back to any previous stage to enhance the performance.

List of Author's Awards, Patents, and Publications

Awards

- **Best Application Paper Award**, "Federated Learning with Diversified Preference for Humor Recognition," *International Workshop on Federated Learning for User Privacy and Data Confidentiality in Conjunction with IJCAI 2020 (FL-IJCAI'20)*.
- **Best Presentation Award**, "Latent-Optimized Adversarial Neural Transfer for Sarcasm Detection", *Pattern Recognition and Machine Intelligence Association (PREMIA) Best Student Paper Awards 2021*.
- **The Certificate of Commendation**, "Federated Learning for Personalized Humor Recognition", *Pattern Recognition and Machine Intelligence Association (PREMIA) Best Student Paper Awards 2022*.
- **WiEST Development Grant**, Women at NTU, Singapore, 2023.

Technology Disclosures

- **Xu Guo**, Han Yu and Chunyan Miao, "Weakly Supervised Learning for Decision Support in Pain Management," Technology Disclosure for Nanyang Technological University (TD-2020-145), 11/05/2020.

Conference Publications

- **Xu Guo**, Boyang Li and Han Yu. Improving the Sample Efficiency of Prompt Tuning with Domain Adaptation. In Findings of EMNLP, 2022.
- Fei Luo, Hangwei Qian, Di Wang, **Xu Guo**, Yan Sun, Eng Sing Lee, Hui Hwang Teong, Ray Tian Rui Lai and Chunyan Miao. Missing Value Imputation for Diabetes Prediction. In IJCNN, pages 1-8, 2022.
- **Xu Guo**, Boyang Li, Han Yu, and Chunyan Miao. Latent-Optimized Adversarial Neural Transfer for Sarcasm Detection. In NAACL, pages 5394–5407, 2021.
- **Xu Guo**, Han Yu, Chunyan Miao and Yiqiang Chen. Agent-based Decision Support for Pain Management in Primary Care Settings. In IJCAI, demo track, pages 6521-6523, 2019.

Journal Articles

- **Xu Guo**, Han Yu, Boyang Li, Hao Wang, Pengwei Xing, Siwei Feng, Zaiqing Nie, and Chunyan Miao. Federated Learning for Personalized Humor Recognition. In ACM Transactions on Intelligent Systems and Technology, Pages 1-18, 2022.
- **Xu Guo**, Han Yu, Yiqiang Chen and Chunyan Miao, "Weakly Supervised Neural Representation Learning through Exploiting Expert Knowledge," International Journal of Information Technology, Volume 25 (1), Pages 1-9, 2019.

Preprints

- **Xu Guo** and Han Yu, "Domain Adaptation and Generalization of Pretrained Language Models: A Survey," arXiv:2211.03154.

Bibliography

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018a. 1, 16

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 1, 3, 16

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. 1, 37, 54, 62, 63, 78, 86, 90, 95

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2020. 1, 16

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Ro{bert}a: A robustly optimized {bert} pretraining approach, 2020. 1, 16

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 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*, Online, July 2020. Association for Computational Linguistics. 1, 16

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21 (140):1–67, 2020a. 1, 16

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021. 1

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017a. 1

Jihang Mao and Wanli Liu. A bert-based approach for automatic humor detection and scoring. In *IberLEF*, 2019. 1, 16, 73, 74

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022. 2, 16

Brian Thompson, Jeremy Gwinnup, Huda Khayrallah, Kevin Duh, and Philipp Koehn. Overcoming catastrophic forgetting during domain adaptation of neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2062–2068, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. 2, 25, 34

Dogu Araci. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*, 2019. 2, 24

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online, November 2020. Association for Computational Linguistics. 2, 24

Timothy Miller, Egoitz Laparra, and Steven Bethard. Domain adaptation in practice: Lessons from a real-world information extraction pipeline. In *Proceedings of the Second Workshop on Domain Adaptation for NLP*, pages 105–110, Kyiv, Ukraine, April 2021. Association for Computational Linguistics. 2

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009. 2, 18, 19

Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016. 2, 19

Alan Ramponi and Barbara Plank. Neural unsupervised domain adaptation in NLP—A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6838–6855, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. 2, 18, 19

Tiezheng Yu, Zihan Liu, and Pascale Fung. AdaptSum: Towards low-resource domain adaptation for abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5892–5904, Online, June 2021a. Association for Computational Linguistics. 3, 25, 34

Minho Ryu, Geonseok Lee, and Kichun Lee. Knowledge distillation for bert unsupervised domain adaptation. *Knowledge and Information Systems*, 64(11):3113–3128, 2022. 3, 27, 34

Dustin Wright and Isabelle Augenstein. Transformer based multi-source domain adaptation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7963–7974, Online, November 2020. Association for Computational Linguistics. 3

Constantinos Karouzos, Georgios Paraskevopoulos, and Alexandros Potamianos. UDALM: Unsupervised domain adaptation through language modeling. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2579–2590, Online, June 2021. Association for Computational Linguistics. 3, 25, 34

Hakan Bilen and Andrea Vedaldi. Universal representations: The missing link between faces, text, planktons, and cat breeds. *arXiv preprint arXiv:1701.07275*, 2017. 3

Entony Lekhtman, Yftah Ziser, and Roi Reichart. 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, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. 3, 24, 34

Xu Guo, Boyang Li, Han Yu, and Chunyan Miao. Latent-optimized adversarial neural transfer for sarcasm detection. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5394–5407, Online, June 2021a. Association for Computational Linguistics. 3, 26, 27, 28, 34

Zhuoyi Wang, Yuqiao Chen, Chen Zhao, Yu Lin, Xujiang Zhao, Hemeng Tao, Yigong Wang, and Latifur Khan. Clear: Contrastive-prototype learning with drift estimation for resource constrained stream mining. WWW '21, page 1351–1362, New York, NY, USA, 2021a. Association for Computing Machinery. ISBN 9781450383127. 3

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989. 3, 25

Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55, 2001. URL `https://people.math.harvard.edu/~ctm/home/text/others/shannon/entropy/entropy.pdf`. 8

Frederick Jelinek. Continuous speech recognition by statistical methods. *Proceedings of the IEEE*, 64(4):532–556, 1976. 8

Alex Franz and Thorsten Brants. All our n-gram are belong to you, 2006. URL `https://ai.googleblog.com/2006/08/all-our-n-gram-are-belong-to-you.html`. 10

Jianfeng Gao and Kai-Fu Lee. Distribution-based pruning of backoff language models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 579–588, Hong Kong, October 2000. Association for Computational Linguistics. doi: 10.3115/1075218.1075291. URL https://aclanthology.org/P00-1073. 10

David Talbot and Miles Osborne. Smoothed Bloom filter language models: Tera-scale LMs on the cheap. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 468–476, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL https://aclanthology.org/D07-1049. 10

Stanley F Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–394, 1999. 11

Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. A neural probabilistic language model. *Advances in neural information processing systems*, 13, 2000. URL https://proceedings.neurips.cc/paper/2000/file/728f206c2a01bf572b5940d7d9a8fa4c-Paper.pdf. 11

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3(null):1137–1155, mar 2003. ISSN 1532-4435. 11

Geoffrey E Hinton. Distributed representations. 1984. 11

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a. 12

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013b. 13, 89

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 13

Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. *Advances in neural information processing systems*, 28, 2015. 14

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. 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 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://aclanthology.org/N18-1202. 14

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. Learned in translation: Contextualized word vectors. *Advances in neural information processing systems*, 30, 2017. 14

Matthew E. Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1756–1765, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1161. URL `https://aclanthology.org/P17-1161`. 14

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017b. 15

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *Paper presented at 3rd International Conference on Learning Representations, ICLR 2015, San Diego, United States.*, 2015. 15

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550. URL `https://aclanthology.org/2020.emnlp-main.550`. 16

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*, 2019. 16

Zhengyan Zhang, Yuxian Gu, Xu Han, Shengqi Chen, Chaojun Xiao, Zhenbo Sun, Yuan Yao, Fanchao Qi, Jian Guan, Pei Ke, et al. Cpm-2: Large-scale cost-effective pre-trained language models. *AI Open*, 2:216–224, 2021. 16

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 16

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022. 16

Erik Nijkamp, Bo Pang, Hiroaki Hayashi, Lifu Tu, Huan Wang, Yingbo Zhou, Silvio Savarese, and Caiming Xiong. Codegen: An open large language model for code with multi-turn program synthesis. *arXiv preprint arXiv:2203.13474*, 2022. 16

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, et al. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, 2022. 16

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 16, 22, 42

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022. 16, 96

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021a. 16

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022. 16

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018b. 17

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006. 17

John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447, Prague, Czech Republic, June 2007. Association for Computational Linguistics. 17

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010. 17

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010. 17

David Grangier and Dan Iter. The trade-offs of domain adaptation for neural language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3802–3813, Dublin, Ireland, May 2022. Association for Computational Linguistics. 17, 20

Vishal M Patel, Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Visual domain adaptation: A survey of recent advances. *IEEE signal processing magazine*, 32(3): 53–69, 2015. 19

Gabriela Csurka. A comprehensive survey on domain adaptation for visual applications. *Domain adaptation in computer vision applications*, pages 1–35, 2017. 19

Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018. 19

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation with multiple sources. *Advances in neural information processing systems*, 21, 2008. 19

Shiliang Sun, Honglei Shi, and Yuanbin Wu. A survey of multi-source domain adaptation. *Information Fusion*, 24:84–92, 2015. 19

Danielle Saunders. Domain adaptation and multi-domain adaptation for neural machine translation: A survey. *Journal of Artificial Intelligence Research*, 75:351–424, 2022. 19

Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897, 2020. 19

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heinz, and Dan Roth. Recent advances in natural language processing via large pre-trained language models: A survey. *arXiv preprint arXiv:2111.01243*, 2021. 19

Junyi Li, Tianyi Tang, Wayne Xin Zhao, and Ji-Rong Wen. Pretrained language models for text generation: A survey. *arXiv preprint arXiv:2105.10311*, 2021. 19

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models. *arXiv preprint arXiv:2203.06904*, 2022. 19, 32, 35

Art B. Owen. *Monte Carlo theory, methods and examples*. 2013. 19

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. 20

Wei Wang, Taro Watanabe, Macduff Hughes, Tetsuji Nakagawa, and Ciprian Chelba. Denoising neural machine translation training with trusted data and online data selection. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 133–143, Brussels, Belgium, October 2018. Association for Computational Linguistics. 20

Marlies van der Wees, Arianna Bisazza, and Christof Monz. Dynamic data selection for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1410, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. 20

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1885–1894. PMLR, 06–11 Aug 2017. 20

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 33:19920–19930, 2020. 20

Xinyi Wang, Ankur Bapna, Melvin Johnson, and Orhan Firat. Gradient-guided loss masking for neural machine translation. *arXiv preprint arXiv:2102.13549*, 2021b. 20

Tasnim Mohiuddin, Philipp Koehn, Vishrav Chaudhary, James Cross, Shruti Bhosale, and Shafiq Joty. Data selection curriculum for neural machine translation. *arXiv preprint arXiv:2203.13867*, 2022. 20

Dan Iter and David Grangier. On the complementarity of data selection and fine tuning for domain adaptation. *arXiv preprint arXiv:2109.07591*, 2021. 20

Xiaofei Ma, Peng Xu, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. Domain adaptation with BERT-based domain classification and data selection. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 76–83, Hong Kong, China, November 2019. Association for Computational Linguistics. 20, 34

Jing Zhang, Zewei Ding, Wanqing Li, and Philip Ogunbona. Importance weighted adversarial nets for partial domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8156–8164, 2018. 20

Zhangjie Cao, Kaichao You, Mingsheng Long, Jianmin Wang, and Qiang Yang. Learning to transfer examples for partial domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2985–2994, 2019. 20

David McClosky, Eugene Charniak, and Mark Johnson. Effective self-training for parsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 152–159, New York City, USA, June 2006. Association for Computational Linguistics. 21

Minghao Chen, Shuai Zhao, Haifeng Liu, and Deng Cai. Adversarial-learned loss for domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3521–3528, 2020. 21, 34

Abdellah El Mekki, Abdelkader El Mahdaouy, Ismail Berrada, and Ahmed Khoumsi. Domain adaptation for Arabic cross-domain and cross-dialect sentiment analysis from contextualized word embedding. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2824–2837, Online, June 2021. Association for Computational Linguistics. 21, 34

Hai Ye, Qingyu Tan, Ruidan He, Juntao Li, Hwee Tou Ng, and Lidong Bing. Feature adaptation of pre-trained language models across languages and domains with robust self-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7386–7399, Online, November 2020. Association for Computational Linguistics. 21, 34

Hong Liu, Jianmin Wang, and Mingsheng Long. Cycle self-training for domain adaptation. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 22968–22981. Curran Associates, Inc., 2021. 21, 34

Ananya Kumar, Tengyu Ma, and Percy Liang. Understanding self-training for gradual domain adaptation. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5468–5479. PMLR, 13–18 Jul 2020. 21

Huazheng Wang, Zhe Gan, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, and Hongning Wang. Adversarial domain adaptation for machine reading comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2510–2520, Hong Kong, China, November 2019a. Association for Computational Linguistics. 21, 26, 34

Ehsan Mohammady Ardehaly and Aron Culotta. Domain adaptation for learning from label proportions using self-training. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, page 3670–3676. AAAI Press, 2016. ISBN 9781577357704. 21

Jianfei Yu, Chenggong Gong, and Rui Xia. Cross-domain review generation for aspect-based sentiment analysis. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4767–4777, Online, August 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.421. URL `https://aclanthology.org/2021.findings-acl.421`. 22, 34

Junjie Li, Jianfei Yu, and Rui Xia. Generative cross-domain data augmentation for aspect and opinion co-extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4219–4229, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.312. URL `https://aclanthology.org/2022.naacl-main.312`. 22, 34

Hao Chen, Rui Xia, and Jianfei Yu. Reinforced counterfactual data augmentation for dual sentiment classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 269–278, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.24. URL https://aclanthology.org/2021.emnlp-main.24. 22, 34

Nitay Calderon, Eyal Ben-David, Amir Feder, and Roi Reichart. DoCoGen: Domain counterfactual generation for low resource domain adaptation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7727–7746, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.533. URL https://aclanthology.org/2022.acl-long.533. 22, 34

Linyi Yang, Lifan Yuan, Leyang Cui, Wenyang Gao, and Yue Zhang. FactMix: Using a few labeled in-domain examples to generalize to cross-domain named entity recognition. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5360–5371, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics. URL https://aclanthology.org/2022.coling-1.476. 22, 34

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *arXiv preprint arXiv:2202.05262*, 2022. 23

Antonia Creswell and Murray Shanahan. Faithful reasoning using large language models. *arXiv preprint arXiv:2208.14271*, 2022. 23

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. 23, 35, 44, 45, 95

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online, August 2021. Association for Computational Linguistics. 23, 35

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68, Dublin, Ireland, May 2022. Association for Computational Linguistics. 23, 35

Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. WARP: Word-level Adversarial ReProgramming. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on*

Natural Language Processing (Volume 1: Long Papers), pages 4921–4933, Online, August 2021. Association for Computational Linguistics. 23, 35

Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou', and Daniel Cer. SPoT: Better frozen model adaptation through soft prompt transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5039–5059, Dublin, Ireland, May 2022. Association for Computational Linguistics. 23, 34, 43, 44

Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. PPT: Pre-trained prompt tuning for few-shot learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8410–8423, Dublin, Ireland, May 2022. Association for Computational Linguistics. 23, 34, 35, 43, 44, 45, 47, 95

Xu Guo, Boyang Li, and Han Yu. Improving the sample efficiency of prompt tuning with domain adaptation. *arXiv preprint arXiv:2210.02952*, 2022a. 23, 28, 34, 95

Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online, August 2021. Association for Computational Linguistics. 23, 44

Timo Schick and Hinrich Schütze. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online, April 2021a. Association for Computational Linguistics. 23

Timo Schick and Hinrich Schütze. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online, June 2021b. Association for Computational Linguistics. 23, 44

Karimi Mahabadi Rabeeh, Zettlemoyer Luke, Henderson James, Mathias Lambert, Saeidi Marzieh, Stoyanov Veselin, and Yazdani Majid. Prompt-free and efficient few-shot learning with language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3638–3652, Dublin, Ireland, May 2022. Association for Computational Linguistics. 23, 43

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020. 24

Iz Beltagy, Kyle Lo, and Arman Cohan. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural*

Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3615–3620, Hong Kong, China, November 2019. Association for Computational Linguistics. 24

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics. 24

Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv preprint arXiv:1904.05342*, 2019a. 24

Kexin Huang, Abhishek Singh, Sitong Chen, Edward T Moseley, Chih-ying Deng, Naomi George, and Charlotta Lindvall. Clinical xlnet: Modeling sequential clinical notes and predicting prolonged mechanical ventilation. *arXiv preprint arXiv:1912.11975*, 2019b. 24

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. BERTweet: A pre-trained language model for English tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, Online, October 2020. Association for Computational Linguistics. 24

Yanzhu Guo, Virgile Rennard, Christos Xypolopoulos, and Michalis Vazirgiannis. Bertweetfr: Domain adaptation of pre-trained language models for french tweets. *arXiv preprint arXiv:2109.10234*, 2021b. 24

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23, 2021. 24

Paweł Budzianowski and Ivan Vulić. Hello, it's GPT-2 - how can I help you? towards the use of pretrained language models for task-oriented dialogue systems. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 15–22, Hong Kong, November 2019. Association for Computational Linguistics. 24

Rong Zhang, Revanth Gangi Reddy, Md Arafat Sultan, Vittorio Castelli, Anthony Ferritto, Radu Florian, Efsun Sarioglu Kayi, Salim Roukos, Avi Sil, and Todd Ward. Multi-stage pre-training for low-resource domain adaptation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5461–5468, Online, November 2020. Association for Computational Linguistics. 24

Yunzhi Yao, Shaohan Huang, Wenhui Wang, Li Dong, and Furu Wei. Adapt-and-distill: Developing small, fast and effective pretrained language models for domains. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, Online, August 2021. Association for Computational Linguistics. 24

Vin Sachidananda, Jason Kessler, and Yi-An Lai. Efficient domain adaptation of language models via adaptive tokenization. In *Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing*, pages 155–165, Virtual, November 2021. Association for Computational Linguistics. 24

Nina Poerner, Ulli Waltinger, and Hinrich Schütze. Inexpensive domain adaptation of pretrained language models: Case studies on biomedical NER and covid-19 QA. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1482–1490, Online, November 2020. Association for Computational Linguistics. 24

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013c. 24

Thuy-Trang Vu, Dinh Phung, and Gholamreza Haffari. Effective unsupervised domain adaptation with adversarially trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6163–6173, Online, November 2020. Association for Computational Linguistics. 24

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online, July 2020. Association for Computational Linguistics. 25

Alexander Rietzler, Sebastian Stabinger, Paul Opitz, and Stefan Engl. Adapt or get left behind: Domain adaptation through bert language model finetuning for aspect-target sentiment classification. *arXiv preprint arXiv:1908.11860*, 2019. 25, 34

Chunning Du, Haifeng Sun, Jingyu Wang, Qi Qi, and Jianxin Liao. Adversarial and domain-aware BERT for cross-domain sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4019–4028, Online, July 2020. Association for Computational Linguistics. 25, 34

Xiaochuang Han and Jacob Eisenstein. 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)*, pages 4238–4248, Hong Kong, China, November 2019. Association for Computational Linguistics. 25, 34

Wissam Antoun, Fady Baly, and Hazem Hajj. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France, May 2020. European Language Resource Association. ISBN 979-10-95546-51-1. 25

Ahmad Beltagy, Abdelrahman Abouelenin, and Omar ElSherief. Arabic dialect identification using BERT-based domain adaptation. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 262–267, Barcelona, Spain (Online), December 2020. Association for Computational Linguistics. 25

Kosuke Nishida, Kyosuke Nishida, Itsumi Saito, Hisako Asano, and Junji Tomita. Unsupervised domain adaptation of language models for reading comprehension. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5392–5399, Marseille, France, May 2020. European Language Resources Association. 25, 34

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113: 54–71, 2019. 25

Haoran Xu, Seth Ebner, Mahsa Yarmohammadi, Aaron Steven White, Benjamin Van Durme, and Kenton Murray. Gradual fine-tuning for low-resource domain adaptation. In *Proceedings of the Second Workshop on Domain Adaptation for NLP*, pages 214–221, Kyiv, Ukraine, April 2021. Association for Computational Linguistics. 25, 34

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. 25

Xisen Jin, Dejiao Zhang, Henghui Zhu, Wei Xiao, Shang-Wen Li, Xiaokai Wei, Andrew Arnold, and Xiang Ren. Lifelong pretraining: Continually adapting language models to emerging corpora. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 1–16, virtual+Dublin, May 2022. Association for Computational Linguistics. 26, 34

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR, 09–15 Jun 2019. 26, 32, 33, 95

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019. 26

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. 26

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. *Advances in neural information processing systems*, 30, 2017. 26, 32

Xudong Wang, Zhaowei Cai, Dashan Gao, and Nuno Vasconcelos. Towards universal object detection by domain attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7289–7298, 2019b. 26, 32

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016. 26

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017. 26

Seanie Lee, Donggyu Kim, and Jangwon Park. Domain-agnostic question-answering with adversarial training. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 196–202, Hong Kong, China, November 2019. Association for Computational Linguistics. 26, 34

Han Zou, Jianfei Yang, and Xiaojian Wu. Unsupervised energy-based adversarial domain adaptation for cross-domain text classification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1208–1218, Online, August 2021. Association for Computational Linguistics. 27, 34

Deepanway Ghosal, Devamanyu Hazarika, Abhinaba Roy, Navonil Majumder, Rada Mihalcea, and Soujanya Poria. KinGDOM: Knowledge-Guided DOMain Adaptation for Sentiment Analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3198–3210, Online, July 2020. Association for Computational Linguistics. 27

Hui Tang, Ke Chen, and Kui Jia. Unsupervised domain adaptation via structurally regularized deep clustering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8725–8735, 2020. 27

Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015. 27

Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. 27

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33:6256–6268, 2020. 27, 34

Jungsoo Park, Gyuwan Kim, and Jaewoo Kang. Consistency training with virtual adversarial discrete perturbation. In *Proceedings of the 2022 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Short Papers)*, 2022. 27, 34, 40

Soyoung Yoon, Gyuwan Kim, and Kyumin Park. SSMix: Saliency-based span mixup for text classification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3225–3234, Online, August 2021. Association for Computational Linguistics. 27

Gangwoo Kim, Hyunjae Kim, Jungsoo Park, and Jaewoo Kang. Learn to resolve conversational dependency: A consistency training framework for conversational question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6130–6141, Online, August 2021. Association for Computational Linguistics. 27

Hong Liu, Mingsheng Long, Jianmin Wang, and Michael Jordan. Transferable adversarial training: A general approach to adapting deep classifiers. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4013–4022. PMLR, 09–15 Jun 2019. 28

Pin Jiang, Aming Wu, Yahong Han, Yunfeng Shao, Meiyu Qi, and Bingshuai Li. Bidirectional adversarial training for semi-supervised domain adaptation. In *IJCAI*, pages 934–940, 2020. 28

Jihan Yang, Ruijia Xu, Ruiyu Li, Xiaojuan Qi, Xiaoyong Shen, Guanbin Li, and Liang Lin. An adversarial perturbation oriented domain adaptation approach for semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12613–12620, 2020. 28

Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014. 28

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13 (1):723–773, 2012. 28

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450. Springer, 2016. 28

Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016. 28

Mohammad Mahfujur Rahman, Clinton Fookes, Mahsa Baktashmotlagh, and Sridha Sridharan. Correlation-aware adversarial domain adaptation and generalization. *Pattern Recognition*, 100:107124, 2020. 28

Juntao Li, Ruidan He, Hai Ye, Hwee Tou Ng, Lidong Bing, and Rui Yan. Unsupervised domain adaptation of a pretrained cross-lingual language model. In Christian

Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3672–3678. International Joint Conferences on Artificial Intelligence Organization, 7 2020. Main track. 29, 34

Alexis Conneau and Guillaume Lample. Cross-lingual language model pretraining. *Advances in neural information processing systems*, 32, 2019. 29

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015. 29

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2019. 29

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 531–540. PMLR, 10–15 Jul 2018. 29

Zhongwei Li, Xuancong Wang, Ai Ti Aw, Eng Siong Chng, and Haizhou Li. Named-entity tagging and domain adaptation for better customized translation. In *Proceedings of the Seventh Named Entities Workshop*, pages 41–46, Melbourne, Australia, July 2018. Association for Computational Linguistics. 30

Rico Sennrich and Barry Haddow. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 83–91, Berlin, Germany, August 2016. Association for Computational Linguistics. 30

Roee Aharoni and Yoav Goldberg. Towards string-to-tree neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 132–140, Vancouver, Canada, July 2017. Association for Computational Linguistics. 30

Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. 30

Min Yang, Zhou Zhao, Wei Zhao, Xiaojun Chen, Jia Zhu, Lianqiang Zhou, and Zigang Cao. Personalized response generation via domain adaptation. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, page 1021–1024, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450350228. 30

Wei-Nan Zhang, Qingfu Zhu, Yifa Wang, Yanyan Zhao, and Ting Liu. Neural personalized response generation as domain adaptation. *World Wide Web*, 22(4):1427–1446, jul 2019. ISSN 1386-145X. 30

Tomáš Šipka, Milan Šulc, and Jiří Matas. The hitchhiker's guide to prior-shift adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1516–1524, 2022. 30

Amr Alexandari, Anshul Kundaje, and Avanti Shrikumar. Maximum likelihood with bias-corrected calibration is hard-to-beat at label shift adaptation. In *International Conference on Machine Learning*, pages 222–232. PMLR, 2020. 30

Novi Quadrianto, Alex J. Smola, Tiberio S. Caetano, and Quoc V. Le. Estimating labels from label proportions. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 776–783, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605582054. 31

Stefan Rueping. Svm classifier estimation from group probabilities. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 911–918, Madison, WI, USA, 2010. Omnipress. ISBN 9781605589077. 31

Felix Yu, Dong Liu, Sanjiv Kumar, Jebara Tony, and Shih-Fu Chang. $\propto$svm for learning with label proportions. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 504–512, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. 31

Xu Guo, Han Yu, Boyang Li, Hao Wang, Pengwei Xing, Siwei Feng, Zaiqing Nie, and Chunyan Miao. Federated learning for personalized humor recognition. *ACM Trans. Intell. Syst. Technol.*, 13(4), may 2022b. ISSN 2157-6904. 31, 34

Chanhee Lee, Kisu Yang, Taesun Whang, Chanjun Park, Andrew Matteson, and Heuiseok Lim. Exploring the data efficiency of cross-lingual post-training in pretrained language models. *Applied Sciences*, 11(5), 2021. ISSN 2076-3417. 31, 34

Paul Michel and Graham Neubig. Extreme adaptation for personalized neural machine translation. In *ACL*, 2018. 31

Joern Wuebker, Patrick Simianer, and John DeNero. Compact personalized models for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 881–886, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. 31

Simone Scardapane, Danilo Comminiello, Amir Hussain, and Aurelio Uncini. Group sparse regularization for deep neural networks. *Neurocomput.*, 241(C):81–89, jun 2017. ISSN 0925-2312. 32

Junshen Chen, Dallas Card, and Dan Jurafsky. Modular domain adaptation. In *Findings of the Association for Computational Linguistics: ACL 2022*, Dublin, Ireland, May 2022. Association for Computational Linguistics. 32, 34

Benyuan Sun, Hongxing Huo, YI YANG, and Bo Bai. Partialfed: Cross-domain personalized federated learning via partial initialization. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 23309–23320. Curran Associates, Inc., 2021. 32

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, Dublin, Ireland, May 2022. Association for Computational Linguistics. 32

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland, May 2022. Association for Computational Linguistics. 32

Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. 32

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. Fast model editing at scale. In *International Conference on Learning Representations*, 2022. 32

Peter Hase, Mona Diab, Asli Celikyilmaz, Xian Li, Zornitsa Kozareva, Veselin Stoyanov, Mohit Bansal, and Srinivasan Iyer. Do language models have beliefs? methods for detecting, updating, and visualizing model beliefs. *arXiv preprint arXiv:2111.13654*, 2021. 32

Asa Cooper Stickland, Alexandre Berard, and Vassilina Nikoulina. Multilingual domain adaptation for NMT: Decoupling language and domain information with adapters. In *Proceedings of the Sixth Conference on Machine Translation*, pages 578–598, Online, November 2021. Association for Computational Linguistics. 32

Alexandra Chronopoulou, Matthew Peters, and Jesse Dodge. Efficient hierarchical domain adaptation for pretrained language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1336–1351, Seattle, United States, July 2022. Association for Computational Linguistics. 32, 34

Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych. AdapterDrop: On the efficiency of adapters in transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. 33

Rabeeh Karimi mahabadi, James Henderson, and Sebastian Ruder. Compacter: Efficient low-rank hypercomplex adapter layers. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. 33

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*, 2022. 33

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 33

Yusheng Su, Xiaozhi Wang, Yujia Qin, Chi-Min Chan, Yankai Lin, Zhiyuan Liu, Peng Li, Juanzi Li, Lei Hou, Maosong Sun, et al. On transferability of prompt tuning for natural language understanding. *arXiv preprint arXiv:2111.06719*, 2021. URL https://arxiv.org/pdf/2111.06719.pdf. 35

Colin Wei, Sang Michael Xie, and Tengyu Ma. Why do pretrained language models help in downstream tasks? an analysis of head and prompt tuning. *Advances in Neural Information Processing Systems*, 34:16158–16170, 2021b. URL https://openreview.net/forum?id=MDMV2SxCboX. 35

Qi Zhu, Bing Li, Fei Mi, Xiaoyan Zhu, and Minlie Huang. Continual prompt tuning for dialog state tracking. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1124–1137, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.80. URL https://aclanthology.org/2022.acl-long.80. 35

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21 (140):1–67, 2020b. URL http://jmlr.org/papers/v21/20-074.html. 37, 42

W Ronny Huang, Zeyad Emam, Micah Goldblum, Liam Fowl, Justin K Terry, Furong Huang, and Tom Goldstein. Understanding generalization through visualizations. 2020. 38

Safa Cicek and Stefano Soatto. Input and weight space smoothing for semi-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019. 38, 40

Dongha Kim, Yongchan Choi, and Yongdai Kim. Understanding and improving virtual adversarial training. *arXiv preprint 1909.06737*, 2019. 38, 40

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015. URL http://arxiv.org/abs/1412.6980. 40

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018. doi: 10.1109/TPAMI.2018.2858821. 40, 43

Takeru Miyato, Andrew M Dai, and Ian Goodfellow. Adversarial training methods for semi-supervised text classification. In *ICLR*, 2016. 40

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics, 2018. URL `http://aclweb.org/anthology/N18-1101`. 42

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075. URL `https://aclanthology.org/D15-1075`. 42

Marie-Catherine de Marneffe, Mandy Simons, and Judith Tonhauser. The commitmentbank: Investigating projection in naturally occurring discourse., 2019. URL `https://semanticsarchive.net/Archive/Tg3ZGI2M/Marneffe.pdf`. 42

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA). URL `http://www.lrec-conf.org/proceedings/lrec2014/pdf/363_Paper.pdf`. 42

Aaron Gokaslan and Vanya Cohen. Openwebtext corpus, 2019. URL `http://Skylion007.github.io/OpenWebTextCorpus`. 43

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. Freelb: Enhanced adversarial training for natural language understanding. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=BygzbyHFvB`. 43

Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4596–4604. PMLR, 10–15 Jul 2018. URL `https://proceedings.mlr.press/v80/shazeer18a.html`. 45

Roberto González-Ibáñez, Smaranda Muresan, and Nina Wacholder. Identifying sarcasm in Twitter: A closer look. In *Proceedings of the 49th Annual Meeting of the*

Association for Computational Linguistics: Human Language Technologies, 2011. 53

Diana G Maynard and Mark A Greenwood. Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis. In *LREC 2014 Proceedings*. ELRA, 2014. 53

Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. A large self-annotated corpus for sarcasm. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation*, 2018. 53

Dmitry Davidov, Oren Tsur, and Ari Rappoport. Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, page 107–116, USA, 2010. Association for Computational Linguistics. ISBN 9781932432831. 53

Elena Filatova. Irony and sarcasm: Corpus generation and analysis using crowdsourcing. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation*, Istanbul, Turkey, 2012. 53

Raymond W. Gibbs. Irony in talk among friends. *Metaphor and Symbol*, 15(1-2):5–27, 2000. 53

Tony Veale, F Amílcar Cardoso, and Rafael Pérez y Pérez. Systematizing creativity: A computational view. In *Computational Creativity*, pages 1–19. Springer, 2019. 53

Sandeepa Kannangara. Mining twitter for fine-grained political opinion polarity classification, ideology detection and sarcasm detection. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 751–752, 2018. 53

Jean E. Fox Tree, J. Trevor D'Arcey, Alicia A. Hammond, and Alina S. Larson. The sarchasm: Sarcasm production and identification in spontaneous conversation. *Discourse Processes*, 57(5-6):507–533, 2020. 53

Ellen Riloff, Ashequl Qadir, Prafulla Surve, LalindraDe Silva, Nathan Gilbert, and Ruihong Huang. Sarcasm as contrast between a positive sentimentand negative situation. In *EMNLP*, 2013. 53

Cynthia Van Hee, Els Lefever, and Véronique Hoste. Semeval-2018 task 3: Irony detection in english tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 39–50, 2018. 53, 62, 66, 67

Silviu Oprea and Walid Magdy. isarcasm: A dataset of intended sarcasm. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020. 53, 62, 66, 67

Tomáš Ptáček, Ivan Habernal, and Jun Hong. Sarcasm detection on Czech and English twitter. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 213–223, 2014. 53, 62

David Bamman and Noah A Smith. Contextualized sarcasm detection on twitter. In *Ninth international AAAI conference on web and social media*. Citeseer, 2015. 53

Aditya Joshi, Vinita Sharma, and Pushpak Bhattacharyya. Harnessing context incongruity for sarcasm detection. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 757–762, 2015. 53

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, volume 37, pages 1180–1189, 2015. 54

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Adversarial multi-task learning for text classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–10, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1001. URL `https://www.aclweb.org/anthology/P17-1001`. 54

Young-Bum Kim, Karl Stratos, and Dongchan Kim. Adversarial adaptation of synthetic or stale data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1297–1307, 2017. 54

Anush Kamath, Sparsh Gupta, and Vitor Carvalho. Reversing gradients in adversarial domain adaptation for question deduplication and textual entailment tasks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5545–5550, 2019. 54

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR, 06–11 Aug 2017. URL `http://proceedings.mlr.press/v70/finn17a.html`. 54, 60, 61

David Balduzzi, Sébastien Racanière, James Martens, Jakob N. Foerster, Karl Tuyls, and Thore Graepel. The mechanics of n-player differentiable games. In Jennifer G. Dy and Andreas Krause, editors, *ICML*, 2018. 54

G. M. Korpelevich. An extragradient method for finding saddle points and for other problems. *Ekonomika i Matematicheskie Metody*, 12:747–756, 1976. 60

Waïss Azizian, Ioannis Mitliagkas, Simon Lacoste-Julien, and Gauthier Gidel. A tight and unified analysis of gradient-based methods for a whole spectrum of games. In *AISTATS*, 2020. 60

Aniruddha Ghosh and Tony Veale. Fracking sarcasm using neural network. In *Proceedings of the 7th workshop on computational approaches to subjectivity, sentiment and social media analysis*, pages 161–169, 2016. 62

Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. Datastories at SemEval-2017 task 4: Deep LSTM with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation*

(SemEval-2017), pages 747–754, Vancouver, Canada, August 2017. Association for Computational Linguistics. 63

Yi Tay, Anh Tuan Luu, Siu Cheung Hui, and Jian Su. Reasoning with sarcasm by reading in-between. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1010–1020, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1093. URL https://www.aclweb.org/anthology/P18-1093. 63

Chuhan Wu, Fangzhao Wu, Sixing Wu, Junxin Liu, Zhigang Yuan, and Yongfeng Huang. Thu_ngn at SemEval-2018 task 3: Tweet irony detection with densely connected LSTM and multi-task learning. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 51–56, 2018. 63, 66, 67

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 65

Tony Veale. *Exploding the creativity myth: The computational foundations of linguistic creativity*. A&C Black, 2012. 73

Simge Aykan and Erhan Nalçacı. Assessing theory of mind by humor: The humor comprehension and appreciation test (tom-hcat). *Frontiers in psychology*, 9, 2018. 73

Rod A Martin, Patricia Puhlik-Doris, Gwen Larsen, Jeanette Gray, and Kelly Weir. Individual differences in uses of humor and their relation to psychological well-being: Development of the humor styles questionnaire. *Journal of research in personality*, 37(1):48–75, 2003. 73

Sonja Heintz and Willibald Ruch. From four to nine styles: An update on individual differences in humor. *Personality and Individual Differences*, 141:7–12, 2019. 73

Nabil Hossain, John Krumm, and Michael Gamon. "President Vows to Cut Hair": Dataset and analysis of creative text editing for humorous headlines. *CoRR, arXiv:1906.00274*, 2019. 73, 83

Diyi Yang, Alon Lavie, Chris Dyer, and Eduard Hovy. Humor recognition and humor anchor extraction. In *EMNLP*, pages 2367–2376, 2015. 73, 74, 89

Peng-Yu Chen and Von-Wun Soo. Humor recognition using deep learning. In *NAACL*, pages 113–117, 2018. 73, 74, 89

Lizhen Liu, Donghai Zhang, and Wei Song. Exploiting syntactic structures for humor recognition. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1875–1883, 2018. 74

Renxian Zhang and Naishi Liu. Recognizing humor on twitter. In *CIKM*, pages 889–898, 2014. 74

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *AISTATS*, pages 1273–1282, 2017. 75, 76, 79

Swaroop Ramaswamy, Rajiv Mathews, Kanishka Rao, and Françoise Beaufays. Federated learning for emoji prediction in a mobile keyboard. *CoRR, arXiv:1906.04329*, 2019. 80

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *CoRR, arXiv:1909.11942*, 2019. 90

Qiang Yang, Yang Liu, Yong Cheng, Yan Kang, Tianjian Chen, and Han Yu. *Federated Learning*. Morgan & Claypool Publishers, 2019. 92

Alejandro Mottini and Amber Roy Chowdhury. What do you mean i'm funny? personalizing the joke skill of a voice-controlled virtual assistant. *arXiv preprint arXiv:1912.03234*, 2019. 92

Jie Lu, Anjin Liu, Fan Dong, Feng Gu, Joao Gama, and Guangquan Zhang. Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, 31(12):2346–2363, 2018. 92

Isvani Frías-Blanco, Jose del Campo-Avila, Gonzalo Ramos-Jiménez, Andre CPLF Carvalho, Agustín Ortiz-Díaz, and Rafael Morales-Bueno. Online adaptive decision trees based on concentration inequalities. *Knowledge-Based Systems*, 104:179–194, 2016. 92

Andrés Cano, Manuel Gómez-Olmedo, and Serafín Moral. A bayesian approach to abrupt concept drift. *Knowledge-Based Systems*, 185:104909, 2019. 92

Ruidan He, Linlin Liu, Hai Ye, Qingyu Tan, Bosheng Ding, Liying Cheng, Jiawei Low, Lidong Bing, and Luo Si. On the effectiveness of adapter-based tuning for pretrained language model adaptation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2208–2222, Online, August 2021. Association for Computational Linguistics. 95