Learning Neural Models for Natural Language Processing in the Face of Distributional Shift

Paul Michel

CMU-LTI-21-013

July 2021

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:

Graham Neubig (Chair)	Carnegie Mellon University
Zico Kolter	Carnegie Mellon University
Zachary Lipton	Carnegie Mellon University
Tatsunori Hashimoto	Stanford University

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Connaître, ce n'est point démonter, ni expliquer.
C'est accéder à la vision.
Mais, pour voir, il convient d'abord de participer.
Cela est dur apprentissage…

ANTOINE DE SAINT EXUPERY

Abstract

The dominating NLP paradigm of training a strong neural predictor to perform one task on a specific dataset has led to state-of-the-art performance in a variety of applications (eg. sentiment classification, span-prediction based question answering or machine translation). However, it builds upon the assumption that the data distribution is stationary, ie. that the data is sampled from a fixed distribution both at training and test time. This way of training is inconsistent with how we as humans are able to learn from and operate within a constantly changing stream of information. Moreover, it is ill-adapted to real-world use cases where the data distribution is expected to shift over the course of a model's lifetime.

The first goal of this thesis is to characterize the different forms this shift can take in the context of natural language processing, and propose benchmarks and evaluation metrics to measure its effect on current deep learning architectures. We then proceed to take steps to mitigate the effect of distributional shift on NLP models. To this end, we develop methods based on parametric reformulations of the distributionally robust optimization framework. Empirically, we demonstrate that these approaches yield more robust models as demonstrated on a selection of realistic problems. In the third and final part of this thesis, we explore ways of efficiently adapting existing models to new domains or tasks. Our contribution to this topic takes inspiration from information geometry to derive a new gradient update rule which alleviate catastrophic forgetting issues during adaptation.

Acknowledgements

This thesis would not have been made possible without the continuous support, involvement and help from my amazing PhD supervisor Graham Neubig. He spent countless hours discussing research with me, helping me with paper writing and was overall very supportive throughout my PhD. I realize now that I was extremely lucky to have such a thoughtful and conscientious advisor. It was a great pleasure to be working with him all these years.

I also want to thank the members of my committee, Zico Kolter, Zachary Lipton and Tatsunori Hashimoto for their insightful feedback which helped shape this document into its current form. I am particularly grateful for Tatsu's involvement, as he became a close collaborator over the last year or so. I must also thank the many anonymous reviewers who gave feedback on the work presented here. Our relationship over the years had its ups and downs, but overall the discussions we had had a net positive impact on my work, and for that I am thankful.

Of course, the work presented here was the product of collaboration with all my co-authors Xian Li and Juan Pino (both of whom I had the pleasure to intern with at Facebook in 2018), Liz Salesky (our discussions at the JSALT workshop in 2019 helped spur the work presented in Chapter 7) and Lucia Specia (also met at JSALT) and the entire team behind the two WMT robustness shared tasks.

I am also very lucky to have had the opportunity to collaborate with a host of brilliant researchers on other papers which didn't make it in this document: in no particular order Jaime Carbonell, Xinyi Wang, Hieu Pham, Omer Levy, Danish, Ameet Talwalkar, Antonios Anastasopoulos, Ziyi Dou and Junjie Hu. I was also very happy to work with Keita Kurita and Lucio Dery, both brilliant students and who made our collaborations together enjoyable first experiences in advising more junior researchers. I am also thankful to the Deepmind language team (and in particular my hosts Dani Yogatama and Sebastian Ruder) who hosted me for a remote internship in the middle of a global pandemic.

I attribute the main reason for my pleasurable PhD experience to the fantastic environment at CMU and at the LTI. I was happy to be surrounded by other remarkable students within our research group (NeuLab), in particular Chunting Zhou, Pengcheng Yin, Chaitanya Malaviya and Shruti Rijhwani, as well as all the others who joined us over the years. I was also lucky to make long lasting friends at CMU: Abhilasha Ravichander, Siddarth Dalmia, Rajat Kulshreshtha, Chirag Nagpal, Evangelia Spiliopoulou, Artidoro Pagnoni, Mariia Ryskina, Shruti Palaskar, Raphael Olivier, Xinya Li, James Fiacco and all the others. I'm also grateful for the wonderful people I met at various conferences (back in the olden days of in-person conferences), in particular Kris Cao,

Kazuya Kawakami, Roee Aharoni, Gail Weiss and Jad Kabbara.

These acknowledgments would not be complete without a mention of the Pittsburgh food and drinks scene. Many a research discussion was had around a drink at Peter's pub, Stack'd, Fuel and Fuddle, Mario's or Hemingway's. I also have fond memories of lunches/dinners/coffee breaks at the Chipotle on Forbes, Sushi Fuku, Quiznos, the indian foodtruck, Porch, Tazza de Oro and Arriviste to mention only a few. They kept my belly full and my wallet flat. Finally, a special mention goes to the Carnegie Museum of Natural History whose incredible dinosaur exhibit (and good coffee) was a welcome breath of fresh air while I was putting the finishing touches to this thesis.

I must also acknowledge the continued support of my parents, Matthieu and Véronique. They have given me every opportunity I needed to go as far as I could in my studies. I realize that this is a privilege that many people I know (and many more I don't) did not have.

Last, but perhaps foremost, I would not have been able to go through these PhD years without the unconditional love and support from my then girlfriend, now wife, Clémence. You endured five long years of distance to make this possible, yet somehow you were always close by my side. This thesis is, of course, dedicated to you.

Contents

Chapter 1

Introduction

The ability to use and comprehend language as a mean of communication is a unique characteristic of human intelligence. Natural language is fundamentally non-stationary: the way we produce utterances is modulated by a myriad of factors that are ever changing based on our interlocutors, social context, emotions, or communication medium used. Our capacity to operate in and adapt to such a dynamic environment is another defining feature of intelligence. Indeed from their youngest age, infants are seldom placed in static situations, yet they are still able to learn emerging patterns from a constant flow of information (see *e.g.* Dupoux (2018)). Hence, it is a natural goal for artificial intelligence research to develop agents that are capable of acquiring natural language understanding and generation abilities in the face of a changing environment.

The development of computer systems which can understand and produce natural language is realized through Natural Language Processing (NLP). In recent years, neural-network based machine learning models have been the source of rapid progress in NLP, and now represent the state-of-the-art in a large number of real-world applications (such as machine translation (Wu et al., 2016), named entity recognition (Lample et al., 2016) or speech recognition (Xiong et al., 2018)), reaching human parity on a variety of benchmarks (Wang et al., 2020a; Rajpurkar et al., 2020)[1]. Yet, these models exhibit a pattern of failures when confronted with changes to their training environment:

- They are highly sensitive to *domain shift*: they perform poorly when confronted with text from a different domain, for example text concerning different topics (Gururangan et al., 2020), demographics (Blodgett et al., 2016; Amodei et al., 2016; Hovy & Søgaard, 2015),

[1]Meaning that they outperform human annotated baselines, we make no claim that these systems have achieved any kind of human-like intelligence.

and even using different data collection processes (Gururangan et al., 2018). In particular, these models often perform poorly when evaluated on sub-populations,[2] and they can latch on to spurious correlations (McCoy et al., 2019).

- They are vulnerable to specific *word or character-level perturbations* (Papernot et al., 2016; Ebrahimi et al., 2018b; Belinkov & Bisk, 2018b). This phenomenon poses security concerns, as this opens up trained models to various types of attacks by nefarious agents: *e.g.* bypassing toxicity detection models (Hosseini et al., 2017).

- When they are trained on new tasks or domains, they rapidly forget previously acquired knowledge, a phenomenon known as *catastrophic forgetting* (McCloskey & Cohen, 1989; Kirkpatrick et al., 2017). This makes adaptation to new data distributions an arduous and unreliable process necessitating significant amounts of domain expertise and regularization (Miceli Barone et al., 2017).

These shortcomings are not only at odds with the natural behaviour observed in intelligent agents mentioned above, but represents a liability for practical users and designers of such systems. Indeed, machine learning models trained on static datasets can fail spectacularly in the face of real-world situations, sometimes with dire consequences (for example, this mis-translation of "good morning" into "attack them" due to dialectic variations; Hern (2017)). More insidiously, the representation disparity between training datasets and the real world can, for example, unfairly affect and further disenfranchise minority groups (Hashimoto et al., 2018; Dixon et al., 2018).

The principal desired characteristic of the machine learning models used in NLP is the ability to learn and generalize from a finite number of training examples. The most common learning paradigm, empirical risk minimization, relies on the assumption that this training data is sampled from a fixed distribution, and generalization is construed as the ability to perform well on unseen samples from this same distribution. The failure cases described above occur when the model operates in a different distribution than the one it was trained on, in which case standard generalization guarantees break down. In this context, the discrepancy between a machine learning model's learning and operating environment can be described as *distributional shift*. The research presented in this thesis focuses on what we believe to be the three major axes of progress towards addressing distributional shift in the context of NLP: *Evaluation*, *Robustness* and *Adaptation*.

We start with **evaluation** because distributional shift is particularly difficult to characterize and measure, especially in regard to natural language. This is partly due to the absence of a canonical

[2]Domains that exist, but are under-represented in their training data (Sagawa et al., 2020).

metric structure of the data. In other words, it is not clear how to efficiently measure the semantic similarity between two sentences and as a result there is no straight-forward way to measure the discrepancy between two samples, let alone two distributions. Therefore, as a natural first step in addressing distributional shift, we propose a new benchmark (Chapter 3) and evaluation metric (Chapter 4) to evaluate domain shift and robustness to adversarial perturbations respectively. With these tools in hand, we set out to construct **robust** models that are trained to be less sensitive to distributional shift, even in the absence of explicit information on the nature of shift. This is done by leveraging the diversity of data within the training distribution to ensure uniform performance over a variety of domains present in the training data (sub-populations). Specifically we formulate a parametric version of the distributionally robust optimization framework which allows for training models that are more robust to sub-population shift (Chapters 5 and 6). Finally, learning in a static environment is fundamentally sub-optimal: we cannot expect our models to perform well on each and every possible future setting, and we must be able to **adapt** them to any new scenario we encounter. Consequently, we research mechanisms by which we are able to fine-tune a trained model on new evidence, without forgetting previously acquired knowledge (Chapter 7).

The detailed outline of our contributions in this thesis is presented below:

- **Chapter 2** lays out a description of the distributional shift phenomenon and discusses previous and current work in this general area.
- **Part I: Measuring and Evaluating Robustness**
  - In **Chapter 3**, we introduce a benchmark for evaluating machine translation models against distributional shift. Specifically, the proposed dataset — dubbed MTNT — provides a testbed for translation of noisy social media text, a challenging domain for common state-of-the-art models trained on news and parliamentary proceedings. This work was published as Michel & Neubig (2018a) at EMNLP 2018[3], Li et al. (2019) at WMT 2019[4] and (Specia et al., 2020) at WMT 2020.[5]
  - In **Chapter 4**, we examine the difficulty of evaluating adversarial perturbations on NLP models, with an emphasis on sequence-to-sequence models. We propose an evaluation framework centered around the idea of evaluating the perturbations' effect on

[3] https://www.aclweb.org/anthology/events/emnlp-2018/
[4] https://www.statmt.org/wmt19/robustness.html
[5] https://www.statmt.org/wmt20/robustness.html

the semantics of the input sentences. The content of this chapter originally appeared as Michel et al. (2019b) at NAACL 2019.[6]

- **Part II: Making Robust Models**
  - **Chapter 5** describes an approach for training models that are robust to distributional shift, using a parametric version of the distributionally robust optimization (DRO) framework. Specifically, we train a model to minimize its expected loss under the distribution defined by another generative model (the "adversary"), which is itself trained to represent the worst-case distribution for the model. In experiments, we find that this approach (called Parametric DRO) yields models that are more robust compared to other baselines in the DRO literature. This chapter was previously published at ICLR 2021[7] as Michel et al. (2021).
  - **Chapter 6** presents an improvement over the P-DRO approach whereby a parametric *re-weighting* of the training data is learned. With appropriate constraints, and a simple but crucial minibatch-level renormalization trick, we find that this results in a simple DRO approach which reliably outperforms other baselines and necessitates little hyperparameter tuning. The work presented in this chapter is currently undergoing peer review.
- **Part III: Adaptation**
  - In **Chapter 7**, we develop a technique for mitigating the catastrophic forgetting issue that arises when adapting a model to a new domain or a new task. The approach, inspired by ideas in information geometry, is demonstrated to drastically reduce forgetting compared to regular gradient descent, and enjoys complementarity with common continual-learning baselines. This chapter contains work that was released as a preprint.

[6] https://www.aclweb.org/anthology/events/naacl-2019/
[7] https://iclr.cc/Conferences/2021

Chapter 2

Background and Literature Review

From a broad perspective, the goal of machine learning is to produce a model that is able to perform a given task, such as predicting an output variable y from an input, x (*e.g.* sentiment given a movie review, or French translation given an English sentence...) given finite amounts of training data (in our example, (x, y) pairs). In this chapter, we formalize this setting for the rest of this thesis.

2.1 Terminology and Definitions

In machine learning, we are confronted with learning to perform a task given a finite number of examples. In the context of this thesis, a task T will be referring to a triplet containing two measurable spaces $\mathcal{X}$ and $\mathcal{Y}$, as well as a probability distribution p with support on the Cartesian product $\mathcal{X} \times \mathcal{Y}$. $\mathcal{X}$ represents the space of all possible inputs (for example the space of all 32×32 images, or the space of all English sentences), while $\mathcal{Y}$ stands for the space of all admissible outputs (for example the set $\{\text{positive}, \text{negative}\}$ in sentiment classification, or all sentences of a target language in translation). Finally, p will represent the true distribution of the data in nature (for example it will assign high probability to a pair of English and French sentences that are translations of each other).

In general, learning a task will consist of training a model to approximate the conditional distribution $p(y \mid x)$, based only on a finite number of training examples $x, y \sim p$. Specifically, we will focus on parametric models, *i.e.* models that are fully determined by a finite, constant (with respect to the size of the training data) number of parameters described by a vector θ that can take its values in Θ, the set of all possible parameters (in practice $\mathbb{R}^d$ or a subset thereof, where d is the total number of parameters).

Traditionally, given a loss function $\ell : \mathcal{X} \times \mathcal{Y} \times \Theta \mapsto \mathbb{R}^+$ measuring the discrepancy between model θ's prediction on x and the reference output y, the modeler will estimate parameters θ^* that better fit the distribution p data by minimizing the expected risk

$$\mathcal{L}_{\text{Expected Risk}}(\theta) = \mathbb{E}_{(x,y)\sim p}\ell_\theta(x, y). \tag{2.1}$$

Note that we abuse notation and write $\ell_\theta(x, y)$ for $\ell(x, y, \theta)$ to distinguish between the dependency over θ, the model parameters, and (x, y), the model's inputs. As implied earlier, one cannot compute the total expectation $\mathbb{E}_{x,y\sim p}$ and we must resort to minimizing the empirical risk computed over a finite training set $(x_1, y_1), \ldots, (x_n, y_n)$

$$\mathcal{L}_{\text{Empirical Risk}}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell_\theta(x_i, y_i) \tag{2.2}$$

2.2 Distributional Shift

2.2.1 What is Distributional Shift

In the empirical risk minimization (ERM) paradigm, machine learning models are trained to minimize their expected loss under a single, static distribution. However, this assumption is not always valid in practical applications. For example, a machine translation system trained on a majority of news article might come to be used to translate social media comments. While the overall task is can arguably be considered the same at a high level from the point of view of a human, there are significant differences in the data distribution (for instance, social media comments may contain more informal language).

In the most general sense, distributional shift (or distribution shift) is simply an instance of the test distribution q being different from the training data distribution p. Of course, this definition is too broad to be realistic, let alone useful in practice.[1] Precisely defining and categorizing what characterizes an acceptable shift is still very much an active area of research.

2.2.2 Categorizing Distributional Shift

While we will not attempt to formulate a complete taxonomy of distributional shifts, we will paint a broad strokes picture of how they are categorized in the literature. We will distinguish two main

[1]Consider, in a binary classification task, setting $q(y \mid x) = 1 - p$. Only the distribution changes, yet the two tasks are directly at odds with each other.

Table 2.1: Examples of "mechanistic" categorizations of distributional shift.

	$p(x, y)$	$\longrightarrow$	$q(x, y)$
Covariate shift	$p(x)\, p(y \mid x)$	$\longrightarrow$	$q(x)\, p(y \mid x)$
Label shift	$p(x \mid y)\, p(y)$	$\longrightarrow$	$p(x \mid y)\, q(y)$
Concept shift	$p(x)\, p(y \mid x)$	$\longrightarrow$	$p(x)\, q(y \mid x)$
Sub-population shift	$\sum_{i=1}^{K} \alpha_i\, p_i(x, y)$	$\longrightarrow$	$\sum_{i=1}^{K} \beta_i\, p_i(x, y)$

schools of thought for distinguishing distributional shift: a "mechanistic" and a "causal" approach.

Mechanistic Categorization

The first approach is to distinguish shifts by describing, in mathematical terms, the mechanics of how the change manifests in the data generation process. The appeal of this categorization is that it allows for a more principled, theoretical approach to tackling distributional shift. A summary of the examples described in this section can be found in Table 2.1.

The most well known instance is *covariate shift* (Shimodaira, 2000; Sugiyama & Müller, 2005; Storkey, 2009; Bickel et al., 2009), where only the distribution over the input space $\mathcal{X}$ (sometimes called the space of "covariates") changes, and the conditional distribution $p(y \mid x)$ is assumed to stay the same. Covariate shift is a rather intuitive assumption, especially in such cases where there is a clear causal relationship between the covariates x and the targets y. It holds an important place within the distribution shift literature because its particular characteristics justify the use of specific algorithms, some of which enjoy appealing properties. For instance, it can be tackled by means of re-weighting with the covariate likelihood ratio $q(x)/p(x)$ (Shimodaira, 2000), and the covariate shift assumption can yield tighter guarantees (Duchi et al., 2019).

Another, diametrically opposed type of shift is *label-shift* (or *prior probability shift* in Storkey (2009)), where only the marginal on $\mathcal{Y}$ changes, and the reverse conditional $p(x \mid y)$ stays fixed. This is most relevant when the causal relationship between x and y is reversed, for example when diagnosing an illness (y) from its symptoms (x). Specific methods have been developed to tackle this scenario (Zhang et al., 2013; Lipton et al., 2018), although their application is generally less common in the NLP literature. Finally, a comparatively less well studied occurrence is *concept-shift*, in which the marginal $p(x)$ is assumed fixed, but the conditional changes to another distribution $q(y \mid x)$ (Quiñonero-Candela et al., 2009). This can result from a mismatch between annotators: for instance in machine translation, different professional annotators may use slightly different translation conventions. Widmer & Kubat (1996) postulates that this is due to the presence

of some "unknown context", such as the two translators having had a different education, which leads to them adhering to two mildly different definitions of the translation problem. A concrete example of this phenomenon in toxicity detection (in the English language) is the "n-word" problem. This racial slur and other historically charged terms such as "queer" are often used disparagingly towards certain underrepresented minorities. However some of them were re-appropriated by minority speakers within their respective communities (Rahman, 2012). Without information about the speaker, it can be difficult to determine the toxicity of an utterance containing such words, even for human annotators (Sap et al., 2019).

Another, more elaborate example of a "mechanistically defined" category of distribution shift is what Koh et al. (2020) calls "sub-population shift". In sub-population shift, it is assumed that the training distribution p can be decomposed as a mixture of K different domains $p = \alpha_1 q_1 + \ldots + \alpha_K q_K$, $\sum_i \alpha_i = 1$. This can be the case by design: for example the training data for the WMT machine translation competition[2] is aggregated from many different machine translation corpora, or the MultiNLI textual entailment dataset (Williams et al., 2018) was created specifically to contain data from different domains. But in many more cases the training data may be an aggregate of various domains, unbeknownst to the dataset's creator: for instance text datasets often contain data from different dialects (*e.g.* American vs. British English, French vs. Canadian French, etc...). The benchmarks aginst which the methods described Part II of this thesis will be evaluated are all examples of sub-population shifts.

Causal Categorization

An alternate view is to distinguish the distribution shifts through the real world phenomena that cause them. This more empirical approach has the merit of describing realistic cases of distribution shift which may not clearly (or provably) fit into the mathematically defined categories presented in the previous section, but are prevalent enough to warrant the development of bespoke solutions.

A predominant cause of distributional shift is change over time (Kelly et al., 1999; Hand, 2006). For instance, Bloomfield (1933) documents nine categories of lexical semantic drift that words may undergo over time[3]. While not laying out a precise typology, McCulloch (2020) also touches on a variety of semantic shift that occurred more recently (and at a comparatively faster pace) with the advent of the internet. The effect of these changes in meaning on word embedding models such as Word2Vec (Mikolov et al. (2013), a staple of the NLP pipeline at the time) were summarized in

[2] `http://statmt.org/wmt21/`

[3] The various causes of this drift are elaborated upon in later studies (Blank & Koch, 2013; Grzega & Schoener, 2007)

Kutuzov et al. (2018). Identifying time as the underlying cause for this phenomenon has the benefit of suggesting solutions tailored to the problem at hand: for example, one could use dynamic word-embedding models (Rudolph & Blei, 2018) to predict upcoming drift, and use this knowledge to augment the training data. In a more recent example, Lazaridou et al. (2021) highlights the specific importance of "dynamic evaluation"[4] (Mikolov et al., 2010) for mitigating loss in performance due to time shift in language models.

In other examples, distributional shift can be described in terms of data collection: a prime example is what Storkey (2009) calls *biased sampling*. Subtle design choices at the time of the creation of the training dataset (such as data source, pre-processing, or annotation scheme) can introduce biases into the training distribution. A recent illustration of biased sampling is the case of crowd-sourced natural language inference[5] datasets (Gururangan et al., 2018): annotators tasked with crafting a contradictory hypothesis from a premise would frequently resort to introducing negation markers (in English: "not", "never", etc…). This introduced a spurious correlation between the presence of such negative words and the "contradiction" label. When confronted with test data that doesn't conform to this pattern (*e.g.* cases of entailment where the hypothesis does include a negation marker), models trained with empirical risk minimization (ERM) break down (Sagawa et al., 2020).

Local and Global Shifts

Aside from the two types of categorization described above, we believe that it is worth making another distinction between what we will call "local" and "global" shifts, purely by virtue of the fact that they are both the subject of considerable, yet distinct areas of the literature. We use the term *local* to refer to cases where the shifted distribution q can be obtained by applying a transformation (deterministic or non-deterministic) to data points in p. Of course, this isn't a very clearly defined category since any distribution can be transported into another one via an optimal transport map. In this case, we refer specifically to shifts that are best explicitly defined by such "local" transformations. This category encompasses phenomena such as character or word-level noise (of the kind described in Belinkov & Bisk (2018b) for instance) or adversarial perturbations (Papernot et al., 2016; Ebrahimi et al., 2018b,a; Cheng et al., 2018a), small modifications to the input of a machine learning model that cause dramatic changes to its output.

[4]The process of continuously training a language model at test time.

[5]Natural language inference is another name for the task of recognizing textual entailment. In its most common formulation, the model is given two sentences, the premise and the hypothesis, and is asked to determine whether the first entails or contradict the second (or whether there is no clear entailment relationship).

The notion of *global* shift will then be used to refer to those type of distributional shift that cannot easily be modeled as local shifts. We make this distinction because, as will be discussed in Chapter 4, local shift can — and should — be addressed specifically.

2.3 Robustness to Distributional Shift

2.3.1 Distributionally Robust Optimization

There is a rather large body of literature devoted to training models that are robust to distributional shift, under the name of distributionally robust optimization (DRO). In DRO, models are trained not to minimize their expected risk under any one distribution p, but rather their worst case risk under a pre-determined family of distributions $\mathcal{Q}$, called the "uncertainty set"

$$\min_{\theta \in \Theta} \max_{q \in \mathcal{Q}} \underbrace{\mathbb{E}_{(x,y)\sim q}\, \ell_\theta(x, y)}_{\mathcal{L}_{\text{DRO}}(\theta)}. \tag{2.3}$$

The DRO loss in Equation 2.3 provides an upper bound on the expected loss of the model under any distribution in the uncertainty set $\mathcal{Q}$, which motivates the use the solution of the min-max problem as a robust model. However this objective is only useful insofar that $\mathcal{Q}$ (1) covers test distributions of interest (corresponding to different domains, demographics, etc.) and (2) is not overly pessimistic. To fulfil this second condition, there should exist some model θ^* that achieves low loss simultaneously on the test distribution as well as $\mathcal{Q}$. This often requires that $\mathcal{Q}$ only contain distributions that are "close" to the training distribution under some divergence metric, or that are restricted to covariate shifts.

In some cases, it is conceivable that $\mathcal{Q}$ is available at training time. For instance, we might have access to some amount of data in K domains of interest, in which case the DRO loss can be written as a maximum over the finite set $\mathcal{Q} = \{q_1, \ldots, q_K\}$. This setting is sometimes referred to as Group-DRO and has been studied in Oren et al. (2019); Sagawa et al. (2020); Zhou et al. (2021).

In the absence of explicit information about the domains of interest, it is up to the practitioner to carefully define this uncertainty set. There is substantial existing work on nonparametric formulations of DRO, where $\mathcal{Q}$ is expressed as a divergence ball centered at the training distribution. Popular examples in the literature are:

- **f-divergence balls**, also called ϕ-divergences or Rényi divergences (Cover, 1999; Van Erven & Harremos, 2014) are a family of divergence metrics of the form $d_f(p, q) = \mathbb{E}_q\, f(\frac{p}{q})$ where

$f : \mathbb{R}_+ \to \mathbb{R}$ is a convex function mapping 1 to 0, and q is assumed to be absolutely continuous to p ($p(x, y) = 0 \Rightarrow q(x, y)$). Well-known representatives include the Kullback-Leibler (KL) divergence ($f(u) = u \log u$), Pearson's χ_2 divergence ($f(u) = \frac{1}{2}(u-1)^2$) or the total variation distance ($f(u) = \frac{1}{2}|u-1|$). They have seen extensive use in the DRO literature (Ben-Tal et al., 2013; Hu & Hong, 2013; Faury et al., 2020).

- **Wasserstein/IPM balls**. The 1-Wasserstein distance or earth mover's distance between two probability distributions p, q on a metric space $\mathcal{Z}$ is defined as $W_1 = \inf_{\pi \in \Pi} \mathbb{E}_{z,z' \sim \pi}\, \rho(z, z')$ where Π is the set of all joint distributions on $\mathcal{Z} \times \mathcal{Z}$ with marginals p and q, and ρ is $\mathcal{Z}$'s canonical metric. Contrary to f-divergences, the Wasserstein metric makes no assumptions of absolute continuity, and it allows practitioners to design problem specific uncertainty sets by choosing the most relevant distance metric ρ. This has made it a popular choice for DRO problems where such metrics are well defined (*e.g.* for real valued inputs), particularly in the context of adversarial robustness (Gao & Kleywegt, 2016; Esfahani & Kuhn, 2018; Sinha et al., 2018). Integral probability metrics (IPMs) are another common class of divergences between probability distributions to which (under some conditions, see Theorem 11.8.2 in Dudley (2018)) the Wasserstein distance belongs. They take the form $\gamma_{\mathcal{F}}(p, q) = \sup_{h \in \mathcal{F}} |\mathbb{E}_p\, h - \mathbb{E}_q\, h|$ where $\mathcal{F}$ is a set of real valued bounded measurable functions on $\mathcal{X} \times \mathcal{Y}$. Although Wasserstein-DRO is are by far more common in the literature, there has been some recent work on general IPM bounded uncertainty sets (Husain, 2020).

- **Moment constraints** consists in uncertainty sets where admissible distribution have to satisfy some (generally linear or quadratic) constraint on their moments, *e.g.* $\| \mathbb{E}_q - \mu \| \leq M$ for some pre-specified values μ and M. They are perhaps one of the oldest formulations of DRO, with roots in operations research (Scarf, 1957; Dupačová, 1987; Delage & Ye, 2010) and with some exceptions (Nguyen et al., 2020), they have seen relatively little use in the modern machine learning context.

- **CVaR constraints** (or α-coverage). Originating from the financial risk management literature (Rockafellar et al., 2000), the conditional value at risk (CVaR) is concerned with guaranteeing good performance under all distributions q that α-covered by p, meaning that for all (x, y), $q(x, y) \leq \alpha p(x, y)$ with $\alpha \in]0, 1]$. This formalizes in the language of probability distributions the idea that q is a subset that covers only a fraction α of the training data. CVaR-constrained uncertainty sets underpin a variety of recently proposed approaches for robust machine learning, even if they are not always explicitly acknowledged as such (Fan

et al., 2017; Curi et al., 2020; Levy et al., 2020).

All these nonparametric approaches are appealing as they require very little domain-specific knowledge, have well-understood theory (Duchi & Namkoong, 2018), and optimization procedures (*e.g.* Hu & Hong (2013) for KL constraints and Levy et al. (2020) for χ_2 and CVaR constraints). We refer to (Rahimian & Mehrotra, 2019) for a more in-depth overview.

Unfortunately, nonparametric DRO algorithms suffer from being overly pessimistic. Their uncertainty sets tend to include distributions that are exceedingly difficult to learn, or not representative of real-world distribution shifts. Furthermore, they often cannot enforce even basic constraints such as covariate shift (Duchi et al., 2020; Hu et al., 2018). Group-structured DRO uncertainty sets (Sagawa et al., 2020) overcome some of these challenges, but as alluded to before, they require significant domain expertise to pre-specify target sub-populations that a model should be robust to.

2.3.2 Measuring Robustness

Over the years, various works have attempted to come up with specific measures of model robustness in order to provide out-of-distribution generalization guarantees. Unfortunately, common divergence metrics such as the KL divergence ($\mathrm{KL}(p\|q) = \mathbb{E}_p \log \frac{q}{p}$; Kullback & Leibler (1951)) or the total variation distance ($d_{\mathrm{TV}}(p,q) = \sup_x |p(x) - q(x)|$; Saks (1937)) generally yield vacuous bounds.

Ben-David et al. (2010) (following a line of work initiated in Kifer et al. (2004) and Ben-David et al. (2007)) introduced a new distance metric called the $\mathcal{H}$-divergence, specifically geared towards yielding out-of-distribution generalization bounds. The central insight of the $\mathcal{H}$-divergence is to induce a distance that is specific to the family of models being trained (the titular $\mathcal{H}$). A key feature of the $\mathcal{H}$-divergence is that it can be approximated with another metric, the proxy $\mathcal{A}$-distance. In broad strokes, the proxy $\mathcal{A}$-distance is proportional to how easy it is to discriminate between the two domains p and q using classifiers from $\mathcal{H}$. More precisely, given two datasets U_p and U_q sampled from p and q respectively, it reads:

$$\hat{d}_{\mathcal{A}}(p,q) = 2(1 - 2\epsilon(U_p, U_q)) \tag{2.4}$$

with

$$\epsilon(U_p, U_q) = \min_{h \in \mathcal{H}} \frac{1}{|U_p| + |U_q|} \sum_{x \in U_p \cup U_q} |h(x) - \mathbb{1}_{x \in U_p}| \tag{2.5}$$

In other words, ϵ is the best error-rate (within $\mathcal{H}$) for distinguishing U_p from U_q. Naturally the exact minimum is in general intractable, but it can be estimated by training a classifier on finite amounts of training data sampled from p and q, making it more conducive to applied research.

Although the $\mathcal{H}$-divergence literature has inspired some work in "unsupervised domain adaptation"[6] (Ganin & Lempitsky (2015) and the follow-up literature on domain adversarial neural networks), it has seen relatively little use in NLP, with some notable exceptions (Blitzer & Pereira, 2007; Rai et al., 2010; Glorot et al., 2011). More recent empirical comparisons (Elsahar & Gallé, 2019; Kashyap et al., 2021) have shown the proxy $\mathcal{A}$-Distance to be competitive when it comes to predicting performance drop in the face of domain shift, possibly heralding a resurgence of the metric. We refer the interested reader to Kashyap et al. (2021) for a complete taxonomy and a more exhaustive overview of domain divergences (written with an NLP audience in mind).

With regards to adversarial perturbation specifically, there is a considerable amount of work devoted to providing certifiable robustness guarantees (see Katz et al. (2017); Sinha et al. (2018); Wong & Kolter (2018); Raghunathan et al. (2018) *inter alia*). Often, these guarantees rely on ideas related to Lipschitz continuity bounds: limiting the change in a model's output as a function of small perturbations in its inputs. These ideas are difficult to transpose to NLP as they rely on some notion of a "small change" in the input, which is usually described in terms of ℓ_2 norm for continuous data such as images, but is harder to formalize in natural languages. Jia et al. (2019) circumvent this issue by adapting an approach from the computer vision literature (Interval Bound Propagation (IBP); Dvijotham et al. (2018)) to specific neural architectures to yield robustness guarantees to word-level perturbations at the embedding level. Overall, guaranteed adversarial robustness in NLP is still an active area of research, with recent work extending these ideas to more modern architectures (Shi et al., 2020) for example.

[6]Training models to be robust to domain shift given unlabeled samples from the target domain

Part I

Measuring and Evaluating Robustness

Chapter 3

MTNT: A Testbed for Machine Translation of Noisy Text

3.1 Introduction

After this overview of the distributional shift problem, in this chapter we present our first contribution: a benchmark dataset for Machine Translation of Noisy Text (MTNT), consisting of noisy user-generated content such as the following comment:

> #nlproc is actualy f*ing hARD tbh 😂

This handcrafted sentence showcases several types of noise that are commonly seen on social media: abbreviations ("#nlproc"), typographical errors ("actualy"), obfuscated profanities ("f*ing"), inconsistent capitalization ("hARD"), Internet slang ("tbh" for "to be honest") and emojis (😂). Although machine translation has achieved significant quality improvements over the past few years due to the advent of Neural Machine Translation (NMT) Kalchbrenner & Blunsom (2013); Sutskever et al. (2014); Bahdanau et al. (2014); Wu et al. (2016), systems are still not robust to noisy input like this Belinkov & Bisk (2018a); Khayrallah & Koehn (2018). For example, Google Translate[1] translates the above example into French as:

> #nlproc est en train de f * ing dur hb

which translates back into English as "#nlproc is in the process of [f * ing] hard hb". This shows that noisy input can lead to erroneous translations that can be misinterpreted or even offensive.

[1] `translate.google.com` as of May 2018

Noise in social media text is a known issue that has been investigated in a variety of previous work Eisenstein (2013); Baldwin et al. (2013). Most recently, Belinkov & Bisk (2018a) have focused on the difficulties that character based NMT models have translating text with character level noise within individual words (from scrambling to simulated human errors such as typos or spelling/conjugation errors). This is a good first step towards noise-robust NMT systems, but as we demonstrate in Section 3.2, word-by-word replacement or scrambling of characters doesn't cover all the idiosyncrasies of language on the Internet.

At this point, despite the obvious utility of creating noise-robust MT systems, and the scientific challenges contained therein, there is currently a bottleneck in that there is no standard open benchmark for researchers and developers of Machine Translation (MT) systems to test the robustness of their models to these and other phenomena found in noisy text on the Internet. In this chapter, we introduce Machine Translation of Noisy Text (MTNT), a new, realistic dataset aimed at testing robustness of MT systems to these phenomena. The dataset contains naturally created noisy source sentences with professionally sourced translations both in a pair of typologically close languages (English and French) and distant languages (English and Japanese). We collect noisy comments from the Reddit[2] online discussion website (Section 3.3) in English, French and Japanese, and commissioned professional translators to translate to and from English, resulting in approximately 1000 test samples and from 6k to 36k training samples in four language pairs (English-French (`en-fr`), French-English (`fr-en`), English-Japanese (`en-ja`) and Japanese-English (`ja-en`)). In addition, we release additional small monolingual corpora in those 3 languages to both provide data for semi-supervised adaptation approaches as well as noisy Language Modeling (LM) experiments. We test standard translation models (Section 3.5) on our data to understand their failure cases and to provide baselines for future work.[3] Finally, we describe the 2019 and 2020 WMT Robustness shared task in which MTNT was used as a benchmark for robust machine translation (Section 3.7). The data is publicly available at `https://pmichel31415.github.io/mtnt/`.

3.2 Noise and Input Variations in Language on the Internet

3.2.1 Examples from Social Media Text

The term "noise" can encompass a variety of phenomena in natural language, with variations across languages (*e.g.*what is a typo in logographic writing systems?) and type of content Baldwin et al.

[2] `www.reddit.com`

[3] Additional language modeling experiments can be found in Appendix 3.6

(2013). To give the reader an idea of the challenges posed to MT and NLP systems operating on this kind of text, we provide a non-exhaustive list of types of noise and more generally input variations that deviate from standard MT training data we've encountered in Reddit comments:

- **Spelling/typographical errors**: "across" → "accross", "receive" → "recieve", "could have" → "could of", "temps" → "tant", "除く" → "覗く"
- **Word omission/insertion/repetition**: "je n'aime pas" → "j'aime pas","je pense" → "moi je pense"
- **Grammatical errors**: "a ton of" → "a tons of", "There are fewer people" → "There are less people"
- **Spoken language**: "want to" → "wanna", "I am" → "I'm", "je ne sais pas" → "chais pas", "何を笑っているの" → "何わろてんねん",
- **Internet slang**: "to be honest" → "tbh", "shaking my head" → "smh", "mort de rire" → "mdr", "笑" → "w"/"草"
- **Proper nouns** (with or without correct capitalization): "Reddit"→ "reddit"
- **Dialects**: African American Vernacular English, Scottish, Provençal, Québécois, Kansai, Tohoku...
- **Code switching**: "This is so cute" → "This is so kawaii", "C'est trop conventionel" → "C'est trop mainstream", "現在捏造中..." → "Now 捏造ing..."
- **Jargon**: on Reddit: "upvote", "downvote", "sub", "gild"
- **Emojis and other unicode characters**: ❤️,😀,😂,😍,😡, 😘, 😭
- **Profanities/slurs** (sometimes masked) "f*ck", "m*rde" ...

3.2.2 Is Translating Noisy Text just another Adaptation Problem?

To a certain extent, translating noisy text is a type of *adaptation*, which has been studied extensively in the context of both Statistical Machine Translation (SMT) and NMT Axelrod et al. (2011); Li et al. (2010); Luong & Manning (2015); Chu et al. (2017); Miceli Barone et al. (2017); Wang et al. (2017); Michel & Neubig (2018b). However, it presents many differences with previous domain

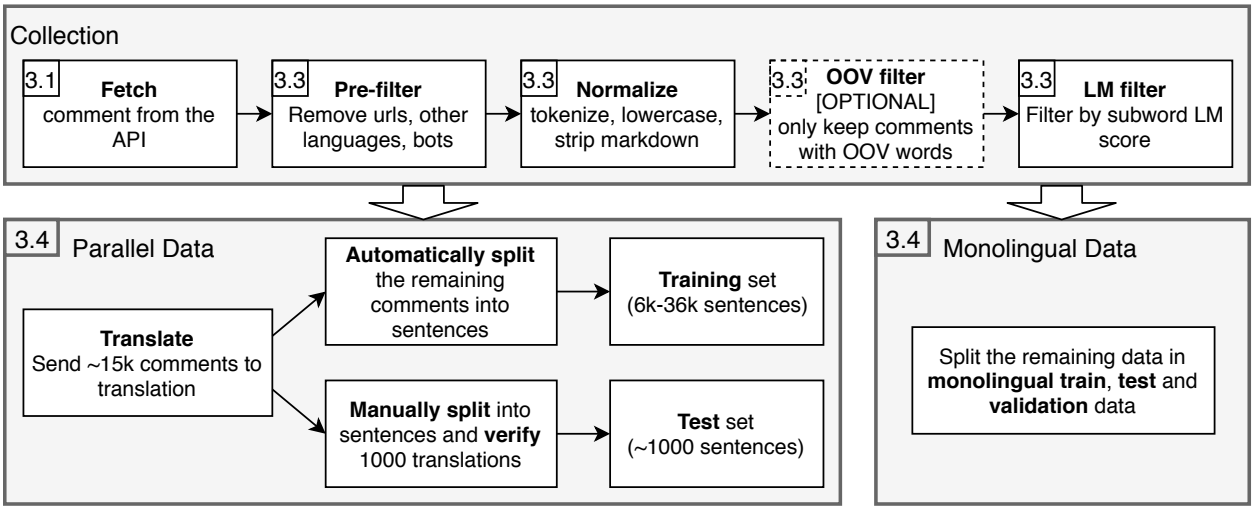


Figure 3.1: Summary of our collection process and the respective sections addressing them. We apply the same procedure for each language.

adaptation problems, where the main goal is to adapt from a particular topic or style. In the case of noisy text, it will not only be the case that a particular word will be translated in a different way than it is in the general domain (e.g. as in the case of "sub"), but also that there will be increased lexical variation (e.g. due to spelling or typographical errors), and also inconsistency in grammar (e.g. due to omissions of critical words or mis-usage). The sum of these differences warrants that noisy MT be treated as a separate instance than domain adaptation, and our experimental analysis in 3.5.4 demonstrates that even after performing adaptation, MT systems still make a large number of noise-related errors.

3.3 Collection Procedure

We first collect noisy sentences in our three languages of interest, English, French and Japanese. We refer to Figure 3.1 for an overview of the data collection and translation process.

We choose Reddit as a source of data because (1) its content is likely to exhibit noise, (2) some of its sub-communities are entirely run in different languages, in particular, English, French and Japanese, and (3) Reddit is a popular source of data in curated and publicly distributed NLP datasets Tan et al. (2016). We collect data using the public Reddit API. [4]

[4]In particular, we use this implementation: `praw.readthedocs.io/en/latest`, and our complete code is available at `http://www.cs.cmu.edu/~pmichel1/mtnt/`.

Note that the data collection and translation is performed at the comment level. We split the parallel data into sentences as a last step.

3.3.1 Data Sources

For each language, we select a set of communities ("subreddits") that we know contain many comments in that language:

English: Since an overwhelming majority of the discussions on Reddit are conducted in English, we don't restrict our collection to any community in particular.

French: `/r/france`, `/r/quebec` and `/r/rance`. The first two are among the biggest French speaking communities on Reddit. The third is a humor/sarcasm based offspring of `/r/france`.

Japanese: `/r/newsokur`, `/r/bakanewsjp`, `/r/newsokuvip`, `/r/lowlevelaware` and `/r/steamr`. Those are the biggest Japanese speaking communities, with over 2,000 subscribers at the time of collection.

We collect comments made during the 03/27/2018-03/29/3018 time period for English, 09/2018-03/2018 for French and 11/2017-03/2018 for Japanese. The large difference in collection time is due to the variance in comment throughput and relative amount of noise between the languages.

3.3.2 Contrast Corpora

Not all comments found on Reddit exhibit noise as described in Section 3.2. Because we would like to focus our data collection on noisy comments, we devise criteria that allow us to distinguish potentially noisy comments from clean ones. Specifically, we compile a *contrast* corpus composed of clean text that we can compare to, and find potentially noisy text that differs greatly from the contrast corpus. Given that our final goal is MT robust to noise, we prefer that these contrast corpora consist of the same type of data that is often used to train NMT models. We select different datasets for each language:

English: The English side of the preprocessed parallel training data provided for the German-English WMT 2017 News translation task,[5] as provided on the website. This amounts to ≈ 5.85 million sentences.

[5] `http://www.statmt.org/wmt17/translation-task.html`

French: The entirety of the French side of the parallel training data provided for the English-French WMT 2015 translation task.[6] This amounts to ≈ 40.86 million sentences.

Japanese: We aggregate three small/medium sized MT datasets: KFTT Neubig (2011), JESC Pryzant et al. (2017) and TED talks Cettolo et al. (2012), amounting to ≈ 4.19 million sentences.

3.3.3 Identifying Noisy Comments

We now describe the procedure used to identify comments containing noise.

Pre-filtering First, we perform three pre-processing to discard comments that do not represent natural noisy text in the language of interest:

1. Comments containing a URL, as detected by a regular expression.
2. Comments where the author's username contains "bot" or "AutoModerator". This mostly removes automated comments from bots.
3. Comments in another language: we run `langid.py`[7] Lui & Baldwin (2012) and discard comments where $p(\text{lang} \mid \text{comment}) > 0.5$ for any language other than the one we are interested in.

This removes cases that are less interesting, i.e. those that could be solved by rule-based pattern matching or are not natural text created by regular users in the target language. Our third criterion in particular discards comments that are blatantly in another language while still allowing comments that exhibit code-switching or that contain proper nouns or typos that might skew the language identification. In preliminary experiments, we noticed that these criteria 14.47, 6.53 and 7.09 % of the collected comments satisfied the above criteria respectively.

Normalization After this first pass of filtering, we pre-process the comments before running them through our noise detection procedure. We first strip Markdown[8] syntax from the comments. For English and French, we normalize the punctuation, lowercase and tokenize the comments using the Moses tokenizer. For Japanese, we simply lowercase the alphabetical characters in the

[6] `http://www.statmt.org/wmt15/translation-task.html`
[7] `https://github.com/saffsd/langid.py`
[8] `https://daringfireball.net/projects/markdown`

comments. Note that this normalization is done for the purpose of noise detection only. The collected comments are released without any kind of preprocessing. We apply the same normalization procedure to the contrast corpora.

Unknown words In the case of French and English, a clear indication of noise is the presence of out-of-vocabulary words (OOV): we record all lowercased words encountered in our reference corpus described in Section 3.3.2 and only keep comments that contain at least one OOV. Since we did not use word segmentation for the Japanese reference corpus, we found this method not to be very effective to select Japanese comments and therefore skipped this step.

Language model scores The final step of our noise detection procedure consists of selecting those comments with a low probability under a language model trained on the reference monolingual corpus. This approach mirrors the one used in Moore & Lewis (2010) and Axelrod et al. (2011) to select data similar to a specific domain using language model perplexity as a metric. We search for comments that have a low probability under a sub-word language model for more flexibility in the face of OOV words. We segment the contrast corpora with Byte-Pair Encoding (BPE) using the sentencepiece[9] implementation. We set the vocabulary sizes to $1,000$, $1,000$ and $4,000$ for English, French and Japanese respectively. We then use a 5-gram Kneser-Ney smoothed language model trained using `kenLM`[10] Heafield et al. (2013) to calculate the log probability, normalized by the number of tokens for every sentence in the reference corpus. Given a reddit comment, we compute the normalized log probability of each of its lines under our subword language model. If for any line this score is below the 1st percentile of scores in the reference corpus, the comment is labeled as noisy and saved.

3.3.4 Creating the Parallel Corpora

Once enough data has been collected, we isolate $15,000$ comments in each language by the following procedure:

- Remove all duplicates. In particular, this handles comments that might have been scraped twice or automatic comments from bots.

[9] `https://github.com/google/sentencepiece`
[10] `https://kheafield.com/code/kenlm/`

- To further weed out outliers (comments that are too noisy, *e.g.*ASCII art, wrong language... or not noisy enough), we discard comments that are on either end of the distribution of normalized LM scores within the set of collected comments. We only keep comments whose normalized score is within the 5-70 percentile for English (resp. 5-60 for French and 10-70 for Japanese). These numbers are chosen by manually inspecting the data.

- Choose $15,000$ samples at random.

We then concatenate the title of the thread where the comment was found to the text and send everything to an external vendor for manual translations. Upon reception of the translations, we noticed a certain amount of variation in the quality of translations, likely because translating social media text, with all its nuances, is difficult even for humans. In order to ensure the highest quality in the translations, we manually filter the data to segment the comments into sentences and weed out poor translations for our test data. We thereby retain around $1,000$ sentence pairs in each direction for the final test set.

We gather the samples that weren't selected for the test sets to be used for training or fine-tuning models on noisy data. We automatically split comments into sentences with a regular expression detecting sentence delimiters, and then align the source and target sentences. Should this alignment fail (*i.e.*the source comment contains a different number of sentences than the target comment after automatic splitting), we revert back to providing the whole comment without splitting. For the training data, we do not verify the correctness of translations as closely as for the test data. Finally, we isolate ≈ 900 samples in each direction to serve as validation data.

Information about the size of the data can be found in Table 3.1c, 3.1a and 3.1b for the test, training and validation sets respectively. We tokenize the English and French data with the Moses Koehn et al. (2007) tokenizer and the Japanese data with Kytea Neubig et al. (2011) before counting the number of tokens in each dataset.

3.3.5 Monolingual Corpora

After the creation of the parallel train and test sets, a large number of unused comments remain in each language, which we provide as monolingual corpora. This additional data has two purposes: first, it serves as a resource for in-domain training using semi-supervised methods relying on monolingual data (e.g. Cheng et al. (2016); Zhang & Zong (2016)). Second, it provides a language modeling dataset for noisy text in three languages.

We select $3,000$ comments at random in each dataset to form a validation set to be used to tune

Table 3.1: Datasets numbers.

	#samples	#src tokens	#trg tokens
en-fr	36,058	841k	965k
fr-en	19,161	661k	634k
en-ja	5,775	281k	506k
ja-en	6,506	172k	128k

(a) Training sets numbers.

#samples	#src tokens	#trg tokens
852	16,957	18,948
886	41,578	46,886
852	40,124	46,886
965	25,010	23,289

(b) Validation sets numbers.

	#samples	#src tokens	#trg tokens
en-fr	1,020	15,919	18,445
fr-en	1,022	16,662	16,038
en-ja	1,002	11,040	20,008
ja-en	1,020	23,997	33,429

(c) Test set numbers.

		#samples	#tok	#char
en	train	81,631	3,99M	18,9M
	dev	3,000	146k	698k
fr	train	26,485	1,52M	7,49M
	dev	3,000	176k	867k
ja	train	32,042	943k	3.9M
	dev	3,000	84k	351k

(d) Monolingual data numbers.

hyper-parameters, and provide the rest as training data. The data is provided with one comment per line. Newlines within individual comments are replaced with spaces. Table 3.1d contains information on the size of the datasets. As with the parallel MT data, we provide the number of tokens after tokenization with the Moses tokenizer for English and French and Kytea for Japanese.

3.4 Dataset Analysis

In this section, we investigate the proposed data to understand how different categories of noise are represented and to show that our test sets contain more noise overall than established MT benchmarks.

3.4.1 Quantifying Noisy Phenomena

We run a series of tests to count the number of occurrences of some of the types of noise described in Section 3.2. Specifically we pass our data through spell checkers to count spelling and grammar errors. Due to some of these tests being impractical to run on a large scale, we limit our analysis to the test sets of MTNT.

We use slightly different procedures depending on the tools available for each language. We test

Table 3.2: Numbers, per 100 tokens, of quantifiable noise occurrences. For each language and category, the dataset with the highest amount of noise is highlighted.

		Spelling	Grammar	Emojis	Profanities
en	newstest2014	0.210	0.189	0.000	0.030
	newsdiscusstest2015	0.621	0.410	0.021	0.076
	MTNT (en-fr)	**2.180**	**0.559**	**0.289**	**0.239**
fr	newstest2014	2.776	0.091	0.000	0.245
	newsdiscusstest2015	1.686	0.457	0.024	0.354
	MTNT	**4.597**	**1.464**	**0.252**	**0.690**
ja	TED	0.011	0.266	0.000	0.000
	KFTT	0.021	0.228	0.000	0.000
	JESC	0.096	0.929	0.090	**0.058**
	MTNT	**0.269**	**1.527**	**0.156**	0.036

for spelling and grammar errors in English data using Grammarly[11], an online resource for English spell-checking. Due to the unavailability of an equivalent of Grammarly in French and Japanese, we test for spelling and grammar error using the integrated spell-checker in Microsoft Word 2013[12]. Note that Word seems to count proper nouns as spelling errors, giving higher numbers of spelling errors across the board in French as compared to English.

For all languages, we also count the number of profanities and emojis using custom-made lists and regular expressions[13]. In order to compare results across datasets of different sizes, we report all counts per 100 words.

The results are recorded in the last row of each section in Table 3.2. In particular, for the languages with a segmental writing system, English and French, spelling errors are the dominant type of noise, followed by grammar error. Unsurprisingly, the former are much less present in Japanese.

3.4.2 Comparison to Existing Machine Translation Test Sets

Table 3.2 also provide a comparison with the relevant side of established MT test sets. For English and French, we compare our data to newstest2014[14] and newsdiscusstest2015[15] test sets. For

[11] https://www.grammarly.com/
[12] https://products.office.com/en-us/microsoft-word-2013
[13] available with our code at https://github.com/pmichel31415/mtnt
[14] http://www.statmt.org/wmt15/dev-v2.tgz
[15] http://www.statmt.org/wmt15/test.tgz

Japanese, we compare with the test sets of the datasets described in Section 3.3.2.

Overall, MTNT contains more noise in all metrics but one (there are more profanities in JESC, a Japanese subtitle corpus). This confirms that MTNT indeed provides a more appropriate benchmark for translation of noisy or non-standard text.

Compared to synthetically created noisy test sets Belinkov & Bisk (2018a) MTNT contains less systematic spelling errors and more varied types of noise (*e.g.*emojis and profanities) and is thereby more representative of naturally occurring noise.

3.5 Machine Translation Experiments

We evaluate standard NMT models on our proposed dataset to assess its difficulty. Our goal is not to train state-of-the art models but rather to test standard off-the-shelf NMT systems on our data, and elucidate what features of the data make it difficult.

3.5.1 Model Description

All our models are implemented in DyNet Neubig et al. (2017) with the XNMT toolkit Neubig et al. (2018). We use approximately the same setting for all language pairs: the encoder is a bidirectional LSTM with 2 layers, the attention mechanism is a multi layered perceptron and the decoder is a 2 layered LSTM. The embedding dimension is 512, all other dimensions are 1024. We tie the target word embeddings and the output projection weights Press & Wolf (2017). We train with Adam Kingma & Ba (2014) with XNMT's default hyper-parameters, as well as dropout (with probability 0.3). We used BPE subwords to handle OOV words. Full configuration details as well as code to reproduce the baselines is available at `https://github.com/pmichel31415/mtnt`.

3.5.2 Training Data

We train our models on standard MT datasets:

- en $\leftrightarrow$ fr: Our training data consists in the europarl-v7[16] and news-commentary-v10[17] corpora, totaling $2,164,140$ samples, $54,611,105$ French tokens and $51,745,611$ English tokens (non-tokenized). We use the newsdiscussdev2015[14] dev set from WMT15 as validation data and evaluate the model on the newsdiscusstest2015[15] and newstest2014[14] test sets.

[16] `http://www.statmt.org/europarl/`
[17] `http://www.statmt.org/wmt15/training-parallel-nc-v10.tgz`

- en $\leftrightarrow$ ja: We concatenate the respective train, validation and test sets of the three corpora mentioned in 3.3.2. In particular we detokenize the Japanese part of each dataset to make sure that any tokenization we perform will be uniform (in practice we remove ASCII spaces). This amounts to $3,900,772$ training samples ($34,989,346$ English tokens without tokenization). We concatenate the dev sets associated with these corpora to serve as validation data and evaluate on each respective test set separately.

Table 3.3: BLEU scores of NMT models on the various datasets.

	en-fr	fr-en
newstest2014	33.52	28.93
newsdiscusstest2015	33.03	30.76
MTNT	21.77	23.27
MTNT (+tuning)	29.73	30.29
	en-ja	ja-en
TED	14.51	13.25
KFTT	20.82	20.77
JESC	15.77	18.00
MTNT	9.02	6.65
MTNT (+tuning)	12.45	9.82

3.5.3 Results

We use `sacreBLEU`[18], a standardized BLEU score evaluation script proposed by Post (2018b), for BLEU evaluation of our benchmark dataset. It takes in detokenized references and hypotheses and performs its own tokenization before computing BLEU score. We specify the `intl` tokenization option. In the case of Japanese text, we run both hypothesis and reference through KyTea before computing BLEU score. We strongly encourage that evaluation be performed in the same manner in subsequent work, and will provide both scripts and an evaluation web site in order to facilitate reproducibility.

Table 3.3 lists the BLEU scores for our models on the relevant test sets in the two language pairs, including the results on MTNT.

[18] `https://github.com/mjpost/sacreBLEU`

Table 3.4: Comparison of our model's output before and after fine-tuning in `fr-en`.

Source	Moi faire la gueule dans le métro me manque, c'est grave ?
Target	I miss sulking in the underground, is that bad?
Our model	I do not know what is going on in the metro, that is a serious matter.
+ fine-tuning	I do not want to be in the metro, it's serious?
Source	* C'est noël / pâques / pentecôte / toussaint : Pick One, je suis pas catho
Target	Christmas / Easter / Pentecost / All Saints: Pick One, I'm not Catholic!
Our model	<unk> It is a pale/poward, a palec<unk>te d'<unk>tat: Pick One, I am not a catho!
+ fine-tuning	<unk> It's no<unk>l / pesc<unk>e /pentecate /mainly: Pick One, I'm not catho!

3.5.4 Analysis of the MT outputs

To better understand the types of errors made by our model, we count the n-grams that are over- and under- generated with respect to the reference translation. Specifically, we compare the count ratios of all 1- to 3-grams in the output and in the reference and look for the ones with the highest (over-generated) and lowest (under-generated) ratio.

We find that in English, the model under-generates the contracted form of the negative ("do not"/"don't") or of auxiliaries ("That is"/"I'm"). Similarly, in French, our model over generates "de votre" (where "votre" is the formal 2nd person plural for "your") and "n'ai pas" which showcases the "ne [...] pas" negation, often dropped in spoken language. Conversely, the informal second person "tu" is under-generated, as is the informal and spoken contraction of "cela", "ça". In Japanese, the model under-generates, among others, the informal personal pronoun 俺 ("ore") or the casual form だ ("da") of the verb です ("desu", to be). In `ja-en` the results are difficult to interpret as the model seems to produce incoherent outputs (*e.g.* "no, no, no...") when the NMT system encounters sentences it has not seen before. The full list of n-grams with the top 5 and bottom 5 count ratios in each language pair is displayed in Table 3.5.

3.5.5 Fine-Tuning

Finally, we test a simple domain adaptation method by fine-tuning our models on the training data described in Section 3.3.4. We perform one epoch of training with vanilla SGD with a learning rate of 0.1 and a batch size of 32. We do not use the validation data at all. As evidenced by the results in the last row of Table 3.3, this drives BLEU score up by 3.17 to 7.96 points depending on the language pair. However large this increase might be, our model still breaks on very noisy sentences. Table 3.4 shows three examples in `fr-en`. Although our model somewhat improves

Table 3.5: Over and under generated n-grams in our model's output for `en-fr`

fr-en	en-fr	ja-en	en-ja
Over generated			
<unk>	<unk>	no, no,	※
it is not	qu'ils	i	が
I do not	de votre	no, no, no,	か？
That is	s'il	so on and	て
not have	n'ai pas	on and so	すか？
Under generated			
it's	tu	I	？
I'm	ça	Is	よ。
I don't	que tu	>	って
>	!	""The	俺
doesn't	as	those	だ。

after fine-tuning, the translations remain inadequate in all cases. In the third case, our model downright fails to produce a coherent output. This shows that despite improving BLEU score, naive domain adaptation by fine-tuning doesn't solve the problem of translating noisy text.

3.6 Language Modeling Experiments

In addition to our MT experiments, we report character-level language modeling results on the monolingual part of our dataset. We use the data described in Section 3.3.5 as training and validation sets. We evaluate the trained model on the source side of our `en-fr`, `fr-en` and `ja-en` test sets for English, French and Japanese respectively.

We report results for two models: a Kneser-Ney smoothed 6-gram model (implemented with KenLM) and an implementation of the AWD-LSTM proposed in Merity et al. (2018)[19]. We report the Bit-Per-Character (bpc) counts in table 3.6. We intend these results to serve as a baseline for future work in language modeling of noisy text in either of those three languages.

[19] `https://github.com/salesforce/awd-lstm-lm`

Table 3.6: Language modeling scores

	6-gram		AWD LSTM	
	dev	test	dev	test
English	2.081	2.179	1.706	1.810
French	1.906	2.090	1.449	1.705
Japanese	5.003	5.497	4.801	5.225

3.7 MTNT in Action: the 2019 and 2020 WMT Robustness Shared Tasks

A year after MTNT was originally published, the dataset was featured in the Robustness shared task[20] at the 2019 conference on machine translation (WMT). Participants were allowed to use large amounts of out-of-domain data from the main WMT translation task[21] or from the three `en-ja` corpora described in Section 3.5, as well as the MTNT dataset itself as a small in-domain corpus. Submitted systems were evaluated on new, blind test sets collected using the same procedure as MTNT *i.e.* scraped from Reddit, filtered out for noisy comments using a sub-word language modeling criterion and translated by professionals.

The shared task attracted 23 submissions from 11 teams. Methods used by competing teams included variants of data cleaning (removal of noisy training samples), the use of placeholders (to account for special characters that could be copied such as emojis), data augmentation strategies (such as back-translation or the addition of filtered data from external sources), domain-aware training (via the addition of domain tags), fine-tuning and ensembling. Table 3.7 showcases an example where specific handling of casing allowed some systems to outperform others on samples containing ALL CAPS (a common phenomenon in user-generated content on social media). A more detailed description of the submissions, as well as an analysis of the results, can be found in the findings paper Li et al. (2019).

The shared task was renewed in 2020, with a broader focus on general domain robustness[22]. In this iteration, participants were asked to train models in one of two language pairs (English-German and English Japanese) using only the data available for the WMT20 news translation task (Barrault et al., 2020). The evaluation data was aggregated from three different domains:

[20] `http://www.statmt.org/wmt19/robustness.html`
[21] `http://www.statmt.org/wmt15/translation-task.html`
[22] `http://www.statmt.org/wmt20/robustness.html`

	Output	BLEU+1
Ref	From Sri Lanka , to Russia , to the United States , to Japan I mean it 's a market THAT GOES EVERYWHERE .	
CUNI	from sri lanka , to russia , to the united states , to japon I mean it 's a market QUI VA PARTOUT .	33.0
NLE	From Sri Lanka , to Russia , to the United States , to Japan I mean it 's a market THAT GOES EVERYWHERE .	100

Table 3.7: An example of handling of casing in two `fr-en` systems, CUNI (Helcl et al., 2019) and NLE (Bérard et al., 2019). The CUNI system does not recognize the ALL CAPS text in the French sentence, and simply copies the source. On the other hand, the NLE system uses a specific *case tagging* technique which enables it to properly translate the relevant phrase, while preserving the original casing.

Wikipedia comments with toxic content[23], speech recognition transcripts (Wang et al., 2020b), and a new test set of reddit comments, collected following again the same procedure as MTNT. Evaluation proceeded in two distinct phases. In the first "zero-shot" phase, submitted systems were simply evaluated against the unseen test sets, thus testing the domain robustness of models trained using only out-of-domain data. In the second "few-shot" phase, participants were given a single week to fine-tune their systems using limited amounts of training data from each domain (including the MTNT training data). Each system output was evaluated using both automatic metrics (BLEU) and human judgements. In the case of the latter, evaluators were not only asked to rate the machine-generated translations on a numerical scale of 1 to 5, but also to indicate the presence of "catastrophic errors" belonging to pre-defined categories such as "introduction of toxicity", "mis-translation of named entities" or "change in units/time/date/numbers"

This edition of the task received 59 submissions by 11 participating teams from both industry and academic institutions. By and large, most teams used similar approaches as the previous year, specifically tagged back-translations, ensembling or fine-tuning on filtered data. In addition, some teams used adaptor modules (Bapna & Firat, 2019) for more efficient fine-tuning, with some level of success. The human evaluation of the results uncovered a predominance of named entity mis-translations among catastrophic errors, followed by sentiment reversal and introduction of toxicity. We refer to the findings paper (Specia et al., 2020) for a full description of the data collection process, evaluation procedure and participating systems.

[23] `https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge`

Acknowledgement *The work discussed in this section was carried out in two different papers, Li et al. (2019) and Specia et al. (2020). The present author was not first author on either of these papers. His contributions were, for Li et al. (2019) to help with the general organization of the task, including establishing the goal of the task, planning, handling submissions and in particular to be responsible for collecting the blind test sets. For Specia et al. (2020), his contributions were more focused on data collection for the reddit test set.*

3.8 Related work

Handling noisy text has received growing attention among various language processing tasks due to the abundance of user generated content on popular social media platforms Crystal (2001); Herring (2003); Danet & Herring (2007). These contents are considered as noisy when compared to news corpora which have been the main data source for language tasks Baldwin et al. (2013); Eisenstein (2013). They pose several unique challenges because they contain a larger variety of linguistic phenomena that are absent in the news domain and that lead to degraded quality when applying an model to out-of-domain data (Ritter et al., 2011; Luong & Manning, 2015). Additionally, they are live examples of the Cmabrigde Uinervtisy (Cambridge University) effect, where state-of-the-art models become brittle while human's language processing capability is more robust (Sakaguchi et al., 2017; Belinkov & Bisk, 2018a).

Efforts to address these challenges have been focused on creating in-domain datasets and annotations (Owoputi et al., 2013; Kong et al., 2014; Blodgett et al., 2017), and domain adaptation training (Luong & Manning, 2015). In MT, improvements were obtained for SMT (Formiga & Fonollosa, 2012). However, the specific challenges for neural machine translation have not been studied until recently Belinkov & Bisk (2018a); Sperber et al. (2017); Cheng et al. (2018b). The first provides empirical evidence of non-trivial quality degradation when source sentences contain natural noise or synthetic noise within words, and the last two explore data augmentation and adversarial approaches of adding noise efficiently to training data to improve robustness.

Our work also contributes to recent advances in evaluating neural machine translation quality with regard to specific linguistic phenomena, such as manually annotated test sentences for English to French translation, in order to identify errors due to specific linguistic divergences between the two languages (Isabelle et al., 2017), or automatically generated test sets to evaluate typical errors in English to German translation (Sennrich, 2017). Our contribution distinguishes itself from this previous work and other similar initiatives (Peterson, 2011) by providing an open test set consisting of naturally occurring text exhibiting a wide range of phenomena related to noisy input text from

contemporaneous social media.

3.9 Discussion

In this chapter, we proposed a new dataset to test MT models for robustness to the types of noise encountered in natural language on the Internet. We contribute parallel training and test data in both directions for two language pairs, English $\leftrightarrow$ French and English $\leftrightarrow$ Japanese, as well as monolingual data in those three languages. We show that this dataset contains more noise than existing MT test sets and poses a challenge to models trained on standard MT corpora. We further demonstrate that these challenges cannot be overcome by a simple domain adaptation approach alone. After its publication, MTNT served as the basis for two shared tasks at the WMT conference in machine translation, fostering research on models and evaluation metrics for this specific problem (Li et al., 2019; Specia et al., 2020). In addition, it was used independently in a variety of subsequent work, either as a benchmark for *e.g.* data augmentation techniques (Karpukhin et al., 2019; Vaibhav et al., 2019) or as an exemplar of domain shift in machine translation (Michel et al., 2019a).

Chapter 4

On Evaluation of Adversarial Perturbations for Sequence-to-Sequence Models

4.1 Introduction

While the previous chapter's focus was on evaluating models under distributional shift to the social media domain (a type of shift which we referred to as a "global" shift in Chapter 2), in this chapter we take a closer look at a different type of distributional shift: adversarial perturbations. At a high level, attacking a machine learning model with adversarial perturbations is the process of making changes to its input to maximize an adversarial goal, such as mis-classification Szegedy et al. (2013) or mis-translation Zhao et al. (2018). These attacks provide insight into the vulnerabilities of machine learning models and their brittleness to samples outside the training distribution. Lack of robustness to these attacks poses security concerns to safety-critical applications, *e.g.* self-driving cars Bojarski et al. (2016).

Adversarial attacks were first defined and investigated for computer vision systems (Szegedy et al. (2013); Goodfellow et al. (2014b); Moosavi-Dezfooli et al. (2016) inter alia), where the input space is continuous, making minuscule perturbations largely imperceptible to the human eye. In discrete spaces such as natural language sentences, the situation is more problematic; even a flip of a single word or character is generally perceptible by a human reader. Thus, most of the mathematical framework in previous work is not directly applicable to discrete text data. Moreover, there is no canonical distance metric for textual data like the ℓ_p norm in real-valued vector spaces such as images, and evaluating the level of semantic similarity between two sentences is a field of research of its own Cer et al. (2017). This elicits a natural question: *what does the term*

"adversarial perturbation" mean in the context of NLP?

We propose a simple but natural criterion for adversarial examples in NLP, particularly untargeted[1] attacks on sequence-to-sequence (seq2seq) models: *adversarial examples should be meaning-preserving on the source side, but meaning-destroying on the target side*. The focus on explicitly evaluating meaning preservation is in contrast to previous work on adversarial examples for seq2seq models Belinkov & Bisk (2018b); Zhao et al. (2018); Cheng et al. (2018a); Ebrahimi et al. (2018a). Nonetheless, this feature is extremely important; given two sentences with equivalent meaning, we would expect a good model to produce two outputs with equivalent meaning. In other words, any meaning-preserving perturbation that results in the model output changing drastically highlights a fault of the model.

A first technical contribution of this chapter is to lay out a method for formalizing this concept of meaning-preserving perturbations (Section 4.2). This makes it possible to evaluate the effectiveness of adversarial attacks or defenses either using gold-standard human evaluation, or approximations that can be calculated without human intervention. We further propose a simple method of imbuing gradient-based word substitution attacks (Section 4.3.1) with simple constraints aimed at increasing the chance that the meaning is preserved (Section 4.3.2).

Our experiments are designed to answer several questions about meaning preservation in seq2seq models. First, we evaluate our proposed "source-meaning-preserving, target-meaning-destroying" criterion for adversarial examples using both manual and automatic evaluation (Section 4.4.2) and find that a less widely used evaluation metric (chrF) provides significantly better correlation with human judgments than the more widely used BLEU and METEOR metrics. We proceed to perform an evaluation of adversarial example generation techniques, finding that chrF does help to distinguish between perturbations that are more meaning-preserving across a variety of languages and models (Section 4.4.3). Finally, we apply existing methods for adversarial training to the adversarial examples with these constraints and show that making adversarial inputs more semantically similar to the source is beneficial for robustness to adversarial attacks and does not decrease test performance on the original data distribution (Section 4.5). A toolkit implementing our evaluation framework is released at `https://github.com/pmichel31415/teapot-nlp`.

[1]Here we use the term untargeted in the same sense as Ebrahimi et al. (2018a): an attack whose goal is simply to decrease performance with respect to a reference translation.

4.2 A Framework for Evaluating Adversarial Attacks

In this section, we present a simple procedure for evaluating adversarial attacks on seq2seq models. We will use the following notation: x and y refer to the source and target sentence respectively. We denote x's translation by model M as y_M. Finally, $\hat{x}$ and $\hat{y}_M$ represent an adversarially perturbed version of x and its translation by M, respectively. The nature of M and the procedure for obtaining $\hat{x}$ from x are irrelevant to the discussion below.

4.2.1 The Adversarial Trade-off

The goal of adversarial perturbations is to produce failure cases for the model M. Hence, the evaluation must include some measure of the *target similarity* between y and y_M, which we will denote $\mathrm{s}_{\text{tgt}}(y, \hat{y}_M)$. However, if no distinction is being made between perturbations that preserve the meaning and those that don't, a sentence like "he's very *friendly*" is considered a valid adversarial perturbation of "he's very *adversarial*", even though its meaning is the opposite. Hence, it is crucial, when evaluating adversarial attacks on MT models, that the discrepancy between the original and adversarial input sentence be quantified in a way that is sensitive to meaning. Let us denote such a *source similarity* score $\mathrm{s}_{\text{src}}(x, \hat{x})$.

Based on these functions, we define the *target relative score decrease* as:

$$\mathrm{d}_{\text{tgt}}(y, y_M, \hat{y}_M) = \begin{cases} 0 \text{ if } \mathrm{s}_{\text{tgt}}(y, \hat{y}_M) \geq \mathrm{s}_{\text{tgt}}(y, y_M) \\ \frac{\mathrm{s}_{\text{tgt}}(y, y_M) - \mathrm{s}_{\text{tgt}}(y, \hat{y}_M)}{\mathrm{s}_{\text{tgt}}(y, y_M)} \text{ otherwise.} \end{cases} \tag{4.1}$$

The choice to report the *relative* decrease in s_{tgt} makes scores comparable across different models or languages[2]. For instance, for languages that are comparatively easy to translate (*e.g.* French-English), s_{tgt} will be higher in general, and so will the gap between $\mathrm{s}_{\text{tgt}}(y, y_M)$ and $\mathrm{s}_{\text{tgt}}(y, \hat{y}_M)$. However this does not necessarily mean that attacks on this language pair are more effective than attacks on a "difficult" language pair (*e.g.* Czech-English) where s_{tgt} is usually smaller.

We recommend that both s_{src} and d_{tgt} be reported when presenting adversarial attack results. However, in some cases where a single number is needed, we suggest reporting the attack's *success* $\mathcal{S} := \mathrm{s}_{\text{src}} + \mathrm{d}_{\text{tgt}}$. The interpretation is simple: $\mathcal{S} > 1 \Leftrightarrow \mathrm{d}_{\text{tgt}} > 1 - \mathrm{s}_{\text{src}}$, which means that the attack has destroyed the target meaning (d_{tgt}) more than it has destroyed the source meaning ($1 - \mathrm{s}_{\text{src}}$).

Importantly, this framework can be extended beyond strictly meaning-preserving attacks. For

[2]Note that we do not allow negative d_{tgt} to keep all scores between 0 and 1.

example, for targeted keyword introduction attacks Cheng et al. (2018a); Ebrahimi et al. (2018a), the same evaluation framework can be used if s_{tgt} (resp. s_{src}) is modified to account for the presence (resp. absence) of the keyword (or its translation in the source). Similarly this can be extended to other tasks by adapting s_{tgt} (*e.g.* for classification one would use the zero-one loss, and adapt the success threshold).

4.2.2 Similarity Metrics

Throughout Section 4.2.1, we have not given an exact description of the semantic similarity scores s_{src} and s_{tgt}. Indeed, automatically evaluating the semantic similarity between two sentences is an open area of research and it makes sense to decouple the definition of adversarial examples from the specific method used to measure this similarity. In this section, we will discuss manual and automatic metrics that may be used to calculate it.

Human Judgment

Judgment by speakers of the language of interest is the *de facto* gold standard metric for semantic similarity. Specific criteria such as adequacy/fluency Ma & Cieri (2006), acceptability Goto et al. (2013), and 6-level semantic similarity Cer et al. (2017) have been used in evaluations of MT and sentence embedding methods. In the context of adversarial attacks, we propose the following 6-level evaluation scheme, which is motivated by previous measures, but designed to be (1) symmetric, like Cer et al. (2017), (2) and largely considers meaning preservation but at the very low and high levels considers fluency of the output[3], like Goto et al. (2013):

> How would you rate the similarity between the meaning of these two sentences?
>
> 0. The meaning is completely different or one of the sentences is meaningless
> 1. The topic is the same but the meaning is different
> 2. Some key information is different
> 3. The key information is the same but the details differ
> 4. Meaning is essentially equal but some expressions are unnatural
> 5. Meaning is essentially equal and the two sentences are well-formed English[a]
>
> [a] Or the language of interest.

[3] This is important to rule out nonsensical sentences and distinguish between clean and "noisy" paraphrases (*e.g.* typos, non-native speech...). We did not give annotators additional instruction specific to typos.

Automatic Metrics

Unfortunately, human evaluation is expensive, slow and sometimes difficult to obtain, for example in the case of low-resource languages. This makes automatic metrics that do not require human intervention appealing for experimental research. This section describes 3 evaluation metrics commonly used as alternatives to human evaluation, in particular to evaluate translation models.[4]

BLEU: Papineni et al. (2002) is an automatic metric based on n-gram precision coupled with a penalty for shorter sentences. It relies on exact word-level matches and therefore cannot detect synonyms or morphological variations.

METEOR: Denkowski & Lavie (2014) first estimates alignment between the two sentences and then computes unigram F-score (biased towards recall) weighted by a penalty for longer sentences. Importantly, METEOR uses stemming, synonymy and paraphrasing information to perform alignments. On the downside, it requires language specific resources.

chrF: Popović (2015) is based on the character n-gram F-score. In particular we will use the chrF2 score (based on the F2-score — recall is given more importance), following the recommendations from Popović (2016). By operating on a sub-word level, it can reflect the semantic similarity between different morphological inflections of one word (for instance), without requiring language-specific knowledge which makes it a good one-size-fits-all alternative.

Because multiple possible alternatives exist, it is important to know which is the best stand-in for human evaluation. To elucidate this, we will compare these metrics to human judgment in terms of Pearson correlation coefficient on outputs resulting from a variety of attacks in Section 4.4.2.

4.3 Gradient-Based Adversarial Attacks

In this section, we overview the adversarial attacks we will be considering in the rest of this chapter.

4.3.1 Attack Paradigm

We perform gradient-based attacks that replace one word in the sentence so as to maximize an adversarial loss function $\mathcal{L}_{\text{adv}}$, similar to the substitution attacks proposed in Ebrahimi et al. (2018b).

[4]Note that other metrics of similarity are certainly applicable within the overall framework of Section 4.2.2, but we limit our examination in this chapter to the three noted here.

Table 4.1: Examples of different adversarial inputs. The substituted word is highlighted.

Original	**Pourquoi** faire cela ?
English gloss	**Why** do this?
Unconstrained	**construisant** (English: building) faire cela ?
kNN	**interrogez** (English: interrogate) faire cela ?
CharSwap	**Puorquoi** (typo) faire cela ?
Original	Si seulement je pouvais me muscler **aussi** rapidement.
English gloss	If only I could build my muscle **this** fast.
Unconstrained	Si seulement je pouvais me muscler **etc** rapidement.
kNN	Si seulement je pouvais me muscler **plsu** (typo for "more") rapidement.
CharSwap	Si seulement je pouvais me muscler **asusi** (typo) rapidement.

General Approach

Precisely, for a word-based translation model M[5], and given an input sentence $w_1, \ldots, w_n$, we find the position i^* and word w^* satisfying the following optimization problem:

$$\underset{1 \leq i \leq n, \hat{w} \in \mathcal{V}}{\arg\max} \; \mathcal{L}_{\text{adv}}(w_0, \ldots, w_{i-1}, \hat{w}, w_{i+1}, \ldots, w_n) \tag{4.2}$$

where $\mathcal{L}_{\text{adv}}$ is a differentiable function which represents our adversarial objective. Using the first order approximation of $\mathcal{L}_{\text{adv}}$ around the original word vectors $\mathbf{w}_1, \ldots, \mathbf{w}_n$[6], this can be derived to be equivalent to optimizing

$$\underset{1 \leq i \leq n, \hat{w} \in \mathcal{V}}{\arg\max} \; [\hat{\mathbf{w}} - \mathbf{w}_i]^\intercal \nabla_{\mathbf{w}_i} \mathcal{L}_{\text{adv}} \, . \tag{4.3}$$

The above optimization problem can be solved by brute-force in $\mathcal{O}(n|\mathcal{V}|)$ space complexity, whereas the time complexity is bottlenecked by a $|\mathcal{V}| \times d$ times $n \times d$ matrix multiplication, which is not more computationally expensive than computing logits during the forward pass of the model. Overall, this naive approach is sufficiently fast to be conducive to adversarial training. We also found that the attacks benefited from normalizing the gradient by taking its sign.

Extending this approach to finding the optimal perturbations for more than 1 substitution would require exhaustively searching over all possible combinations. However, previous work Ebrahimi et al. (2018a) suggests that greedy search is a good enough approximation.

[5]Note that this formulation is also valid for character-based models (see Ebrahimi et al. (2018a)) and subword-based models. For subword-based models, additional difficulty would be introduced due to changes to the input resulting in different subword segmentations. This poses an interesting challenge that is beyond the scope of the current work.

[6]More generally we will use the bold $\mathbf{w}$ when talking about the embedding vector of word w

The Adversarial Loss $\mathcal{L}_{\text{adv}}$

We want to find an adversarial input $\hat{x}$ such that, assuming that the model has produced the correct output $y_1, \ldots, y_{t-1}$ up to step $t-1$ during decoding, the probability that the model makes an error at the next step t is maximized.

In the log-semiring, this translates into the following loss function:

$$\mathcal{L}_{\text{adv}}(\hat{x}, y) = \sum_{t=1}^{|y|} \log(1 - p(y_t \mid \hat{x}, y_1, \ldots, y_{t-1})) \tag{4.4}$$

4.3.2 Enforcing Semantically Similar Adversarial Inputs

In contrast to previous methods, which don't consider meaning preservation, we propose simple modifications of the approach presented in Section 4.3.1 to create adversarial perturbations at the word level that are more likely to preserve meaning. The basic idea is to restrict the possible word substitutions to similar words. We compare two sets of constraints:

kNN: This constraint enforces that the word be replaced only with one of its 10 nearest neighbors in the source embedding space. This has two effects: first, the replacement will be likely semantically related to the original word (if words close in the embedding space are indeed semantically related, as hinted by Table 4.1). Second, it ensures that the replacement's word vector is close enough to the original word vector that the first order assumption is more likely to be satisfied.

CharSwap: This constraint requires that the substituted words must be obtained by swapping characters. Word internal character swaps have been shown to not affect human readers greatly McCusker et al. (1981), hence making them likely to be meaning-preserving. Moreover we add the additional constraint that the substitution must not be in the vocabulary, which will likely be particularly meaning-destroying on the target side for the word-based models we test here. In such cases where word-internal character swaps are not possible or can't produce OOV words, we resort to the naive strategy of repeating the last character of the word. The exact procedure used to produce this kind of perturbations is described in Listing 4.3.2. Note that for a word-based model, every OOV will look the same (a special <unk> token), however the choice of OOV will still have an influence on the output of the model because we use unk-replacement.

```
1 def make_oov(
2   word,
3   vocab,
4   max_scrambling,
```

```
):
  """Modify a word to make it OOV
  (while keeping the meaning)"""
  # If the word has >3 letters
  # try scrambling them
  L = len(word)
  if L > 3:
    # For a fixed number of steps
    for _ in range(max_scrambling):
      # Swap two adjacent letters
      # in the middle of the word
      pos = random.randint(1, L - 3)
      word = word[:pos]
      word += word[pos+1] + word[pos]
      word += word[pos+2:]
      # If we got an OOV already just
      # return it
      if word not in vocab:
        return word
  # If nothing worked, or the word is
  # too short for scrambling, just
  # repeat the last letter ad nauseam
  char = word[-1]
  while word in vocab:
    word = word + char
  return word
```

In contrast, we refer the base attack without constraints as **Unconstrained** hereforth. Table 4.1 gives qualitative examples of the kind of perturbations generated under the different constraints.

For subword-based models, we apply the same procedures at the subword-level on the original segmentation. We then de-segment and re-segment the resulting sentence (because changes at the subword or character levels are likely to change the segmentation of the resulting sentence).

4.4 Experiments

Our experiments serve two purposes. First, we examine our proposed framework of evaluating adversarial attacks (Section 4.2), and also elucidate which automatic metrics correlate better with human judgment for the purpose of evaluating adversarial attacks (Section 4.4.2). Second, we use this evaluation framework to compare various adversarial attacks and demonstrate that adversarial attacks that are explicitly constrained to preserve meaning receive better assessment scores (Section

4.4.3).

4.4.1 Experimental setting

Data: Following previous work on adversarial examples for seq2seq models Belinkov & Bisk (2018b); Ebrahimi et al. (2018a), we perform all experiments on the IWSLT2016 dataset Cettolo et al. (2016) in the {French,German,Czech}→English directions (`fr-en`, `de-en` and `cs-en`). We compile all previous IWSLT test sets before 2015 as validation data, and keep the 2015 and 2016 test sets as test data. The data is tokenized with the Moses tokenizer Koehn et al. (2007). The exact data statistics can be found in Table 4.2.

Table 4.2: IWSLT2016 data statistics.

	#train	#valid	#test
`fr-en`	220.4k	6,824	2,213
`de-en`	196.9k	11,825	2,213
`cs-en`	114.4k	5,716	2,213

MT Models: We perform experiments with two common NMT models. The first is an LSTM based encoder-decoder architecture with attention (Luong et al., 2015). It uses 2-layer encoders and decoders, and dot-product attention. We set the word embedding dimension to 300 and all others to 500. The second model is a self-attentional Transformer Vaswani et al. (2017), with 6 1024-dimensional encoder and decoder layers and 512 dimensional word embeddings. Both the models are trained with Adam Kingma & Ba (2014), dropout Srivastava et al. (2014) of probability 0.3 and label smoothing Szegedy et al. (2016) with value 0.1. We experiment with both word based models (vocabulary size fixed at 40k) and subword based models (BPE Sennrich et al. (2016) with 30k operations). For word-based models, we perform `<unk>` replacement, replacing `<unk>` tokens in the translated sentences with the source words with the highest attention value during inference. The full experimental setup and source code are available at `https://github.com/pmichel31415/translate/tree/paul/pytorch_translate/research/adversarial/experiments`.

Automatic Metric Implementations: To evaluate both sentence and corpus level BLEU score, we first de-tokenize the output and use `sacreBLEU`[7] Post (2018b) with its internal `intl` tokenization, to keep BLEU scores agnostic to tokenization. We compute METEOR using the official

[7] `https://github.com/mjpost/sacreBLEU`

implementation[8]. ChrF is reported with the `sacreBLEU` implementation on detokenized text with default parameters. A toolkit implementing the evaluation framework described in Section 4.2.1 for these metrics is released at `https://github.com/pmichel31415/teapot-nlp`.

4.4.2 Correlation of Automatic Metrics with Human Judgment

Table 4.3: Correlation of automatic metrics to human judgment of adversarial source and target sentences. "*" indicates that the correlation is significantly better than the next-best one.

Language	BLEU	METEOR	chrF
French	0.415	0.440	**0.586***
English	0.357	0.478*	**0.497**

We first examine which of the automatic metrics listed in Section 4.2.2 correlates most with human judgment for our adversarial attacks. For this experiment, we restrict the scope to the case of the LSTM model on `fr-en`. For the French side, we randomly select 900 sentence pairs $(x, \hat{x})$ from the validation set, 300 for each of the Unconstrained, kNN and CharSwap constraints. To vary the level of perturbation, the 300 pairs contain an equal amount of perturbed input obtained by substituting 1, 2 and 3 words. On the English side, we select 900 pairs of reference translations and translations of adversarial input $(y, \hat{y}_M)$ with the same distribution of attacks as the source side, as well as 300 (y, y_M) pairs (to include translations from original inputs). This amounts to 1,200 sentence pairs in the target side.

These sentences are sent to English and French speaking annotators to be rated according to the guidelines described in Section 4.2.2. Each sample (a pair of sentences) is rated by two independent evaluators. If the two ratings differ, the sample is sent to a third rater (an auditor and subject matter expert) who makes the final decision.

Finally, we compare the human results to each automatic metric with Pearson's correlation coefficient. The correlations are reported in Table 4.3. As evidenced by the results, chrF exhibits higher correlation with human judgment, followed by METEOR and BLEU. This is true both on the source side (x vs $\hat{x}$) and in the target side (y vs $\hat{y}_M$). We evaluate the statistical significance of this result using a paired bootstrap test for $p < 0.01$. Notably we find that chrF is significantly better than METEOR in French but not in English. This is not too unexpected because METEOR has access to more language-dependent resources in English (specifically synonym information) and thereby can make more informed matches of these synonymous words and phrases. Moreover

[8] `http://www.cs.cmu.edu/~alavie/METEOR/`

Table 4.4: Correlation of automatic metrics to human judgment of semantic similarity between original and adversarial source sentences, broken down by number of perturbed words. "*" indicates that the correlation is significantly better than the next-best one.

# edits	BLEU	METEOR	chrF
1	0.351	0.352	**0.486***
2	0.403	0.424	**0.588***
3	0.334	0.393	**0.560***

Table 4.5: Correlation of automatic metrics to human judgment of semantic similarity between original and adversarial source sentences, broken down by type of constraint on the perturbation. "*" indicates that the correlation is significantly better than the next-best one.

Constraint	BLEU	METEOR	chrF
Unconstrained	0.274	0.572	**0.599**
CharSwap	0.274	0.319	**0.383**
kNN	0.534	0.584	**0.606**

the French source side contains more "character-level" errors (from CharSwap attacks) which are not picked-up well by word-based metrics like BLEU and METEOR.

We provide a breakdown of the correlation coefficients of automatic metrics with human judgment for source-side meaning-preservation, both in terms of number of perturbed words (Table 4.4) and constraint (Table 4.5). While those coefficients are computed on a much smaller sample size, and their differences are not all statistically significant with $p < 0.01$, they exhibit the same trend as the results from Table 4.3 (BLEU $<$ METEOR $<$ chrF). In particular Table 4.4 shows that the good correlation of chrF with human judgment is not only due to the ability to distinguish between different number of edits.

Thus, in the following, we report attack results both in terms of chrF in the source ($\mathrm{s}_{\mathrm{src}}$) and relative decrease in chrF (RDchrF) in the target (d_{tgt}).

4.4.3 Attack Results

We can now compare attacks under the three constraints Unconstrained, kNN and CharSwap and draw conclusions on their capacity to preserve meaning in the source and destroy it in the target. Attacks are conducted on the validation set using the approach described in Section 4.3.1 with 3 substitutions (this means that each adversarial input is at edit distance at most 3 from the original input). Results (on a scale of 0 to 100 for readability) are reported in Table 4.6 for both word-

Table 4.6: Target RDchrF and source chrF scores for all the attacks on all our models (word- and subword-based LSTM and Transformer).

		LSTM			Transformer		
	Language pair	cs-en	de-en	fr-en	cs-en	de-en	fr-en
		Target RDChrF			Target RDChrF		
Word-based	Original chrF	45.68	49.43	57.49	47.66	51.08	58.04
	Unconstrained	25.38	25.54	25.59	25.24	25.00	24.68
	CharSwap	24.11	24.94	23.60	21.59	23.23	21.75
	kNN	15.00	15.59	15.22	20.74	19.97	18.59
		Source chrF			Source chrF		
	Unconstrained	70.14	72.39	74.29	69.03	71.93	73.23
	CharSwap	82.65	84.40	86.62	84.13	85.97	87.02
	kNN	78.08	78.11	77.62	74.94	77.92	77.88
		Target RDChrF			Target RDChrF		
Subword-based	Original chrF	48.30	52.42	59.08	49.70	54.01	59.65
	Unconstrained	25.79	26.03	26.96	23.97	25.07	25.28
	CharSwap	18.65	19.15	19.75	16.98	18.38	17.85
	kNN	15.00	16.26	17.12	19.02	18.58	18.63
		Source chrF			Source chrF		
	Unconstrained	69.32	72.12	73.57	68.66	71.51	72.65
	CharSwap	85.84	87.46	87.98	85.79	87.07	87.99
	kNN	76.17	77.74	78.03	73.05	75.91	76.54

and subword- based LSTM and Transformer models. To give a better idea of how the different variables (language pair, model, attack) affect performance, we give a graphical representation of these same results in Figure 4.1 for the word-based models. The rest of this section discusses the implication of these results.

Source chrF Highlights the Effect of Adding Constraints: Comparing the kNN and CharSwap rows to Unconstrained in the "source" sections of Table 4.6 clearly shows that constrained attacks have a positive effect on meaning preservation. Beyond validating our assumptions from Section 4.3.2, this shows that source chrF is useful to carry out the comparison in the first place[9]. To give a point of reference, results from the manual evaluation carried out in Section 4.4.2 show that that 90% of the French sentence pairs to which humans gave a score of 4 or 5 in semantic similarity have a chrF > 78.

[9]It can be argued that using chrF gives an advantage to CharSwap over kNN for source preservation (as opposed to METEOR for example). We find that this is the case for Czech and German (source METEOR is higher for kNN) but not French. Moreover we find (see *e.g.* Table 4.4) that chrF correlates better with human judgement even for kNN.

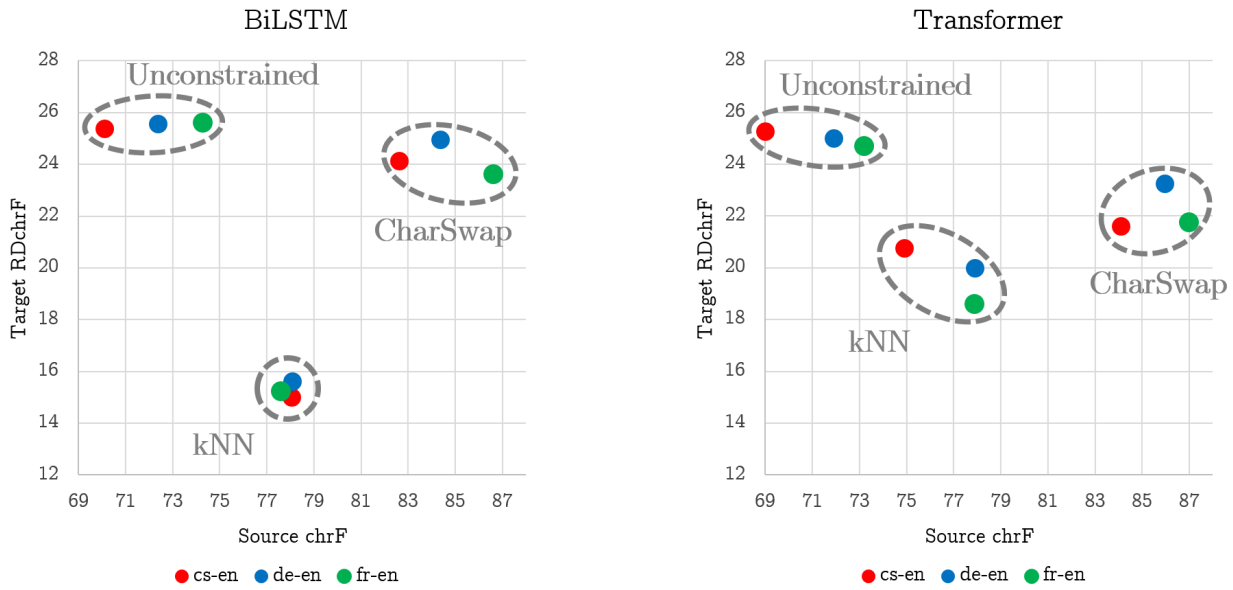


Figure 4.1: Graphical representation of the results in Table 4.6 for word-based models. High source chrF and target RDchrF (upper-right corner) indicates a good attack.

Different Architectures are not Equal in the Face of Adversity: Inspection of the target-side results yields several interesting observations. First, the high RDchrF of CharSwap for word-based model is yet another indication of their known shortcomings when presented with words out of their training vocabulary, even with `<unk>`-replacement. Second, and perhaps more interestingly, Transformer models appear to be less robust to small embedding perturbations (kNN attacks) compared to LSTMs. Although the exploration of the exact reasons for this phenomenon is beyond the scope of this work, this is a good example that RDchrF can shed light on the different behavior of different architectures when confronted with adversarial input. Overall, we find that the CharSwap constraint is the only one that consistently produces attacks with > 1 average success (as defined in Section 4.2.1) according to Table 4.6. Table 4.7 contains two qualitative examples of this attack on the LSTM model in `fr-en`.

4.5 Adversarial Training with Meaning-Preserving Attacks

4.5.1 Adversarial Training

Adversarial training Goodfellow et al. (2014b) augments the training data with adversarial examples. Formally, in place of the negative log likelihood (NLL) objective on a sample x, y,

Table 4.7: Example of CharSwap attacks on the `fr-en` LSTM. The first example is a successful attack (high source chrF and target RDchrF) whereas the second is not.

Successful attack (source chrF = 80.89, target RDchrF = 84.06)	
Original	Ils le réinvestissent directement en engageant plus de procès.
Adv. src	Ilss le réinvestissent dierctement en engagaent plus de procès.
Ref.	They plow it right back into filing more troll lawsuits.
Base output	They direct it directly by engaging more cases.
Adv. output	.. de plus.
Unsuccessful attack (source chrF = 54.46, target RDchrF = 0.00)	
Original	C'était en Juillet 1969.
Adv. src	C'étiat en Jiullet 1969.
Ref.	This is from July, 1969.
Base output	This was in July 1969.
Adv. output	This is. in 1969.

$\mathcal{L}(x, y) = NLL(x, y)$, the loss function is replaced with an interpolation of the NLL of the original sample x, y and an adversarial sample $\hat{x}, y$:

$$\mathcal{L}'(x, y) = (1 - \alpha)NLL(x, y) + \alpha NLL(\hat{x}, y) \tag{4.5}$$

Ebrahimi et al. (2018a) suggest that while adversarial training improves robustness to adversarial attacks, it can be detrimental to test performance on non-adversarial input. We investigate whether this is still the case when adversarial attacks are largely meaning-preserving.

In our experiments, we generate $\hat{x}$ by applying 3 perturbations on the fly at each training step. To maintain training speed we do not solve Equation (4.2) iteratively but in one shot by replacing the argmax by top-3. Although this is less exact than iterating, this makes adversarial training time less than $2\times$ slower than normal training. We perform adversarial training with perturbations without constraints (Unconstrained-adv) and with the CharSwap constraint (CharSwap-adv). All experiments are conducted with the word-based LSTM model.

4.5.2 Results

Test performance on non-adversarial input is reported in Table 4.8. In keeping with the rest of the paper, we primarily report chrF results, but also show the standard BLEU as well.

Language pair	cs-en	de-en	fr-en
Base	44.21 (22.89)	49.30 (28.61)	55.67 (35.28)
	$\alpha = 1.0$		
Unconstrained-adv	41.38 (21.51)	46.15 (27.06)	53.39 (33.96)
CharSwap-adv	43.74 (23.00)	48.85 (28.45)	55.60 (35.33)
	$\alpha = 0.5$		
Unconstrained-adv	43.68 (22.93)	48.60 (28.30)	55.55 (35.25)
CharSwap-adv	44.57 (23.66)	49.14 (28.66)	55.88 (35.63)

Table 4.8: ChrF (BLEU) scores on the original test set before/after adversarial training of the word-based LSTM model.

We observe that when $\alpha = 1.0$, *i.e.* the model only sees the perturbed input during training[10], the Unconstrained-adv model suffers a drop in test performance, whereas CharSwap-adv's performance is on par with the original. This is likely attributable to the spurious training samples $(\hat{x}, y)$ where y is not an acceptable translation of $\hat{x}$ introduced by the lack of constraint. This effect disappears when $\alpha = 0.5$ because the model sees the original samples as well.

Not unexpectedly, Table 4.9 indicates that CharSwap-adv is more robust to CharSwap constrained attacks for both values of α, with 1.0 giving the best results. On the other hand, Unconstrained-adv is similarly or more vulnerable to these attacks than the baseline. Hence, we can safely conclude that adversarial training with CharSwap attacks improves robustness while not impacting test performance as much as unconstrained attacks.

4.6 Related work

Following seminal work on adversarial attacks by Szegedy et al. (2013), Goodfellow et al. (2014b) introduced gradient-based attacks and adversarial training. Since then, a variety of attack Moosavi-Dezfooli et al. (2016) and defense Cissé et al. (2017); Kolter & Wong (2017) mechanisms have been proposed. Adversarial examples for NLP specifically have seen attacks on sentiment Papernot et al. (2016); Samanta & Mehta (2017); Ebrahimi et al. (2018b), malware Grosse et al. (2016),

[10]This setting is reminiscent of word dropout Iyyer et al. (2015).

Language pair	cs-en	de-en	fr-en
Base	24.11	24.94	23.60
		$\alpha = 1.0$	
Unconstrained-adv	25.99	26.24	25.67
CharSwap-adv	16.46	17.19	15.72
		$\alpha = 0.5$	
Unconstrained-adv	26.52	27.26	24.92
CharSwap-adv	20.41	20.24	16.08

Table 4.9: Robustness to CharSwap attacks on the validation set with/without adversarial training (RDchrF). Lower is better.

gender Reddy & Knight (2016) or toxicity Hosseini et al. (2017) classification to cite a few.

In MT, methods have been proposed to attack word-based Zhao et al. (2018); Cheng et al. (2018a) and character-based Belinkov & Bisk (2018b); Ebrahimi et al. (2018a) models. However these works side-step the question of meaning preservation in the source: they mostly focus on target side evaluation. Finally there is work centered around meaning-preserving adversarial attacks for NLP via paraphrase generation Iyyer et al. (2018) or rule-based approaches Jia & Liang (2017); Ribeiro et al. (2018); Naik et al. (2018); Alzantot et al. (2018). However the proposed attacks are highly engineered and focused on English.

4.7 Conclusion

This chapter highlights the importance of performing *meaning-preserving* adversarial perturbations for NLP models (with a focus on seq2seq). We proposed a general evaluation framework for adversarial perturbations and compared various automatic metrics as proxies for human judgment to instantiate this framework. We then confirmed that, in the context of MT, "naive" attacks do not preserve meaning in general, and proposed alternatives to remedy this issue. Finally, we have shown that a careful choice of more meaning-preserving perturbations is beneficial for adversarial training.

Since the original publication of the content in this chapter in 2019, a number of works have followed-up, either by adapting the evaluation framework to other tasks, *e.g.* semantic parsing in Huang et al. (2021), or by building upon it for designing more imperceptible adversarial perturbations, for instance using the proposed evaluation metrics as rewards for reinforcement learning based perturbation generation (Zou et al., 2020), or pushing further the concept of indistinguishable

perturbations with encoding specific character substitutions (Boucher et al., 2021).

Part II

Making Robust Models

Chapter 5

Modeling the Second Player in Distributionally Robust Optimization

5.1 Introduction

Up to this point, we have principally been concerned with *evaluating* the effect that various types of distributional shifts have on existing models. In this chapter and the next, we now tackle the problem of *training* models that are more robust against distributional shifts.

The tendency of machine learning models to exhibit drops in accuracy when confronted with data from domains that are absent from or under-represented in their training data often arises from the objective function of ERM. Recall that in ERM, the parameters θ of the model are learned by minimizing the expectation of a loss function ℓ under a data distribution p (or, specifically in practice, an associated empirical data distribution $\hat{p}$)

$$\mathcal{L}_{\text{ERM}}(\theta) = \mathbb{E}_{(x,y)\sim\hat{p}}\ell(x, y, \theta). \tag{5.1}$$

When the model encounters data sampled from a different distribution $q_{\text{test}} \neq p$, performance can suffer significantly. As described in Chapter 2, distributionally robust optimization (DRO) provides a natural solution to this issue by replacing the expected risk under a single distribution p with the *worst* expected risk over a pre-determined family of distributions $\mathcal{Q}$ (the "uncertainty set")

$$\mathcal{L}_{\text{DRO}}(\theta) = \max_{q\in\mathcal{Q}} \mathbb{E}_{(x,y)\sim q}\, \ell(x, y, \theta). \tag{5.2}$$

If $\mathcal{Q}$ contains q_{test}, the DRO objective upper bounds the expected risk under q_{test}. However, *a priori*

knowledge of possible test distributions is not always available or easy to acquire. For example, training a model to be robust to some demographic attributes ($\mathcal{Q} = \{q_{\text{demographic 1}}, q_{\text{demographic 2}}, \ldots\}$) requires collecting and annotating data with the necessary information, an expensive and ethically fraught endeavour. In the absence of such information, one has to resort to defining the uncertainty set analytically, drawing on one's intuition of what constitutes a possible test distribution given the observed training distribution, such as using moment constraints (Delage & Ye, 2010; Nguyen et al., 2020), f-divergence (Ben-Tal et al., 2013; Hu & Hong, 2013; Faury et al., 2020), Wasserstein/IPM (Sinha et al., 2018; Husain, 2020) balls, or coarse-grained mixture models (Oren et al., 2019; Hu et al., 2018). However, the need for keeping the inner supremum in Eq. (5.2) tractable limits the possible choices.

In this chapter, we propose that the uncertainty set be instead defined as a family of parametric generative models. The resulting DRO objective (§5.2) is a differentiable game with two players: the original model $\ell(x, y; \theta)$ and a model of its worst-case distribution $q_\psi(x, y)$, the titular "second player" which we hereafter refer to as *the adversary*. Using this formulation — which we call P-DRO — allows for more flexibility in the choice of the adversary's architecture (and so the uncertainty set). Unfortunately, finding a solution of this game via direct application of simultaneous gradient descent (Singh et al., 2000) is difficult (Balduzzi et al., 2018). In particular, direct gradient descent on the uncertainty set suffers from instability due to the large variance of the gradients (Greensmith et al., 2004), and hyper-parameter selection is not straightforward.

To address these challenges, we make two main contributions (§5.3): first, we propose a new relaxation of the DRO game's inner maximization problem (with KL constraints). The resulting objective is more amenable to simultaneous gradient update than the original zero-sum game and significantly improves training stability, while still yielding useful adversaries. Second, we develop a principled approach for selecting hyper-parameters: we leverage the learned adversaries to decide which of any two given models trained with P-DRO is more robust than the other.

We do an in-depth set of experiments analyzing the effect of our proposed changes on both a toy task as well as a more realistic, yet still synthetic sentiment classification task (§5.4). Finally, we show that in the more realistic setting of toxicity detection, P-DRO yields models that are more robust to changes in demographic groups, even though these groups are unknown at training time, opening up applications in combatting dataset bias (§5.5). Code to reproduce our experiments can be found at `https://github.com/pmichel31415/P-DRO`.

5.2 Parameterizing the Uncertainty Set

Consider a model parameterized by $\theta \in \mathbb{R}^{d_{\text{model}}}$. Minimizing the DRO objective described in Eq. (5.2) over the uncertainty set $\mathcal{Q}$ turns the optimization problem into the min-max (or zero-sum) game

$$\min_{\theta \in \mathbb{R}^d} \max_{q \in \mathcal{Q}} \mathbb{E}_{(x,y)\sim q}\, \ell(x, y, \theta). \tag{5.3}$$

The first player controls the parameters θ, whilst the second player controls the worst-case distribution q. In the absence of explicit information on groups of interest (such as demographics, domain, etc.), an adequate choice of the uncertainty set $\mathcal{Q}$ is critical to the success of DRO. This is in fact very much an active area of research (Sinha et al. (2018); Duchi & Namkoong (2018); Oren et al. (2019), see Rahimian & Mehrotra (2019) for a survey). $\mathcal{Q}$ must be sufficiently large to contain test distributions of interest, but if it is too large it may contain "adversarial" distributions on which no model can perform well. Moreover, the design of $\mathcal{Q}$ is also circumscribed by the necessity of keeping the min-max problem tractable, particularly in the context of stochastic optimization. In Hu & Hong (2013) and Duchi et al. (2016) for example, the choice of f-divergence balls allows the use of duality arguments to reformulate (5.3) as a more manageable min-min problem. Others, like Hu et al. (2018) or Oren et al. (2019), propose using mixture models, the simplicity of which enables them to solve the inner maximization problem efficiently.

Instead, we propose to explicitly model the second player in the DRO game as a parametric model q_ψ of the data. Of course, not all parameterizations $\psi \in \mathbb{R}^{d_{\text{adv}}}$ of a given generative model represent useful distributions, and we require that the adversary stay "close" to the underlying true data distribution p. As a measure of distance between q_ψ and p, we choose the KL (Kullback & Leibler, 1951) divergence due to its wide acceptance in the machine learning community, as well as its appealing properties in the context of DRO.[1] The KL upper bound, κ, is left as a parameter to be decided by the experimenter. We refer to the resulting DRO formulation as Parametric DRO

$$\min_{\theta} \max_{\substack{\psi \\ \text{KL}(q_\psi \| p) \leq \kappa}} \underbrace{\mathbb{E}_{(x,y)\sim q_\psi} \ell(x, y, \theta)}_{\mathcal{L}_{\text{P-DRO}}(\theta, \psi)}. \tag{5.4}$$

5.3 Optimizing P-DRO

The min-max problem in Eq. (5.4) belongs to a class of games called "differentiable games" (another famous representative being generative adversarial networks (Goodfellow et al., 2014a)). We

[1]For instance: $\text{KL}(q\|p) < +\infty$ implies that q stays within the support of p

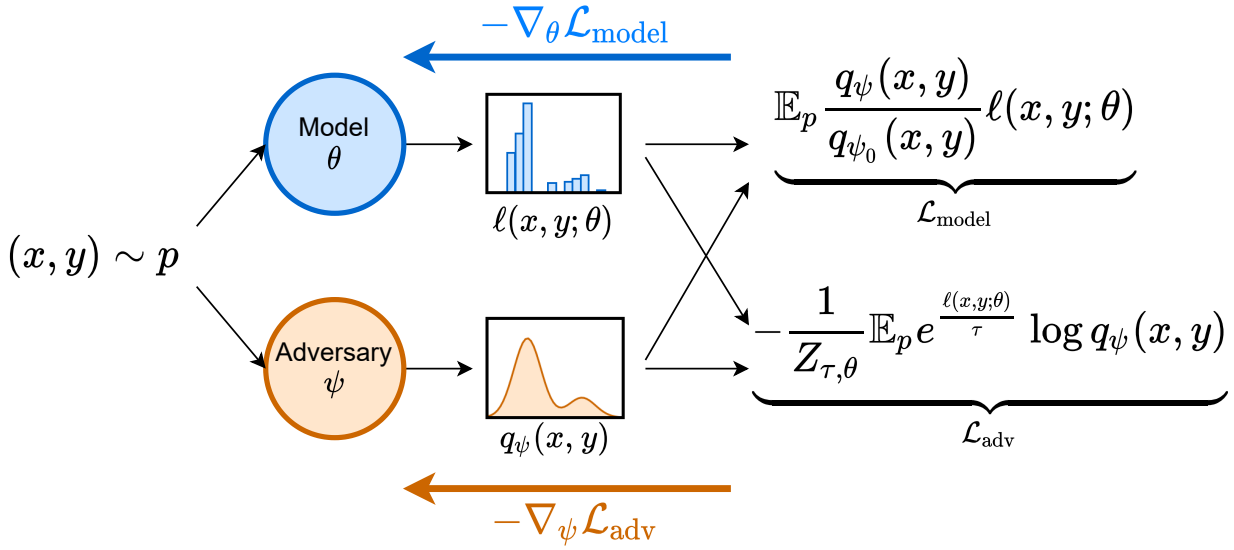


Figure 5.1: Summary of P-DRO: At every step of training, (x, y) pairs are sampled from the data distribution p and fed to both the model θ and the adversary ψ. For every sample, the model produces loss values $\ell(x, y; \theta)$ and the adversary produces densities $q_\psi(x, y)$. Both are combined into $\mathcal{L}_{\text{model}}$ and $\mathcal{L}_{\text{adv}}$, which are used to update the θ and ψ respectively, via simultaneous gradient updates.

can search for a solution of this game with simultaneous gradient descent (Singh et al., 2000), *i.e.* by simultaneously updating θ and ψ with $-\nabla_\theta \mathcal{L}_{\text{P-DRO}}$ and $\nabla_\psi \mathcal{L}_{\text{P-DRO}}$ respectively. Unfortunately, in general, there is no theoretical guarantee that simultaneous gradient descent will converge to a Nash equilibrium[2] (Balduzzi et al., 2018), nor that any such equilibrium even exists if the objective is non-convex in θ (or non-concave in ψ). The success of GANs and the follow-up literature (Wang et al., 2019c) serves as an encouraging example that gradient based methods can yield useful solutions despite the pessimistic theoretical results. In this section, we discuss difficulties that arise when optimizing θ and ψ jointly, and propose modifications of the objective to address them.

5.3.1 Training the Model θ

We could train the model θ by taking negative gradient steps on $\mathbb{E}_{(x,y)\sim q_\psi} \ell(x, y; \theta)$. This gradient can be estimated by sampling examples from q_ψ and averaging the gradient of their losses.

[2]Nash equilibria (Osborne & Rubinstein, 1994) can be thought of the game theoretic analog of global minima in optimization.

Unfortunately, this objective requires that q_ψ is well-behaved at all iterations, as it is the only source of supervision for θ. If q_ψ is initialized incorrectly or begins producing unrealistic (x, y), the quality of θ degrades as it begins to learn a predictor on invalid training examples from q_ψ. As an alternative, we opt to compute the gradients for θ with importance sampling, *i.e.* rewriting $\mathcal{L}_{\text{P-DRO}}$ as $\mathbb{E}_{(x,y)\sim p} \frac{q_\psi(x,y)}{p(x,y)} \ell(x, y; \theta)$, which ensures that all (x, y) samples will be derived from the training set itself. Unfortunately, the true density p is unknown to us. As an approximation, we replace $\frac{q_\psi(x,y)}{p(x,y)}$ with the likelihood ratio between q_ψ and the maximum likelihood estimate of p, $q_{\psi_0} \coloneqq \arg\max_{q_\psi} \mathbb{E}_{(x,y)\sim p} \log q_\psi(x, y)$. This changes the min-max problem to

$$\min_{\theta} \max_{\substack{\psi \\ \text{KL}(q_\psi \| p) \leq \kappa}} \underbrace{\mathbb{E}_{(x,y)\sim p} \frac{q_\psi(x, y)}{q_{\psi_0}(x, y)} \ell(x, y, \theta)}_{\mathcal{L}_{\text{model}}} . \tag{5.5}$$

This becomes a simple expected loss objective, which we can estimate by sampling from the empirical distribution $\hat{p}$. In experiments, we find that with this formulation we are able to train robust θ even when q_ψ is only a mediocre generative model (see Appendix 5.6.2). To further stabilize training at the beginning of the optimization process, we initialize ψ with ψ_0, making the objective exactly the same as ERM for the first gradient step.

5.3.2 Training the Adversary ψ

According to Eq. (5.5) the adversary ψ must maximize $E_{(x,y)\sim q_\psi} \frac{p(x,y)}{q_{\psi_0}(x,y)} \ell(x, y, \theta)$ within a KL ball of fixed radius. This is challenging for several reasons: first, enforcing the bound is intractable for complex families of adversaries where *e.g.* projecting onto the KL ball is another difficult optimization problem of its own. Second, maximizing the expectation with respect to the parameters of the distribution q_ψ is prone to instability due to large gradient variance (Greensmith et al., 2004).

Lagrangian Relaxation To address the first difficulty, we loosen the strict KL constraint and instead consider the Lagrangian relaxation $\mathbb{L}$

$$\mathbb{L}(\psi, \tau) = \mathbb{E}_{(x,y)\sim q_\psi} \frac{p(x, y)}{q_{\psi_0}(x, y)} \ell(x, y, \theta) - \tau \left(\text{KL}(q_\psi \| p) - \kappa \right) . \tag{5.6}$$

We fix the Lagrangian multiplier $\tau > 0$ as treat it as a "temperature" hyper-parameter. With some reorganization (which we develop in Appendix A.1), we can show that

$$\mathbb{L}(\psi, \tau) = -\tau \text{KL}(q_\psi \| q^*_{\tau,\theta}) + C. \tag{5.7}$$

Where $q^*_{\tau,\theta} \propto p(x,y) e^{\frac{p(x,y)}{q_{\psi_0}(x,y)} \frac{\ell(x,y;\theta)}{\tau}}$ and C is a constant in ψ. In other words, maximizing $\mathbb{L}$ in ψ is equivalent to minimizing the KL divergence between q_ψ and $q^*_{\tau,\theta}$. One difficulty with this objective is that $q^*_{\tau,\theta}$ depends upon the unknown probability density $p(x,y)$. We avoid this problem by treating the density ratio $\frac{p(x,y)}{q_{\psi_0}(x,y)}$ as a constant, which is closely related to assumptions that have been used successfully in past formulations of DRO (Oren et al., 2019). Empirically, we find that incorporating q_{ψ_0} as a surrogate for p is a serviceable approximation, as demonstrated in Section 5.4.

Reversing the KL Minimizing the KL divergence in this direction is difficult for several reasons. First, it entails optimizing an expectation in q_ψ over ψ, which is difficult due to the large variance of the gradients (Greensmith et al., 2004). Second, computing this KL necessitates access to the true theoretical density $p(x,y)$ in order to compute $q^*_{\tau,\theta}(x,y)$ in the argument of the expectation, but this quantity is unknown in practice.[3] To sidestep these issues, we elect to minimize the reverse direction $\text{KL}(q^*_{\tau,\theta} \| q_\psi)$ instead. Due to the KL divergence being non-symmetric, this is a rather crude approximation[4], the implications of which are discussed in Norouzi et al. (2016). However, we find that this approach dramatically stabilizes the gradient dynamics while still yielding good adversaries, as observed empirically in Section 5.4.7. Discarding the entropy term (constant in ψ), the resulting problem is equivalent to minimizing

$$\mathcal{L}_{\text{adv}}(\psi, \tau) \coloneqq -\frac{1}{Z_{\tau,\theta}} \mathbb{E}_p \, e^{\frac{\ell(x,y;\theta)}{\tau}} \log q_\psi(x,y) \tag{5.8}$$

in ψ, where $Z_{\tau,\theta} = \mathbb{E}_p \, e^{\frac{\ell(x,y;\theta)}{\tau}}$ is the normalizer of q^*. In this case, we can estimate this expectation by substituting the empirical distribution $\hat{p}$ for p in the expectation.

[3]Note that substituting the empirical distribution $\hat{p}$ for p poses issues here, because q_ψ is not absolutely continuous with respect to $\hat{p}$.

[4]For instance, the optimum of the reverse KL doesn't necessarily match that of the forward KL within the parametric confusion set $\mathcal{Q}$

Computing the Normalizer Approximating the inverse normalizer $\frac{1}{Z_{\tau,\theta}}$ in a minibatch yields a biased estimator. On the other hand, computing $Z_{\tau,\theta}$ over the entire training data at each step is prohibitive since it requires computing the loss of every single example. As a middle ground, we keep a running normalizer $\tilde{Z}_k$ computed from the average of the normalizers over a fixed number k of consecutive minibatches. In other words, if B_i and θ_i denote the minibatch and adversary parameters at step i respectively, the normalizer at step t will be

$$\tilde{Z}_k = \frac{1}{\sum_{i=t-k}^{t} |B_i|} \sum_{i=t-k}^{t} \sum_{x,y \in B_i} e^{\frac{\ell(x,y;\theta_i)}{\tau}}. \tag{5.9}$$

If k is too low, there is a risk of under-estimating the normalizer, especially if the distribution of weights contains infrequent high weight samples. On the other hand, if k is too high there is a risk of using "stale" weights in the normalizer. In experiments, we treat k as a hyper-parameter.

5.3.3 Optimal Stopping

When should one stop training a model with P-DRO? In ERM it is customary to stop training after the empirical risk — periodically evaluated on a held out validation dataset — stops decreasing. This is particularly important to prevent over-fitting to the training data. However, it is not an appropriate criterion for P-DRO, since the model is not trained to minimize empirical risk in the first place. A more pertinent choice is to compare the robust validation losses

$$\mathcal{L}_{\text{robust,valid}}(\theta) = \max_{q_\psi \in \mathcal{Q}} \underbrace{\frac{1}{|D_{\text{valid}}|} \sum_{x,y \in D_{\text{valid}}} \frac{q_\psi(x,y)}{q_{\psi_0}(x,y)} \ell(x,y;\theta)}_{:=\mathcal{L}_{\text{valid}}(\theta,\psi)}. \tag{5.10}$$

However, finding the inner supremum for each of the T evaluation checkpoints $\theta_1 \dots \theta_T$ is expensive as it requires solving T independent optimization problems. Instead, we leverage the existence of adversaries ψ_t associated with each model θ_t, as well as the initial adversary ψ_0 and take the maximum over the $T+1$ adversaries $\{\psi_0, \dots, \psi_T\}$. Since our relaxation of the P-DRO objective loosens the KL constraint, we need weed out adversaries which might violate it. Specifically, we estimate the $\text{KL}(q_\psi \| p) = \mathbb{E}_p\, q_\psi/p \log q_\psi/p$ on the validation set, using q_ψ/q_{ψ_0} as a stand-in for q_ψ/p, and reject all adversaries for which the result is greater than a threshold, which we set to $\log 10$ based on preliminary experiments detailed in Appendix 5.6.1.[5] We refer to this stopping

[5]To simplify notation, this additional constraint is implicit in the rest of this section.

criterion as **Minmax**.

Computing the full min-max necessitates keeping track of T models and $T+1$ adversaries, which is ponderous when the model is large. As a solution, we propose an approximation, **Greedy-Minmax**, in which we only keep one best model θ^*. At each evaluation step T, we compare θ_T to θ^*, and update θ^* to whichever achieves lower robust validation loss over the $T+1$ adversaries $\psi_0, \ldots, \psi_T$.

By keeping track of only one additional model, and using the weights $\frac{q_{\psi_t}(x_i,y_i)}{q_{\psi_0}(x_i,y_i)}$ of individual examples in D_{valid} as sufficient statistics for computing the loss against each adversary, Greedy-Minmax can be achieved with space complexity $2d_{\text{model}} + T|D_{\text{valid}}|$, which is much more efficient than the $T(d_{\text{model}} + d_{\text{adv}})$ of Minmax.

5.3.4 Hyper-parameter Selection

Our proposed P-DRO method relies on 3 different hyper-parameters (in addition to the model's hyper-parameters): the adversary learning rate λ, the temperature τ and the size of the re-normalizing window k. As a consequence, we need a reliable criterion for deciding which of two configurations is better. This model comparison bears many similarities with the stopping problem described above. Therefore, we resort to a similar solution: given two models θ_1, θ_2 trained with P-DRO, and their respective adversaries $\{\psi_0^1, \ldots, \psi_T^1\}$, $\{\psi_0^2, \ldots, \psi_T^2\}$ (for instance, the adversaries associated with θ_1 and θ_2 at periodic checkpoints during training), we select the best model following

$$\theta^* = \arg\min_{\theta \in \{\theta_1, \theta_2\}} \max_{\psi \in \{\psi_0^1, \ldots, \psi_T^1, \psi_0^2, \ldots, \psi_T^2\}} \mathcal{L}_{\text{valid}}(\theta, \psi). \tag{5.11}$$

5.4 Experimental Analysis of P-DRO

Before moving on to a real world scenario in Section 5.5, we first demonstrate that P-DRO is able to learn robust models in a synthetic NLP task, and perform ablation studies to examine the importance of the various modifications described in Section 5.3.

5.4.1 Experimental Setting

For analysis purposes, we design a simple NLP task amenable to DRO. We specifically choose NLP as a domain due to the striking success of language models as generative models of textual data (Sundermeyer et al., 2012; Radford et al., 2018), which can be used to model the uncertainty

set. We base our task off of the binary version of the Stanford Sentiment Treebank dataset (SST-2; Socher et al. (2013)), which we modify to introduce spurious correlation. Specifically, we introduce a distractor token to some sentences. The distractor we use consists of prepending "so , " to the sentence ("i hated this movie" $\longrightarrow$ "so , I hated this movie"), which doesn't change the underlying sentiment. The resulting samples can be categorized in 4 "groups" depending on their label (positive or negative) and the presence or absence of the distractor. In particular, we add this distractor to 95% of the negative reviews and 5% of the positive reviews in the training and validation set, so that the presence of the distractor strongly correlates with negative sentiment (a similar construction is proposed in (Utama et al., 2020)). In the test data, we modify 50% of all sentences for each class equitably to ensure that there is enough data in each group, but we report "average" test accuracy by re-weighting the group accuracies to mimick the training distribution. We call this modified task **BiasedSST**.

5.4.2 Model Settings

In all experiments, we split the text into sub-word tokens using the tokenizer described in (Devlin et al., 2018). For the classifier, we train a simple one layer BiLSTM model with embedding/hidden dimension 300. During training, we sample minibatches that contain at most 64 sentences or 2500 tokens, whichever is greater, in order to prevent GPU memory overflow in case of long sentences. We train all models with Adam (Kingma & Ba, 2014) with an initial learning rate of 2×10^{-5}, which we decay linearly at each step until the end of training. We validate the models every epoch. For BERT, we start from the `bert-base-uncased` checkpoint.

5.4.3 Adversary Settings

In all experiments, we use an auto-regressive transformer model based on the successful GPT-2 language model (Radford et al., 2019) architecture but with 6 layers, a dimension of 512 and 8 attention heads (we experiment with a smaller, LSTM based adversary in Section 5.6.2). In order to model the input output pair (x, y), we pre-pend a special label-specific token to sentences before running them through the language model. We train the adversary with vanilla stochastic gradient descent, which we found more stable in experiments.

To initialize the adversary (to obtain ψ_0), we first pre-train the model on a generic, relatively large language modeling dataset, WikiText-103 (Merity et al., 2017). We also use a batch size of 64 samples or 2500 tokens, and train with Adam for 10 epochs, with a fixed learning rate of 3×10^{-4}.

Then, we fine-tune this model on each dataset, this time minimizing the negative log-likelihood of the (x, y) pair, using the same hyper-parameters but a smaller learning rate (10^{-5}). We find that, due to the small to medium size of the datasets under consideration, this LM pretraining step helped achieve lower error on the generative modeling task.

5.4.4 Baselines

In addition to ERM, we also compare three other approaches. First, to appreciate how well the model could perform if the groups were known at training time, we train with Group-DRO on the oracle groups using an exponentiated-gradients based online algorithm (**Oracle DRO**; Sagawa et al. (2020)). Second, we implement **Topic DRO** (Oren et al., 2019), a method for DRO on NLP where the uncertainty set is determined by mixtures of a topic model. Finally, we compare to non-parametric DRO with a KL constrained uncertainty set (Hu & Hong, 2013; Hu et al., 2018), which we adapt to fit our online mini-batch training setting (**NonParam**). Below, we outline experimental details for the latter two.

Topic DRO

To train the topic model for Topic DRO, we first pre-process the text by removing all punctuation, urls and user mentions (for twitter data). Importantly, we remove stop-words for our toxicity experiments but *not* for our BiasedSST experiment. This is because the distractor token we use ("so") belongs to most English stop words lists, and removing it would completely prevent the topic model from picking up on the groups of interest. We then estimate the parameters of the model with Gensim[6] and use similar settings as Oren et al. (2019) ($\alpha = 0.1$, $\beta = 1.0$), setting the number of topics to 10.

For both Oracle-DRO and Topic DRO, we use the algorithm proposed in Sagawa et al. (2020) to estimate the worst-case group (either oracle group or topic in Topic DRO) online during training. We perform grid-search over $\{1, 0.1, 0.01\}$ to find the best learning rate for the group weights update. For Oracle DRO, the best model is simply selected by robust validation accuracy. For Topic DRO, unless specified otherwise, we select the model with the lowest worst-case error over all topics.

[6] `https://radimrehurek.com/gensim/`

NonParam

In the KL-constrained non-parametric setting, the min-max problem reads

$$\min_{\theta} \max_{\substack{q \text{ s.t.} \\ \mathrm{KL}(q_\psi \| p) \leq \kappa}} \mathbb{E}_{(x,y)\sim q} \ell(x, y, \theta). \tag{5.12}$$

Here, κ is the desired radius of the KL ball, and is treated as a hyper-parameter. The solution of the inner maximum has an analytical solution of the form $q^*_\theta = \frac{a}{Z_{\theta,\tau^*}} p(x,y) e^{\frac{\ell(x,y;\theta)}{\tau^*}}$ (see Hu & Hong (2013); Hu et al. (2018) for details) with $Z_{\theta,\tau^*} = \mathbb{E}_p e^{\frac{\ell(x,y;\theta)}{\tau^*}}$ and τ^* such that

$$\mathrm{KL}(q^* \| p) = \mathbb{E}_p \frac{e^{\frac{\ell(x,y;\theta)}{\tau^*}}}{Z_{\theta,\tau^*}} \left(\frac{\ell(x,y;\theta)}{\tau^*} - \log Z_{\theta,\tau^*} \right) = \kappa.$$

Note that both computing Z_{θ,τ^*} and $\mathrm{KL}(q^* \| p)$ require taking expectations over p. In our setting, where $\ell(x, y; \theta)$ is the output of a large neural network, we cannot afford to take this expectation over the entire training data at each step. Instead, we fall back to taking the average over each mini-batch. We find τ^* with binary search in $\log_{10}$ space within the $[10^{-10}, 10^{10}]$ interval and clip to the lowest or highest value should the result lie outside the search interval.

In all experiments, we try 4 different values for κ: 0.01, 0.1, 1 and 10. Unless indicated otherwise, we perform early stopping and hyper-parameter selection using our Minmax criterion using the non-parametric weights as adversaries on the validation data.

5.4.5 P-DRO can Learn Robust Models

Table 5.1: Average and robust accuracies on BiasedSST. Underlining indicates statistically significant difference compared to ERM ($p < 0.05$)

	Robust	Average
ERM	2.15 ± 0.97	95.09 ± 0.16
Topic DRO	5.18 ± 1.46	95.00 ± 0.10
NonParam	28.11 ± 2.16	92.45 ± 1.55
P-DRO	34.98 ± 9.39	84.21 ± 2.11
Oracle DRO	67.71 ± 3.03	77.91 ± 4.49

We train 7 models with P-DRO on BiasedSST using different hyper-parameters for the adversary. We start from configuration $\lambda = 10^{-4}$, $\tau = 0.01$, $k = 5$, and for each hyper-parameter we run a configuration with a smaller and a higher value, keeping all other hyper-parameters the same. We train for 50 epochs and select the best model using the strategies described in Section 5.3.

We report the worst-case ("robust") accuracy over all groups on the test set, as well the average accuracy in Table 5.1 (we report the mean and standard deviation over 5 runs). We find that both Topic DRO, NonParam and P-DRO are more robust than ERM, but the latter outperforms the former two close to 30 and 7 points respectively, achieving 52% of Oracle DRO's robust accuracy, while not leveraging any information on the oracle groups.

5.4.6 Optimal Stopping and Hyper-parameter Selection Ablation

To understand the importance of the optimal stopping and hyper-parameter selection strategy described in Section 5.3.3, we perform an ablation on the BiasedSST dataset comparing 4 strategies:

- **Average**: models are selected based on their average zero-one loss (*i.e.* error rate) on the unmodified validation set. This is the baseline stopping criterion.
- **Minmax**: selection based on the adversaries (as described in Section 5.3.3), with and without the **KL constraint**, as well as its variant **Greedy-Minmax** for stopping.
- **Oracle**: in this setting the groups are known (in the validation set), and models are selected based on their error rate on the worst performing group. This is the optimal criterion for the group-DRO setting we are considering.

To compare stopping criterions experiments, we only consider one set of hyper-parameters: $\lambda = 10^{-4}$, $k = 5$ and $\tau = 0.01$. From the robust validation accuracies reported in Table 5.2a, we first observe that Average stopping results in a robust accuracy of 0, highlighting the necessity for a suitable stopping criterion. We find that Minmax, especially with a KL constraint, is a much better strategy, recovering $\approx 60\%$ of the performance achievable with Oracle stopping. Notably, the Greedy-Minmax variant which we use in practice reaches very close results (< 1 point difference) despite its requiring to keep track of only 2 out of the 50 model checkpoints at any time.

To understand the effectiveness of the Minmax strategy for selecting hyper-parameters. We take the models trained in Section 5.4.5, but select the best hyper-parameters using the different strategies described above. Results, shown in Table 5.2b, confirm that Minmax (with the KL constraint) is a better choice than Average for selecting hyper-parameters, even though the improvement is not as striking as for stopping.

Table 5.2: Effect of different optimal stopping and hyper-parameter selection strategies on robust validation accuracy.

(a) Optimal stopping

Criterion	Robust accuracy
Average	0.00 ± 0.00
Minmax	25.22 ± 13.01
+KL constraint	31.30 ± 10.07
+Greedy-Minmax	**32.17** ± ± 11.20
Oracle	50.95 ± 5.01

(b) Hyper-parameter selection

Criterion	Robust accuracy
Average	31.03 ± 12.16
Minmax	28.62 ± 12.37
+KL constraint	**35.65** ± 11.47
Oracle	38.26 ± 13.01

5.4.7 Importance of $\mathcal{L}_{\text{adv}}$

Finally, we investigate the importance of modifying the adversary's objective as described in Section 5.3.2. For this experiment, we devise a simpler toy task on which directly training the constrained DRO objective is possible. Specifically, we consider the two-dimensional binary classification problem pictured in Figure 5.2. The training data consists of 10,000 points partitioned in two normally distributed "domains" with a 1:50 sampling ratio and different classification boundaries. We train a logistic regression model, which cannot perfectly fit the training data and must trade-off between accuracy on each domain. For the sake of simplicity, we only model the input variables x[7] as isotropic normal distributions with fixed variance: the adversaries' parameter $\psi \in \mathbb{R}^2$ represents the location of the Gaussian (we fix the variance to the empirical variance of the data).

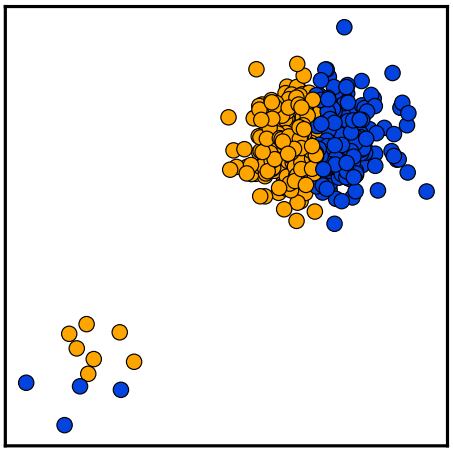


Figure 5.2: A toy classification task.

We compare 3 different versions of P-DRO: first, naive simultaneous gradient descent on the zero-sum game, without any constraint on the adversary (**bare P-DRO**), then the same, but with an approximation of the explicit KL constraint between q_ψ and q_{ψ_0} (**+KL constraint**. Even in this simplest setting, the exact KL between q_ψ (a gaussian) and p (a mixture of gaussians) does not

[7]In other words, we set $q_\psi(x, y) = p(y \mid x)q_\psi(x)$, where $p(y \mid x)$, is the true conditional which will be canceled out in the ratio $\frac{q_\psi(x,y)}{q_{\psi_0}(x,y)}$.

have an analytical expression (Hershey & Olsen, 2007). Instead, we fall back on enforcing the KL constraint between q_ψ and q_{ψ_0}, both isotropic gaussians with the same standard deviation. Let μ and $\mu_0 \in \mathbb{R}^2$ denote their respective mean, and $\sigma > 0$ their standard deviation. In this context, their KL divergence reduces to:

$$\mathrm{KL}(q_\psi \| q_{\psi_0}) = \mathrm{KL}(q_{\psi_0} \| q_\psi) = \frac{1}{2\sigma^2} \|\mu - \mu_0\|^2$$

In other words, the KL divergence is equivalent to the euclidean distance between the distributions' means. We use this fact to project ψ (in the KL sense) onto $B_\kappa = \{\hat{\psi} \mid \mathrm{KL}(q_{\hat{\psi}} \| q_{\psi_0}) < \kappa\}$:

$$\begin{aligned} \mathrm{proj}_{B_\kappa}(\psi) &\coloneqq \arg\min_{\hat{\psi} \in B_\kappa} \mathrm{KL}(q_\psi \| q_{\hat{\psi}}) \\ &= \psi_0 + \frac{\sqrt{2\kappa}\sigma}{\|\psi - \psi_0\|}(\psi - \psi_0) \end{aligned}$$

Finally we report results using our relaxation and the KL reversal described in Section 5.3.2 (+$\mathcal{L}_{\text{adv}}$). For each setting, we report the average and robust accuracy with mean and standard deviation over 10 runs. For the KL constraint and the relaxation, we report the best results among 4 values of the KL bound κ and the temperature τ respectively.

Table 5.3: Ablation of P-DRO to train the linear model on the toy task. We report accuracy on both domains, as well as robust accuracy.

	Average	Robust
ERM	**84.66** ± 0.10	49.75 ± 0.05
bare P-DRO	49.97 ± 0.10	49.85 ± 0.10
+kl constraint	76.63 ± 9.93	58.41 ± 9.25
+$\mathcal{L}_{\text{adv}}$	76.97 ± 0.43	**64.32** ± 1.31

In Table 5.3, we observe that bare P-DRO is too unstable and systematically diverges. The addition of a KL constraint mitigates this behaviour, but the zero-sum objective is still unstable, as evidenced by the high standard deviations. Finally, we find that the addition of $\mathcal{L}_{\text{rev}}$ stabilizes the training process greatly, leading to consistently high robust accuracy.

5.5 P-DRO in Practice: Case Study of Toxicity Detection

In this section, we demonstrate the effectiveness of P-DRO in the more realistic setting of toxicity detection, the task of recognizing various forms of toxic language (eg. hate speech or offensive language). Identifying online abuse on the internet is a crucial challenge, and has garnered much

interest in the NLP community (Schmidt & Wiegand, 2017; Fortuna & Nunes, 2018). However, recent work (Sap et al., 2019) has shown that there is strong correlation between toxic labels and the presence of certain markers of dialects of English spoken by minority groups. This correlation is in turn amplified by hate speech classifiers trained on such data, leading to biased prediction.

Our results on BiasedSST suggest that P-DRO can provide one solution to preventing models from absorbing spurious correlations present in their training data, even in the absence of protected attributes (such as language variety here).

5.5.1 Experimental Setting

Following Sap et al. (2019) and Xia et al. (2020), we perform experiments on two datasets: **DWMW17** (Davidson et al., 2017), a corpus of 25K tweets classified in three categories: *hate speech* (6%), *offensive* (76%) and *neither* (18%), and **FDCL18** (Founta et al., 2018), a 100k sized dataset, also collected from Twitter and annotated with an additional *spam* label, with the following breakdown by categories: *hateful* (5%), *abusive* (27%), *normal* (54%) and *spam* (14%).

The released version of these datasets does not contain information on the dialect of each user. In order to be able to evaluate our models, and to train an Oracle DRO baseline, we follow Sap et al. (2019) and use annotations provided by the dialect classifier described in Blodgett et al. (2016) to label each example as one of four English varieties: White-aligned, African American, Hispanic, and Other. Note that, as these are automatically obtained labels, the groups may not exactly correspond to the actual racial sociolects, however Sap et al. (2019) does report that they correlate highly with self-reported race, and they serve as a useful proxy in the absence of manual annotation.

We formulate the group-DRO problem by separating each dataset into independent groups identified by both language variety and label, for a total of 12 and 16 groups for DWMW17 and FDCL18 respectively. Some of these groups are severely under-represented in the test set. In order to make our robust accuracy results reliable yet still representative of the under-represented groups, we combine groups that contain less than 100 samples into a single group to compute robust test accuracies.

On DWMW17, we train the same BiLSTM model as described in Section 5.4.6. To illustrate the applicability of P-DRO to other model architectures, we pick BERT (Devlin et al., 2018), a large scale pre-trained model as a classifier on FDCL18. In both cases, we adopt the Transformer architecture described in Section 5.4.6 as the adversary. We train the adversary with a temperature of $\tau = 0.01$ and a normalizing window $k = 10$. To demonstrate the efficacy of automatic hyper-

Table 5.4: Robust test accuracy on the DWMW17 and FDCL18 toxicity detection tasks.

(a) No group information.

	DWMW17		FDCL18	
	Robust	Average	Robust	Average
ERM	53.19 ± 1.70	69.44 ± 0.53	19.57 ± 7.00	**81.56** ± 0.26
Topic DRO	45.26 ± 3.47	61.68 ± 5.02	16.48 ± 5.46	80.49 ± 0.49
NonParam	54.13 ± 1.14	**70.54** ± 0.64	17.54 ± 6.41	81.20 ± 0.11
P-DRO	**69.06** ± 1.70	69.69 ± 2.50	**30.25** ± 10.13	79.91 ± 1.41
Oracle DRO	74.50 ± 1.74	65.79 ± 0.76	55.23 ± 3.97	72.43 ± 2.61

(b) With group information on the validation set.

	Robust	Average	Robust	Average
ERM	53.15 ± 0.87	69.64 ± 1.01	34.07 ± 3.20	78.78 ± 0.38
Topic DRO	52.02 ± 1.26	**69.11** ± 0.49	34.82 ± 3.73	**79.59** ± 0.85
NonParam	49.41 ± 5.60	58.53 ± 5.71	43.13 ± 6.97	69.51 ± 3.07
P-DRO	**63.05** ± 4.25	63.07 ± 3.92	**47.61** ± 4.53	74.82 ± 1.90
Oracle DRO	74.50 ± 1.74	65.79 ± 0.76	55.23 ± 3.97	72.43 ± 2.61

parameter selection in the P-DRO setting, we delegate the choice of the adversary's learning rate λ to grid-search, training 3 models with $\lambda \in \{10^{-5}, 10^{-4}, 10^{-3}\}$ and selecting the best using the Minmax criterion described in Section 5.3.4. We also report numbers for Oracle DRO and Topic DRO. Results are averaged over 5 runs, each with a different random seed.

5.5.2 Can P-DRO Produce more Robust Models?

Table 5.4a reports the robust test accuracies of all models on both tasks. Importantly, except for Oracle DRO, none of the methods compared here necessitate any knowledge of the groups, neither in the training nor validation data. We observe that in both settings P-DRO is able to achieve higher robust accuracy than ERM, Topic DRO and NonParam.

This suggests P-DRO as a useful option in case no group information whatsoever is available. However, in practice, it may be feasible to annotate at least a small amount of data with group information. To emulate this scenario, we perform the same experiment, but assume that group annotations are available on the validation data, which we use to determine optimal stopping and hyper-parameters. Results for this setting are reported in Table 5.4b. We find that, while the use

of robust validation accuracy yields more robust models even for ERM (especially on FDCL18), P-DRO is still the best alternative that doesn't require group annotation on the training data.

5.6 Additional Experiments

5.6.1 Minmax Validation KL Threshold

The Monte-Carlo estimate of $\mathrm{KL}(q_\psi \| p)$ on the validation set is $\frac{1}{|D_{\text{valid}}|} \sum_{x,y \in D_{\text{valid}}} \frac{q_\psi(x,y)}{p(x,y)} \log \frac{q_\psi(x,y)}{p(x,y)}$. Similarly to Section 5.3, we approximate the (unknown) likelihood ratio $\frac{q_\psi(x,y)}{p(x,y)}$ with $\frac{q_\psi(x,y)}{q_{\psi_0}(x,y)}$.

We want to reject all adversaries where this approximated KL is greater than some threshold, κ_{valid}, but how do we choose a good value for κ_{valid}? Consider an adversary which selects a fraction of the validation data of size $\alpha|D_{\text{valid}}|$ for some $\alpha \in (0, 1]$. In such a case, the likelihood ratio is $1/\alpha$ on this subset and 0 everywhere else, and the resulting KL estimate will be $\log \alpha$. In other words, choosing a threshold of κ_{valid} means allowing the adversary to potentially select any subset of size at least $1/e^{\kappa_{\text{valid}}}$ of the original data. Our heuristic choice, $\log 10$, corresponds to allowing subsets of size at least 10% of $|D_{\text{valid}}|$.

Of course, this is only a heuristic because the adversary can reweight the validation set non-uniformly. To assess the effect of κ_{valid} on Greedy-Minmax, we compute the average robust validation error of the selected model across 5 runs for 3 different values of the adversary's learning rate. Results on BiasedSST, depicted in Figure 5.3, show that adversaries with higher learning rate are more sensitive to the choice of threshold, but all values of κ_{valid} between $\log 5$ and $\log 20$ seem to work for these settings.

5.6.2 P-DRO Experiments with an LSTM Adversary

We replicate the experiments BiasedSST experiments in Section 5.4, but this time using a smaller generative model, which is unlikely to generate good samples. Specifically, we use a one layer LSTM model (Hochreiter & Schmidhuber, 1997) with embedding and hidden dimension 256. We only perform grid-search over $\lambda \in [10^{-5}, 10^{-4}, 10^{-3}]$ and select the best with Minmax.

Once pre-trained on the BiasedSST dataset, this model achieves a perplexity of 227.0, more than 4 times worse than the transformer model we use in other experiments (49.8). However, as evidenced by its robust accuracy displayed in Table 5.5, P-DRO is still able to learn a robust model. We take this as evidence that the re-weighting introduced in Section 5.3 helps stabilize training even when q_ψ is not a perfect model of the data.

Figure 5.3: Evolution of the robust validation accuracy of the model selected by Greedy-Minmax as a function of the KL threshold κ_{valid}

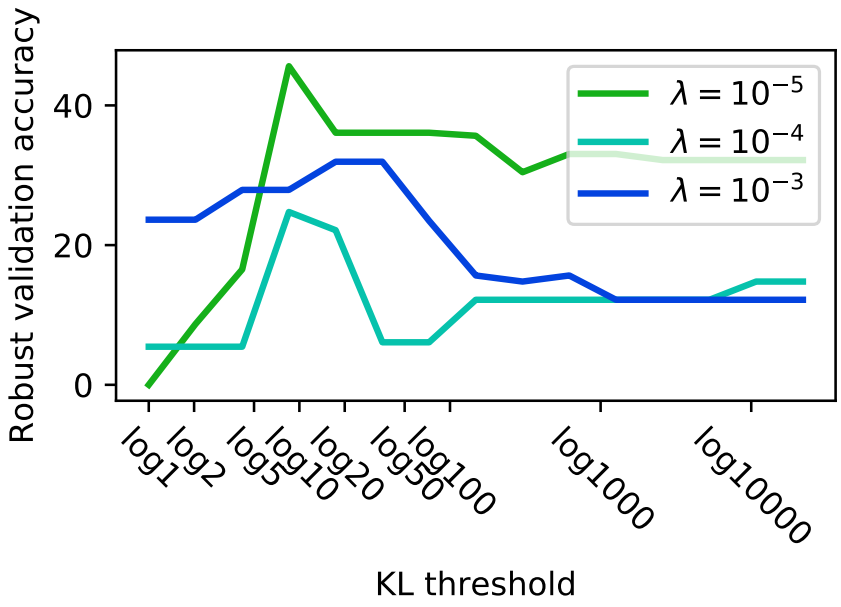


Table 5.5: Average and robust accuracies on BiasedSST when P-DRO is trained with an LSTM adversary. Underlining indicates statistically significant difference compared to ERM ($p < 0.05$)

	Robust	Average
ERM	2.15 $\pm$ 0.97	95.09 $\pm$ 0.16
Topic DRO	5.18 $\pm$ 1.46	95.00 $\pm$ 0.10
NonParam	28.11 $\pm$ 2.16	92.45 $\pm$ 1.55
P-DRO	43.68 $\pm$ 4.93	86.58 $\pm$ 1.77
Oracle DRO	67.71 $\pm$ 3.03	77.91 $\pm$ 4.49

Table 5.6: Effect of hyper-parameters on robust validation accuracy on BiasedSST

	Robust accuracy	
	Minmax stopping	Oracle stopping
$\lambda = 10^{-5}$	28.62 $\pm$ 12.37	45.10 $\pm$ 4.50
$\lambda = 10^{-4}$	44.74 $\pm$ 3.24	50.43 $\pm$ 5.05
$\lambda = 10^{-3}$	25.57 $\pm$ 10.33	38.70 $\pm$ 2.97
$\tau = 0.1$	39.72 $\pm$ 5.55	50.00 $\pm$ 4.98
$\tau = 0.01$	44.74 $\pm$ 3.24	50.43 $\pm$ 5.05
$\tau = 0.001$	44.74 $\pm$ 3.24	50.87 $\pm$ 5.09
$k = 1$	41.98 $\pm$ 4.48	49.60 $\pm$ 5.39
$k = 5$	44.74 $\pm$ 3.24	50.43 $\pm$ 5.05
$k = 10$	32.17 $\pm$ 11.20	50.95 $\pm$ 5.01

5.6.3 Influence of hyper-parameters on P-DRO

We study the influence of the 3 hyper-parameters τ (temperature), k (size of the renormalization window) and λ (learning rate of the adversary) on the performance of P-DRO. All experiments are run on the BiasedSST dataset, and the analysis proceeds as follows: we start from configuration $\tau = 0.01$, $k = 5$ and $\lambda = 10^{-4}$ and vary each of the hyper-parameters independently. We report two numbers for each configuration: robust accuracy of the best model (1) using Greedy-Minmax stopping and (2) using Oracle stopping. The latter is useful to disentangle the effect of the stopping criterion.

As seen in the results shown in Table 5.6, we find that τ has the least effect on robust accuracies. While the renormalization window parameter k has some effect on optimal stopping, the best robust accuracy achieved by the model (with oracle stopping) varies little. We observe the adversary's learning rate λ to be the most sensitive hyper-parameter, which is why we restrict our grid-search to λ in Section 5.5.

5.7 Conclusion

In this chapter, we have shown that there is promise in using parametric families of neural generative models for defining the uncertainty set in distributionally robust optimization. While we only perform experiments on NLP tasks, this approach can, in theory, be applied in any modality.

However, the reliance of P-DRO on good quality generative models limits its adoption in applications where good quality generative models are unavailable, or when such model cannot produce exact densities efficiently. In the following chapter, we will address these issues by parameterizing the likelihood ratio q_ψ/p directly, an alternative formulation which poses different implementation challenges.

Chapter 6

Distributionally Robust Models with Parametric Likelihood Ratios

In chapter 5 we proposed P-DRO, a promising approach where the uncertainty set is defined by a parametric family of generative models, which allows for more flexibility in defining the uncertainty set. Despite P-DRO's demonstrated ability to train models that are robust to distributional shift, it suffers from several drawbacks: first, it necessitates training an auxiliary generative model of the data. This can be difficult for several reasons. First, this limits the applicability of the method to domains with generative models that allow for exact probability computations. Moreover, even when such generative models are available, they are often more computationally demanding than their discriminative counterparts. In language models for instance, probabilities for sequences of text are obtained by iteratively producing conditional probabilities over all tokens in the vocabulary. This additional step results in considerable computational overhead compared to discriminative models. Second, it is challenging to use in practice due to its reliance on a number of hyperparameters and approximations to the objective function.

In this chapter, we propose a new approach for DRO, called R-PDRO, based on a key modification of the P-DRO algorithm: instead of modeling the worst-case distributions directly, we parametrize the likelihood ratio between the training distribution and the worst-case distribution. This removes the dependency on an unwieldy generative model, making the method useful for more applications. While likelihood-ratio formulations of DRO have been tried in prior work (Sagawa et al., 2020), we show that they are particularly effective for parametric, neural-network based adversaries. Our approach relies on two simple ideas: a mini-batch level normalization strategy to enforce likelihood ratio constraints and a KL divergence penalty to limit the scope of

the uncertainty set. R-PDRO consistently achieves equal or better robust sub-population accuracy compared to P-DRO and other baselines on a variety of standard benchmarks in image and text classification. In addition, we find it is both faster than P-DRO and depends on fewer hyperparameters. Additional ablation and analysis experiments demonstrate that our minibatch normalization strategy is necessary for high performance, and that the parametric adversaries enabled by R-PDRO are substantially more resistant to label noise compared to traditional non-parametric approaches to DRO.

6.1 Parametric Likelihood Ratio

6.1.1 DRO as a Likelihood Ratio Optimization Problem

In the situation that all distributions in $\mathcal{Q}$ are absolutely continuous with respect to p (i.e. for all $q \in \mathcal{Q}$, $q(x, y) > 0$ only if $p(x, y) > 0$) the inner maximum in Equation 2.3 can be rewritten purely as a function of the likelihood ratio $\frac{q}{p}$

$$\mathbb{E}_{(x,y)\sim q}\, \ell_\theta(x, y) = \mathbb{E}_{(x,y)\sim p}\, \frac{q(x, y)}{p(x, y)} \ell_\theta(x, y). \tag{6.1}$$

Such absolute continuity assumptions are standard in f-divergence and group DRO methods, which both rely upon re-weighting the training distributions. In fact, the KL divergence constraint in P-DRO presupposes absolute continuity.

This suggests that the inner maximum can be re-written as an optimization problem on functions $r : \mathcal{X} \times \mathcal{Y} \longrightarrow \mathbb{R}_+$ within the uncertainty set $\mathcal{R} \in \{r \mid pr \in \mathcal{Q}\}$

$$\min_\theta \max_{r\in\mathcal{R}} \mathbb{E}_{(x,y)\sim p}\, r(x, y)\ell_\theta(x, y). \tag{6.2}$$

This reparametrization of the problem will allow us to replace the parametric family of generative models with a parametric family over probability ratios.

6.1.2 Ratio-based P-DRO

The likelihood-ratio formulation described above is appealing for P-DRO because it enables the use of discriminative style neural architectures for parameterizing the ratio r, which opens up many more options for defining the parametric uncertainty set. Specifically, we can set the adversary to

be any parametric function $r_\psi : \mathcal{X} \times Y \longrightarrow \mathbb{R}^+$ verifying $\mathbb{E}_{x,y\sim p}\, r_\psi(x,y) = 1$. The key insight that we use to realize our proposed method is that we do not need to limit the choice of adversaries to those that implicitly satisfy this normalization condition (*i.e.* generative models). Instead, we can pick any adversary and treat normalization as an additional constraint (the "normalization constraint").

Note that in this case, the KL constraint takes the simple form $\mathrm{KL}(pr_\psi \| p) = \mathbb{E}_{pr_\psi} \log \frac{pr_\psi}{p} = \mathbb{E}_p\, r_\psi \log r_\psi$. The final min-max problem, which we dub ratio-based P-DRO (R-PDRO), is as follows:

$$\min_{\theta} \max_{\substack{\psi \\ \mathbb{E}_p\, r_\psi \log r_\psi \leq \kappa \\ \mathbb{E}_p r_\psi = 1}} \underbrace{\mathbb{E}_{(x,y)\sim p} r_\psi(x,y)\ell_\theta(x,y)}_{\mathcal{L}_{\text{R-PDRO}}}. \tag{6.3}$$

As in P-DRO, we can look for equilibria of this differentiable min-max game by performing simultaneous gradient updates (Singh et al., 2000) to θ and ψ in directions $-\nabla_\theta \mathcal{L}_{\text{R-PRDO}}$ and $+\nabla_\psi \mathcal{L}_{\text{R-PRDO}}$ respectively. Although finding global equilibria is not guaranteed in high dimensional non-convex settings (Balduzzi et al., 2018), empirical evidence from Chapter 5 suggests that models trained in this manner still reach useful solutions.

In experiments, we adopt an exponential parameterization $r_\psi(x,y) = e^{f_\psi(x,y)}$ where f_ψ is the output of any parametric model with values in $\mathbb{R}$. Similarly to P-DRO, we do not explicitly enforce the KL constraint (due to the difficulty of projecting onto the KL ball), and instead relax it in the form of a KL term $\tau\, \mathbb{E}_p\, r_\psi \log r_\psi$ added to the loss function. The regularization strength τ is treated as a hyper-parameter.

6.1.3 Enforcing the normalization constraint

In addition to the KL constraint, R-PDRO necessitates that r_ψ satisfies a normalization constraint $\mathbb{E}_p\, r_\psi = 1$ to ensure that pr_ψ is a proper probability distribution over $\mathcal{D}_{\text{train}}$. If that were not the case, the adversary r_ψ could artificially increase the weighted expectation $\mathbb{E}_p\, r_\psi \ell_\theta$ by assigning a total weight greater than 1 to the entire dataset.

Existing methods for ratio based DRO such as Sagawa et al. (2020) achieve this by either projecting r_ψ onto the simplex $\{r \mid \mathbb{E}_p\, r = 1\}$ after each gradient update on ψ, or by directly parametrizing the ratio as $r_\psi(x,y) = e^{f_\psi(x,y)} / \mathbb{E}_p\, e^{f_\psi}$. Unfortunately, these solutions are computationally infeasible in practical scenarios with large datasets. Indeed, they necessitate computing the entire expectation over p, which can be costly when each $f_\psi(x,y)$ is the output of a neural model.

We propose two simple, yet effective solutions for addressing this issue in the context of mini-

batch training where we can only compute f_ψ for small number of samples $(x_1, y_1), \ldots, (x_n, y_n)$ at each step of training.

Self-normalization is inspired by the idea of "self-normalization" developed for globally normalized structured prediction models (Andreas et al., 2015; Goyal et al., 2019). It consists in adding a relaxation of the normalization constraint to the objective. Specifically, following Goyal et al. (2019) we add a squared penalty on the log normalizer at the minibatch level. Ignoring the KL penalty, this regularized objective takes the following form:

$$\mathcal{L}_{\text{self-norm}}(\theta, \psi) = \frac{1}{n}\sum_{i=1}^{n} r_\psi(x_i, y_i)\ell_\theta(x_i, y_i) - \beta \log\left(\frac{1}{n}\sum_{i=1}^{n} r_\psi(x_i, y_i)\right)^2. \tag{6.4}$$

The hyper-parameter β controls the regularization strength. Intuitively, this penalizes adversary that assign too much (or too little) total weight to the minibatch. However, the choice of an optimal β adds an additional degree of freedom to R-PDRO, which suggests our second option as a simpler alternative.

Batch-level normalization consists of using the normalized ratio at the minibatch level by setting

$$\tilde{r}_\psi(x_i, y_i) = \frac{e^{f_\psi(x_i, y_i)}}{\sum_{j=1}^{n} e^{f_\psi(x_j, y_j)}} \tag{6.5}$$

for each sample (x_i, y_i) in the minibatch. An obvious downside of this approach is that the weight of each sample now depends on the minibatch it was sampled from. This can be problematic for small batch sizes: as an extreme example, for a batch size of 1, this normalization scheme assigns the same weight of 1 to every sample, making the objective equivalent to ERM.

However, minibatch approximations have proven effective for other forms of DRO (Hu et al., 2018; Levy et al., 2020) and there is some evidence that they can yield accurate estimates for higher batch sizes (Cortes et al., 2010). In practice we find that this approach yields good results for commonly used batch sizes, is generally more stable than the self-normalization penalty, and does not introduce any additional hyper-parameters. In most of our experiments, we adopt this approach unless specified otherwise. In that case, the R-PDRO objective on a minibatch becomes

$$\mathcal{L}_{\text{batch-level norm}}(\theta, \psi) = \underbrace{\sum_{i=1}^{n} \tilde{r}_\psi(x_i, y_i)\ell_\theta(x_i, y_i)}_{\text{expected loss under } pr_\psi} - \tau \underbrace{\sum_{i=1}^{n} \tilde{r}_\psi(x_i, y_i) \log \tilde{r}_\psi(x_i, y_i)}_{\text{KL penalty}}. \tag{6.6}$$

The KL term serves to penalize ratios that deviate too much from 1.[1] The only hyper-parameter that needs to be tuned is the KL regularization strength τ.

6.2 Experiments

6.2.1 Datasets

We perform experiments on four datasets: two text classification datasets used in chapter 5, Biased SST and FDCL18 and two image classification datasets from Sagawa et al. (2020), Waterbirds and CelebA. Specific details for each dataset follow these previous works, as described below:

Biased SST is based on the SST-2 sentiment classification dataset (Radford et al., 2018), but modified to introduce spurious correlation between a distractor token ("So,") and positive labels in around 95% of the dataset. In this setting models trained with ERM can very easily learn this spurious correlation, which hurts performance on the small sub-population that doesn't suffer from this bias.

FDCL18 A toxicity detection dataset of tweets labeled as *hateful* (5%), *abusive* (27%), *normal* (54%) and *spam* (14%). The group-DRO problem is formulated by partitioning the evaluation data along labels as dialectal annotation obtained automatically with an off-the shelf classifier (Blodgett et al., 2016; Sap et al., 2019). In particular these dialects align closely with self-reported race, and Sap et al. (2019) found that machine learning models trained on such toxicity detection datasets tend to exhibit bias towards certain labels depending on the demographics of the tweets' authors, particularly with minorities. In order to report more reliable accuracy numbers, all groups containing less than 100 samples are aggregated when computing test accuracies.

Waterbirds An image classification dataset where the task is to predict "land bird" or "water bird" with the confounding factor of the background; most water (resp. land) bird pictures have water (resp. land) on the background.

CelebA A popular face recognition dataset originally published by Liu et al. (2015). The group-DRO problem is formulated as a task of predicting the hair color ("Blond" or "Dark") across groups formed by the combination of the label and the (binary) gender of the subject. Due to the

[1]Note that the penalty is subtracted because we are maximizing with respect to ψ

Table 6.1: Robust and average test accuracies on the Biased SST and FDCL18 datasets. Underlined numbers indicates statistically significant difference compared to ERM ($p < 0.05$). Bold numbers indicates the best number in each column (barring Oracle DRO).

	Biased SST		FDCL18	
	Robust	Average	Robust	Average
ERM	2.15 ± 0.97	**95.09** ± 0.16	19.57 ± 7.00	**81.56** ± 0.26
Topic DRO	5.18 ± 1.46	95.00 ± 0.10	16.48 ± 5.46	80.49 ± 0.49
NonParam	28.11 ± 2.16	92.45 ± 1.55	17.54 ± 6.41	81.20 ± 0.11
P-DRO	34.98 ± 9.39	84.21 ± 2.11	30.25 ± 10.13	79.91 ± 1.41
R-PDRO	**50.70** ± 7.33	86.60 ± 2.96	**53.52** ± 1.66	76.62 ± 1.43
Oracle DRO	67.71 ± 3.03	77.91 ± 4.49	55.23 ± 3.97	72.43 ± 2.61

Table 6.2: Robust and average test accuracies on the Waterbirds and CelebA datasets. Underlined numbers indicates statistically significant difference compared to ERM ($p < 0.05$). Bold numbers indicates the best number in each column (barring Oracle DRO).

	Waterbirds		CelebA	
	Robust	Average	Robust	Average
ERM	68.32 ± 2.02	89.23 ± 0.36	40.33 ± 2.29	**95.89** ± 0.05
NonParam	72.21 ± 0.95	**90.54** ± 0.72	43.33 ± 3.58	95.72 ± 0.10
R-PDRO	**73.49** ± 3.01	90.15 ± 0.74	**55.78** ± 9.15	93.10 ± 3.87
Oracle DRO	85.60 ± 0.95	89.12 ± 1.20	89.22 ± 0.90	92.59 ± 0.40

spurious correlation between blond hair/female and dark hair/male, models trained with ERM tend to achieve lower accuracies on underrepresented groups such as "blond-haired male" which totals only 0.85% of the training data.

6.2.2 Experimental Setting

On BiasedSST and FDCL18 we follow the experimental setting of chapter 5 and train a BiLSTM and BERT-base model Devlin et al. (2018) respectively. On the image classification datasets we train Resnet-50 architectures (He et al., 2016) pre-trained on ImageNet (Deng et al., 2009) as in Sagawa et al. (2020).

Since the adversary in R-PDRO can be any discriminative architecture, we opt for the natural solution of using a similar architecture for this model. For instance on Biased SST, we take

$f_\psi(x, y)$ as the raw logit output by a BiLSTM architecture identical to that of the classifier (without the final softmax layer). We do the same for the other datasets, with the exception of FDCL18 where we use a smaller DistillBERT model (Sanh et al., 2019) for efficiency. We use the same learning rate and optimizer for both model and adversary and only vary the KL penalty weight $\tau \in \{0.001, 0.01, 0.1, 1.0\}$. We perform optimal stopping using the Minmax criterion proposed in chapter 5: every epoch T, we determine the best model by explicitly solving a greedy approximation of the min-max game between all previously checkpointed adversaries and models on the validation dataset $\mathcal{D}_{\text{valid}}$.

$$\theta^* = \underset{\theta_{i=1\ldots T}}{\arg\min} \max_{\psi_{j=1\ldots T}} \frac{1}{|D_{\text{valid}}|} \sum_{x,y \in D_{\text{valid}}} \tilde{r}_{\psi_j}(x, y) \ell_{\theta_i}(x, y) \tag{6.7}$$

A similar strategy is applied for hyper-parameter selection. Importantly, we substitute the 0-1 loss for ℓ_θ in Equation 6.7 as we found in preliminary experiments on BiasedSST that it consistently produced better results.

We compare our results to 5 different methods experimented with in chapter 5:

- **ERM**: standard training to minimize the average loss.
- **NonParam**: Non-parametric DRO with a KL constrained uncertainty set (Hu & Hong, 2013; Hu et al., 2018). In this particular case, the inner maximization problem has an analytical solution of the form $q(x, y) \propto e^{\frac{\ell_\theta(x,y)}{\tau^*}}$ where τ^* is chosen so that the KL divergence with the training distribution is smaller than a pre-defined bound κ. As a result, examples with high losses are up-weighted.
- **Topic-DRO**: a variation on Topic CVaR (Oren et al., 2019) using the online algorithm from Sagawa et al. (2020) to minimize the worst case loss on a collection of pseudo domains automatically generated via topic modeling.[2] We use this baseline for the text datasets only (Biased SST and FDCL18).
- **P-DRO**: the parametric DRO approach proposed in chapter 5. For image datasets, preliminary experiments using auto-regressive models for image modeling (Van Oord et al., 2016) proved to be prohibitively slow. Therefore, we only report P-DRO on text datasets as in chapter 5.

[2]This baseline was inaccurately referred to as "Topic CVaR" in chapter 5

- **Oracle DRO**: an online algorithm for minimizing the worst-case loss on the true uncertainty set (Sagawa et al., 2020). Contrary to all other methods, Oracle DRO presupposes that we know the groups of interest at training time. As such, it is not directly comparable and serves to provide an upper bound of the robust accuracy that can be achieved.

Across all experiments, we report results with mean and standard deviation across 5 reruns with different seeds.

Dataset-specific Hyper-parameters

All hyper-parameters listed below are constant across all methods:

Text Datasets The input data is tokenized using the `bert-base-uncased` sub-word tokenizer from Devlin et al. (2018). We train both classifier and adversary with Adam (Kingma & Ba, 2014) using a learning rate of 2×10^{-5}, linearly decay the learning rate to 0 at each step. We train with batches of size 64 (or containing up to 2500 tokens, whichever is lower) for 50 and 20 epochs for BiasedSST and FDCL18 respectively, evaluating model on the validation data every epoch.

Image Datasets On both datasets, images are rescaled to 224×224 pixels and pixel values are normalized to have mean 0 and variance 1 across all 3 color channels on the training data. At training time, we augment the data by randomly cropping or flipping the images horizontally. We train using regular stochastic gradient descent using a constant learning rate of 10^{-3} and a batch size of 32. We train for 75 and 13 epochs on Waterbirds and CelebA respectively (those numbers were chosen to match the number of steps trained to Sagawa et al. (2020) despite the smaller batch size), and validate every 100 (for Waterbirds) and 1000 (for CelebA) training steps.

Method Specific Hyper-parameters

For NonParam we follow the adaptation of Hu et al. (2018) used in chapter 5 and choose the optimal temperature τ^* based on minibatch level estimates of the KL divergence. We treat the KL bound κ as a hyper-parameter. We adapt the Minmax stopping criterion of P-DRO to the non-parametric adversaries as we found it yielded more robust models than those selected with average validation accuracy. We sweep over $\kappa \in \{0.01, 0.1, 1.0, 10.0\}$

For R-PDRO we perform optimal stopping using the Minmax criterion with a KL threshold of $\log 10$ in all experiments, to match the value recommended for P-DRO. Specifically, we estimate

the KL divergence of checkpointed adversaries ψ_i on the validation data as follows:

$$\frac{1}{|\mathcal{D}_{\text{valid}}|} \sum_{x,y \in \mathcal{D}_{\text{valid}}} \hat{r}_\psi(x,y) \log \hat{r}_\psi(x,y) \tag{6.8}$$

and reject adversaries for which this quantity exceeds $\log 10$.

6.2.3 Results

For all methods, we report the worst accuracy over all groups in the test set (the "robust accuracy"). Models that are robust to sub-population shift will have higher robust accuracies. Since there is often a trade-off between being robust to distribution shift and performing well on the training distribution, we also report the standard accuracy on the original test set (the "average accuracy")

As shown in Table 6.1, R-PDRO works particularly well on the two text classification tasks, beating all baselines by a large margin on both BiasedSST and FDCL18. In fact, its robust accuracy almost matches that of Oracle DRO on FDCL18, despite the fact that the former doesn't use an any group information at training time. Compared to P-DRO, we find that results are not only better, but also more consistent as evidenced by the lower standard deviation.

Results are also generally good on image datasets (see Table 6.2). On Waterbirds both the NonParam baseline and R-PDRO perform much better than ERM, with a slight edge for R-PDRO (although the latter exhibits a higher variance across reruns). On CelebA, R-PDRO largely outperforms the baselines.

6.3 Analysis and Ablations

We perform a series of analysis experiments and ablation studies to better (1) identify how the parametric representation of the ratio provides improvements over non-parametric alternatives and (2) understand the importance of the various choices made with regards to the renormalization scheme described in Section 6.1. Throughout this section, experiments are performed on the BiasedSST dataset.

6.3.1 Why are Parametric Adversaries Better? The Case of Label Noise

Our experimental results in Section 6.2 shows that parametric approaches such as P-DRO and R-PDRO consistently outperform their non-parametric counterparts. A possible explanation for this

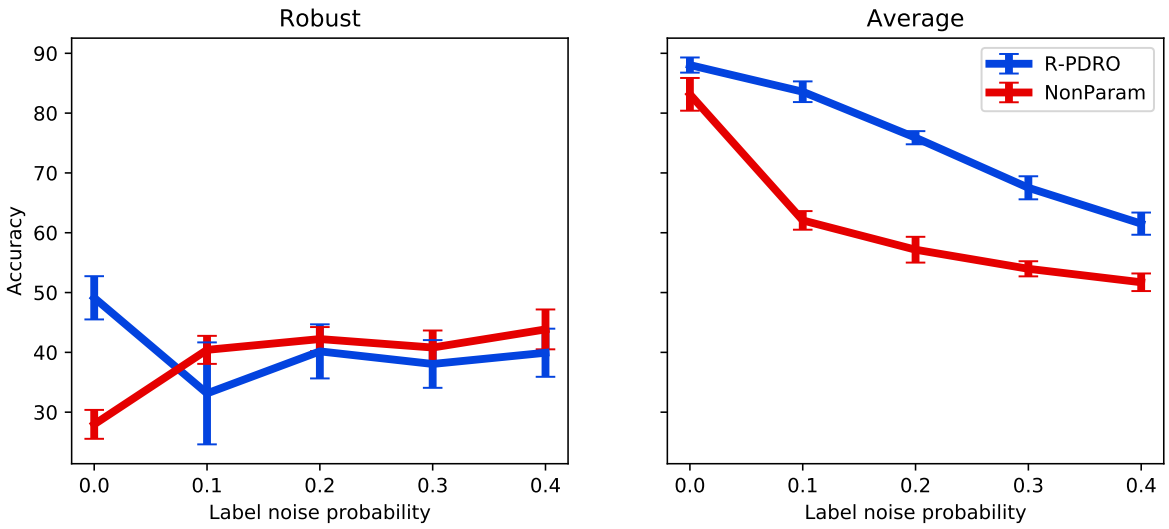


Figure 6.1: Effect of label noise on parametric and non parametric DRO.

phenomenon is that for non-parametric approaches, the optimal weights generally only depends on the loss of the model. This can be problematic because the non-parametric worst-case distribution will indiscriminately up-weight noisy samples that have high loss. On the other hand, we hypothesize that it is more difficult for the parametric adversary to "fit to the noise" and that it tends to focus more on systematic patterns of failures of the model.

To corroborate this hypothesis, we perform experiments by adding increasing amounts of noise to the Biased SST. Specifically, for each example in the training set we replace the original label with a random label with probability $p_{\text{noise}} = 0, 0.1, \ldots, 0.5$.[3] We then train models on these increasingly noisy datasets using both a parametric (R-PDRO) and non-parametric (NonParam) approach. To simplify experiments we only run one hyper-parameter configuration for each ($\tau = 0.1$ and $\kappa = 1$ for R-PDRO and NonParam respectively) and report the test accuracies of the model with the highest robust accuracy on the validation set. Results are averaged over 5 runs with different random seeds.

As showcased in Figure 6.1, while R-PDRO's robust accuracy seems to suffer under the effect of noise, we find that its average accuracy is a lot more resilient to the introduction of label noise compared to NonParam, losing around 12 points when $p_{\text{noise}} = 0.2$ ($88.04 \rightarrow 75.89$), versus 26 for NonParam ($83.14 \rightarrow 57.16$). This further supports our hypothesis that non-parametric adversaries tend to fit to these noisy examples, which decreases the overall quality of the resulting classifier.

[3] $p_{\text{noise}} = 0$ corresponds to the original dataset.

6.3.2 Optimization with Simultaneous Gradient Updates Plays a Crucial Role

Despite the aforementionned results, it remains unclear *why* R-PDRO learns re-weightings that are less sensitive to label noise or difficult examples compared to R-PDRO. Indeed, since the non-parametric adversary is the optimal solution of the inner minimization problem in Eq. 6.2, it stands to reason that (large, sometimes over-parameterized) parametric adversaries from R-PDRO would converge to the same solutions as NonParam.

Our hypothesis is that the simultaneous updates to both model and adversary parameters prevent the parametric adversary from converging towards such non-parametric solutions, and provides some implicit regularization against up-weighting examples that are noisy or too difficult.

To verify this hypothesis, we conduct a toy experiment where we allow the adversary to take additional gradient steps in-between each update to the classifier. At the limit, this would allow the adversary to find an optimum of the inner maximization problem at each step of training the classifier (which for large enough adversaries, might come close to the non-parametric solution).

For computational efficiency, these experiments are performed on the toy setting described in Chapter 5, Section 5.4.7: a linear model is trained on a binary classification problem with two domains, one of which is severely under-represented. For our adversary, we experiment with a linear adversary, as well as larger multilayer perceptrons with one hidden layer and 2 (MLP-2) and 4 (MLP-4) hidden units. In Figure 6.2, we report the average robust accuracy (across 5 reruns) for classifiers trained with R-PDRO when the adversary is allowed to take more steps than the classifier.

We observe that R-PDRO's robust accuracy suffers from giving the adversary too much time to catch up with the classifier: as the number of updates to the adversary increases, robust accuracy decreases. This effect is amplified in larger adversaries (*e.g.* MLP-4).

This experiment underlines the importance of the simultaneous gradient updates, which prevent large, over-parameterized adversaries from converging to the sub-optimal non-parametric solution.

6.3.3 Batch-level Normalization vs Self-normalization

In Section 6.1.3, we discussed two alternatives for enforcing the normalization constraint on the ratios ($\mathbb{E}_p r_\psi = 1$): a regularization-based approach ("self-normalization") and batch level renormalization. In Figure 6.3a, we show the effect of self-normalization with different values of the regularization weight β for a fixed value of the KL penalty $\tau = 0.1$. We find that batch-level nor-

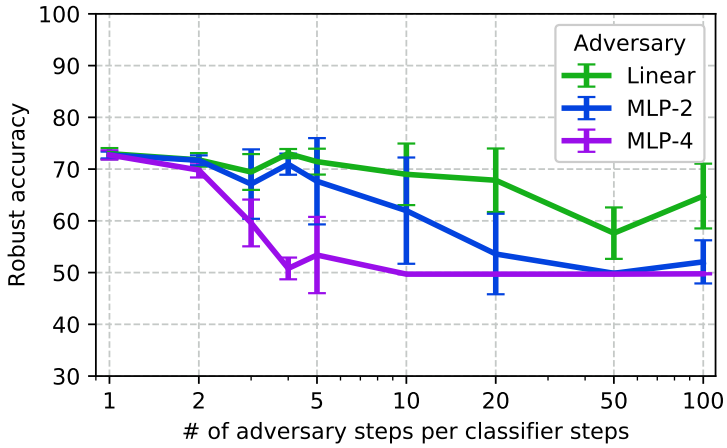


Figure 6.2: Evolution of R-PDRO's robust accuracy as the adversary is allowed to take more gradient steps than the classifier in a toy setting.

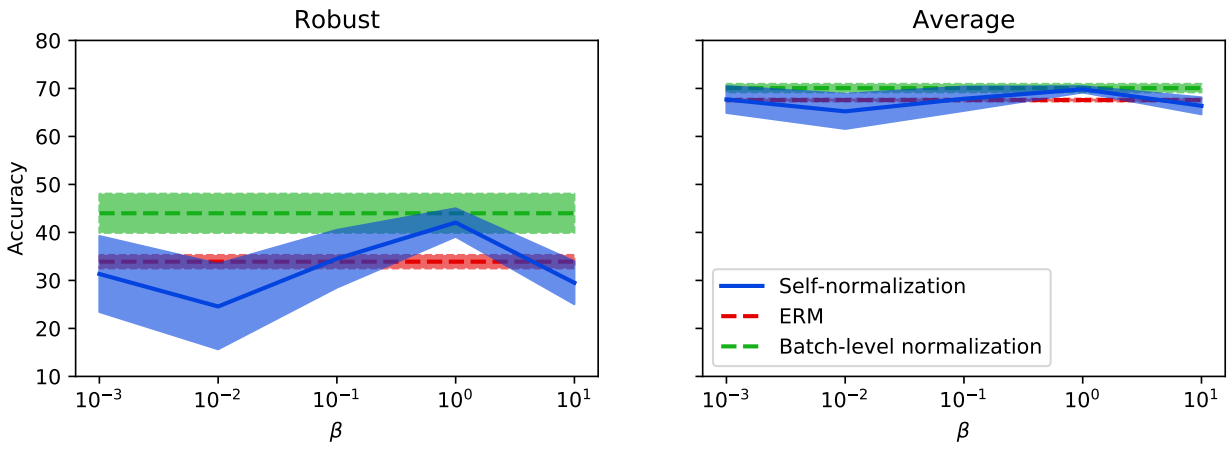
malization achieves a slightly better robust accuracy than the best self-normalization configuration.

In chapter 5, the Minmax stopping criterion was identified as a major contributing factor to the performance of parametric DRO approaches. To understand how it affects each normalization strategy, we perform the same experiment as above but this time *without* the Minmax criterion, selecting instead the model with the highest robust accuracy on the validation set ("Oracle stopping"), which provides an upper bound to the robust accuracy that can be achieved. Results in Figure 6.3b show that although accuracies are generally closer, we again find that batch-level normalization matches the best self-normalization penalty. This indicates that batch-level normalization not only performs as well as self-normalization (without the need for tuning the additional hyper-parameter β), but also that it interacts better with the Minmax stopping criterion, making it a preferable alternative overall.

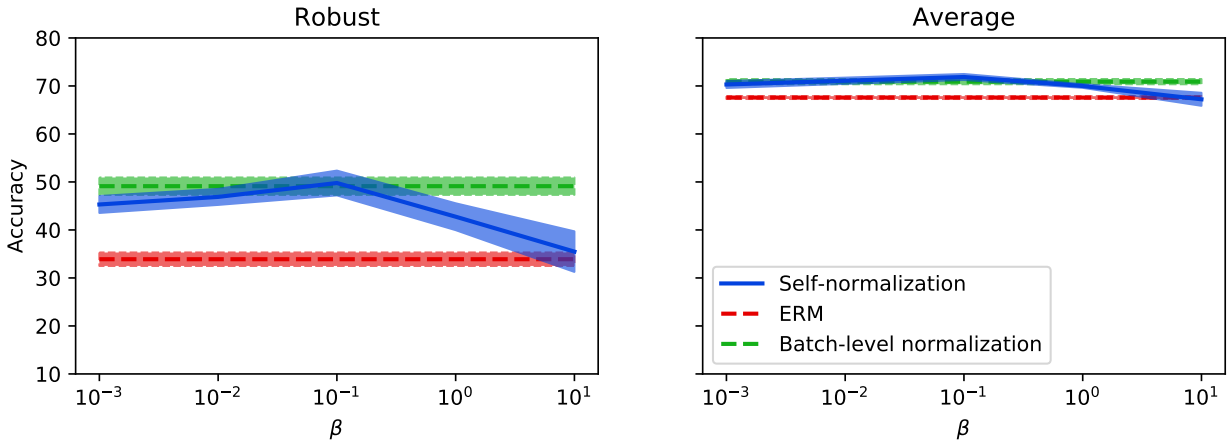
6.3.4 Effect of Batch Size on Re-normalization

As pointed out in Section 6.1.3, a potential downside of the minibatch-level normalization approach is that the effective weight of each sample then depends on the other samples within the minibatch. For example, consider an adversary that assigns high weight to only 5% of the training data. With a small enough batch size, it is likely that some batches may not contain any example of the high weight subpopulation, in which case minibatch level renormalization will overestimate the weight of the sample in the minibatch.

To assess the severity of this issue, we run R-PDRO on Biased SST with $\tau = 0.1$ and vary



(a) Minmax stopping



(b) Oracle stopping

Figure 6.3: Effect of self-normalization coefficient β on robust and average accuracy. For two different stopping strategy: Minmax stopping (used in experiments in Section 6.2) and Oracle stopping. We report results of ERM (which corresponds to $\beta = \infty$) and batch level renormalization for comparison.

the batch size in $\{4, 8, 16, 32, 64, 128\}$. Each configuration is run 3 times, and we report average and standard deviation of the robust and average test accuracies in Table 6.3. Results suggest that while robust accuracy indeed deteriorates for lower batch sizes (4 and 8), results are consistently good for batch sizes upwards of 16, a reasonable number considering that larger batch sizes are often preferred in the literature (Popel & Bojar, 2018; Goyal et al., 2017).

Table 6.3: Effect of batch size on R-PDRO performance.

Batch size	Robust	Average
64 (ERM)	32.97 $\pm$ 2.34	92.42 $\pm$ 0.38
4	37.50 $\pm$ 2.37	92.32 $\pm$ 0.75
8	38.08 $\pm$ 2.06	91.61 $\pm$ 0.48
16	41.67 $\pm$ 3.53	91.24 $\pm$ 0.34
32	42.32 $\pm$ 0.72	89.77 $\pm$ 1.42
64	44.15 $\pm$ 6.83	88.30 $\pm$ 2.33
128	42.25 $\pm$ 7.90	88.21 $\pm$ 2.02

6.4 Conclusion

In this chapter we have proposed a parametric, likelihood ratio based approach to distributionally robust optimization of machine learning models. With the proposed method, we can use any type of parametric function estimator to define the uncertainty set of the DRO min-max game. We showed that with a careful renormalization strategy, the proposed method (R-PDRO) can be used to train robust models. It depends on very few hyper-parameters and consistently performs well on a number of benchmarks, making it an appealing off-the-shelf option. Finally we have shown that such parametric approaches are more resilient to the presence of noise in the training data when compared to their non-parametric alternatives.

The main downside of R-PDRO is the computational overhead of jointly training a second neural model. An interesting direction for future work is to improve its efficiency through parallel computation or by sharing parameters between the classifier and the adversary.

Part III

Adaptation

Chapter 7

Regularizing Trajectories to Mitigate Catastrophic Forgetting

> It is good to have an end to journey toward;
> but it is the journey that matters, in the end.
>
> *Ursula K. Le Guin*

7.1 Introduction

The methods presented in previous chapters are able to train models that are more robust to distributional shifts because they can identify sub-population of the training data corresponding to potential domains where the model performs poorly. However, there are cases where the model might be confronted with a completely unseen test distribution, or even with a new task. In such cases where the test environment cannot be anticipated, an alternative solution is to *adapt* the model, using the limited amounts of data that are available in the target domain or task. However, continual learning of multiple tasks is hampered by catastrophic forgetting (McCloskey & Cohen, 1989; Ratcliff, 1990), the tendency of previously acquired knowledge to be overwritten when learning a new task.

Modern techniques to mitigate catastrophic forgetting can be roughly categorized into 3 lines of work (see Parisi et al. (2019) for a comprehensive overview): 1. regularization-based approaches, where forgetting is mitigated by the addition of a penalty term in the learning objective (Kirkpatrick et al. (2017); Chaudhry et al. (2018a), *inter alia*), 2. dynamic architectures approaches, which

incrementally increase the model's capacity to accomodate the new tasks (Rusu et al., 2016), and 3. memory-based approaches, which retain data from learned tasks for later reuse (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2018b, 2019). Among these, regularization-based approaches are particularly appealing because they do not increase the model size and do not require access to past data. This is particularly relevant to real-world scenarios where keeping data from previous training tasks may be impractical because of infrastructural or privacy-related reasons. Moreover, they are of independent intellectual interest because of their biological inspiration rooted in the idea of synaptic consolidation (Kirkpatrick et al., 2017).

In these regularization-based approaches, a good regularizer will ensure that, when learning a new task, gradient descent will ultimately converge to parameters that yield good results on the new task while preserving performance on previously learned tasks. Critically, this is predicated upon successful optimization of the regularized objective, a fact that has been largely taken for granted in previous work. Non-convexity of the loss function, along with noise in the data (due to small or biased datasets) or in the gradients (due to stochastic gradient descent), can yield optimization trajectories — and ultimately convergence points — that are highly non-deterministic, even for the same starting parameters. As we demonstrate in this chapter, this can cause unintended catastrophic forgetting along the optimization path. This is illustrated in a toy setting in Figure 7.1: a two parameter model is trained to perform task T_2 (an arbitrary bi-modal loss function) after having learned task T_1 (a logistic regression task). Standard finetuning, even in the presence of a regularized objective (EWC; Kirkpatrick et al. (2017)), quickly changes the loss of T_1 and tends to converge to a solution with high T_1 loss.

We propose to remedy this issue by regularizing not the objective function but *the optimization trajectory itself*, specifically by preconditioning gradient descent with the empirical Fisher information of previously learned tasks (§7.3). This yields what we refer to as a *co-natural* gradient, an update rule inspired by the natural gradient (Amari, 1997), but taking the Fisher information of *previous tasks* as a natural Riemannian metric[1] of the parameter space, instead of the Fisher information of the task being optimized for. When we introduce our proposed co-natural gradient for the toy example of Figure 7.1, the learning trajectory follows a path that changes the loss on T_1 much more slowly, and tends to converges to the optimum that incurs the lowest performance degradation on T_1.

We test the validity of our approach in a continual learning scenario (§7.4). We show that the co-natural gradient consistently reduces forgetting in a variety of existing continual learning

[1]Informally, the reader can think of a Riemannian metric as a function that assigns an inner product $u, v \mapsto g_{\mathbf{x}}(u, v)$ to each point x in the space, thus inducing a localized notion of distance and curvature.

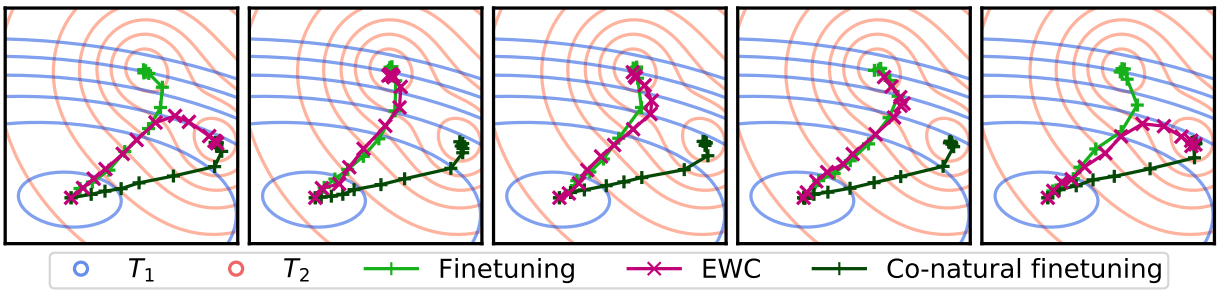


Figure 7.1: **On the importance of trajectories: an example with 2-dimensional logistic regression.** Having learned task T_1, the model is trained on T_2 with two different objectives: minimizing the loss on T_2 (Finetuning) and a regularized objective (EWC; Kirkpatrick et al. (2017)). We add a small amount of Gaussian noise to gradients in order to simulate the stochasticity of the trajectory. Plain finetuning and EWC often converge to a solution with high loss for T_1, but the co-natural optimization trajectory *consistently* converges towards the optimum with lowest loss for T_1. Each individual plot corresponds to a separate run using a different random seed.

approaches by a large factor, and greatly improves performance over simple finetuning, without modification to the training objective. As an additional contribution, we assemble a new collection of 13 tasks for evaluating continual learning on text classification.

We further investigate the special case of transfer learning in a two-task, low-resource scenario. In this specific case, control over the optimization trajectory is particularly useful because the optimizer has to rely on early stopping to prevent overfitting to the meager amount of training data in the target task. We show that the co-natural gradient yields the best trade-offs between source and target domain performance over a variety of hyper-parameters (§7.6).

The work presented in this chapter bears many similarities to previous related works on Bayesian formulations of continual learning which used similar inverse-Fisher conditioning to reduce forgetting: NVCL (Tseran et al., 2018) and GNG (Chen et al., 2018). However, we our work makes significant contributions over these previous approaches

First, these prior works were developed in the framework of Bayesian continual learning. While the central idea of using the Fisher to modulate the update is similar, we adopt a broader approach and successfully apply this technique to a variety of other techniques outside the Bayesian scaffolding. Second, we introduce a damping coefficient, which allows control over the regularization effect of the co-natural gradient, as evidenced by our analysis in Appendix A.4 specifically, and more generally by the results in Section 4 and 5, where our results are obtained by performing grid-search over this additional hyper-parameter. Third, we provide empirical evidence of the co-

natural gradients' effectiveness in a variety of experimental settings from relatively small, toy-like models and datasets (*e.g.* Split CIFAR or Omniglot) to more realistic scenarios (*e.g.* MiniImageNet, BERT on text classification and our MT experiments). In comparison, the aforementioned previous work only test small 2 layers MLP models on MNIST and FashionMNIST. We believe that this establishes our work as a strong, independent contribution.

7.2 Background and Notations

We first give a brief overview of the continual learning paradigm and existing approaches for overcoming catastrophic forgetting.

7.2.1 Notation

Let us define a task as a triplet containing an input space $\mathcal{X}$ and an output space $\mathcal{Y}$, both measurable spaces, as well as a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$. In general, learning a task will consist of training a model to approximate the conditional distribution $p(y \mid x)$ induced by $\mathcal{D}$.

Consider a probabilistic model p_θ parametrized by $\theta \in \mathbb{R}^d$ where d is the size of the model, trained to perform a *source* task $S = \langle \mathcal{X}_S, \mathcal{Y}_S, \mathcal{D}_S \rangle$ to some level of performance, yielding parameters θ_S. In the most simple instance of continual learning, we are tasked with learning a second *target* task $T = \langle \mathcal{X}_T, \mathcal{Y}_T, \mathcal{D}_T \rangle$. In general in a multitask setting, it is not the case that the input or output spaces are the same. The discrepancy between input/output space can be addressed in various ways, *e.g.* by adding a minimal number of task-specific parameters (for example, different softmax layers for different label sets). To simplify exposition, we set these more specific considerations aside for now, and assume that $\mathcal{X}_S = \mathcal{X}_T$ and $\mathcal{Y}_S = \mathcal{Y}_T$.

At any given point during training for task T, our objective will be to minimize the loss function $\mathcal{L}_T(\theta)$ – generally the expected negative log-likelihood $\mathbb{E}_{x,y\sim\mathcal{D}_T}[-\log p_\theta(y \mid x)]$. Typically, this will be performed by iteratively adding incremental update vectors $\delta \in \mathbb{R}^d$ to the parameters $\theta \longleftarrow \theta + \delta$.

7.2.2 Existing Approaches for Continual Learning

In this chapter, we focus on those models that have a fixed architecture over the course of continual learning. The study of continual learning for models of fixed capacity can be split into two distinct (but often overlapping) streams of work:

Regularization-based approaches introduce a penalty in the loss function $\mathcal{L}_T$, often quadratic, pulling the weights θ back towards θ_S:

$$\mathcal{L}_T(\theta) = \underbrace{\mathbb{E}_{\mathcal{D}_T} - \log p_\theta(y \mid x)}_{\text{NLL on task } T} + \underbrace{\lambda(\theta - \theta_S)^T \Omega_S (\theta - \theta_S)}_{\text{Regularization term}} \tag{7.1}$$

where Ω_S is a matrix, generally diagonal, that encodes the respective importance of each parameter with respect to task S, and λ is a regularization strength hyper-parameter. Various choices have been proposed for Ω_S; the diagonal empirical Fisher information matrix (Kirkpatrick et al., 2017), or path-integral based importance measures (Zenke et al., 2017; Chaudhry et al., 2018a). More elaborate regularizers have been proposed based on *e.g.* a Bayesian formulation of continual learning (Nguyen et al., 2017; Ahn et al., 2019) or a distillation term (Li & Hoiem, 2016; Dhar et al., 2019). The main advantage of these approaches is that they do not rely on having access to training data of previous tasks.

Memory-based approaches store data from previously seen tasks for re-use in continual learning, either as a form of constraint, by *e.g.* ensuring that training on the new task doesn't increase the loss on previous tasks (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2018b), or for replay *i.e.* by retraining on instances from previous tasks (Rebuffi et al., 2017; Chaudhry et al., 2019; Aljundi et al., 2019b,a). Various techniques have been proposed for the selection of samples to store in the memory (Chaudhry et al., 2019; Aljundi et al., 2019b) or for retrieval of the samples to be used for replay Aljundi et al. (2019a).

All of these methods rely on stochastic gradient descent to optimize their regularized objective or to perform experience replay, with the notable exception of GEM (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2018b), where the gradients are projected onto the orthogonal complement of previous task's gradients. However, this method has been shown to perform poorly in comparison with simple replay (Chaudhry et al., 2019), and it still necessitates access to data from previous tasks.

7.3 Regularizing the Trajectory

After briefly recalling how the usual update is obtained in gradient descent, we derive a new, *conatural* update designed to better preserve the distribution induced by the model over previous tasks.

7.3.1 Warm up: the Standard Gradient Descent Update

At point θ in the parameter space, gradient descent finds the optimal update δ that is (1) small and (2) locally minimizes the difference in loss $\mathcal{L}(\theta + \delta) - \mathcal{L}(\theta)$ ($\approx \delta^\intercal \nabla_\theta \mathcal{L}$ at the first order). Traditionally this can be formulated as minimizing the Lagrangian:

$$\mathbb{L}(\delta) = \underbrace{\delta^\intercal \nabla_\theta \mathcal{L}_T}_{\substack{\text{first order} \\ \text{loss minimization term}}} + \underbrace{\mu \|\delta\|^2}_{\text{“small update” term}} \tag{7.2}$$

with Lagrangian multiplier $\mu > 0$. Minimizing $\mathbb{L}$ for δ yields the well-known optimal update δ^*:

$$\delta^* = -\frac{1}{2\mu} \nabla_\theta \mathcal{L}_T \tag{7.3}$$

where $\frac{1}{2\mu}$ corresponds to the learning rate (see Appendix B for the full derivation).

7.3.2 KL Regularization of Trajectories

The $\|\delta\|^2$ term in $\mathbb{L}$ implicitly expresses the underlying assumption that the best measure of distance between parameters θ and $\theta + \delta$ is the Euclidean distance. In a continual learning setting however, the quantity we are most interested in preserving is the probability distribution that θ models on the source task S:

$$p^S_\theta(x, y) = p_\theta(y \mid x) p^S(x) \tag{7.4}$$

Therefore, a more *natural* distance between θ and $\theta + \delta$ is the Kullback-Leibler divergence $\mathrm{KL}(p^S_\theta \| p^S_{\theta+\delta})$ (Kullback & Leibler, 1951). For preventing catastrophic forgetting along the optimization path, we incorporate incorporate this KL term into the Lagrangian $\mathbb{L}$ itself:

$$\mathbb{L}(\delta) = \delta^\intercal \nabla_\theta \mathcal{L}_T + \mu \|\delta\|^2 + \nu \mathrm{KL}(p^S_\theta \| p^S_{\theta+\delta}) \tag{7.5}$$

Doing so means that the optimization trajectory will tend to follow the direction that changes the distribution of the model the least. Notably, this is not a function of the previous objective $\mathcal{L}_S$, so knowledge of the original training objective is not necessary during continual learning (which is typically the case in path-integral based regularization methods (Zenke et al., 2017) or experience replay (Chaudhry et al., 2019)).

7.3.3 Co-natural Gradient Optimization

Presuming that δ is small, we can perform a second order Taylor approximation of the function $\delta \mapsto \mathrm{KL}(p^S_\theta \| p^S_{\theta+\delta})$ around 0. Considering that both the zeroeth and first order terms are null because 0 is a global minimizer of $\delta \mapsto \mathrm{KL}(p^S_\theta \| p^S_{\theta+\delta})$, this reduces the Lagrangian to a quadratic optimization problem (we refer the reader to Pascanu & Bengio (2013) for a more detailed derivation.):

$$\mathbb{L}(\delta) = \delta^\intercal \nabla_\theta \mathcal{L}_T + \mu \|\delta\|^2 + \frac{1}{2}\nu \delta^\intercal F^S_\theta \delta \tag{7.6}$$

where F^S_θ is the Hessian of the KL divergence around θ. A crucial, well-known property of this matrix is that it coincides with the Fisher information matrix[2] $\mathbb{E}_{x,y\sim p_\theta}[(\nabla \log p^S_\theta)(\nabla \log p^S_\theta)^T]$ (the expectation being taken over the model's distribution p_θ; see Appendix B for details). This is appealing from a computational perspective because the Fisher can be computed by means of first order derivatives only.

Minimizing for δ yields the following optimal update:

$$\delta^* = -\lambda \left[F^S_\theta + \alpha I\right]^{-1} \nabla_\theta \mathcal{L}_T \tag{7.7}$$

where coefficients μ and ν are folded into two hyper-parameters: the learning rate λ and a damping coefficient α (the step-by-step derivation can be found in Appendix B). In practice, especially with low damping coefficients, it is common to obtain updates that are too large (typically when some parameters have no effect on the KL divergence). To address this, we re-normalize δ^* to have the same norm as the original gradient, $\|\nabla \mathcal{L}_T\|$. Moreover, to improve numerical stability in the case of degenerate Fisher matrices, we add a very small damping term of 10^{-12} to all experiments.

For computational reasons, we will make 3 key practical approximations to the Fisher:

1. $F^S_\theta \approx F^S_{\theta_S}$: we maintain the Fisher computed at θ_S, instead of recomputing F_S at every step of training. This relieves us of the computational burden of updating the Fisher for every new value of θ. This approximation (shared by previous work, *e.g.* Kirkpatrick et al. (2017);Chaudhry et al. (2018a)) is only valid insofar as θ_S and θ are close. Empirically we observe that this still leads to good results.

2. F^S is diagonal: this is a common approximation in practice with two appealing properties. First, this makes it realistic to store the d diagonal Fisher coefficients in memory. Second,

[2]Hence our use of the letter F to designate the Hessian

this trivializes the inverse operation (simply invert the diagonal elements).

3. Empirical Fisher: this common approximation replaces the expectation under the model's distribution by the expected log-likelihood of the *true* distribution: $E_{x,y\sim p^S}[(\nabla \log p^S_\theta)(\nabla \log p^S_\theta)^T]$ (mind the subscript). This is particularly useful in tasks with a large or unbounded number of classes (*e.g.* structured prediction), where summing over all possible outputs is intractable. We can then compute the diagonal of the empirical Fisher using Monte Carlo sampling: $\frac{1}{N}\sum_{i=1}^{N}[\nabla \log p^S_\theta(y_i \mid x_i)]^2$ with (x_i, y_i) sampled from $\mathcal{D}_S$ (we use $N = 1000$ for all experiments).

This formulation bears many similarities with the natural gradient from Amari (1997), which also uses the KL divergence as a metric for choosing the optimal update δ^* in gradient descent. There is a however a crucial difference, both in execution and purpose: where the natural gradient uses knowledge of the curvature of the KL divergence of $\mathcal{D}_T$ to *speed-up* convergence, our proposed method leverages the curvature of the KL divergence on $\mathcal{D}_S$ to *slow-down* divergence from $p^S_{\theta_S}$. To highlight the resemblance and complementarity between these two concepts, we refer to the new update as the *co-natural* gradient.

7.3.4 Beyond Two Tasks

In a continual learning scenario, we are confronted with a large number of tasks $T_1 \dots T_n$ presented in sequential order. When learning T_n, we can change the Lagrangian $\mathbb{L}$ from 7.5 to incorporate the constraints for all previous tasks $T_1 \dots T_{n-1}$:

$$\mathbb{L}(\delta) = \delta^\intercal \nabla_\theta \mathcal{L}_{T_n} + \mu\|\delta\|^2 + \sum_{i=1}^{n-1} \nu_i \mathrm{KL}(p^{T_i}_\theta \| p^{T_i}_{\theta+\delta}) \tag{7.8}$$

This in turn changes the Fisher in Eq. 7.8 to $\tilde{F}_{n-1} := \frac{1}{2}\sum_{i=1}^{n-1} \nu_i F^{T_i}$. The choice of the coefficients ν_i is crucial. Setting all ν_i to the same value, *i.e.* assigning the same importance to all tasks is suboptimal for a few reasons. First and foremost, it is unreasonable to expect of a model with finite capacity to remember an unbounded number of tasks (as tasks "fill-up" the model capacity, $\tilde{F}_{n-1}$ is likely to become more "homogeneous"). Second, as training progresses and θ changes, our approximation that $F^{T_i}_\theta \approx F^{T_i}_{\theta_{T_i}}$ is less and less likely to hold.

We address this issue in the same fashion as Schwarz et al. (2018), by keeping a rolling expo-

Table 7.1: Average accuracies and forgetting after all tasks have been learnt, with and without the co-natural gradient. Results are reported in percentages ($\pm$ the standard deviation over 5 re-runs). Bold print indicates statistically significant difference between standard and co-natural ($p < 0.05$).

(a) **Split CIFAR**

	Finetuning	EWC	ER
	Average accuracy ↑		
Standard	31.56 ±2.11	50.63 ±2.30	59.30 ±0.92
Co-natural	**55.97** ±1.48	**54.81** ±2.06	**64.72** ±0.85
	Forgetting ↓		
Standard	35.48 ±2.46	11.56 ±1.93	9.20 ±0.99
Co-natural	**5.04** ±1.24	**4.77** ±1.11	**2.52** ±0.58

(b) **Omniglot**

	Finetuning	EWC	ER
	Average accuracy ↑		
Standard	20.21 ±3.97	**71.16** ±0.51	70.41 ±3.05
Co-natural	**67.84** ±4.85	67.39 ±1.45	**78.04** ±1.79
	Forgetting ↓		
Standard	73.52 ±3.80	22.27 ±0.76	12.60 ±2.76
Co-natural	**12.09** ±4.69	**8.61** ±1.93	**4.95** ±0.39

(c) **MiniImageNet**

	Finetuning	EWC	ER
	Average accuracy ↑		
Standard	35.36 ±1.75	62.19 ±1.49	65.93 ±0.49
Co-natural	**63.91** ±1.54	63.89 ±1.11	**71.00** ±0.67
	Forgetting ↓		
Standard	42.08 ±1.56	6.98 ±1.65	12.57 ±0.61
Co-natural	**8.86** ±2.03	6.96 ±1.28	**4.72** ±0.89

(d) **Text Classification**

	Finetuning	EWC	ER
	Average accuracy ↑		
Standard	62.13 ±2.90	67.79 ±1.18	**73.57** ±0.42
Co-natural	**68.74** ±1.49	68.07 ±1.30	69.35 ±1.45
	Forgetting ↓		
Standard	13.79 ±2.84	0.75 ±0.39	2.16 ±0.23
Co-natural	**0.73** ±0.18	0.50 ±0.37	**0.50** ±0.25

nential average of the Fisher matrices:

$$\tilde{F}_n^{\gamma} = \gamma F_{T_n} + (1-\gamma)\tilde{F}_{n-1}^{\gamma} \tag{7.9}$$

In this case, previous tasks are gracefully forgotten at an exponential rate controlled by γ. We account for the damping α term in Eq. 7.7 by setting $\tilde{F}_0 := \frac{\alpha}{\gamma} I$. In preliminary experiments, we have found $\gamma = 0.9$ to yield consistently good results, and use this value in all presented experiments.

7.4 Continual Learning Experiment Setting

7.4.1 Setup, Evaluation and Baselines

To examine our hypothesis that controlling the optimization trajectory with the co-natural gradient reduces catastrophic forgetting, we follow the experimental procedure from Chaudhry et al. (2019):

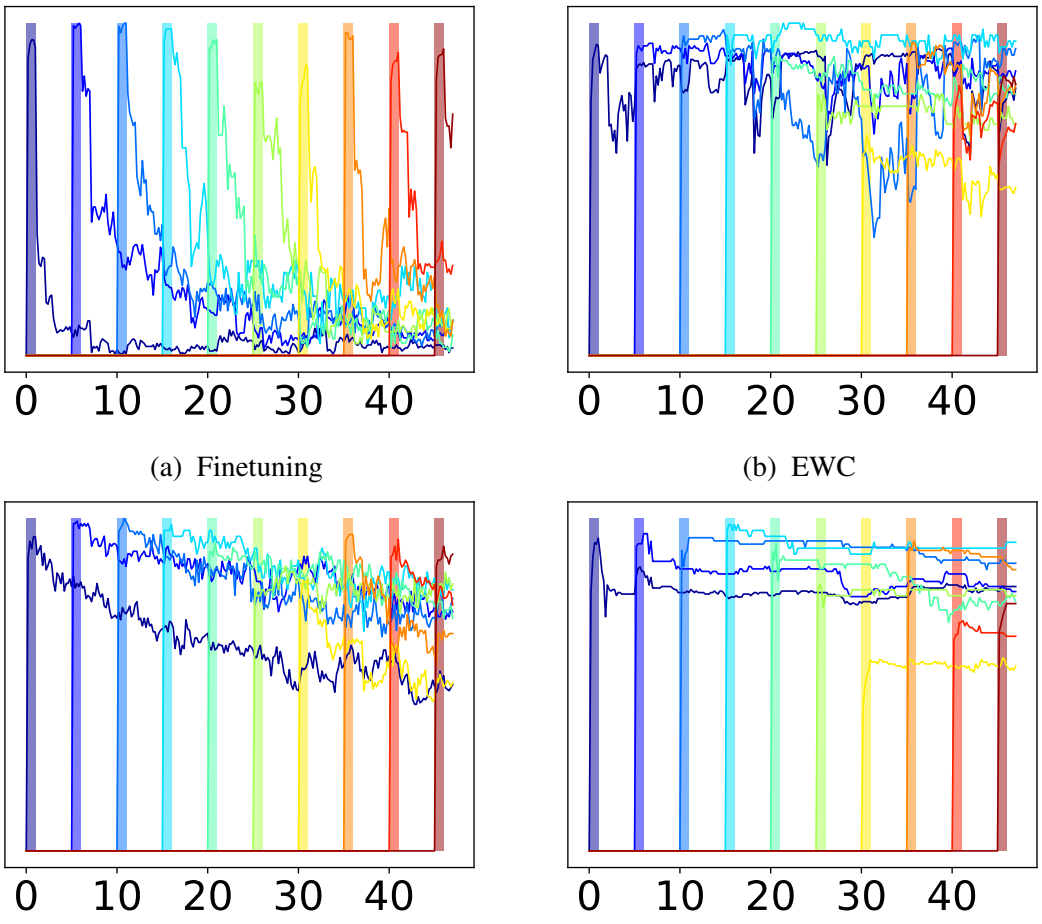
given a collection of tasks, we create a "validation set" of 3 tasks used to select the best hyper-parameters, and keep the remaining tasks for evaluation. This split is chosen at random and kept the same across all experiments. In most settings, the nature and possibly the number of classes changes from task to task. We account for this by training a separate "task head" for each task: an affine transform projecting the features onto the number of classes, followed by a softmax layer. We apply continual learning only to the remaining, "feature-extraction" part of the model.

We report results using two common metrics for continual learning: **average accuracy**, the accuracy at the end of training averaged over all tasks, and **forgetting**. Forgetting is defined in Chaudhry et al. (2018a) as the difference in performance on a task between the current model and the best performing model on this task. Formally if A_t^T represents the accuracy on task T at step t of training, the forgetting F_t^T at step t is defined as $F_t^T = \max_{\tau<t} A_\tau^T - A_t^T$. Low forgetting means that the model tend to keep the same level of performance on a task it has previously learned.

We implement the proposed co-natural update rule on top of 3 baselines:

- **Finetuning**: Simply train the model on the task at hand, without any form of regularization.
- **EWC**: Proposed by Kirkpatrick et al. (2017), it is a simple but effective quadratic regularization approach. While neither the most recent nor sophisticate regularization technique, it is a natural baseline for us to compare to in that it also consists in a Fisher-based penalty — albeit in the loss function instead of the optimization dynamics. We also use the rolling Fisher described in Section 7.3.4, making our EWC baseline equivalent to the superior online EWC introduced by Schwarz et al. (2018).
- **ER**: Experience replay with a fixed sized episodic memory proposed by Chaudhry et al. (2019). While not directly comparable to EWC in that it presupposes access to data from previous tasks, ER is a simple approach that boasts the best performances on a variety of benchmarks (Chaudhry et al., 2019). In all experiments, we use a memory of size 1,000 populated with reservoir sampling.

Training proceeds as follows: we perform exhaustive search on all possible hyper-parameter configurations using the validation tasks. Every configuration is reran 5 times (3 for text classification) with a different random seed and assigned a score based on the average task score at the end of training (averaged over reruns). We then evaluate the best hyper-parameters by continual training on the evaluation tasks. Results are reported over 5 random restarts, and we control for statistical significance using a paired t-test (we pair together runs with the same task ordering). For all experiments we train with plain stochastic gradient descent.



(a) Finetuning (b) EWC

(c) ER (d) Co-natural Finetuning

Figure 7.2: Evolution of task performance over the course of continual learning on one ordering of Omniglot. For visibility we only show accuracies for every fifth task. The rectangular shaded regions delineate the period during which each task is being trained upon; with the exception of ER, this is the only period the model has access to the data for this task.

7.4.2 Datasets

We use the four following task suites in our experiments:

Split CIFAR

The CIFAR100 dataset, split into 20 independent 5-way classification tasks with 2500 training examples and 500 test examples each. This split, performed at random, is kept the same across all

experiments, only the order of these tasks is changed. Following Chaudhry et al. (2018b), we use a smaller version of the ResNet architecture (He et al., 2016) train on each task for 1 epoch with batch size 10.

Omniglot

The Omniglot dataset (Lake et al., 2015) consists of 50 independent character recognition datasets on different alphabet. We adopt the setting of Schwarz et al. (2018) and consider each alphabet as a separate task, and split each task such that every character is present 12, 4 and 4 times in the training, validation and test set respectively (out of the 20 images for each character).[3]

On this dataset we use the same small CNN architecture as Schwarz et al. (2018). We augment the training data by randomly shifting and rotating images and train on each task for 2500 steps (120 to 417 epochs depending on the alphabet) with batch size 32. We ignore the validation data and simply evaluate on the test set at the end of training.

Split MiniImageNet

The MiniImageNet dataset (a subset of the popular ImageNet (Deng et al., 2009) dataset[4]; Vinyals et al. (2016)). We split the dataset into 20 disjoint 5-way classification tasks, similarly to Split CIFAR, and use the same smaller ResNet. We train on each task for 500 steps with batch size 32 ($= 6.4$ epochs). We ignore the validation data and simply evaluate on the test set at the end of training.

Text Classification

The most recent work on continual learning of language tasks (de Masson d'Autume et al., 2019) relies on a relatively small set of tasks (5 classification tasks from Zhang et al. (2015)). This small number of tasks is not amenable to our experimental setup described above, where 3 tasks are reserved for validation. Therefore we assemble a larger collection of text classification datasets from three sources: the GLUE benchmark (Wang et al., 2019b), the SuperGLUE benchmark (Wang et al., 2019a) and the text classification datasets used in de Masson d'Autume et al. (2019). To keep

[3]Note that this is a different setting than the usual meta-learning scenario that Omniglot is used for.

[4]Similarly to Omniglot, MiniImageNet was originally intended as a meta-learning benchmark and therefore its standard train/validation/test split consists of disjoint classes. We perform a custom transversal split so that the dataset can be used as a standard 100-way classification task. The accuracies reported here are not to be compared with the meta-learning literature.

things relatively simple, we only keep tasks that 1. are single sentence or sentence pair classification tasks and 2. have more than 1000 training examples. Specifically, we use the following tasks from each dataset (reported with the training data size):

- **GLUE**
  - CoLA (Warstadt et al., 2018): 8.6K
  - MultiNLI (Williams et al., 2018): 392.7K
  - MRPC (Dolan & Brockett, 2005): 3.7K
  - QNLI (Rajpurkar et al., 2016): 108.4K
  - QQP (Iyer et al., 2017): 363.8K
  - RTE (Dagan et al., 2005; Haim et al., 2006; Giampiccolo et al., 2007; Bentivogli et al., 2009): 2.5K
  - SST-2 (Socher et al., 2013): 67.3K
- **SuperGLUE**
  - BoolQ (Clark et al., 2019): 9.4K
- **de Masson d'Autume et al. (2019)**
  - AG News (Gulli, 2005) (115.0K)
  - Amazon Reviews Full (McAuley & Leskovec, 2013) (115.0K)
  - DBPedia (Lehmann et al., 2015) (115.0K)
  - Yahoo Answers (Zhang et al., 2015) (115.0K)
  - Yelp Reviews Full (Yelp, 2015) (115.0K)

Note that the RTE dataset from SuperGLUE also satisfies our task type and dataset size constraint, however according to Wang et al. (2019a) it is an exact duplicate of GLUE's RTE dataset, therefore we don't include it. de Masson d'Autume et al. (2019) performed some mild preprocessing and subsampling of the datasets in Zhang et al. (2015), however they did not release the final data. We perform a similar preprocessing step: for datasets where each sample consists in several pieces of text (typically review title and body in the Amazon and Yelp datasets), we concatenate all

into one utterance, joined with a period and a space (". "). We subsample the datasets to 115K training samples, 5K validation samples and 7.6K test samples.

We randomly select three out of these 13 tasks to serve as the validation split upon which to perform hyper-parameter search, namely: **BoolQ**, **MRPC** and **SST-2**. Since test sets are not available for the GLUE and SuperGLUE benchmark, we compute the final scores on the validation datasets. Note that in all our experiments, validation data is not used during training, therefore there is no leakage of the ultimate evaluation dataset in the training procedure

Instructions to download and code to preprocess the data will be made available at `https://github.com/pmichel31415/translate/tree/paul/pytorch_translate/research/adversarial/experiments` to facilitate reproduction of our results and future work on continual learning for text classification. Following recent practice in text classification, we use a large model that has already been pre-trained in an unsupervised fashion (specifically `bert-base-uncased`[5] from Devlin et al. (2018)), and fine-tune the model on the supervised classification tasks. We train on each task for 1000 steps with batch size 16 (from 7 to < 1 epochs due to the large variance in dataset sizes).

7.4.3 Grid-search Parameters

For each all image-related tasks, we perform grid-search over the following parameter values:

- Learning rate (all methods): 0.1, 0.03, 0.01
- EWC regularization strength (EWC, Co-natural EWC): 0.5, 1, 5
- Fisher damping coefficient (Co-natural finetuning, Co-natural EWC, Co-natural ER): 0, 1, 0.1

For text classification with BERT specifically, preliminary experiments showed that all methods benefitted from lower learning rate as well as lower regularization (likely due to the fact that optimization is starting from a pretrained model). We therefore use the following values:

- Learning rate (all methods): 0.05, 0.01, 0.005
- EWC regularization strength (EWC, Co-natural EWC): 0.5, 0.1, 0.05

[5]We use the open-source implementation from Wolf et al. (2019) with the default configuration.

- Fisher damping coefficient (Co-natural finetuning, Co-natural EWC, Co-natural ER): 1, 10, 100

For ER, we simply set the replay batch size to the same value as standard training (10 and 32 for Split CIFAR and Omniglot respectively). Note that whenever applicable, we re-normalize the diagonal Fisher so that the sum of its weights is equal to the number of parameters in the model. This is so that the hyper-parameter choice is less dependent on the size of the model. In particular this means that the magnitude of each diagonal element is much bigger, which is why we do grid-search over smaller regularization parameters for EWC than is common in the literature.

7.5 Continual Learning Results

7.5.1 The Co-natural Gradient Helps Prevent Forgetting

The upper half of Tables 7.1a, 7.1b, 7.1c and 7.1d reports the average accuracy of all the tasks at the end of training (higher is better). We observe that the co-natural gradient always improves greatly over simple finetuning, and occasionally over EWC and ER. We note that simple co-natural finetuning (without any other form of regularization) sometimes matches or exceeds the performance of EWC even though it requires strictly fewer resources (there is no need to store the previous parameters as in EWC, or data in ER).

Even more appreciable is the effect of the co-natural trajectories on forgetting, as shown in the lower half of Table 7.1. As evidenced by the results in the lowest rows, using the co-natural gradient on top of finetuning and ER systematically results in large drops in forgetting across all datasets. With EWC, the conclusion is more nuanced. While the co-natural gradient never increases forgetting, the effect is not always statistically significant.

To get a qualitative assessment of the learning trajectories that yield such results, we visualize the accuracy curves of 10 out of the 47 evaluation tasks of Omniglot in Figure 7.2. We observe that previous approaches do poorly at keeping stable levels of performance over a long period of time (especially for tasks learned early in training), a problem that is largely alleviated by the co-natural preconditioning. This seems to come at the cost of more intransigence (Chaudhry et al., 2018a), *i.e.* some of the later tasks are not being learnt properly. In models of fixed capacity, there is a natural trade-off between intransigence and forgetting (see also the "stability-plasticity" dilemma in neuroscience Grossberg (1982)). Our results position the co-natural gradient as a strong low-forgetting/moderate intransigence basis for future work.

7.5.2 Sensitivity to Damping

The main hyper-parameter for the co-natural gradient is the damping parameter α from Eq. 7.7. In our previous experiments, the value of α is chosen according to grid search on a small number of tasks. While this is a realistic setting for practical applications, in this section we perform a smaller, targeted experiment to examine the effect of α on catastrophic forgetting.

We focus on the 17 evaluation tasks of the Split CIFAR dataset and set the learning rate to 0.1. We observe how much the model "forgets" about the first task it has observed after training on all others. Specifically, we train on the first task, compute its Fisher F_1, and then proceed to train on the 16 remaining tasks with the co-natural gradient using $F_1 + \alpha$ (in particular we do not regularize for the other tasks). We observe how the value of alpha affects the final forgetting of the first task at the end of training, as well as the model's ability to learn new tasks (sometimes referred to as "intransigence" in the literature Chaudhry et al. (2018a)). As a measure of the latter, we report the maximum accuracy achieved by the model averaged over the 16 remaining tasks.

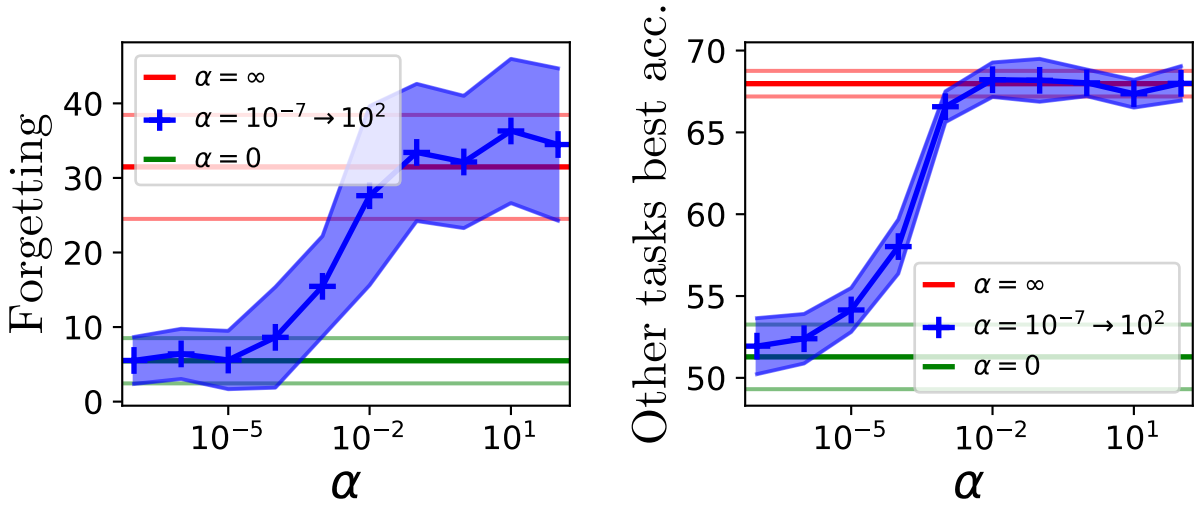
We evaluate values of α from 10^{-7} to 10^{2} (following a geometric progression), as well as the two extremal values $\alpha = 0$ ("pure" co-natural gradient[6]) and $\alpha = \infty$ (simple finetuning). All results are averaged over 10 random restarts (with different model initialization and task order).

We observe in Figure 7.3a that forgetting monotonically increases with the damping coefficient. Similarly, Figure 7.3b shows how increasing α results in lower intransigence. While these two observations are expected, it is interesting to note the existence of a "sweet spot" around $\alpha \in [10^{-5}, 10^{-4}]$ where damped co-natural gradient is significantly less intransigent than the undamped co-natural gradient while simultaneously not being significantly less robust to forgetting (all hypotheses are tested for with $p < 0.05$).

7.6 Low-Resource Adaptation Experiments

In this section we take a closer look at the specific case of adapting a model from a single task to another, when we only have access to a minimal amount of data in the target task. In this case, controlling the learning trajectory is particularly important because the model is being trained on an unreliable sample of the true distribution of the target task, and we have to rely on early-stopping to prevent overfitting. We show that using the co-natural gradient during adaptation helps both at preserving source task performance and reach higher overall target task performance.

[6]As mentioned in Section 7.3, we do actually add a small damping term $\varepsilon = 10^{-12}$ to all experiments.



(a) Forgetting

(b) Intransigence

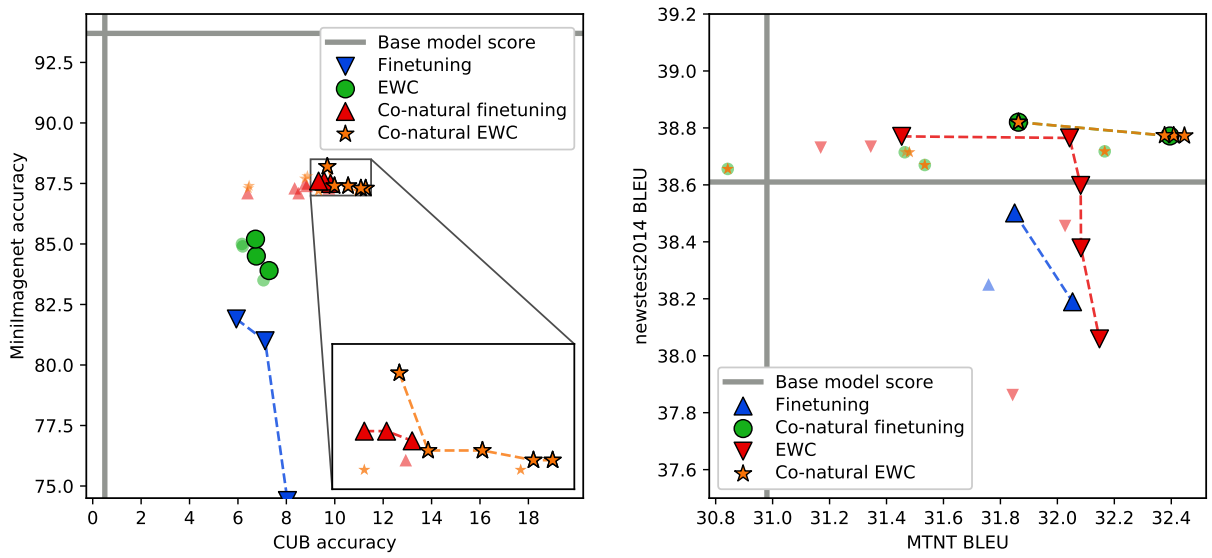
Figure 7.3: Effect of damping on forgetting and intransigence. The horizontal axis (measuring α) follows a logarithmic scale. The width of the bands on each side of the curves represent the standard deviation over the 10 re-runs.

7.6.1 Experimental Setting

We perform experiments on two different scenarios:

Image classification We take MiniImagenet as a source task and CUB (a 200-way birds species classification dataset; Welinder et al. (2010)) as a target task. To guarantee a strong base model despite the small size of MiniImageNet, we start off from a ResNet18 model (He et al., 2016) pretrained on the full ImageNet, which we retrofit to MiniImageNet by replacing the last fully connected layer with a separate linear layer regressed over the MiniImageNet training data. To simulate a low-resource setting, we sub-sample the CUB training set to 200 images ($\approx$ 1 per class). Scores for these tasks are reported in terms of accuracy.

Machine translation We consider adaptation of an English to French model trained on WMT15 (a dataset of parallel sentences crawled from parliamentary proceedings, news commentary and web page crawls; Bojar et al. (2015)) to MTNT (a dataset of Reddit comments; Michel & Neubig (2018a)). Our model is a Transformer (Vaswani et al., 2017) pretrained on WMT15. Similarly to CUB, we simulate a low-resource setting by taking a sub-sample of 1000 sentence pairs as a



(a) MiniImageNet to CUB adaptation

(b) WMT to MTNT fine-tuning

Figure 7.4: Low-resource adaptation results. The source (resp. target) task performance is represented on the vertical (resp. horizontal) axis. Pareto optimal configurations for each method are highlighted and the frontier is represented with dashed lines. The solid gray lines indicate the score of the original model trained on the source task.

training set. Scores for these two datasets are reported in terms of BLEU score (Papineni et al., 2002).[7]

Here we do not allow any access to data in the source task when training on the target task. We compare four methods **Finetuning** (our baseline), **Co-natural finetuning**, **EWC** (which has been proven effective for domain adaptation, see Thompson et al. (2019)) and **Co-natural EWC**.

Given that different methods might lead to different trade-offs between source and target task performance, with some variation depending on the hyper-parameters (*e.g.* learning rate, regularization strength...), we take inspiration from Thompson et al. (2019) and graphically report results for all hyper-parameter configuration of each method on the 2 dimensional space defined by the score on source and target tasks[8]. Additionally, we highlight the Pareto frontier of each method *i.e.* the set of configurations that are not strictly worse than any other configuration for the same model.

[7]We use sacrebleu (Post, 2018a) with `-tok intl` as recommended by Michel & Neubig (2018a).

[8]For CUB in particular we report the average accuracy of every configuration over 5 runs, each with a different 200-sized random subset of the data.

7.6.2 Results

The adaptation results for both scenarios are reported in Figure 7.4. We find that in both cases, the co-natural gradient not only helps preserving the source task performance, but to some extent it also allows the model to reach better performance on the target task as well. We take this to corroborate our starting hypothesis: while introducing a regularizer does help, controlling the optimization dynamics actively helps counteract overfitting to the very small amount of training data, because the co-natural pre-conditioning makes it harder for stochastic gradient descent to push the model towards directions that would also hurt the source task.

7.7 Conclusion

We have presented the co-natural gradient, a technique that regularizes the optimization trajectory of models trained in a continual setting. We have shown that the co-natural gradient stands on its own as an efficient approach for overcoming catastrophic forgetting, and that it effectively complements and stabilizes other existing techniques at a minimal cost. We believe that the co-natural gradient — and more generally, trajectory regularization — can serve as a solid bedrock for building agents that learn without forgetting.

Chapter 8

Conclusion

In this thesis, we have explored ways to mitigate the difficulty of learning neural models for natural language processing when confronted with changes in the data distribution. We presented an overview of the problem and the current state of the literature, and made contributions along three core axes: evaluation, robustness and adaptation.

Because benchmarks and evaluation metrics are our compass towards successfully addressing distributional shift, in Part I we first developed a new dataset (MTNT, Chapter 3) for evaluating the ability of machine translation models (usually trained on news articles or parliamentary proceedings) to handle a shift to the social media domain. This confirmed that common neural architectures suffer significant performance drops when confronted with text data from social media sources. In Chapter 4, we tackled the distinct, but equally important issue of evaluating adversarial perturbation on textual inputs, with a focus on sequence-to-sequence models. Specifically, we proposed an evaluation framework which we validated against human evaluations, and showed its benefits for the purpose of carrying out effective adversarial training strategies. Collecting datasets and designing evaluation metrics is a time-consuming, often thankless and ultimately sisyphean endeavour: new models inevitably "overfit" to the available benchmarks on which they are evaluated, which necessitates creating new datasets, new metrics *etc*... But as our work illustrates, every contribution helps: for example, our MTNT dataset helped assess the robustness of modern MT models, and was used as the basis for two subsequent robust machine translation shared task, helping to drive innovation in that area.

In Part II we focused on the central problem of training models that are robust to distributional shifts. Our approach took inspiration from existing work in distributionally robust optimization, wherein models are trained to minimize their expected risk against the most difficult data dis-

tribution within a pre-determined family (the "uncertainty set"). In our work we confronted the difficulty of defining this uncertainty set appropriately for practical applications. Our approach (called Parametric DRO) revolved around modeling the uncertainty set as a parametric family of generative models (Chapter 5) or likelihood ratios (Chapter 6) to strike a better balance between the extreme pessimism or the limited expressivity of existing alternatives in the literature. In P-DRO, a robust classifier learns to minimize its expected loss under a parametric adversary which is itself trained to model the sub-population on which the classifier performs the worst. We empirically demonstrated that both variants could be used to derive models that were significantly more robust to sub-population shifts compared with relevant baselines. While the work presented in these two chapters is more recent, we hope that it can serve as a strong foundation towards more distributionally robust models in NLP.

Finally, due to the inherent limitations of our DRO approach, which anticipates distributional shifts using only a static dataset, in Part III we turned to the problem of *adapting* existing models to new data distributions or tasks. We used ideas from information geometry to derive a gradient update rule (dubbed the "co-natural gradient") for mitigating the catastrophic forgetting effect, where adapting neural models to new tasks or domains causes significant drops in performance in previously learned tasks (Chapter 7). The effectiveness of this method was validated on a diverse selection of continual-learning benchmarks, as well as a machine translation domain adaptation problem using our previously proposed MTNT dataset as a test bed.

Beyond the conclusions directly derived from our contributions described above, we identify two higher level takeaways from our time pursuing research on the specific topic of distributional shift. The first is that *proper evaluation is of paramount importance*. This idea has permeated every chapter of this thesis. Of course, this was the central motivation for the entirety of Part I, but we have also made a conscious effort to conduct fair and consistent evaluation in chapters 5, 6 and 7. This is particularly important in the specialised sub-areas that are distributionally robust optimization and continual learning, because due to their cross-sectional nature, they tend to attract practitioners with diverse research backgrounds. This often results in inconsistent evaluation standards, which can can hamper progress (Gulrajani & Lopez-Paz, 2020; Farquhar & Gal, 2018).

Our second key takeaway is that, for practical applications of machine learning (such as NLP), *research progress should be guided by, but not limited to what is predicted by theory*. This is especially relevant in the context of deep learning, where theory lags behind the state of the art (Nagarajan & Kolter, 2019)[1]. In our work on Parametric DRO in Part II, we experimented with highly non-convex-concave differentiable games which were not guaranteed to even converge. Further-

[1] And of course, advancing theory is a worthwhile and crucial pursuit in and of itself.

more, we used a number of (sometimes extreme) approximations to make our methods useable in practical NLP scenarios. Despite this, we found that they outperformed more principled, but perhaps less flexible baselines from the non-parametric DRO literature. This is not to say that research should proceed by blindly throwing ideas on the metaphorical wall and "seeing what sticks" (an inefficient, not to mention computationally unsustainable process). But there is a balance to strike, and we can achieve significant progress by taking inspiration from more conservative lines of work and porting them over to more applied research areas, without being restrained by mathematical rigour.

Future Work

During our investigations throughout the course of producing the work described in this thesis, we have come to identify open questions which we believe would provide fertile ground for future research. In the next few paragraphs, we elaborate on these possible future directions towards close the gap between human and machine learning in NLP in the face of distributional shift

In our own work, we have mostly been concerned with "interpolative" versions of distributional robustness, wherein we assume that our training data contains enough diversity for us to simulate potential distributional shifts. This setting has its uses, as demonstrated by our positive results in a variety of realistic benchmarks in Part 2.3. But there is value in considering a more difficult scenario where we might have to extrapolate and guess at what unseen domains might look like. While there is some existing work in generating difficult or out-of-domain examples in computer vision (Yue et al., 2019; Zhou et al., 2020), this remains an under-explored area of the literature, especially when it comes to natural language processing. Leveraging the generative capabilities of strong parametric adversaries trained with P-DRO to hallucinate unseen, yet difficult examples could be a viable path to progress in this challenging setting.

With respect to the adaptation aspect, we believe that it is time for the community to move beyond the episodic setting commonly considered in continual learning and start tackling the "streaming" setting where the test (or training!) distribution is liable to change continuously over time. This scenario is more representative of how we as humans interact with our non-stationary environment, and it poses exciting new challenges, not only terms of the learning algorithms needed in such environments, but also with regards to how these same algorithms should be evaluated. How do "episodic" continual learning algorithm such as the co-natural gradient (Chapter 7) fare against continuous distribution shift? And can the variety of data distributions observed over time in turn guide the construction of more accurate uncertainty sets in DRO? These are only some of the many

open questions in this so far relatively uncharted research area. While there has been preliminary work on this general topic very recently (Lazaridou et al., 2021; Hombaiah et al., 2021), there is still a long road ahead, as it were.

Finally, as large models pre-trained with a language modeling objective have become ever more prominent in NLP research over the past few years (Radford et al., 2019; Dai et al., 2019; Raffel et al., 2019), the distinction between domain and task has become blurrier. For example, in Radford et al. (2019) and Raffel et al. (2019), multiple tasks are reformulated as a single, unified language modeling or question answering task. We hypothesize that this trend is only going to increase because, as its name indicates, natural language is a natural communication protocol for us to interact with and query our models. With this in mind, we posit that creating models that are both robust and able to adapt to shift in the data distribution is a critical requirement for developing artificial agents that are capable of tackling a variety of tasks in diverse environments. We hope that the work presented in this thesis towards training models that are better equipped to deal with distributional shift provides a useful step in this direction.

Bibliography

Ahn, H., Lee, D., Cha, S., and Moon, T. Uncertainty-based continual learning with adaptive regularization. In *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Aljundi, R., Caccia, L., Belilovsky, E., Caccia, M., Charlin, L., and Tuytelaars, T. Online continual learning with maximally interfered retrieval. In *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2019a.

Aljundi, R., Lin, M., Goujaud, B., and Bengio, Y. Gradient based sample selection for online continual learning. In *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2019b.

Alzantot, M., Sharma, Y., Elgohary, A., Ho, B.-J., Srivastava, M., and Chang, K.-W. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2890–2896, 2018.

Amari, S.-i. Neural learning in structured parameter spaces-natural riemannian gradient. In *Proceedings of the 9th Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 127–133, 1997.

Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G., Chen, J., Chen, J., Chen, Z., Chrzanowski, M., Coates, A., Diamos, G., Ding, K., Du, N., Elsen, E., Engel, J., Fang, W., Fan, L., Fougner, C., Gao, L., Gong, C., Hannun, A., Han, T., Johannes, L., Jiang, B., Ju, C., Jun, B., LeGresley, P., Lin, L., Liu, J., Liu, Y., Li, W., Li, X., Ma, D., Narang, S., Ng, A., Ozair, S., Peng, Y., Prenger, R., Qian, S., Quan, Z., Raiman, J., Rao, V., Satheesh, S., Seetapun, D., Sengupta, S., Srinet, K., Sriram, A., Tang, H., Tang, L., Wang, C., Wang, J., Wang, K., Wang, Y., Wang, Z., Wang, Z., Wu, S., Wei, L., Xiao, B., Xie, W., Xie, Y., Yogatama, D., Yuan, B., Zhan, J., and Zhu,

Z. Deep speech 2 : End-to-end speech recognition in english and mandarin. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pp. 173–182, 2016. URL http://proceedings.mlr.press/v48/amodei16.html.

Andreas, J., Rabinovich, M., Jordan, M. I., and Klein, D. On the accuracy of self-normalized log-linear models. In *Proceedings of the 29th Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 1783–1791, 2015.

Axelrod, A., He, X., and Gao, J. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 355–362, 2011.

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*, 2014.

Balduzzi, D., Racaniere, S., Martens, J., Foerster, J., Tuyls, K., and Graepel, T. The mechanics of n-player differentiable games. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 354–363, 2018. URL http://proceedings.mlr.press/v80/balduzzi18a/balduzzi18a.pdf.

Baldwin, T., Cook, P., Lui, M., MacKinlay, A., and Wang, L. How noisy social media text, how diffrnt social media sources? In *Proceedings of the 6th International Joint Conference on Natural Language Processing (IJCNLP)*, pp. 356–364, 2013. URL http://www.aclweb.org/anthology/I13-1041.

Bapna, A. and Firat, O. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

Barrault, L., Biesialska, M., Bojar, O., Costa-jussà, M. R., Federmann, C., Graham, Y., Grundkiewicz, R., Haddow, B., Huck, M., Joanis, E., Kocmi, T., Koehn, P., Lo, C.-k., Ljubešić, N., Monz, C., Morishita, M., Nagata, M., Nakazawa, T., Pal, S., Post, M., and Zampieri, M. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the 5th Conference on Machine Translation (WMT)*, 2020.

Belinkov, Y. and Bisk, Y. Synthetic and natural noise both break neural machine translation. *International Conference on Learning Representations*, 2018a.

Belinkov, Y. and Bisk, Y. Synthetic and natural noise both break neural machine translation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018b.

Ben-David, S., Blitzer, J., Crammer, K., Pereira, F., et al. Analysis of representations for domain adaptation. *Proceedings of the 20th Annual Conference on Neural Information Processing Systems (NIPS)*, 2007.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

Ben-Tal, A., Den Hertog, D., De Waegenaere, A., Melenberg, B., and Rennen, G. Robust solutions of optimization problems affected by uncertain probabilities. *Management Science*, 59(2):341–357, 2013.

Bentivogli, L., Clark, P., Dagan, I., and Giampiccolo, D. The fifth pascal recognizing textual entailment challenge. In *TAC*, 2009.

Bérard, A., Calapodescu, I., and Roux, C. Naver labs europe's systems for the wmt19 machine translation robustness task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pp. 526–532, 2019.

Bickel, S., Brückner, M., and Scheffer, T. Discriminative learning under covariate shift. *Journal of Machine Learning Research*, 10(Sep):2137–2155, 2009.

Blank, A. and Koch, P. *Historical semantics and cognition*, volume 13. Walter de Gruyter, 2013.

Blitzer, J. and Pereira, F. Domain adaptation of natural language processing systems. *University of Pennsylvania*, pp. 1–106, 2007.

Blodgett, S. L., Green, L., and O'Connor, B. Demographic dialectal variation in social media: A case study of African-American English. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1119–1130, 2016. URL `https://www.aclweb.org/anthology/D16-1120`.

Blodgett, S. L., Wei, J., and O'Connor, B. A dataset and classifier for recognizing social media english. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pp. 56–61, 2017.

Bloomfield, L. *Language*. Allen & Unwin, 1933.

Bojar, O., Chatterjee, R., Federmann, C., Haddow, B., Huck, M., Hokamp, C., Koehn, P., Logacheva, V., Monz, C., Negri, M., Post, M., Scarton, C., Specia, L., and Turchi, M. Findings of the 2015 workshop on statistical machine translation. In *Proceedings of the 10th Workshop on Statistical Machine Translation (WMT)*, pp. 1–46, 2015.

Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

Boucher, N., Shumailov, I., Anderson, R., and Papernot, N. Bad characters: Imperceptible nlp attacks. *arXiv preprint arXiv:2106.09898*, 2021.

Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., and Specia, L. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 1–14, 2017.

Cettolo, M., Girardi, C., and Federico, M. Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pp. 261–268, 2012.

Cettolo, M., Niehues, J., Stüker, S., Bentivogli, L., Cattoni, R., and Federico, M. The iwslt 2016 evaluation campaign. In *Proceedings of the 2016 International Workshop on Spoken Language Translation (IWSLT)*, 2016.

Chaudhry, A., Dokania, P. K., Ajanthan, T., and Torr, P. H. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the 16th European Conference on Computer Vision (ECCV)*, pp. 532–547, 2018a.

Chaudhry, A., Ranzato, M., Rohrbach, M., and Elhoseiny, M. Efficient lifelong learning with a-gem. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018b.

Chaudhry, A., Rohrbach, M., Elhoseiny, M., Ajanthan, T., Dokania, P. K., Torr, P. H., and Ranzato, M. Continual learning with tiny episodic memories. *arXiv preprint arXiv:1902.10486*, 2019.

Chen, Y., Diethe, T., and Lawrence, N. Facilitating bayesian continual learning by natural gradients and stein gradients. In *Continual Learning Workshop@ NeurIPS*, 2018.

Cheng, M., Yi, J., Zhang, H., Chen, P.-Y., and Hsieh, C.-J. Seq2sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples. *arXiv preprint arXiv:1803.01128*, 2018a.

Cheng, Y., Xu, W., He, Z., He, W., Wu, H., Sun, M., and Liu, Y. Semi-supervised learning for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1965–1974, 2016.

Cheng, Y., Tu, Z., Meng, F., Zhai, J., and Liu, Y. Towards robust neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018b.

Chu, C., Dabre, R., and Kurohashi, S. An empirical comparison of domain adaptation methods for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 385–391, 2017.

Cissé, M., Bojanowski, P., Grave, E., Dauphin, Y., and Usunier, N. Parseval networks: Improving robustness to adversarial examples. In *International Conference on Machine Learning*, 2017.

Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M., and Toutanova, K. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 2924–2936, 2019.

Cortes, C., Mansour, Y., and Mohri, M. Learning bounds for importance weighting. In *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems (NIPS)*, volume 10, pp. 442–450. Citeseer, 2010.

Cover, T. M. *Elements of information theory*. John Wiley & Sons, 1999.

Crystal, D. *Language and the Internet*. Cambridge University Press, 2001.

Curi, S., Levy, K. Y., Jegelka, S., and Krause, A. Adaptive sampling for stochastic risk-averse learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Proceedings of the 34th Annual Conference on Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 1036–1047. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/0b6ace9e8971cf36f1782aa982a708db-Paper.pdf.

Dagan, I., Glickman, O., and Magnini, B. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pp. 177–190. Springer, 2005.

Dai, Z., Yang, Z., Yang, Y., Cohen, W. W., Carbonell, J., Le, Q. V., and Salakhutdinov, R. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.

Danet, B. and Herring, S. *The Multilingual Internet: Language, Culture, and Communication Online*. Oxford University Press., New York, 2007.

Davidson, T., Warmsley, D., Macy, M., and Weber, I. Automated hate speech detection and the problem of offensive language. In *Proceedings of the 11th International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2017.

de Masson d'Autume, C., Ruder, S., Kong, L., and Yogatama, D. Episodic memory in lifelong language learning. In *Advances in Neural Information Processing Systems*, pp. 13122–13131, 2019.

Delage, E. and Ye, Y. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations research*, 58(3):595–612, 2010.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *Proceedings of the 22nd IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, 2009.

Denkowski, M. and Lavie, A. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*, 2014.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2018.

Dhar, P., Singh, R. V., Peng, K.-C., Wu, Z., and Chellappa, R. Learning without memorizing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5138–5146, 2019.

Dixon, L., Li, J., Sorensen, J., Thain, N., and Vasserman, L. Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 67–73, 2018.

Dolan, W. B. and Brockett, C. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the The 3rd International Workshop on Paraphrasing (IWP)*, 2005. URL `http://aclweb.org/anthology/I05-5002`.

Duchi, J. and Namkoong, H. Learning models with uniform performance via distributionally robust optimization. *arXiv preprint arXiv:1810.08750*, 2018. URL `https://arxiv.org/pdf/1810.08750.pdf`.

Duchi, J., Glynn, P., and Namkoong, H. Statistics of robust optimization: A generalized empirical likelihood approach. *arXiv preprint arXiv:1610.03425*, 2016.

Duchi, J., Hashimoto, T., and Namkoong, H. Distributionally robust losses for latent covariate mixtures. *arXiv preprint arXiv:2007.13982*, 2020.

Duchi, J. C., Hashimoto, T., and Namkoong, H. Distributionally robust losses against mixture covariate shifts. *Under revision in Operations Research*, 2019.

Dudley, R. M. *Real analysis and probability*. CRC Press, 2018.

Dupačová, J. The minimax approach to stochastic programming and an illustrative application. *Stochastics: An International Journal of Probability and Stochastic Processes*, 20(1):73–88, 1987.

Dupoux, E. Cognitive science in the era of artificial intelligence: A roadmap for reverse-engineering the infant language-learner. *Cognition*, 173:43–59, 2018.

Dvijotham, K., Gowal, S., Stanforth, R., Arandjelovic, R., O'Donoghue, B., Uesato, J., and Kohli, P. Training verified learners with learned verifiers. *arXiv preprint arXiv:1805.10265*, 2018.

Ebrahimi, J., Lowd, D., and Dou, D. On adversarial examples for character-level neural machine translation. In *COLING*, 2018a.

Ebrahimi, J., Rao, A., Lowd, D., and Dou, D. Hotflip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 31–36, 2018b.

Eisenstein, J. What to do about bad language on the internet. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 359–369, 2013. URL http://www.aclweb.org/anthology/N13-1037.

Elsahar, H. and Gallé, M. To annotate or not? predicting performance drop under domain shift. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2163–2173, 2019.

Esfahani, P. M. and Kuhn, D. Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations. *Mathematical Programming*, 171 (1):115–166, 2018.

Fan, Y., Lyu, S., Ying, Y., and Hu, B.-G. Learning with average top-k loss. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 497–505, 2017.

Farquhar, S. and Gal, Y. Towards robust evaluations of continual learning. *arXiv preprint arXiv:1805.09733*, 2018.

Faury, L., Tanielian, U., Dohmatob, E., Smirnova, E., and Vasile, F. Distributionally robust counterfactual risk minimization. In *Proceedings of the 34th Meeting of the Association for Advancement of Artificial Intelligence (AAAI)*, volume 34, pp. 3850–3857, 2020.

Formiga, L. and Fonollosa, J. A. R. Dealing with input noise in statistical machine translation. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING)*, pp. 319–328, 2012. URL http://www.aclweb.org/anthology/C12-2032.

Fortuna, P. and Nunes, S. A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)*, 51(4):1–30, 2018.

Founta, A.-M., Djouvas, C., Chatzakou, D., Leontiadis, I., Blackburn, J., Stringhini, G., Vakali, A., Sirivianos, M., and Kourtellis, N. Large scale crowdsourcing and characterization of twitter abusive behavior. In *Proceedings of the 12th International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2018.

Ganin, Y. and Lempitsky, V. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pp. 1180–1189. PMLR, 07–09 Jul 2015. URL http://proceedings.mlr.press/v37/ganin15.pdf.

Gao, R. and Kleywegt, A. J. Distributionally robust stochastic optimization with wasserstein distance. *arXiv preprint arXiv:1604.02199*, 2016.

Giampiccolo, D., Magnini, B., Dagan, I., and Dolan, B. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pp. 1–9. Association for Computational Linguistics, 2007.

Glorot, X., Bordes, A., and Bengio, Y. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, 2011.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Proceedings of the 28th Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 2672–2680, 2014a. URL `http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf`.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014b.

Goto, I., Chow, K.-P., Lu, B., Sumita, E., and T'sou, B. K.-Y. Overview of the patent machine translation task at the ntcir-10 workshop. In *NII Test Collection for IR Systems*, 2013.

Goyal, K., Dyer, C., and Berg-Kirkpatrick, T. An empirical investigation of global and local normalization for recurrent neural sequence models using a continuous relaxation to beam search. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 1724–1733, 2019.

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Greensmith, E., Bartlett, P. L., and Baxter, J. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5(Nov):1471–1530, 2004.

Grossberg, S. T. *Studies of Mind and Brain: Neural Principles of Learning, Perception, Development, Cognition, and Motor Control*, volume 70. Springer Science & Business Media, 1982.

Grosse, K., Papernot, N., Manoharan, P., Backes, M., and McDaniel, P. Adversarial perturbations against deep neural networks for malware classification. *arXiv preprint arXiv:1606.04435*, 2016.

Grzega, J. and Schoener, M. English and general historical lexicology. *Eichstätt-Ingolstadt: Katholische Universität*, 2007.

Gulli, A. Ag's corpus of news articles. `http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html`, 2005.

Gulrajani, I. and Lopez-Paz, D. In search of lost domain generalization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

Gururangan, S., Swayamdipta, S., Levy, O., Schwartz, R., Bowman, S., and Smith, N. A. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 107–112, 2018.

Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., and Smith, N. A. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8342–8360, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.740. URL `https://www.aclweb.org/anthology/2020.acl-main.740`.

Haim, R. B., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B., and Szpektor, I. The second pascal recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, 2006.

Hand, D. J. Classifier technology and the illusion of progress. *Statistical science*, 21(1):1–14, 2006.

Hashimoto, T., Srivastava, M., Namkoong, H., and Liang, P. Fairness without demographics in repeated loss minimization. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 1929–1938. PMLR, 2018.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Heafield, K., Pouzyrevsky, I., Clark, J. H., and Koehn, P. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 690–696, 2013. URL `https://kheafield.com/papers/edinburgh/estimate_paper.pdf`.

Helcl, J., Libovickỳ, J., and Popel, M. Cuni system for the wmt19 robustness task. In *Proceedings of the 4th Conference on Machine Translation (WMT)*, 2019.

Hern, A. Facebook translates 'good morning' into 'attack them', leading to arrest. https://www.theguardian.com/technology/2017/oct/24/facebook-palestine-israel-translates-good-morning-attack-them-arrest, 2017. URL https://www.theguardian.com/technology/2017/oct/24/facebook-palestine-israel-translates-good-morning-attack-them-arrest. Accessed: 2020-04-17.

Herring, S. C. Media and language change: Introduction. *Journal of Historical Pragmatics*, 4(1): 1–17, 2003.

Hershey, J. and Olsen, P. Approximating the kullback leibler divergence between gaussian mixture models. *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 4:IV–317–IV–320, 2007.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Hombaiah, S. A., Chen, T., Zhang, M., Bendersky, M., and Najork, M. Dynamic language models for continuously evolving content. *arXiv preprint arXiv:2106.06297*, 2021.

Hosseini, H., Kannan, S., Zhang, B., and Poovendran, R. Deceiving google's perspective api built for detecting toxic comments. *arXiv preprint arXiv:1702.08138*, 2017.

Hovy, D. and Søgaard, A. Tagging performance correlates with author age. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 483–488, 2015. URL https://www.aclweb.org/anthology/P15-2079.

Hu, W., Niu, G., Sato, I., and Sugiyama, M. Does distributionally robust supervised learning give robust classifiers? In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 2029–2037, 2018. URL http://proceedings.mlr.press/v80/hu18a/hu18a.pdf.

Hu, Z. and Hong, L. J. Kullback-leibler divergence constrained distributionally robust optimization. *Available at Optimization Online*, 2013.

Huang, S., Li, Z., Qu, L., and Pan, L. On robustness of neural semantic parsers. In *Proceedings of the 16th European Chapter of the Association for Computational Linguistics (EACL)*, pp. 3333–3342, 2021.

Husain, H. Distributional robustness with ipms and links to regularization and gans. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

Isabelle, P., Cherry, C., and Foster, G. A challenge set approach to evaluating machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2486–2496, 2017. URL https://www.aclweb.org/anthology/D17-1263.

Iyer, S., Dandekar, N., and Csernai, K. First quora dataset release: Question pairs. https://www.quora.com/q/quoradata/First-Quora-Dataset-Release-Question-Pairs, 2017.

Iyyer, M., Manjunatha, V., Boyd-Graber, J., and Daumé III, H. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1681–1691, 2015.

Iyyer, M., Wieting, J., Gimpel, K., and Zettlemoyer, L. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1875–1885, 2018.

Jia, R. and Liang, P. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2021–2031, 2017.

Jia, R., Raghunathan, A., Göksel, K., and Liang, P. Certified robustness to adversarial word substitutions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2019.

Kalchbrenner, N. and Blunsom, P. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1700–1709, 2013.

Karpukhin, V., Levy, O., Eisenstein, J., and Ghazvininejad, M. Training on synthetic noise improves robustness to natural noise in machine translation. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT)*, pp. 42–47, 2019.

Kashyap, A. R., Hazarika, D., Kan, M.-Y., and Zimmermann, R. Domain divergences: A survey and empirical analysis. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2021.

Katz, G., Barrett, C., Dill, D. L., Julian, K., and Kochenderfer, M. J. Reluplex: An efficient smt solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, pp. 97–117. Springer, 2017.

Kelly, M. G., Hand, D. J., and Adams, N. M. The impact of changing populations on classifier performance. In *Proceedings of the 5th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)*, 1999.

Khayrallah, H. and Koehn, P. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation (WNMT)*, pp. 74–83, 2018.

Kifer, D., Ben-David, S., and Gehrke, J. Detecting change in data streams. In *Proceedings of the 30th International Conference on Very Large Data Bases*, 2004.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 177–180, 2007.

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., et al. Wilds: A benchmark of in-the-wild distribution shifts. *arXiv preprint arXiv:2012.07421*, 2020.

Kolter, J. Z. and Wong, E. Provable defenses against adversarial examples via the convex outer adversarial polytope. *arXiv preprint arXiv:1711.00851*, 2017.

Kong, L., Schneider, N., Swayamdipta, S., Bhatia, A., Dyer, C., and Smith, N. A. A dependency parser for tweets. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1001–1012, 2014. URL `http://www.aclweb.org/anthology/D14-1108`.

Kullback, S. and Leibler, R. A. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

Kutuzov, A., Øvrelid, L., Szymanski, T., and Velldal, E. Diachronic word embeddings and semantic shifts: a survey. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, 2018.

Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. Human-level concept learning through probabilistic program induction. *Science*, 350:1332–1338, 2015.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 260–270, 2016.

Lazaridou, A., Kuncoro, A., Gribovskaya, E., Agrawal, D., Liska, A., Terzi, T., Gimenez, M., d'Autume, C. d. M., Ruder, S., Yogatama, D., et al. Pitfalls of static language modelling. *arXiv preprint arXiv:2102.01951*, 2021.

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., Van Kleef, P., Auer, S., et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web – Interoperability, Usability, Applicability*, 6(2):167–195, 2015.

Levy, D., Carmon, Y., Duchi, J. C., and Sidford, A. Large-scale methods for distributionally robust optimization. *Proceedings of the 34th Annual Conference on Neural Information Processing Systems (NeurIPS)*, 33, 2020.

Li, M., Zhao, Y., Zhang, D., and Zhou, M. Adaptive development data selection for log-linear model in statistical machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, pp. 662–670, 2010.

Li, X., Michel, P., Anastasopoulos, A., Belinkov, Y., Durrani, N., Firat, O., Koehn, P., Neubig, G., Pino, J., and Sajjad, H. Findings of the first shared task on machine translation robustness. In *Proceedings of the 4th Conference on Machine Translation (WMT)*, pp. 91–102, 2019. URL https://www.aclweb.org/anthology/W19-5303.

Li, Z. and Hoiem, D. Learning without forgetting. *Proceedings of the 14th European Conference on Computer Vision (ECCV)*, 2016.

Lipton, Z. C., Wang, Y.-X., and Smola, A. Detecting and correcting for label shift with black box predictors. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, December 2015.

Lopez-Paz, D. and Ranzato, M. Gradient episodic memory for continual learning. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 6467–6476, 2017.

Lui, M. and Baldwin, T. langid.py: An off-the-shelf language identification tool. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 25–30, 2012. URL http://www.aclweb.org/anthology/P12-3005.

Luong, M.-T. and Manning, C. D. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the 2015 International Workshop on Spoken Language Translation (IWSLT)*, 2015.

Luong, T., Pham, H., and Manning, C. D. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1412–1421, 2015.

Ma, X. and Cieri, C. Corpus support for machine translation at ldc. In *Language Resources and Evaluation Conference*, 2006.

McAuley, J. and Leskovec, J. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys)*, pp. 165–172, 2013.

McCloskey, M. and Cohen, N. J. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pp. 109–165. Elsevier, 1989.

McCoy, T., Pavlick, E., and Linzen, T. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 3428–3448, 2019.

McCulloch, G. *Because internet: Understanding the new rules of language*. Riverhead Books, 2020.

McCusker, L. X., Gough, P. B., and Bias, R. G. Word recognition inside out and outside in. *Journal of Experimental Psychology: Human Perception and Performance*, 7(3):538, 1981.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

Merity, S., Keskar, N. S., and Socher, R. Regularizing and optimizing lstm language models. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

Miceli Barone, A. V., Haddow, B., Germann, U., and Sennrich, R. Regularization techniques for fine-tuning in neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1490–1495, 2017.

Michel, P. and Neubig, G. MTNT: A testbed for machine translation of noisy text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 543–553, 2018a.

Michel, P. and Neubig, G. Extreme adaptation for personalized neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018b. URL `https://www.aclweb.org/anthology/P18-2050.pdf`.

Michel, P., Levy, O., and Neubig, G. Are sixteen heads really better than one? In *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2019a.

Michel, P., Li, X., Neubig, G., and Pino, J. On evaluation of adversarial perturbations for sequence-to-sequence models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 3103–3114, 2019b. URL `https://www.aclweb.org/anthology/N19-1314`.

Michel, P., Hashimoto, T., and Neubig, G. Modeling the second player in distributionally robust optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Mikolov, T., Karafiát, M., Burget, L., Černockỳ, J., and Khudanpur, S. Recurrent neural network based language model. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association (InterSpeech)*, 2010.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2013.

Moore, R. C. and Lewis, W. Intelligent selection of language model training data. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 220–224, 2010.

Moosavi-Dezfooli, S.-M., Fawzi, A., and Frossard, P. Deepfool: A simple and accurate method to fool deep neural networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2574–2582, 2016.

Nagarajan, V. and Kolter, J. Z. Uniform convergence may be unable to explain generalization in deep learning. *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Naik, A., Ravichander, A., Sadeh, N., Rose, C., and Neubig, G. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 2340–2353, 2018.

Neubig, G. The Kyoto free translation task. http://www.phontron.com/kftt, 2011.

Neubig, G., Nakata, Y., and Mori, S. Pointwise prediction for robust, adaptable japanese morphological analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 529–533, 2011.

Neubig, G., Dyer, C., Goldberg, Y., Matthews, A., Ammar, W., Anastasopoulos, A., Ballesteros, M., Chiang, D., Clothiaux, D., Cohn, T., et al. Dynet: The dynamic neural network toolkit. *arXiv preprint arXiv:1701.03980*, 2017.

Neubig, G., Sperber, M., Wang, X., Felix, M., Matthews, A., Padmanabhan, S., Qi, Y., Sachan, D. S., Arthur, P., Godard, P., Hewitt, J., Riad, R., and Wang, L. XNMT: The extensible neural machine translation toolkit. In *Conference of the Association for Machine Translation in the Americas (AMTA) Open Source Software Showcase*, 2018.

Nguyen, C. V., Li, Y., Bui, T. D., and Turner, R. E. Variational continual learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

Nguyen, V. A., Si, N., and Blanchet, J. Robust bayesian classification using an optimistic score ratio. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

Norouzi, M., Bengio, S., Jaitly, N., Schuster, M., Wu, Y., Schuurmans, D., et al. Reward augmented maximum likelihood for neural structured prediction. In *Proceedings of the 30th Annual Conference on Neural Information Processing Systems (NIPS)*, volume 29, pp. 1723–1731, 2016.

Oren, Y., Sagawa, S., Hashimoto, T., and Liang, P. Distributionally robust language modeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4227–4237, 2019. URL https://www.aclweb.org/anthology/D19-1432.

Osborne, M. J. and Rubinstein, A. *A Course in Game Theory*. The MIT Press, 1994. ISBN 0262150417.

Owoputi, O., O'Connor, B., Dyer, C., Gimpel, K., Schneider, N., and Smith, N. A. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 380–390, 2013. URL https://www.aclweb.org/anthology/N13-1039.

Papernot, N., McDaniel, P., Swami, A., and Harang, R. Crafting adversarial input sequences for recurrent neural networks. In *Military Communications Conference, MILCOM 2016-2016 IEEE*, pp. 49–54. IEEE, 2016.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 311–318, 2002.

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. Continual lifelong learning with neural networks: A review. *Neural Networks*, 2019.

Pascanu, R. and Bengio, Y. Revisiting natural gradient for deep networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2013.

Peterson, K. Openmt12 evaluation, 2011.

Popel, M. and Bojar, O. Training tips for the transformer model. *arXiv preprint arXiv:1804.00247*, 2018.

Popović, M. chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pp. 392–395, 2015.

Popović, M. chrf deconstructed: beta parameters and n-gram weights. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pp. 499–504, 2016. doi: 10.18653/v1/W16-2341.

Post, M. A call for clarity in reporting BLEU scores. In *Proceedings of the 3rd Conference on Machine Translation (WMT)*, pp. 186–191, 2018a.

Post, M. A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*, 2018b.

Press, O. and Wolf, L. Using the output embedding to improve language models. In *Proceedings of the 15th European Chapter of the Association for Computational Linguistics (EACL)*, pp. 157–163, 2017.

Pryzant, R., Chung, Y., Jurafsky, D., and Britz, D. Jesc: Japanese-english subtitle corpus. *ArXiv e-prints*, 2017.

Quiñonero-Candela, J., Sugiyama, M., Lawrence, N. D., and Schwaighofer, A. *Dataset shift in machine learning*. Mit Press, 2009.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. Improving language understanding with unsupervised learning. Technical report, Technical report, OpenAI, 2018.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog*, 1:8, 2019.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019. URL https://arxiv.org/pdf/1910.10683.pdf.

Raghunathan, A., Steinhardt, J., and Liang, P. Certified defenses against adversarial examples. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

Rahimian, H. and Mehrotra, S. Distributionally Robust Optimization: A Review. *arXiv preprint arXiv:1908.05659*, 2019.

Rahman, J. The n word: Its history and use in the african american community. *Journal of English Linguistics*, 40(2):137–171, 2012.

Rai, P., Saha, A., Daumé III, H., and Venkatasubramanian, S. Domain adaptation meets active learning. In *Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing*, 2010.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2383–2392, 2016.

Rajpurkar, P., Jia, R., and Liang, P. Squad 2.0 benchmark leaderboard. https://rajpurkar.github.io/SQuAD-explorer/, 2020. URL https://rajpurkar.github.io/SQuAD-explorer/. Accessed: 2020-04-16.

Ratcliff, R. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*, 97(2):285, 1990.

Rebuffi, S.-A., Kolesnikov, A., Sperl, G., and Lampert, C. H. icarl: Incremental classifier and representation learning. In *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2001–2010, 2017.

Reddy, S. and Knight, K. Obfuscating gender in social media writing. In *Proceedings of the First Workshop on NLP and Computational Social Science*. Association for Computational Linguistics, 2016.

Ribeiro, M. T., Singh, S., and Guestrin, C. Semantically equivalent adversarial rules for debugging nlp models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 856–865, 2018.

Ritter, A., Clark, S., Etzioni, O., et al. Named entity recognition in tweets: an experimental study. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1524–1534. Association for Computational Linguistics, 2011.

Rockafellar, R. T., Uryasev, S., et al. Optimization of conditional value-at-risk. *Journal of risk*, 2: 21–42, 2000.

Rudolph, M. and Blei, D. Dynamic embeddings for language evolution. In *Proceedings of the 27th International Conference on World Wide Web (WWW)*, 2018.

Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., and Hadsell, R. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020. URL `https://arxiv.org/pdf/1911.08731.pdf`.

Sakaguchi, K., Duh, K., Post, M., and Van Durme, B. Robsut wrod reocginiton via semi-character recurrent neural network. In *Proceedings of the 31st Meeting of the Association for Advancement of Artificial Intelligence (AAAI)*, pp. 3281–3287, 2017.

Saks, S. Theory of the integral monografie matematyczne. *Vol. VII., Warszawa-Lwow*, 1937.

Samanta, S. and Mehta, S. Towards crafting text adversarial samples. *arXiv preprint arXiv:1707.02812*, 2017.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Sap, M., Card, D., Gabriel, S., Choi, Y., and Smith, N. A. The risk of racial bias in hate speech detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1668–1678, 2019. URL `https://www.aclweb.org/anthology/P19-1163`.

Scarf, H. E. *A Min-Max Solution of an Inventory Problem*. RAND Corporation, Santa Monica, CA, 1957.

Schmidt, A. and Wiegand, M. A survey on hate speech detection using natural language processing. In *Proceedings of the 5th International Workshop on Natural Language Processing for Social Media (SocialNLP)*, pp. 1–10, 2017. URL https://www.aclweb.org/anthology/W17-1101.

Schwarz, J., Czarnecki, W., Luketina, J., Grabska-Barwinska, A., Teh, Y. W., Pascanu, R., and Hadsell, R. Progress & compress: A scalable framework for continual learning. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 4535–4544, 2018.

Sennrich, R. How grammatical is character-level neural machine translation? assessing mt quality with contrastive translation pairs. In *Proceedings of the 15th European Chapter of the Association for Computational Linguistics (EACL)*, pp. 376–382, 2017. URL http://www.aclweb.org/anthology/E17-2060.

Sennrich, R., Haddow, B., and Birch, A. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1715–1725, 2016. URL http://www.aclweb.org/anthology/P16-1162.

Shi, Z., Zhang, H., Chang, K.-W., Huang, M., and Hsieh, C.-J. Robustness verification for transformers. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

Shimodaira, H. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

Singh, S. P., Kearns, M. J., and Mansour, Y. Nash convergence of gradient dynamics in general-sum games. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, UAI '00, pp. 541–548, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. ISBN 1558607099.

Sinha, A., Namkoong, H., and Duchi, J. Certifying some distributional robustness with principled adversarial training. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the*

2013 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1631–1642, 2013.

Specia, L., Li, Z., Pino, J., Chaudhary, V., Guzmán, F., Neubig, G., Durrani, N., Belinkov, Y., Koehn, P., Sajjad, H., et al. Findings of the wmt 2020 shared task on machine translation robustness. In *Proceedings of the 5th Conference on Machine Translation (WMT)*, 2020.

Sperber, M., Niehues, J., and Waibel, A. Toward robust neural machine translation for noisy input sequences. In *Proceedings of the 2017 International Workshop on Spoken Language Translation (IWSLT)*, pp. 90–96, 2017. URL `http://workshop2017.iwslt.org/downloads/iwslt2017_proceeding_v2.pdf`.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Storkey, A. When training and test sets are different: characterizing learning transfer. *Dataset shift in machine learning*, pp. 3–28, 2009.

Sugiyama, M. and Müller, K.-R. Input-dependent estimation of generalization error under covariate shift. *Statistics & Decisions*, 23(4/2005):249–279, 2005.

Sundermeyer, M., Schlüter, R., and Ney, H. Lstm neural networks for language modeling. In *Proceedings of the 13th Annual Conference of the International Speech Communication Association (InterSpeech)*, 2012.

Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks. In *Proceedings of the 28th Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 3104–3112, 2014.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016.

Tan, C., Niculae, V., Danescu-Niculescu-Mizil, C., and Lee, L. Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions. In *Proceedings of the 25th International Conference on World Wide Web (WWW)*, pp. 613–624. International World Wide Web Conferences Steering Committee, 2016.

Thompson, B., Gwinnup, J., Khayrallah, H., Duh, K., and Koehn, P. 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 (NAACL-HLT)*, 2019.

Tseran, H., Khan, M. E., Harada, T., and Bui, T. D. Natural variational continual learning. In *Continual Learning Workshop@ NeurIPS*, 2018.

Utama, P. A., Moosavi, N., and Gurevych, I. Towards debiasing nlu models from unknown biases. *arXiv preprint arXiv:2009.12303*, 2020.

Vaibhav, V., Singh, S., Stewart, C., and Neubig, G. Improving robustness of machine translation with synthetic noise. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2019.

Van Erven, T. and Harremos, P. Rényi divergence and kullback-leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014.

Van Oord, A., Kalchbrenner, N., and Kavukcuoglu, K. Pixel recurrent neural networks. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pp. 1747–1756. PMLR, 2016.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 5998–6008, 2017.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. Matching networks for one shot learning. In *Proceedings of the 30th Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 3630–3638, 2016.

Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Pro-*

ceedings of the 33rd Annual Conference on Neural Information Processing Systems (NeurIPS), pp. 3261–3275, 2019a.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019b.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. Glue benchmark leaderboard. https://gluebenchmark.com/leaderboard, 2020a. URL `https://gluebenchmark.com/leaderboard`. Accessed: 2020-04-16.

Wang, C., Pino, J., Wu, A., and Gu, J. Covost: A diverse multilingual speech-to-text translation corpus. pp. 4197–4203, 2020b.

Wang, R., Utiyama, M., Liu, L., Chen, K., and Sumita, E. Instance weighting for neural machine translation domain adaptation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1483–1489, 2017.

Wang, Z., She, Q., and Ward, T. E. Generative adversarial networks in computer vision: A survey and taxonomy. *arXiv preprint arXiv:1906.01529*, 2019c.

Warstadt, A., Singh, A., and Bowman, S. R. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*, 2018.

Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., and Perona, P. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

Widmer, G. and Kubat, M. Learning in the presence of concept drift and hidden contexts. *Machine learning*, 23(1):69–101, 1996.

Williams, A., Nangia, N., and Bowman, S. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 1112–1122, 2018.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., and Brew, J. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv preprint arXiv:1910.03771*, 2019.

Wong, E. and Kolter, Z. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pp. 5286–5295. PMLR, 2018.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

Xia, M., Field, A., and Tsvetkov, Y. Demoting racial bias in hate speech detection. In *Proceedings of the 9th International Workshop on Natural Language Processing for Social Media (SocialNLP)*, pp. 7–14, 2020. URL https://www.aclweb.org/anthology/2020.socialnlp-1.2.

Xiong, W., Wu, L., Alleva, F., Droppo, J., Huang, X., and Stolcke, A. The microsoft 2017 conversational speech recognition system. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 5934–5938, 2018.

Yelp. Yelp dataset challenge. https://www.yelp.com/dataset/challenge, 2015.

Yue, X., Zhang, Y., Zhao, S., Sangiovanni-Vincentelli, A., Keutzer, K., and Gong, B. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Zenke, F., Poole, B., and Ganguli, S. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pp. 3987–3995, 2017.

Zhang, J. and Zong, C. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1535–1545, 2016.

Zhang, K., Schölkopf, B., Muandet, K., and Wang, Z. Domain adaptation under target and conditional shift. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pp. 819–827, 2013.

Zhang, X., Zhao, J., and LeCun, Y. Character-level convolutional networks for text classification. In *Proceedings of the 29th Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 649–657, 2015.

Zhao, Z., Dua, D., and Singh, S. Generating natural adversarial examples. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

Zhou, C., Ma, X., Michel, P., and Neubig, G. Examining and combating spurious features under distribution shift. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

Zhou, K., Yang, Y., Hospedales, T., and Xiang, T. Deep domain-adversarial image generation for domain generalisation. In *Proceedings of the 34th Meeting of the Association for Advancement of Artificial Intelligence (AAAI)*, 2020.

Zou, W., Huang, S., Xie, J., Dai, X., and Chen, J. A reinforced generation of adversarial examples for neural machine translation. In *Proceedings of the 8th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 3486–3497, 2020.

Appendix

Appendix A

Supplemental Material for Chapter 5

A.1 Reorganizing the Lagrangian $\mathbb{L}(\psi, \tau)$

Let us write the Lagrangian $\mathbb{L}$ explicitly:

$$\mathbb{L}(\psi, \tau) = \mathbb{E}_{(x,y)\sim q_\psi} \frac{p(x,y)}{q_{\psi_0}(x,y)} \ell(x,y,\theta) - \tau \left(\mathrm{KL}(q_\psi \| p) - \kappa\right) \tag{A.1}$$

$$= \mathbb{E}_{(x,y)\sim q_\psi} \frac{p(x,y)}{q_{\psi_0}(x,y)} \ell(x,y,\theta) - \tau\, \mathbb{E}_{(x,y)\sim q_\psi} \log \frac{q_\psi(x,y)}{p(x,y)} + \tau\kappa \tag{A.2}$$

$$= \tau\, \mathbb{E}_{(x,y)\sim q_\psi} \log \left(\frac{p(x,y) e^{\frac{p(x,y)}{q_{\psi_0}(x,y)} \frac{\ell(x,y,\theta)}{\tau}}}{q_\psi(x,y)} \right) + \tau\kappa \tag{A.3}$$

$$= \tau(\kappa - \mathrm{KL}(q_\psi \| q^*_{\tau,\theta})) + \log \left(\mathbb{E}_{(x,y)\sim p}\, e^{\frac{p(x,y)}{q_{\psi_0}(x,y)} \frac{\ell(x,y,\theta)}{\tau}} \right) \tag{A.4}$$

This last step requires that the log moment generating function of ℓ under p exist for τ. In most scenarios we consider, ℓ is typically the negative log likelihood of a neural network model, which is generally bounded. Therefore the moment generating function is defined everywhere.

Note that the KL term is the only one dependent on ψ, therefore maximizing $\mathbb{L}$ for ψ is equivalent to maximizing $-\mathrm{KL}(q_\psi \| q^*_{\tau,\theta})$, in other words minimizing $\mathrm{KL}(q_\psi \| q^*_{\tau,\theta})$

Appendix B

Supplemental Material for Chapter 7

B.1 The Standard Gradient Update (Equation 7.3)

We derive the standard update in Eq. 7.3 by solving the Lagrangian $\mathbb{L}$ in Eq. 7.2 for δ. Given that its first and second derivatives with respect to δ are:

$$\nabla\mathbb{L} = \nabla\mathcal{L}_T + 2\mu\delta$$
$$\nabla^2\mathbb{L} = 2\mu I$$

the problem is trivially strictly convex and its global minimizer δ^* satisfies:

$$\nabla\mathbb{L}\Big|_{\delta^*} = 0 \iff \delta^* = -\frac{1}{2\mu}\nabla\mathcal{L}_T$$

□

B.2 Equivalence of the Hessian of the KL Divergence and the Fisher Information Matrix

To simplify notation, let us perform the change of variables $\theta+\delta \to x$. We show that the Hessian of the KL coincides with the Fisher on θ: in other words, $\nabla^2\mathrm{KL}(p_\theta\|p_x)\Big|_{x=\theta} = \mathbb{E}_{p_\theta}[(\nabla \log p_\theta)(\nabla \log p_\theta)^\intercal]$. Under mild regularity assumptions[1], we can write (all derivatives are taken with respect to variable

[1]Essentially allowing us to interchange derivatives and integrals.

x):

$$\begin{aligned}\nabla^2 \mathrm{KL}(p_\theta \| p_x) &= \underbrace{\nabla^2 \mathbb{E}_{p_\theta}[\log p_\theta]}_{=0} - \nabla^2 \mathbb{E}_{p_\theta}[\log p_x] \\ &= -\mathbb{E}_{p_\theta}[\nabla^2 \log p_x]\end{aligned}$$

Now note that $\nabla^2 \log p_x$ can be rewritten via standard derivatives manipulations as $\frac{\nabla^2 p_x}{p_x} - \frac{(\nabla p_x)(\nabla p_x)^\intercal}{p_x^2}$. This leads to:

$$\nabla^2 \mathrm{KL}(p_\theta \| p_x) = -\mathbb{E}_{p_\theta}\left[\frac{\nabla^2 p_x}{p_x}\right] + \mathbb{E}_{p_\theta}\left[\left(\frac{\nabla p_x}{p_x}\right)\left(\frac{\nabla p_x}{p_x}\right)^\intercal\right]$$

When taken at θ, the first term evaluates to[2] :

$$\begin{aligned}\mathbb{E}_{p_\theta}\left[\frac{\nabla^2 p_x}{p_x}\right]\Big|_{x=\theta} &= \left[\int p_\theta(z)\frac{\nabla^2 p_x(z)}{p_x(z)}dz\right]\Big|_{x=\theta} \\ &= \int p_\theta(z)\frac{\nabla^2 p_\theta(z)}{p_\theta(z)}dz \\ &= \int \nabla^2 p_\theta(z)dz \\ &= \nabla^2 \underbrace{\int p_\theta(z)dz}_{=1} = 0\end{aligned}$$

By using the identity $\frac{\nabla p_x}{p_x} = \nabla \log p_x$ and evaluating at $x = \theta$, the second term gives us:

$$\nabla^2 \mathrm{KL}(p_\theta \| p_x)\Big|_{x=\theta} = \mathbb{E}_{p_\theta}[(\nabla \log p_\theta)(\nabla \log p_\theta)^\intercal]$$

□

B.3 Obtaining the Co-natural Update (Equation 7.7)

We solve the Lagrangian from Eq. 7.6 in a similar fashion as in B.1. First we compute its gradient and Hessian with respect to δ:

[2]We abuse notation and write $\nabla p_x\big|_{x=\theta}$ as ∇p_θ

$$\begin{aligned}\nabla\mathbb{L} &= \nabla\mathcal{L}_T + 2\mu\delta + \nu F_\theta^S \delta \\ &= \nabla\mathcal{L}_T + (\nu F_\theta^S + 2\mu I)\delta \\ \nabla^2\mathbb{L} &= (\nu F_\theta^S + 2\mu I)\end{aligned}$$

While not as straightforwardly as the one in B.1, this problem is also strongly convex: indeed F_θ^S is positive semi-definite (as an expectation of PSD matrices) and the addition of μI ensures that $\nabla^2\mathbb{L}$ is positive definite. We find the unique solution by solving:

$$\begin{aligned}\nabla\mathbb{L}\Big|_{\delta^*} = 0 &\iff \nabla\mathcal{L}_T + (\nu F_\theta^S + 2\mu I)\delta^* = 0 \\ &\iff \delta^* = -[\nu F_\theta^S + 2\mu I]^{-1}\nabla\mathcal{L}_T\end{aligned}$$

Set $\lambda := \frac{1}{\nu}$ and $\alpha := \frac{\mu}{\nu}$ to get Eq. 7.7 $\square$