Towards Data-Efficient Machine Learning

Qizhe Xie

July 2020
CMU-ML-20-107

Machine Learning Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Eduard Hovy (Chair), Carnegie Mellon University
Tom Mitchell, Carnegie Mellon University
Ruslan Salakhutdinov, Carnegie Mellon University
Quoc Le, Google Brain

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

This material is based upon work supported by DARPA/AFRL FA87501220342 and PricewaterhouseCoopers. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Defense Advanced Research Projects Agency, Air Force Research Laboratory, or PricewaterhouseCoopers.

Keywords: Semi-supervised Learning, Transfer Learning, Deep Learning

Abstract

Deep learning works well when the problem is regular enough and there is abundant training data to adequately and in a representative way reflect all the regularity. As the ambition of researchers grows, problems with less regularity are being addressed, where more data is needed to achieve great performance. In addition, as researchers push the boundary of deep learning, state-of-the-art models become more and more data-hungry due to the growing capacity. Hence, data annotation is necessary to train deep learning models to perform well. However, data annotation is a costly process that requires a significant amount of work for each new task of interest.

To tackle this difficulty, we present algorithms that can leverage other kinds of information to achieve a better performance given a certain amount of data. In this thesis, we show how to leverage several kinds of information including: (1) unlabeled data; (2) data from another domain; (3) prior knowledge. First, when unlabeled data of the domain of interest is available, semi-supervised learning can effectively improve the performance of deep learning models by regularizing the models to make consistent predictions for similar examples; Second, when data from another domain is available, transfer learning or domain adaptation can be applied to transfer general knowledge or task-specific knowledge learned from another domain to the domain of interest; Last, with prior knowledge, we can inject targeted inductive biases into the models and make use of external knowledge bases.

With three possible directions, one might wonder what direction should be taken given a new task. To offer practical suggestions to researchers and practitioners, we analyze the effectiveness, the applicability and the engineering difficulty of each algorithm. Specifically, we present the performance of different algorithms on different problems and study whether different algorithms can be combined together for improved performance, analyze whether an algorithm can be applied to a broad range of tasks or is restricted to certain tasks and discuss the required engineering efforts for each algorithm.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

In recent years, the rise of deep learning has led to significant breakthroughs in a wide variety of domains. While these advancements are largely credited to the improved capacity of novel neural architectures, the amount of training data plays an equally critical role. Specifically, the early success of deep learning often relies on the construction of large-scale labeled datasets, such as ImageNet that contains millions of images for classification, WMT with millions of paralleled sentences for machine translation or a speech recognition dataset reaching over 10,000 hours of audio. In addition, the success of deep learning motivates researchers to tackle more and more challenging problems such as question answering, reasoning and chatbots. In these difficult problems, training models with a large capacity on a large amount of data is needed to achieve a great performance. For example, the recently released chatbot Meena [1] uses a neural network with 2.6 billion parameters and is trained on public domain social media conversation datasets with 40 billion words. Hence, a large amount of labeled data is typically required for deep learning to work well.

However, data annotation is an expensive process that requires a large amount of work for each new task of interest. For better data efficiency, we present algorithms to leverage several kinds of information that is easier to obtain. We show that deep learning can still work well with limited data and additional easy-to-obtain information. We show methods that leverage (1) unlabeled data by semi-supervised learning; (2) data from another domain by transfer learning; (3) external knowledge that is useful for the task at hand.

1.2 Algorithms for Data-Efficient Learning

1.2.1 Semi-supervised Learning

Among many possible directions, exploiting unlabeled examples via semi-supervised learning [33] is arguably one of the most widely considered directions. Most recent efforts in semi-supervised learning with deep model can be grouped into four categories: (1) graph-based label propagation via graph convolution [126] and graph embeddings [274], (2) modeling prediction target as latent variables [125], (3) consistency training [8, 135, 171] and (4) self-training or co-training [22, 224, 291].

Unsupervised Data Augmentation In Chapter 2, we present a method called unsupervised data augmentation or UDA that makes use of better data augmentation in consistency training. UDA brings substantial improvements across six language and three vision tasks under the same consistency training framework.

In a nutshell, the consistency training methods simply regularize the model prediction to be less sensitive to *small* noise applied to data examples (labeled or unlabeled). In the simplest form, given an observed example x, consistency training methods first create a noised version $\hat{x}$ (e.g. by adding Gaussian noise or dropout), and enforce the two model predictions of x and $\hat{x}$ to be similar. Intuitively, a good model should be invariant to any superficial and small noise that does not change the label of an example. Under this generic framework, methods in this category differ mostly in how the perturbed sample $\hat{x}$ is created, which in turn influences the sample efficiency. Notably, this category of method is extremely simple and can be directly applied to unlabeled samples without changing the model architecture for most problems.

Besides exploiting unlabeled examples, another more direct alternative to alleviating supervision scarcity is to perform data augmentation based on labeled examples. Typically, given a labeled pair (x, y), data augmentation utilizes the prior knowledge of the data domain to construct a transformation which maps the original example x to an augmented example $\hat{x}$ that shares the same label y as the original input. For example, for image classification, flipping or rotating an image can lead to a new image with the same class.

Despite the distinct motivations of smoothness enforcing and data augmentation, the key component in both methods is a noise or perturbation mapping that produces a new example from an original example. In comparison, one desirable property of data augmentation is that it makes sure $\tilde{x}$ shares the same label. On the other hand, consistency training can be directly applied to unlabeled data while data augmentation requires label information. Hence, we investigate the role of noise injection in consistency training and observe that advanced data augmentation methods, specifically those work best in supervised learning [48, 131, 234, 294], also perform well in semi-supervised learning. There is indeed a strong correlation between the performance of data augmentation operations in supervised learning and their performance in consistency training. We substitute the traditional noise injection methods with high quality data augmentation methods in order to improve consistency training. UDA leads to significant improvements on text classification and image classification tasks in low-data regime.

Noisy Student Training for ImageNet After presenting a method that leads to substantial improvements in low-data regime where the amount of training data is small, we turn to ImageNet where we have a large amount of labeled training data. On ImageNet, state-of-the-art (SOTA) vision models are still trained with supervised learning which requires a large corpus of labeled images to work well. By showing the models only labeled images, we limit ourselves from making use of unlabeled images available in much larger quantities to improve accuracy and robustness of SOTA models.

In Chapter 3, we present a method called Noisy Student Training that uses unlabeled images to improve the SOTA ImageNet accuracy. We show that the accuracy gain has an outsized impact on robustness (out-of-distribution generalization). We use a much larger corpus of unlabeled images, where a large fraction of images do not belong to ImageNet training set distribution (i.e., they do not belong to any category in ImageNet). Noisy Student Training has three main steps: (1) train a teacher model on labeled images, (2) use the teacher to generate pseudo labels on unlabeled images, and (3) train a student model on the combination of labeled images and pseudo labeled images. We iterate this algorithm a few times by treating the student as a teacher to relabel the unlabeled data and training a new student.

Noisy Student Training improves self-training and distillation in two ways. First, it makes the student larger than, or at least equal to, the teacher so the student can better learn from a larger dataset. Second, it adds noise to the student so the noised student is forced to learn harder from the pseudo labels. To noise the student, we use input noise such as RandAugment data augmentation [49] and model noise such as dropout [241] and stochastic depth [104] during training.

Using Noisy Student Training, together with 300M unlabeled images, we improve EfficientNet's [250] ImageNet top-1 accuracy to 88.4. This accuracy is 2.0 better than the previous SOTA results which requires 3.5B weakly labeled Instagram images. Not only our method improves standard ImageNet accuracy, it also improves classification robustness on much harder test sets by large margins.

Noisy Student Training for a Complex Reading Comprehension Dataset RACE After studying semi-supervised learning's effectiveness on classification tasks. We are interested in whether semi-supervised learning can lead to performance improvements on more complex tasks such as reasoning and machine comprehension. In Chapter 4, we first present a reading comprehension dataset that contains questions used to evaluate human's reasoning abilities and that requires significantly more reasoning than existing reading comprehension datasets. Then we evaluate the performance of Noisy Student Training on this task. We find that Noisy Student Training leads to significant improvements even for this complex task.

1.2.2 Transfer Learning

Transfer learning can also effectively reduce the need for using a large amount of annotation data by transferring knowledge learned on another task or domain. It improves performance of our model on a target task by using extra data available on a source task. Approaches in this direction can be broadly categorized into two groups where the source task and the target task are either similar or different. First, when the source task and the target task are similar, shared statistical regularities can be better learned on the combined data on two tasks. For example, for the task of sentiment classifications, book review data can be useful for the task of movie review [28, 67, 68, 262]. Second, when the source task and the target task are different, the source task is usually used to pretrain a neural representation to learn task-agnostic knowledge about text, images or videos. Pretraining has given rises to a lot of breakthroughs in natural language processing recently [50, 56, 101, 198, 201, 289].

Transfer Learning by Parameter Sharing between Similar Sub-tasks In Chapter 5, we present ITransF, a method that learns a parameter sharing mechanism for transferring knowledge between sub-tasks for knowledge base completion. Specifically, as different sub-tasks, many relations share common statistical regularities. At the core of ITransF is a sparse attention mechanism, which learns to compose shared concept parameters into relation-specific parameters, leading to a better generalization property. ITransF improves mean rank and Hits@10 on two benchmark datasets on knowledge base completion, over all previous approaches of the same kind. In addition, the parameter sharing is clearly indicated by the learned sparse attention vectors, enabling us to interpret how knowledge transfer is carried out.

Transfer Learning by Domain-invariant Representation Learning In Chapter 6, we present a method that uses adversarial training to learn domain-invariant representation to perform transfer learning between similar domains. Adversarial training has been shown to able to learn an invariant representation across domains [28, 67, 68, 262] and enables classifiers trained on the source domain to be applicable to the target domain. Moment discrepancy regularizations can also effectively remove domain-specific information [28, 298] for the same purpose. By learning language-invariant representations, classifiers trained on the source language can be applied to the target language [39, 281]. In our work, the representation learning process is formulated as an adversarial minimax game. We analyze the optimal equilibrium of such a game and find that it amounts to maximizing the uncertainty of inferring the detrimental factor given the representation while maximizing the certainty of making task-specific predictions. We show that the proposed framework induces an invariant representation, and leads to better generalization evidenced by the improved performance on machine translation.

Transfer Learning by Pretraining Different from the previously mentioned approaches where transfer learning happens between similar tasks, pretraining transfers knowledge from a source task that is different from the target task. The pretraining task is used to learn a task-agnostic representation of words, sentences and images. Pretraining has been widely used in natural language processing. Collobert and Weston [45] demonstrated that word embeddings learned by language modeling can improve the performance significantly on semantic role labeling. Later, the pretraining of word embeddings was simplified and substantially scaled in Word2Vec [166] and Glove [197]. More recently, Dai and Le [50], Devlin et al. [56], Howard and Ruder [101], Peters et al. [198], Radford et al. [201], Yang et al. [289] have shown that pre-training using language modeling and denoising auto-encoding leads to significant improvements on many tasks in the language domain.

In Chapter 7, we first introduce a cloze test dataset CLOTH and then show that a language model pretrained on the 1-Billion-Word corpus, a large scale language modeling corpus, can lead to significant improvements on the CLOTH dataset. Specifically, the pretrained model achieves an accuracy of 70.7, significantly outperforming the model trained only on the CLOTH dataset which has an accuracy of 48.5. This shows that pretraining can leverage a large amount of unlabeled data to learn general knowledge about natural language.

1.2.3 Making use of External Knowledge

Lastly, when we have prior knowledge about the task at hand, we can improve models' performance by creating models that reflect or make use of the prior knowledge. The prior knowledge can be inductive biases that one has about the current task or external world knowledge stored as knowledge bases.

Inductive Biases as External Knowledge In Chapter 8, with the prior knowledge that token-level training signals provides better credit assignments than sentence-level training signals, we present methods that lead to improved performance for text generation by breaking down the sentence-level training signals into token-level signals. Specifically, we use the sentence-level training signal provided by RAML [183] and establish a theoretical equivalence between the token-level counterpart of RAML and the entropy regularized reinforcement learning. Motivated by this connection, we present two sequence prediction algorithms with improved performance.

Knowledge Bases as External Knowledge In Chapter 9, we present a method to incorporate structured knowledge information from knowledge bases to enable the model to understand entities that are usually not well covered in raw text. Prior work typically incorporate prior knowledge in the form of knowledge base embeddings [36, 285] or enhance pretraining with external knowledge bases [312]. However, different downstream tasks usually use different knowledge bases and switching tasks would also require retraining the knowledge base embedding and fine-tuning representation learning model. We present a method that incorporates external structured knowledge from knowledge bases by Graph Convolutional Network.

1.3 A Comprehensive View for Data-Efficient Learning

Given a new task, one might wonder what would be the best strategy to achieve good empirical results. Specifically, what algorithm should one use? With this question in mind, we analyze the effectiveness, the applicability and the engineering difficulty of each algorithm. More specifically, we evaluate different algorithms on different problems, discuss whether they can be applied to a variety of tasks or are restricted to certain tasks and briefly discuss the engineering efforts required for each algorithm.

Effectiveness In terms of effectiveness, we have the following observations:

- Semi-supervised learning is very effective both for natural language processing tasks and computer vision tasks. On image classification tasks, Noisy Student Training achieves 88.4 top-1 accuracy on ImageNet, which is 2.0 percent better than the state-of-the-art model that requires 3.5B weakly labeled Instagram images, as shown in Section 3.3. Noisy Student Training also improves ImageNet-A top-1 accuracy from 61.0 to 83.7 and improves the performance on RACE from 81.7 to 83.7 as shown in Section 4.6. Similarly, UDA leads to significant improvements given limited labeled data. It achieves an error rate of 5.43 on CIFAR-10 with only 250 labeled examples, which is similar to the error rate of 4.2 using 50,000 labeled examples, as shown in Section 2.5.2. On natural language processing tasks, UDA reduces the error rate from 43.27 to 25.23 for IMDb with 20 labeled examples and from 50.80 to 41.35 for Yelp-5 with 2,500 labeled examples as shown in Section 2.5.3.

- Transfer learning from pretraining brings substantial gains on many natural language tasks. For example, using a pretrained language model improves the accuracy from 48.7 to 70.7 on the CLOTH dataset as shown in Section 7.4.1. On text classifications, it reduces the error rate from 43.27 to 11.72 for IMDb with 20 labeled examples and from 50.80 to 38.90 for Yelp-5 with 2,500 labeled examples as shown in Section 2.5.3. There is also a growing interest in studying pretraining for computer vision tasks [92, 259, 302]. Transfer learning between similar tasks brings respectful but smaller improvements: it improves BLEU score from 35.2 to 36.1 for a French-to-English translation task and 27.3 to 28.1 for a German-to-English translation task in Section 6.5.2 and improves the Hits@10 from 79.7 to 81.4 for knowledge base completion in Section 5.3.3.
- Using external knowledge bases is useful for tasks related to entities and relations. Specifically, it improves the F1 score from 53.18 to 57.28 on relation extraction as shown in Section 9.4.3. Injecting prior knowledge into model design improves the BLEU score from 30.90 to 31.44 on image captioning and from 28.04 to 28.30 on German-to-English translation as shown in Section 8.7.3.
- Lastly, different algorithms usually bring complementary benefits and they can be combined to achieve better performance. For example, semi-supervised learning is complementary to transfer learning. When initialized with BERT, semi-supervised learning can still significantly reduce the error rate from 11.72 to 4.78 for IMDb with 20 examples and from 38.90 to 33.54 for Yelp-5 with 2,500 examples as shown in Section 2.5.3. Using external knowledge bases also boost the the F1 score from 53.18 to 57.28 achieved by transfer learning using BERT, as shown in Section 8.7.3.

Applicability Then, we discuss whether an algorithm is applicable to a variety of tasks.

- Semi-supervised learning is widely applicable to many tasks ranging from text tasks to vision tasks. As shown in Section 2.5, Section 3.3 and Section 4.6, semi-supervised learning methods UDA and Noisy Student Training are effective for 7 language datasets and 3 computer vision datasets. Semi-supervised learning only requires using unlabeled data which is usually available at large quantities for real-world applications.
- Transfer learning from pretraining works well for natural language processing tasks. It is used for 10 natural language processing datasets as shown in Section 2.5, Section 7.4.1 and Section 9.4. We do not explore using transfer learning on vision tasks in this thesis but there has been a growing interest in this line of research [37, 89, 92, 259, 302]. An advantage of this approach is that it does not require extra task-specific data that is similar to the task at hand and hence it can be easily used for any tasks. In comparison, it is harder to apply transfer learning from similar tasks since it requires task-specific data though it is also effective for 4 datasets as shown in Section 5.3.3 and Section 6.5.2.
- It is not very straightforward to use external knowledge in many cases. Usually, different tasks require different external knowledge. For example, tasks of the medical domain and legal domain may require the use of completely different knowledge bases. In addition, it

is not clear whether most tasks can benefit from external knowledge.

Engineering Difficulty One might also be interested in the engineering difficulty to apply different algorithms.

- Semi-supervised learning, especially the Noisy Student Training method, requires very little engineering efforts since it does not require changing the underlying model or architecture.
- For natural language processing, transfer learning from pretraining is also easy to use and has become the standard practice nowadays. In comparison, transfer learning from similar tasks is harder since it requires significant changes to the architecture or learning algorithm.
- Using external knowledge also requires significant changes to the architecture or learning algorithm.

Considering the effectiveness, applicability and the engineering difficulties, we have the following recommendations for readers: (1) Semi-supervised learning should be used by default when unlabeled data is available since it leads to large performance improvements to both natural language processing tasks and computer vision tasks, while requiring little engineering; (2) Transfer learning from pretraining models should be used by default for natural language processing since it brings significant improvements and is easy to use; (3) Whether external knowledge should be used should be determined on a case-by-case basis since each task may require different external knowledge and it is significantly more costly in engineering efforts than using transfer learning or semi-supervised learning.

Part I

Data-Efficient Learning by Semi-supervised Learning

Chapter 2

Semi-supervised learning by Unsupervised Data Augmentation

In this Chapter, we present a semi-supervised learning method called Unsupervised Data Augmentation (UDA) that can effectively improve the model's performance on various domains given limited annotated data.

2.1 Introduction

A fundamental weakness of deep learning is that it typically requires a lot of labeled data to work well. Semi-supervised learning (SSL) [33] is one of the most promising paradigms of leveraging unlabeled data to address this weakness. The recent works in SSL are diverse but those that are based on consistency training [8, 135, 205, 251] have shown to work well on many benchmarks.

In a nutshell, consistency training methods simply regularize model predictions to be invariant to small noise applied to either input examples [43, 171, 219] or hidden states [8, 135]. This framework makes sense intuitively because a good model should be robust to any small change in an input example or hidden states. Under this framework, different methods in this category differ mostly in how and where the noise injection is applied. Typical noise injection methods are additive Gaussian noise, dropout noise or adversarial noise.

In this work, we investigate the role of noise injection in consistency training and observe that advanced data augmentation methods, specifically those work best in supervised learning [48, 131, 234, 294], also perform well in semi-supervised learning. There is indeed a strong correlation between the performance of data augmentation operations in supervised learning and their performance in consistency training. We, hence, substitute the traditional noise injection methods with high quality data augmentation methods in order to improve consistency training. To emphasize the use of better data augmentation in consistency training, we name our method Unsupervised Data Augmentation or UDA.

We evaluate UDA on a wide variety of language and vision tasks. On six text classification tasks, our method achieves significant improvements over state-of-the-art models. Notably, on IMDb, UDA with 20 labeled examples outperforms the state-of-the-art model trained on 1250x more labeled data. On standard semi-supervised learning benchmarks CIFAR-10 and SVHN, UDA outperforms all existing semi-supervised learning methods by significant margins and achieves an error rate of 5.4 and 2.72 with 250 labeled examples respectively. Finally, we also find UDA to be beneficial when there is a large amount of supervised data. For instance, on ImageNet, UDA leads to improvements of top-1 accuracy from 58.84 to 68.78 with 10% of the labeled set and from 78.43 to 79.05 when we use the full labeled set and an external dataset with 1.3M unlabeled examples.

Our key contributions and findings can be summarized as follows:

- First, we show that state-of-the-art data augmentations found in supervised learning can also serve as a superior source of noise under the consistency enforcing semi-supervised framework. *See results in Table 2.1 and Table 2.2.*
- Second, we show that UDA can match and even outperform purely supervised learning that uses orders of magnitude more labeled data. *See results in Table 2.6 and Figure 2.4.*
- Third, we show that UDA combines well with transfer learning, e.g., when fine-tuning from BERT (*see Table 2.6*), and is effective at high-data regime, e.g. on ImageNet (*see Table 3.3*).
- Lastly, we also provide a theoretical analysis of how UDA improves the classification performance and the corresponding role of the state-of-the-art augmentation in Section 8.2.

2.2 Unsupervised Data Augmentation (UDA)

In this section, we first formulate our task and then present the key method and insights behind UDA. Throughout this work, we focus on classification problems and will use x to denote the input and y^* to denote its ground-truth prediction target. We are interested in learning a model $p_\theta(y \mid x)$ to predict y^* based on the input x, where θ denotes the model parameters. Finally, we will use $p_L(x)$ and $p_U(x)$ to denote the distributions of labeled and unlabeled examples respectively and use f^* to denote the perfect classifier that we hope to learn.

2.2.1 Background: Supervised Data Augmentation

Data augmentation aims at creating novel and realistic-looking training data by applying a transformation to an example, without changing its label. Formally, let $q(\hat{x} \mid x)$ be the augmentation transformation from which one can draw augmented examples $\hat{x}$ based on an original example x. For an augmentation transformation to be valid, it is required that any example $\hat{x} \sim q(\hat{x} \mid x)$ drawn from the distribution shares the same ground-truth label as x. Given a valid augmentation transformation, we can simply minimize the negative log-likelihood on augmented examples.

Supervised data augmentation can be equivalently seen as constructing an augmented labeled set from the original supervised set and then training the model on the augmented set. Therefore, the augmented set needs to provide additional inductive biases to be more effective. How to design the augmentation transformation has, thus, become critical.

In recent years, there have been significant advancements on the design of data augmentations for NLP [294], vision [48, 131] and speech [84, 190] in supervised settings. Despite the promising results, data augmentation is mostly regarded as the “cherry on the cake” which provides a steady but limited performance boost because these augmentations has so far only been applied to a set of labeled examples which is usually of a small size. Motivated by this limitation, via the consistency training framework, we extend the advancement in supervised data augmentation to semi-supervised learning where abundant unlabeled data is available.

2.2.2 Unsupervised Data Augmentation

As discussed in the introduction, a recent line of work in semi-supervised learning has been utilizing unlabeled examples to enforce smoothness of the model. The general form of these works can be summarized as follows:

- Given an input x, compute the output distribution $p_\theta(y \mid x)$ given x and a noised version $p_\theta(y \mid x, \epsilon)$ by injecting a small noise ϵ. The noise can be applied to x or hidden states.
- Minimize a divergence metric between the two distributions $\mathcal{D}\left(p_\theta(y \mid x) \parallel p_\theta(y \mid x, \epsilon)\right)$.

This procedure enforces the model to be insensitive to the noise ϵ and hence smoother with respect to changes in the input (or hidden) space. From another perspective, minimizing the consistency loss gradually propagates label information from labeled examples to unlabeled ones.

In this work, we are interested in a particular setting where the noise is injected to the input x, i.e., $\hat{x} = q(x, \epsilon)$, as considered by prior works [135, 171, 219]. But different from existing work, we focus on the unattended question of how the form or "quality" of the noising operation q can influence the performance of this consistency training framework. Specifically, to enforce consistency, prior methods generally employ simple noise injection methods such as adding Gaussian noise, simple input augmentations to noise unlabeled examples. In contrast, we hypothesize that stronger data augmentations in supervised learning can also lead to superior performance when used to noise unlabeled examples in the semi-supervised consistency training framework, since it has been shown that more advanced data augmentations that are more diverse and natural can lead to significant performance gain in the supervised setting.

Following this idea, we use a rich set of state-of-the-art data augmentations verified in various supervised settings to inject noise and optimize the same consistency training objective on unlabeled examples. When jointly trained with labeled examples, we utilize a weighting factor λ to balance the supervised cross entropy and the unsupervised consistency training loss, which is illustrated in Figure 2.1. Formally, the full objective can be written as follows:

$$\min_\theta \ \mathcal{J}(\theta) = \mathop{\mathbb{E}}_{x \sim p_L(x)} \left[-\log p_\theta(f^*(x) \mid x)\right] + \lambda \mathop{\mathbb{E}}_{x \sim p_U(x)} \mathop{\mathbb{E}}_{\hat{x} \sim q(\hat{x} \mid x)} \left[\mathrm{CE}\left(p_{\tilde{\theta}}(y \mid x) \| p_\theta(y \mid \hat{x})\right)\right]$$

where CE denotes cross entropy, $q(\hat{x} \mid x)$ is a data augmentation transformation and $\tilde{\theta}$ is a *fixed* copy of the current parameters θ indicating that the gradient is not propagated through $\tilde{\theta}$, as suggested by VAT [171]. We set λ to 1 for most of our experiments and use different batch sizes for the supervised data and the unsupervised data. In the vision domain, simple augmentations including cropping and flipping are applied to labeled examples. To minimize the discrepancy between supervised training and prediction on unlabeled examples, we apply the same simple augmentations to unlabeled examples for computing $p_{\tilde{\theta}}(y \mid x)$.

Discussion. Before detailing the augmentation operations used in this work, we first provide some intuitions on how more advanced data augmentations can provide extra advantages over simple ones used in earlier works from three aspects:

- **Valid noise**: Advanced data augmentation methods that achieve great performance in supervised learning usually generate realistic augmented examples that share the same ground-truth labels with the original example. Thus, it is safe to encourage the consistency between predictions on the original unlabeled example and the augmented unlabeled examples.
- **Diverse noise**: Advanced data augmentation can generate a diverse set of examples since it can

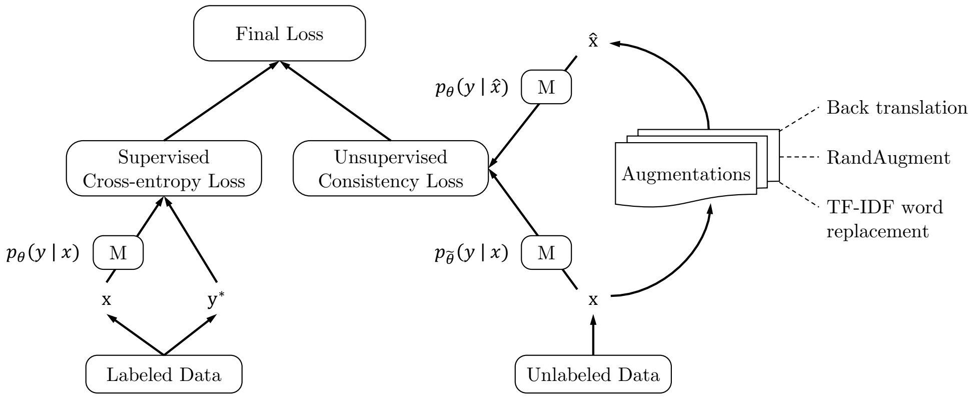


Figure 2.1: Training objective for UDA, where M is a model that predicts a distribution of y given x.

make large modifications to the input example without changing its label, while simple Gaussian noise only make local changes. Encouraging consistency on a diverse set of augmented examples can significantly improve the sample efficiency.

- **Targeted inductive biases**: Different tasks require different inductive biases. Data augmentation operations that work well in supervised training essentially provides the missing inductive biases.

2.2.3 Augmentation Strategies for Different Tasks

We now detail the augmentation methods, tailored for different tasks, that we use in this work.

RandAugment for Image Classification. We use a data augmentation method called RandAugment [49], which is inspired by AutoAugment [48]. AutoAugment uses a search method to combine all image processing transformations in the Python Image Library (PIL) to find a good augmentation strategy. In RandAugment, we do not use search, but instead uniformly sample from the same set of augmentation transformations in PIL. In other words, RandAugment is simpler and requires no labeled data as there is no need to search for optimal policies.

In our implementation of RandAugment, each sub-policy is composed of two operations, where each operation is represented by the transformation name, probability, and magnitude that is specific to that operation. For example, a sub-policy can be [(Sharpness, 0.6, 2), (Posterize, 0.3, 9)].

For each operation, we randomly sample a transformation from 15 possible transformations, a magnitude in $[1, 10)$ and fix the probability to 0.5. Specifically, we sample from the following 15 transformations: Invert, Cutout, Sharpness, AutoContrast, Posterize, ShearX, TranslateX, TranslateY, ShearY, Rotate, Equalize, Contrast, Color, Solarize, Brightness. We find this setting to work well in our first try and did not tune the magnitude range and the probability. Tuning these hyperparameters might result in further gains in accuracy.

Back-translation for Text Classification. When used as an augmentation method, back-translation [60, 225] refers to the procedure of translating an existing example x in language A into another language B and then translating it back into A to obtain an augmented example $\hat{x}$. As observed by [294], back-translation can generate diverse paraphrases while preserving the semantics of the original sentences, leading to significant performance improvements in question answering. In our case, we use back-translation to paraphrase the training data of our text classification tasks.[1]

We find that the diversity of the paraphrases is important. Hence, we employ random sampling with a tunable temperature instead of beam search for the generation. As shown in Figure 2.2, the paraphrases generated by back-translation sentence are diverse and have similar semantic meanings. More specifically, we use WMT'14 English-French translation models (in both directions) to perform back-translation on each sentence. To facilitate future research, we have open-sourced our back-translation system together with the translation checkpoints.

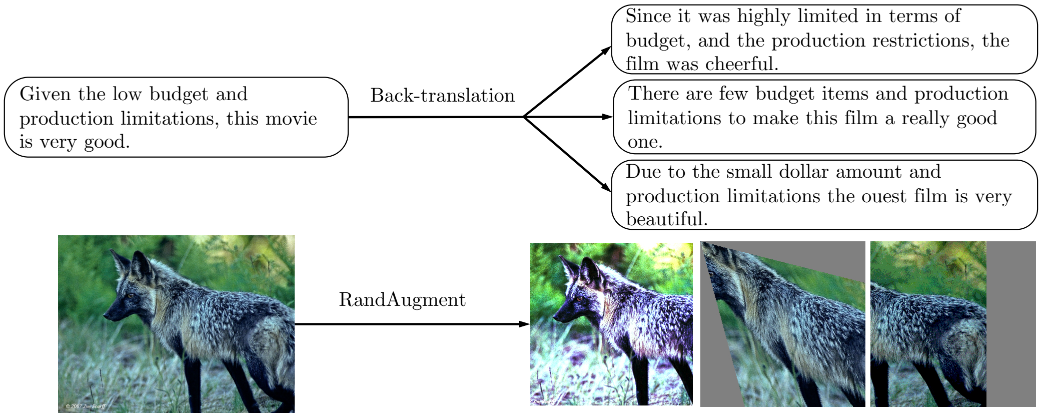

Given the low budget and production limitations, this movie is very good.
Back-translation
Since it was highly limited in terms of budget, and the production restrictions, the film was cheerful.
There are few budget items and production limitations to make this film a really good one.
Due to the small dollar amount and production limitations the ouest film is very beautiful.
RandAugment

Figure 2.2: Augmented examples using back-translation and RandAugment.

Word replacing with TF-IDF for Text Classification. While back-translation is good at maintaining the global semantics of a sentence, there is little control over which words will be retained. This requirement is important for topic classification tasks, such as DBPedia, in which some keywords are more informative than other words in determining the topic. We, therefore, an augmentation method that replaces uninformative words with low TF-IDF scores while keeping those with high TF-IDF values.

Ideally, we would like the augmentation method to generate both diverse and valid examples. Hence, the augmentation is designed to retain keywords and replace uninformative words with other uninformative words. We use BERT's word tokenizer since BERT first tokenizes sentences into a sequence of words and then tokenize words into subwords although the model uses subwords as input.

Specifically, Suppose $\mathrm{IDF}(w)$ is the IDF score for word w computed on the whole corpus, and $\mathrm{TF}(w)$ is the TF score for word w in a sentence. We compute the TF-IDF score as $\mathrm{TFIDF}(w) = \mathrm{TF}(w)\mathrm{IDF}(w)$. Suppose the maximum TF-IDF score in a sentence x is $C = \max_i \mathrm{TFIDF}(x_i)$. To make the probability of having a word replaced to negatively correlate with its TF-IDF score, we set the probability to $\min(p(C - \mathrm{TFIDF}(x_i))/Z, 1)$, where p is a hyperparameter that controls the magnitude of the augmentation and $Z = \sum_i (C - \mathrm{TFIDF}(x_i))/|x|$ is the average score. p is set to 0.7 for experiments on DBPedia.

When a word is replaced, we sample another word from the whole vocabulary for the replacement. Intuitively, the sampled words should not be keywords to prevent changing the ground-truth labels of the sentence. To measure if a word is keyword, we compute a score of each word

[1] We also note that while translation uses a labeled dataset, the translation task itself is quite distinctive from a text classification task and does not make use of any text classification label. In addition, back-translation is a general data augmentation method that can be applied to many tasks with the same model checkpoints.

on the whole corpus. Specifically, we compute the score as $S(w) = \mathrm{freq}(w)\mathrm{IDF}(w)$ where $\mathrm{freq}(w)$ is the frequency of word w on the whole corpus. We set the probability of sampling word w as $(\max_{w'} S(w') - S(w))/Z'$ where $Z' = \sum_w \max_{w'} S(w') - S(w)$ is a normalization term.

Discussion on Trade-off Between Diversity and Validity for Data Augmentation. Despite that state-of-the-art data augmentation methods can generate diverse and valid augmented examples as discussed in section 2.2.2, there is a trade-off between diversity and validity since diversity is achieved by changing a part of the original example, naturally leading to the risk of altering the ground-truth label. We find it beneficial to tune the trade-off between diversity and validity for data augmentation methods. For text classification, we tune the temperature of random sampling. On the one hand, when we use a temperature of 0, decoding by random sampling degenerates into greedy decoding and generates perfectly valid but identical paraphrases. On the other hand, when we use a temperature of 1, random sampling generates very diverse but barely readable paraphrases. We find that setting the Softmax temperature to $0.7, 0.8$ or 0.9 leads to the best performances.

2.2.4 Additional Training Techniques

In this section, we present additional techniques targeting at some commonly encountered problems.

Sharpening Predictions. We find it helpful to mask out examples that the current model is not confident about and to use a low Softmax temperature to sharpen predictions when computing the target distribution on unlabeled examples. Specifically, in each minibatch, the consistency loss term is computed only on examples whose highest probability among classification categories is greater than a threshold.

Domain-relevance Data Filtering. Ideally, we would like to make use of out-of-domain unlabeled data since it is usually much easier to collect, but the class distributions of out-of-domain data are mismatched with those of in-domain data, which can result in performance loss if directly used [184]. To obtain data relevant to the domain for the task at hand, we adopt a common technique for detecting out-of-domain data. We use our baseline model trained on the in-domain data to infer the labels of data in a large out-of-domain dataset and pick out examples that the model is most confident about. Specifically, for each category, we sort all examples based on the classified probabilities of being in that category and select the examples with the highest probabilities.

Training Signal Annealing for Low-data Regime In semi-supervised learning, we often encounter a situation where there is a huge gap between the amount of unlabeled data and that of labeled data. Hence, the model often quickly overfits the limited amount of labeled data while still underfitting the unlabeled data. To tackle this difficulty, we introduce a new training technique, called Training Signal Annealing (TSA), which gradually releases the "training signals" of the labeled examples as training progresses. Intuitively, we only utilize a labeled example if the model's confidence on that example is lower than a predefined threshold which increases according to a schedule. Specifically, at training step t, if the model's predicted probability for the correct category $p_\theta(y^* \mid x)$ is higher than a threshold η_t, we remove that example from the loss function. Suppose K is the number of categories, by gradually increasing η_t from $\frac{1}{K}$ to 1, the threshold η_t serves as a ceiling to prevent over-training on easy labeled examples.

We consider three increasing schedules of η_t with different application scenarios. Let T be the total number of training steps, the three schedules are shown in Figure 2.3. Intuitively, when the model is prone to overfit, e.g., when the problem is relatively easy or the number of labeled examples is very limited, the exp-schedule is most suitable as the supervised signal is mostly released at the end of training. In contrast, when the model is less likely to overfit (e.g., when we have abundant labeled examples or when the model employs effective regularization), the log-schedule can serve well.

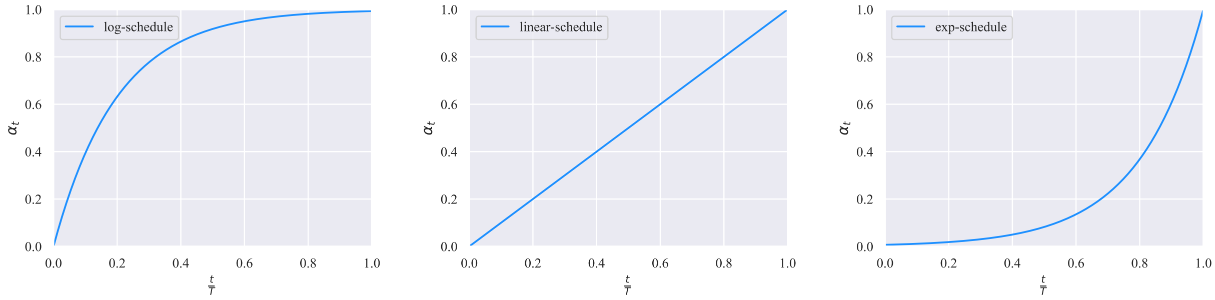


Figure 2.3: Three schedules of TSA. We set $\eta_t = \alpha_t * (1 - \frac{1}{K}) + \frac{1}{K}$. α_t is set to $1 - \exp(-\frac{t}{T} * 5)$, $\frac{t}{T}$ and $\exp((\frac{t}{T} - 1) * 5)$ for the log, linear and exp schedules.

2.3 Theoretical Analysis

In this section, we theoretically analyze why UDA can improve the performance of a model and the required number of labeled examples to achieve a certain error rate. Following previous sections, we will use f^* to denote the perfect classifier that we hope to learn, use p_U to denote the marginal distribution of the unlabeled data and use $q(\hat{x} \mid x)$ to denote the augmentation distribution.

To make the analysis tractable, we make the following simplistic assumptions about the data augmentation transformation:

- **In-domain** augmentation: data examples generated by data augmentation have non-zero probability under p_U, i.e., $p_U(\hat{x}) > 0$ for $\hat{x} \sim q(\hat{x} \mid x), x \sim p_U(x)$.
- **Label-preserving** augmentation: data augmentation preserves the label of the original example, i.e., $f^*(x) = f^*(\hat{x})$ for $\hat{x} \sim q(\hat{x} \mid x), x \sim p_U(x)$.
- **Reversible** augmentation: the data augmentation operation can be reversed, i.e., if $q(\hat{x} \mid x) > 0$ then $q(x \mid \hat{x}) > 0$.

As the first step, we hope to provide an intuitive sketch of our formal analysis. Let us define a graph G_{p_U} where each node corresponds to a data sample $x \in X$ and an edge $(\hat{x}, x)$ exists in the graph *if and only if* $q(\hat{x} \mid x) > 0$. Due to the label-preserving assumption, it is easy to see that examples with different labels must reside on different components (disconnected sub-graphs) of the graph G_{p_U}. Hence, for an N-category classification problems, the graph has N components (sub-graphs) when all examples within each category can be traversed by the augmentation operation. Otherwise, the graph will have more than N components.

Given this construction, notice that for each component C_i of the graph, as long as there is a single labeled example in the component, i.e. $(x^*, y^*) \in C_i$, one can propagate the label of the node to the rest of the nodes in C_i by traversing C_i via the augmentation operation $q(\hat{x} \mid x)$. More importantly, if one only performs *supervised data augmentation*, one can only propagate the label information to the directly connected neighbors of the labeled node. In contrast, performing *unsupervised data augmentation* ensures the traversal of the entire sub-graph C_i. This provides the first high-level intuition how UDA could help.

Taking one step further, in order to find a perfect classifier via such label propagation, it requires that there exists at least one labeled example in each component. In other words, the number of components lower bounds the minimum amount of labeled examples needed to learn a perfect classifier. Importantly, number of components is actually decided by the quality of the augmentation operation: an ideal augmentation should be able to reach all other examples of the same category given a starting instance. This well matches our discussion of the benefits of state-of-the-art data augmentation methods in generating more diverse examples. Effectively, the augmentation diversity leads to more neighbors for each node, and hence reduces the number of components in a graph.

With the intuition described, we state our formal results. Without loss of generality, assume there are k components in the graph. For each component $C_i (i = 1, \ldots, k)$, let P_i be the total probability mass that an observed labeled example fall into the i-th component, i.e., $P_i = \sum_{x \in C_i} p_L(x)$. The following theorem characterizes the relationship between UDA error rate and the amount of labeled examples.

Theorem 1. *Under UDA, let $Pr(\mathcal{A})$ denote the probability that the algorithm cannot infer the label of a new test example given m labeled examples from P_L. $Pr(\mathcal{A})$ is given by*

$$Pr(\mathcal{A}) = \sum_i P_i (1 - P_i)^m.$$

In addition, $O(k/\epsilon)$ labeled examples can guarantee an error rate of $O(\epsilon)$, i.e.,

$$m = O(k/\epsilon) \implies Pr(\mathcal{A}) = O(\epsilon).$$

Proof. Let x' be the sampled test example. Then the probability of event $\mathcal{A}$ is

$$Pr(\mathcal{A}) = \sum_i Pr(\mathcal{A} \text{ and } x' \in C_i) = \sum_i P_i (1 - P_i)^m$$

To bound the probability, we would like to find the maximum value of $\sum_i P_i (1 - P_i)^m$. We can define the following optimization function:

$$\min_P - \sum_{c_i} P_i (1 - P_i)^m$$
$$\text{s.t.} \sum_{c_i} P_i = 1$$

The problem is a convex optimization problem and we can construct its the Lagrangian dual function:

$$\mathcal{L} = \sum_i P_i (1 - P_i)^m - \lambda (\sum_i P_i - 1)$$

Using the KKT condition, we can take derivatives to P_i and set it to zero. Then we have

$$\lambda = (1 - mP_i)(1 - P_i)^{m-1}$$

Hence $P_i = P_j$ for any $i \neq j$. Using the fact that $\sum_i P_i = 1$, we have

$$P_i = \frac{1}{k}$$

Plugging the result back into $Pr(\mathcal{A}) = \sum_i P_i(1 - P_i)^m$, we have

$$Pr(\mathcal{A}) \leq (1 - \frac{1}{k})^m = \exp(m \log(1 - \frac{1}{k})) \leq \exp(-\frac{m}{k})$$

Hence when $m = O(\frac{k}{\epsilon})$, we have

$$Pr(\mathcal{A}) = O(\epsilon)$$

□

From the theorem, we can see the number of components, i.e. k, directly governs the amount of labeled data required to reach a desired performance. As we have discussed above, the number of components effectively relies on the quality of an augmentation function, where better augmentation functions result in fewer components. This echoes our discussion of the benefits of state-of-the-art data augmentation operations in generating more diverse examples. Hence, with state-of-the-art augmentation operations, UDA is able to achieve good performance using fewer labeled examples.

In addition, we can quantify how many unlabeled examples are needed so that we have k connected components and all examples can be classified correctly given a minimum number of labeled examples. For a given component C_i, as long as our sampled unlabeled examples and labeled examples constitute a spanning tree of C_i, we can correctly classify all labeled examples within the component. We can define a vertex cut as a set of nodes/unlabeled examples that the component is not connected after removing the nodes in the set. Let α_{C_i} be the value of the minimum vertex cut (i.e., the minimum, over all vertex cuts of C_i, of the sum of weights on the vertices). For our sample to contain a "spanning tree" of C_i, and therefore to include all of labeled examples in C_i as one component, it must have at least one vertex / unlabeled example in that vertex cut. The expected number of unlabeled examples needed for this to occur is at least $\frac{1}{\alpha_{C_i}}$. Considering all components, at least $\min_H \frac{1}{\alpha_{C_i}}$ unlabeled examples are needed.

2.4 Experiment Details

2.4.1 Text Classifications

Datasets. In our semi-supervised setting, we randomly sampled labeled examples from the full supervised set[2] and use the same number of examples for each category. For unlabeled data, we

[2] `http://bit.ly/2kRWoof`, `https://ai.stanford.edu/~amaas/data/sentiment/`

use the whole training set for DBPedia, the concatenation of the training set and the unlabeled set for IMDb and external data for Yelp-2, Yelp-5, Amazon-2 and Amazon-5 [165][3]. Note that for Yelp and Amazon based datasets, the label distribution of the unlabeled set might not match with that of labeled datasets since there are different number of examples in different categories. Nevertheless, we find it works well to use all the unlabeled data.

Preprocessing. We find the sequence length to be an important factor in achieving good performance. For all text classification datasets, we truncate the input to 512 subwords since BERT is pretrained with a maximum sequence length of 512. Further, when the length of an example is greater than 512, we keep the last 512 subwords instead of the first 512 subwords as keeping the latter part of the sentence lead to better performances on IMDb.

Fine-tuning BERT on in-domain unsupervised data. We fine-tune the BERT model on in-domain unsupervised data using the code released by BERT. We try learning rate of 2e-5, 5e-5 and 1e-4, batch size of 32, 64 and 128 and number of training steps of 30k, 100k and 300k. We pick the fine-tuned models by the BERT loss on a held-out set instead of the performance on a downstream task.

Random initialized Transformer. For the experiments with randomly initialized Transformer, we adopt hyperparameters for BERT base except that we only use 6 hidden layers and 8 attention heads. We also increase the dropout rate on the attention and the hidden states to 0.2, When we train UDA with randomly initialized architectures, we train UDA for 500k or 1M steps on Amazon-5 and Yelp-5 where we have abundant unlabeled data.

BERT hyperparameters. Following the common BERT fine-tuning procedure, we keep a dropout rate of 0.1, and try learning rate of 1e-5, 2e-5 and 5e-5 and batch size of 32 and 128. We also tune the number of steps ranging from 30 to 100k for various data sizes.

UDA hyperparameters. We set the weight on the unsupervised objective λ to 1 in all of our experiments. We use a batch size of 32 for the supervised objective since 32 is the smallest batch size on v3-32 Cloud TPU Pod. We use a batch size of 224 for the unsupervised objective when the Transformer is initialized with BERT so that the model can be trained on more unlabeled data. We find that generating one augmented example for each unlabeled example is enough for $\text{BERT}_{\text{FINETUNE}}$.

All experiments in this part are performed on a v3-32 Cloud TPU Pod.

2.4.2 Semi-supervised learning benchmarks CIFAR-10 and SVHN

Hyperparameters for Wide-ResNet-28-2. We train our model for 500K steps. We apply Exponential Moving Average to the parameters with a decay rate of 0.9999. We use a batch size of 64 for labeled data and a batch size of 448 for unlabeled data. The softmax temperature is set to 0.4. The softmax threshold is set to 0.8. We use a cosine learning rate decay schedule: $\cos(\frac{7t}{8T} * \frac{\pi}{2})$ where t is the current step and T is the total number of steps. We use a SGD optimizer with nesterov momentum with the momentum hyperparameter set to 0.9. In order to reduce training time, we generate augmented examples before training and dump them to disk. For CIFAR-10, we generate 100 augmented examples for each unlabeled example. Note that generating aug-

[3] `https://www.kaggle.com/yelp-dataset/yelp-dataset`, `http://jmcauley.ucsd.edu/data/amazon/`

mented examples in an online fashion is always better or as good as using dumped augmented examples since the model can see different augmented examples in different epochs, leading to more diverse samples. We report the average performance and the standard deviation for 10 runs. Experiments in this part are performed on a Tesla V100 GPU.

Hyperparameters for Shake-Shake and PyramidNet. For the experiments with Shake-Shake, we train UDA for 300k steps and use a batch size of 128 for the supervised objective and use a batch size of 512 for the unsuperivsed objective. For the experiments with PyramidNet+ShakeDrop, we train UDA for 700k steps and use a batch size of 64 for the supervised objective and a batch size of 128 for the unsupervised objective. For both models, we use a learning rate of 0.03 and use a cosine learning decay with one annealing cycle following AutoAugment. Experiments in this part are performed on a v3-32 Cloud TPU v3 Pod.

2.4.3 ImageNet

10% Labeled Set Setting. Unless otherwise stated, we follow the standard hyperparameters used in an open-source implementation of ResNet.[4] For the 10% labeled set setting, we use a batch size of 512 for the supervised objective and a batch size of 15,360 for the unsupervised objective. We use a base learning rate of 0.3 that is decayed by 10 for four times and set the weight on the unsupervised objective λ to 20. We mask out unlabeled examples whose highest probabilities across categories are less than 0.5 and set the Softmax temperature to 0.4. The model is trained for 40k steps. Experiments in this part are performed on a v3-64 Cloud TPU v3 Pod.

Full Labeled Set Setting. For experiments on the full ImageNet, we use a batch size of 8,192 for the supervised objective and a batch size of 16,384 for the unsupervised objective. The weight on the unsupervised objective λ is set to 1. We use entropy minimization to sharpen the prediction. We use a base learning rate of 1.6 and decay it by 10 for four times. Experiments in this part are performed on a v3-128 Cloud TPU v3 Pod.

2.5 Experiments

In this section, we evaluate UDA on a variety of language and vision tasks. For language, we rely on six text classification benchmark datasets, including IMDb, Yelp-2, Yelp-5, Amazon-2 and Amazon-5 sentiment classification and DBPedia topic classification [158, 306]. For vision, we employ two smaller datasets CIFAR-10 [130], SVHN [176], which are often used to compare semi-supervised algorithms, as well as ImageNet [55] of a larger scale to test the scalability of UDA.

2.5.1 Correlation between Supervised and Semi-supervised Performances

As the first step, we try to verify the fundamental idea of UDA, i.e., there is a positive correlation of data augmentation's effectiveness in supervised learning and semi-supervised learning. Based on Yelp-5 (a language task) and CIFAR-10 (a vision task), we compare the performance

[4] https://github.com/tensorflow/tpu/tree/master/models/official/resnet

of different data augmentation methods in either fully supervised or semi-supervised settings. For Yelp-5, apart from back-translation, we include a simpler method Switchout [270] which replaces a token with a random token uniformly sampled from the vocabulary. For CIFAR-10, we compare RandAugment with two simpler methods: (1) cropping & flipping augmentation and (2) Cutout.

Based on this setting, Table 2.1 and Table 2.2 exhibit a strong correlation of an augmentation's effectiveness between supervised and semi-supervised settings. This validates our idea of stronger data augmentations found in supervised learning can always lead to more gains when applied to the semi-supervised learning settings.

Augmentation (# Sup examples)	Sup (50k)	Semi-Sup (4k)
Crop & flip	5.36	10.94
Cutout	4.42	5.43
RandAugment	**4.23**	**4.32**

Table 2.1: Error rates on CIFAR-10.

Augmentation (# Sup examples)	Sup (650k)	Semi-sup (2.5k)
✗	38.36	50.80
Switchout	37.24	43.38
Back-translation	**36.71**	**41.35**

Table 2.2: Error rate on Yelp-5.

2.5.2 Algorithm Comparison on Vision Semi-supervised Learning Benchmarks

With the correlation established above, the next question we ask is how well UDA performs compared to existing semi-supervised learning algorithms. To answer the question, we focus on the most commonly used semi-supervised learning benchmarks CIFAR-10 and SVHN.

Vary the size of labeled data. Firstly, we follow the settings in [184] and employ Wide-ResNet-28-2 [88, 297] as the backbone model and evaluate UDA with varied supervised data sizes.

In Table 2.3, we show results of Pseudo-Label [138], Π-Model [135], Mean Teacher [251], VAT [171] and MixMatch [19]. Fully supervised learning using 50,000 examples achieves an error rate of 4.23 and 5.36 with or without RandAugment. The performance of the baseline models are reported by MixMatch [19]. The comparison with MixMatch and VAT is further plotted in Figure 2.4a.

Methods / # Sup	250	500	1,000	2,000	4,000
Pseudo-Label	49.98 ± 1.17	40.55 ± 1.70	30.91 ± 1.73	21.96 ± 0.42	16.21 ± 0.11
Π-Model	53.02 ± 2.05	41.82 ± 1.52	31.53 ± 0.98	23.07 ± 0.66	17.41 ± 0.37
Mean Teacher	47.32 ± 4.71	42.01 ± 5.86	17.32 ± 4.00	12.17 ± 0.22	10.36 ± 0.25
VAT	36.03 ± 2.82	26.11 ± 1.52	18.68 ± 0.40	14.40 ± 0.15	11.05 ± 0.31
MixMatch	11.08 ± 0.87	9.65 ± 0.94	7.75 ± 0.32	7.03 ± 0.15	6.24 ± 0.06
UDA (RandAugment)	**5.43 ± 0.96**	**4.80 ± 0.09**	**4.75 ± 0.10**	**4.73 ± 0.14**	**4.32 ± 0.08**

Table 2.3: Error rate (%) for CIFAR-10.

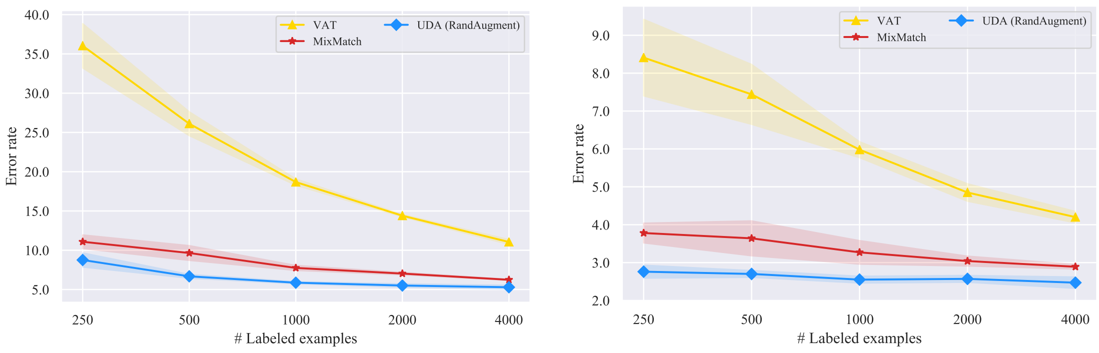
In Table 2.4, we similarly show results for compared methods of Figure 2.4b and results of methods mentioned above. Fully supervised learning using 73,257 examples achieves an error rate of 2.28 and 2.84 with or without RandAugment. The performance of the baseline models are reported by MixMatch [19]. The comparison with MixMatch and VAT is further plotted in Figure 2.4b.

Methods / # Sup	250	500	1,000	2,000	4,000
Pseudo-Label	21.16 ± 0.88	14.35 ± 0.37	10.19 ± 0.41	7.54 ± 0.27	5.71 ± 0.07
Π-Model	17.65 ± 0.27	11.44 ± 0.39	8.60 ± 0.18	6.94 ± 0.27	5.57 ± 0.14
Mean Teacher	6.45 ± 2.43	3.82 ± 0.17	3.75 ± 0.10	3.51 ± 0.09	3.39 ± 0.11
VAT	8.41 ± 1.01	7.44 ± 0.79	5.98 ± 0.21	4.85 ± 0.23	4.20 ± 0.15
MixMatch	3.78 ± 0.26	3.64 ± 0.46	3.27 ± 0.31	3.04 ± 0.13	2.89 ± 0.06
UDA (RandAugment)	**2.72 ± 0.40**	**2.27 ± 0.09**	**2.23 ± 0.07**	**2.20 ± 0.06**	**2.28 ± 0.10**

Table 2.4: Error rate (%) for SVHN.

We have the following observations for the performance on these two task:

- First, UDA consistently outperforms the two baselines given different sizes of labeled data.
- Moreover, the performance difference between UDA and VAT shows the superiority of data augmentation based noise. The difference of UDA and VAT is essentially the noise process. While the noise produced by VAT often contain high-frequency artifacts that do not exist in real images, data augmentation mostly generates diverse and realistic images.



(a) CIFAR-10

(b) SVHN

Figure 2.4: Comparison with two semi-supervised learning methods on CIFAR-10 and SVHN with varied number of labeled examples.

Comparisons with published results on CIFAR-10 and SVHN Here, we directly compare UDA with previously published results under different model architectures. Following previous work, 4k and 1k labeled examples are used for CIFAR-10 and SVHN respectively. As shown in Table 2.5, given the same architecture, UDA outperforms all published results by significant margins and nearly matches the fully supervised performance, which uses 10x more labeled examples. This shows the huge potential of state-of-the-art data augmentations under the consistency training framework in the vision domain.

Method	Model	# Param	CIFAR-10 (4k)	SVHN (1k)
Π-Model [135]	Conv-Large	3.1M	12.36 ± 0.31	4.82 ± 0.17
Mean Teacher [251]	Conv-Large	3.1M	12.31 ± 0.28	3.95 ± 0.19
VAT + EntMin [171]	Conv-Large	3.1M	10.55 ± 0.05	3.86 ± 0.11
SNTG [154]	Conv-Large	3.1M	10.93 ± 0.14	3.86 ± 0.27
VAdD [192]	Conv-Large	3.1M	11.32 ± 0.11	4.16 ± 0.08
Fast-SWA [4]	Conv-Large	3.1M	9.05	-
ICT [266]	Conv-Large	3.1M	7.29 ± 0.02	3.89 ± 0.04
Pseudo-Label [138]	WRN-28-2	1.5M	16.21 ± 0.11	7.62 ± 0.29
LGA + VAT [110]	WRN-28-2	1.5M	12.06 ± 0.19	6.58 ± 0.36
mixmixup [85]	WRN-28-2	1.5M	10	-
ICT [266]	WRN-28-2	1.5M	7.66 ± 0.17	3.53 ± 0.07
MixMatch [19]	WRN-28-2	1.5M	6.24 ± 0.06	2.89 ± 0.06
Mean Teacher [251]	Shake-Shake	26M	6.28 ± 0.15	-
Fast-SWA [4]	Shake-Shake	26M	5.0	-
MixMatch [19]	WRN	26M	4.95 ± 0.08	-
UDA (RandAugment)	WRN-28-2	1.5M	4.32 ± 0.08	**2.23 ± 0.07**
UDA (RandAugment)	Shake-Shake	26M	3.7	-
UDA (RandAugment)	PyramidNet	26M	**2.7**	-

Table 2.5: Comparison between methods using different models where PyramidNet is used with ShakeDrop regularization. On CIFAR-10, with only 4,000 labeled examples, UDA matches the performance of fully supervised Wide-ResNet-28-2 and PyramidNet+ShakeDrop, where they have an error rate of 5.4 and 2.7 respectively when trained on 50,000 examples without RandAugment. On SVHN, UDA also matches the performance of our fully supervised model trained on 73,257 examples without RandAugment, which has an error rate of 2.84.

2.5.3 Evaluation on Text Classification Datasets

Next, we further evaluate UDA in the language domain. Moreover, in order to test whether UDA can be combined with the success of unsupervised representation learning, such as BERT [56], we further consider four initialization schemes: (a) random Transformer; (b) BERT$_{\text{BASE}}$; (c) BERT$_{\text{LARGE}}$; (d) BERT$_{\text{FINETUNE}}$: BERT$_{\text{LARGE}}$ fine-tuned on in-domain unlabeled data[5]. Under each of these four initialization schemes, we compare the performances with and without UDA.

The results are presented in Table 2.6 where we would like to emphasize three observations:

- First, even with very few labeled examples, UDA can offer decent or even competitive performances compared to the SOTA model trained with full supervised data. Particularly, on binary sentiment analysis tasks, with only 20 supervised examples, UDA outperforms the previous SOTA trained with full supervised data on IMDb and is competitive on Yelp-2 and Amazon-2.
- Second, UDA is complementary to transfer learning / representation learning. As we can see, when initialized with BERT and further finetuned on in-domain data, UDA can still significantly reduce the error rate from 6.50 to 4.20 on IMDb.

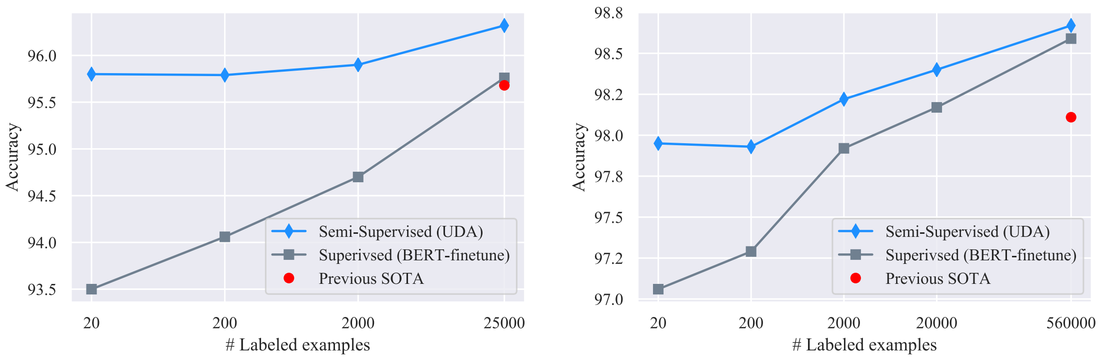
[5] One exception is that we do not pursue BERT$_{\text{FINETUNE}}$ on DBPedia as fine-tuning BERT on DBPedia does not yield further performance gain. This is probably due to the fact that DBPedia is based on Wikipedia while BERT is already trained on the whole Wikipedia corpus.

Fully supervised baseline						
Datasets (# Sup examples)	IMDb (25k)	Yelp-2 (560k)	Yelp-5 (650k)	Amazon-2 (3.6m)	Amazon-5 (3m)	DBpedia (560k)
Pre-BERT SOTA	*4.32*	2.16	29.98	3.32	34.81	0.70
BERT$_{\text{LARGE}}$	4.51	*1.89*	*29.32*	*2.63*	*34.17*	*0.64*

Semi-supervised setting							
Initialization	**UDA**	IMDb (20)	Yelp-2 (20)	Yelp-5 (2.5k)	Amazon-2 (20)	Amazon-5 (2.5k)	DBpedia (140)
Random	✗	43.27	40.25	50.80	45.39	55.70	41.14
	✓	25.23	8.33	41.35	16.16	44.19	7.24
BERT$_{\text{BASE}}$	✗	18.40	13.60	41.00	26.75	44.09	2.58
	✓	5.45	2.61	33.80	3.96	38.40	1.33
BERT$_{\text{LARGE}}$	✗	11.72	10.55	38.90	15.54	42.30	1.68
	✓	4.78	2.50	33.54	3.93	37.80	1.09
BERT$_{\text{FINETUNE}}$	✗	6.50	2.94	32.39	12.17	37.32	-
	✓	**4.20**	**2.05**	**32.08**	**3.50**	**37.12**	-

Table 2.6: Error rates on text classification datasets. In the fully supervised settings, the pre-BERT SOTAs include ULMFiT [101] for Yelp-2 and Yelp-5, DPCNN [115] for Amazon-2 and Amazon-5, Mixed VAT [216] for IMDb and DBPedia. All of our experiments use a sequence length of 512.

- Finally, we also note that for five-category sentiment classification tasks, there still exists a clear gap between UDA with 500 labeled examples per class and BERT trained on the entire supervised set. Intuitively, five-category sentiment classifications are much more difficult than their binary counterparts. This suggests a room for further improvement in the future.



(a) IMDb

(b) Yelp-2

Figure 2.5: Accuracy on IMDb and Yelp-2 with different number of labeled examples. In the large-data regime, with the full training set of IMDb, UDA also provides robust gains.

Experiments on Text Classification with Varied Label Set Sizes We also try different data sizes on text classification tasks. As show in Figure 2.5, UDA leads to consistent improvements across all labeled data sizes on IMDb and Yelp-2.

2.5.4 Scalability Test on the ImageNet Dataset

Then, to evaluate whether UDA can scale to problems with a large scale and a higher difficulty, we now turn to the ImageNet dataset with ResNet-50 being the underlying architecture. Specifically, we consider two experiment settings with different natures:

- We use 10% of the supervised data of ImageNet while using all other data as unlabeled data. As a result, the unlabeled exmaples are entirely in-domain.
- In the second setting, we keep all images in ImageNet as supervised data. Then, we use the domain-relevance data filtering method to filter out 1.3M images from JFT [42, 98]. Hence, the unlabeled set is not necessarily in-domain.

The results are summarized in Table 3.3. In both 10% and the full data settings, UDA consistently brings significant gains compared to the supervised baseline. This shows UDA is not only able to scale but also able to utilize out-of-domain unlabeled examples to improve model performance. In parallel to our work, S4L [302] and CPC [92] also show significant improvements on ImageNet.

Methods	SSL	10%	100%
ResNet-50	✗	55.09 / 77.26	77.28 / 93.73
w. RandAugment		58.84 / 80.56	78.43 / 94.37
UDA (RandAugment)	✓	**68.78 / 88.80**	**79.05 / 94.49**

Table 2.7: Top-1 / top-5 accuracy on ImageNet with 10% and 100% of the labeled set. We use image size 224 and 331 for the 10% and 100% experiments respectively.

2.5.5 Ablation Studies

Ablation Studies for Unlabeled Data Size Here we present an ablation study for unlabeled data sizes. As shown in Table 2.8 and Table 2.9, given the same number of labeled examples, reducing the number of unsupervised examples clearly leads to worse performance. In fact, having abundant unsupervised examples is more important than having more labeled examples since reducing the unlabeled data amount leads to worse performance than reducing the labeled data by the same ratio.

# Unsup / # Sup	250	500	1,000	2,000	4,000
50,000	5.43 ± 0.96	4.80 ± 0.09	4.75 ± 0.10	4.73 ± 0.14	4.32 ± 0.08
20,000	11.01 ± 1.01	9.46 ± 0.14	8.57 ± 0.14	7.65 ± 0.17	7.31 ± 0.24
10,000	23.17 ± 0.71	18.43 ± 0.43	15.46 ± 0.58	12.52 ± 0.13	10.32 ± 0.20
5,000	35.41 ± 0.75	28.35 ± 0.60	22.06 ± 0.71	17.36 ± 0.15	13.19 ± 0.12

Table 2.8: Error rate (%) for CIFAR-10 with different amounts of labeled data and unlabeled data.

# Unsup / # Sup	250	500	1,000	2,000	4,000
73,257	2.72 ± 0.40	2.27 ± 0.09	2.23 ± 0.07	2.20 ± 0.06	2.28 ± 0.10
20,000	5.59 ± 0.74	4.43 ± 0.15	3.81 ± 0.11	3.86 ± 0.14	3.64 ± 0.20
10,000	17.13 ± 12.85	7.59 ± 1.01	5.76 ± 0.29	5.17 ± 0.12	5.40 ± 0.12
5,000	31.58 ± 7.39	12.66 ± 0.81	6.28 ± 0.25	8.35 ± 0.36	7.76 ± 0.28

Table 2.9: Error rate (%) for SVHN with different amounts of labeled data and unlabeled data.

Ablations Studies on RandAugment We hypothesize that the success of RandAugment should be credited to the diversity of the augmentation transformations, since RandAugment works very well for multiple different datasets while it does not require a search algorithm to find out the most effective policies. To verify this hypothesis, we test UDA's performance when we restrict the number of possible transformations used in RandAugment. As shown in Figure 2.6, the performance gradually improves as we use more augmentation transformations.

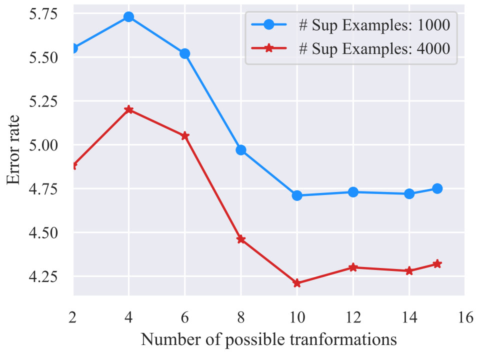


Figure 2.6: Error rate of UDA on CIFAR-10 with different numbers of possible transformations in RandAugment. UDA achieves lower error rate when we increase the number of possible transformations, which demonstrates the importance of a rich set of augmentation transformations.

Ablation Studies for TSA We study the effect of TSA on two tasks with different amounts of unlabeled data on Yelp-5 where we have only 2.5k labeled examples and 6m unlabeled examples.

As shown in Table 2.10, on Yelp-5, where there is a lot more unlabeled data than labeled data, TSA reduces the error rate from 50.81 to 41.35 when compared to the baseline without TSA. More specifically, the best performance is achieved when we choose to postpone releasing the supervised training signal to the end of the training, i.e, exp-schedule leads to the best performance.

Ablation Studies for Data Augmentation on DBPedia Lastly, we study the performance of different augmentations on DBPedia with 140 labeled examples. We initialize our model using BERT$_{\text{LARGE}}$ and compare back-translation and TF-IDF based word replacement. As shown in Table 2.11, using back-translation hurts the performance on DBPedia. We found that keywords are often deleted after applying back-translation, which may lead to the significant performance

TSA schedule	Error rate
✗	50.81
log-schedule	49.06
linear-schedule	45.41
exp-schedule	**41.35**

Table 2.10: Ablation study for Training Signal Annealing (TSA) on Yelp-5. The shown numbers are error rates.

Augmentation	Error rate
✗	1.68
Back Translation	4.49
TF-IDF Word-Based Replacement	1.09

Table 2.11: Ablation study for augmentation on DBPedia.

drop. In comparison, TF-IDF based word replacement can preserve the keywords which has a high TF-IDF score, leading to improved performance.

2.6 Related Work

Semi-supervised Learning. Due to the long history of semi-supervised learning (SSL), we refer readers to [33] for a general review. More recently, many efforts have been made to renovate classic ideas into deep neural instantiations. For example, graph-based label propagation [316] has been extended to neural methods via graph embeddings [274, 287] and later graph convolutions [126]. Similarly, with the variational auto-encoding framework and reinforce algorithm, classic graphical models based SSL methods with target variable being latent can also take advantage of deep architectures [125, 157, 288]. Besides the direct extensions, it was found that training neural classifiers to classify out-of-domain examples into an additional class [221] works very well in practice. Later, Dai et al. [52] shows that this can be seen as an instantiation of low-density separation.

Existing works in consistency training does make use of data augmentation [135, 219]; however, they only apply weak augmentation methods such as random translations and cropping. In parallel to our work, ICT [266] and MixMatch [19] also show improvements for semi-supervised learning. These methods employ mixup [304] on top of simple augmentations such as flipping and cropping; instead, UDA emphasizes on the use of state-of-the-art data augmentations, leading to significantly better results on CIFAR-10 and SVHN. In addition, UDA is also applicable to language domain and can also scale well to more challenging vision datasets, such as ImageNet.

Other works in the consistency training family mostly differ in how the noise is defined: Pseudo-ensemble [8] directly applies Gaussian noise and Dropout noise; VAT [170, 171] defines the noise by approximating the direction of change in the input space that the model is most sensitive to; Cross-view training [43] masks out part of the input data. Apart from enforcing consistency on the input examples and the hidden representations, another line of research enforces consistency on the model parameter space. Works in this category include Mean Teacher [251], fast-Stochastic Weight Averaging [4] and Smooth Neighbors on Teacher Graphs [154].

Apart from enforcing consistency on the noised input examples and the hidden representations, another line of research enforces consistency under different model parameters, which is complementary to our method. For example, Mean Teacher [251] maintains a teacher model with parameters being the ensemble of a student model's parameters and enforces the consistency between the predictions of the two models. Recently, Athiwaratkun et al. [4] present fast-SWA that improves Mean Teacher by encouraging the model to explore a diverse set of plausible parameters. In addition to parameter-level consistency, SNTG [154] also enforces input-level consistency by constructing a similarity graph between unlabeled examples.

Data Augmentation. Also related to our work is the field of data augmentation research. Besides the conventional approaches and two data augmentation methods mentioned in Section 2.2.1, a recent approach MixUp [304] goes beyond data augmentation from a single data point and performs interpolation of data pairs to achieve augmentation. Recently, Hernández-García and König [96] have shown that data augmentation can be regarded as a kind of explicit regularization methods similar to Dropout.

Diverse Back Translation. Diverse paraphrases generated by back-translation has been a key component in the significant performance improvements in our text classification experiments. We use random sampling instead of beam search for decoding similar to the work by Edunov et al. [60]. There are also recent works on generating diverse translations [91, 128, 230] that might lead to further improvements when used as data augmentations.

Unsupervised Representation Learning. Apart from semi-supervised learning, unsupervised representation learning offers another way to utilize unsupervised data. Collobert and Weston [45] demonstrated that word embeddings learned by language modeling can improve the performance significantly on semantic role labeling. Later, the pre-training of word embeddings was simplified and substantially scaled in Word2Vec [166] and Glove [197]. More recently, Dai and Le [50], Devlin et al. [56], Howard and Ruder [101], Peters et al. [198], Radford et al. [201] have shown that pre-training using language modeling and denoising auto-encoding leads to significant improvements on many tasks in the language domain. There is also a growing interest in self-supervised learning for vision [92, 259, 302].

Consistency Training in Other Domains. Similar ideas of consistency training has also been applied in other domains. For example, recently, enforcing adversarial consistency on unsupervised data has also been shown to be helpful in adversarial robustness [30, 175, 242, 301]. Enforcing consistency w.r.t data augmentation has also been shown to work well for representation learning [103, 292]. Invariant representation learning [144, 220] applies the consistency loss not only to the predicted distributions but also to representations and has been shown significant improvements on speech recognition.

2.7 Discussions

In this work, we show that data augmentation and semi-supervised learning are well connected: better data augmentation can lead to significantly better semi-supervised learning. Our method, UDA, employs state-of-the-art data augmentation found in supervised learning to generate diverse and realistic noise and enforces the model to be consistent with respect to these noise. For text, UDA combines well with representation learning, e.g., BERT, and is very effective in low-data regime where state-of-the-art performance is achieved on IMDb with only 20 examples. For vision, UDA outperforms prior work by a clear margin and nearly matches the performance of the fully supervised models trained on the full labeled sets which are one order of magnitude larger. Lastly, UDA can effectively leverage out-of-domain unlabeled data and achieve improved performances on ImageNet where we have a large amount of supervised data.

Chapter 3

Semi-supervised Learning by Noisy Student Training

In this chapter, we present a semi-supervised learning method that leads to great performance gains at high-data regime and achieves the state-of-the-art results on ImageNet. This is the first work that uses unlabeled data to achieve state-of-the-art on ImageNet. Apart from the great performance, this method is very easy to use and does not require changing the underlying architectures.

3.1 Introduction

Deep learning has shown remarkable successes in image recognition in recent years [88, 131, 236, 247, 250]. However state-of-the-art (SOTA) vision models are still trained with supervised learning which requires a large corpus of labeled images to work well. By showing the models only labeled images, we limit ourselves from making use of unlabeled images available in much larger quantities to improve accuracy and robustness of SOTA models.

Here, we use unlabeled images to improve the SOTA ImageNet accuracy and show that the accuracy gain has an outsized impact on robustness (out-of-distribution generalization). For this purpose, we use a much larger corpus of unlabeled images, where a large fraction of images do not belong to ImageNet training set distribution (i.e., they do not belong to any category in ImageNet). We train our model with Noisy Student Training, a semi-supervised learning approach, which has three main steps: (1) train a teacher model on labeled images, (2) use the teacher to generate pseudo labels on unlabeled images, and (3) train a student model on the combination of labeled images and pseudo labeled images. We iterate this algorithm a few times by treating the student as a teacher to relabel the unlabeled data and training a new student.

Noisy Student Training improves self-training and distillation in two ways. First, it makes the student larger than, or at least equal to, the teacher so the student can better learn from a larger dataset. Second, it adds noise to the student so the noised student is forced to learn harder from the pseudo labels. To noise the student, we use input noise such as RandAugment data augmentation [49] and model noise such as dropout [241] and stochastic depth [104] during training.

Using Noisy Student Training, together with 300M unlabeled images, we improve EfficientNet's [250] ImageNet top-1 accuracy to 88.4%. This accuracy is 2.0% better than the previous SOTA results which requires 3.5B weakly labeled Instagram images. Not only our method improves standard ImageNet accuracy, it also improves classification robustness on much harder test sets by large margins: ImageNet-A [94] top-1 accuracy from 61.0% to 83.7%, ImageNet-C [93] mean corruption error (mCE) from 45.7 to 28.3 and ImageNet-P [93] mean flip rate (mFR) from 27.8 to 12.2. Our main results are shown in Table 3.1.

	ImageNet top-1 acc.	ImageNet-A top-1 acc.	ImageNet-C mCE	ImageNet-P mFR
Prev. SOTA	86.4%	61.0%	45.7	27.8
Ours	**88.4%**	**83.7%**	**28.3**	**12.2**

Table 3.1: Summary of key results compared to previous state-of-the-art models [161, 257]. Lower is better for mean corruption error (mCE) and mean flip rate (mFR).

3.2 Noisy Student Training

Algorithm 1 gives an overview of Noisy Student Training. The inputs to the algorithm are both labeled and unlabeled images. We use the labeled images to train a teacher model using the standard cross entropy loss. We then use the teacher model to generate pseudo labels on unlabeled images. The pseudo labels can be soft (a continuous distribution) or hard (a one-hot distribution). We then train a student model which minimizes the combined cross entropy loss on both labeled images and unlabeled images. Finally, we iterate the process by putting back the student as a teacher to generate new pseudo labels and train a new student. The algorithm is also illustrated in Figure 3.1.

The algorithm is an improved version of self-training, a method in semi-supervised learning (e.g., [224, 291]), and distillation [98]. More discussions on how our method is related to prior work are included in Section 3.6.

Our key improvements lie in adding noise to the student and using student models that are equal to or larger than the teacher. This makes our method different from Knowledge Distillation [98] where adding noise is not the core concern and a small model is often used as a student to be faster than the teacher. One can think of our method as Knowledge Expansion in which we want the student to be better than the teacher by giving the student model enough capacity and difficult environments in terms of noise to learn through.

Noising Student – When the student is deliberately noised it is actually trained to be consistent to the more powerful teacher that is not noised when it generates pseudo labels. In our experiments, we use two types of noise: input noise and model noise. For input noise, we use data augmentation with RandAugment [49]. For model noise, we use dropout [241] and stochastic depth [104].

When applied to unlabeled data, noise has a compound benefit of enforcing local smoothness in the decision function on both labeled and unlabeled data. Different kinds of noise have

Algorithm 1 Noisy Student Training.

```
Require: Labeled images $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$ and unlabeled images $\{\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_m\}$.
1: Learn teacher model $\theta_*^t$ which minimizes the cross entropy loss on labeled images
       $\frac{1}{n}\sum_{i=1}^{n}\ell(y_i, f^{noised}(x_i, \theta^t))$
2: Use an unnoised teacher model to generate soft or hard pseudo labels for unlabeled images
       $\tilde{y}_i = f(\tilde{x}_i, \theta_*^t), \forall i = 1, \cdots, m$
3: Learn an **equal-or-larger** student model $\theta_*^s$ which minimizes the cross entropy loss on labeled images
   and unlabeled images with **noise** added to the student model
       $\frac{1}{n}\sum_{i=1}^{n}\ell(y_i, f^{noised}(x_i, \theta^s)) + \frac{1}{m}\sum_{i=1}^{m}\ell(\tilde{y}_i, f^{noised}(\tilde{x}_i, \theta^s))$
4: Iterative training: Use the student as a teacher and go back to step 2.
```

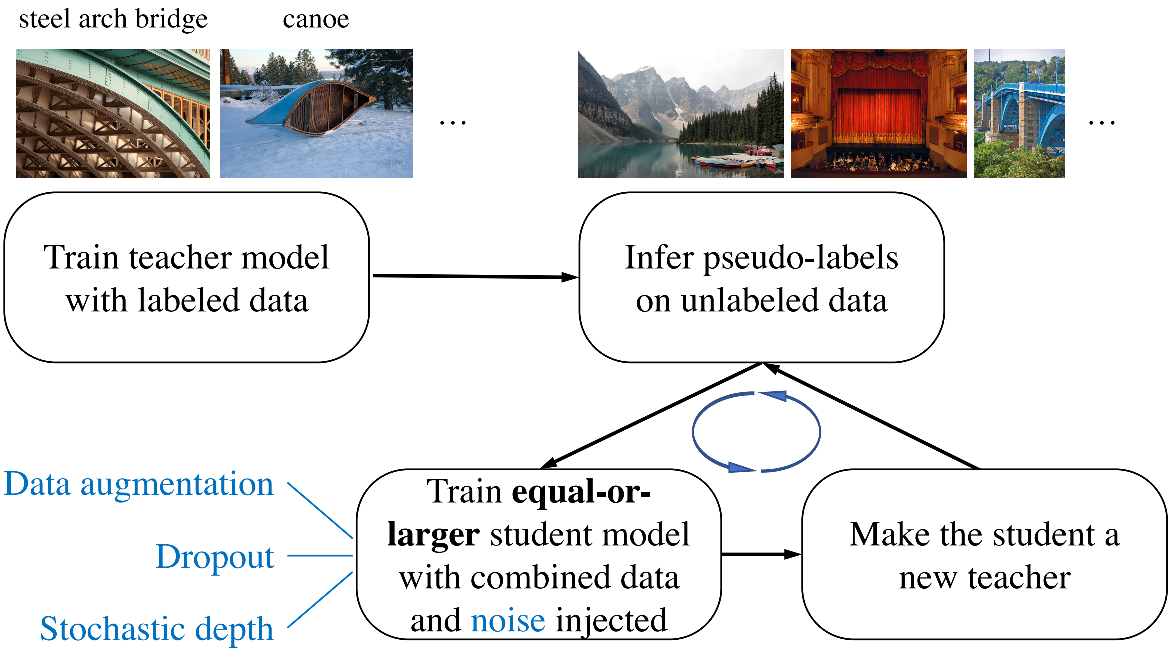

steel arch bridge
canoe
...
...
Train teacher model with labeled data
Infer pseudo-labels on unlabeled data
Data augmentation
Dropout
Stochastic depth
Train equal-or-larger student model with combined data and noise injected
Make the student a new teacher

Figure 3.1: Illustration of the Noisy Student Training. (All shown images are from ImageNet.)

different effects. With data augmentation noise, the student must ensure that an image, when translated for example, should have the same category as a non-translated image. This invariant constraint encourages the student model to learn beyond the teacher to make predictions with more difficult images. When dropout and stochastic depth function are used as noise, the teacher behaves like an ensemble at inference time (during which it generates pseudo labels), whereas the student behaves like a single model. In other words, the student is forced to mimic a more powerful ensemble model. We present an ablation study on the effects of noise in Section 3.4.1.

Other Techniques – Noisy Student Training also work better with an additional trick: data filtering and balancing, similar to [278, 284]. Specifically, we filter images that the teacher model has low confidences on since they are usually out-of-domain images. To ensure that the distribution of the unlabeled images match that of the training set, we also need to balance the number of unlabeled images for each class, as all classes in ImageNet have a similar number of labeled images. For this purpose, we duplicate images in classes where there are not enough images. For classes where we have too many images, we take the images with the highest confidence.[1]

Finally, we emphasize that our method can be used with soft or hard pseudo labels as both work well in our experiments. Soft pseudo labels, in particular, work slightly better for out-of-domain unlabeled data. Thus in the following, for consistency, we report results with soft pseudo labels unless otherwise indicated.

Comparisons with Existing SSL Methods. Apart from self-training, another important line of work in semi-supervised learning [33, 317] is based on consistency training [8, 20, 135, 171, 205, 251, 278] and pseudo labeling [3, 109, 138, 232]. Although they have produced promising results, in our preliminary experiments, methods based on consistency regularization and pseudo labeling work less well on ImageNet. Instead of using a teacher model trained on labeled data to generate pseudo-labels, these methods do not have a separate teacher model and use the model being trained to generate pseudo-labels. In the early phase of training, the model being trained has low accuracy and high entropy, hence consistency training regularizes the model towards high entropy predictions, and prevents it from achieving good accuracy. A common workaround is to use entropy minimization, to filter examples with low confidence or to ramp up the consistency loss. However, the additional hyperparameters introduced by the ramping up schedule, confidence-based filtering and the entropy minimization make them more difficult to use at scale. The self-training / teacher-student framework is better suited for ImageNet because we can train a good teacher on ImageNet using labeled data.

3.3 Experiments

In this section, we will first describe our experiment details. We will then present our ImageNet results compared with those of state-of-the-art models. Lastly, we demonstrate the surprising improvements of our models on robustness datasets (such as ImageNet-A, C and P) as well as under adversarial attacks.

[1] The benefits of data balancing is significant for small models while less significant for larger models.

3.3.1 Experiment Details

Labeled dataset. We conduct experiments on ImageNet 2012 ILSVRC challenge prediction task since it has been considered one of the most heavily benchmarked datasets in computer vision and that improvements on ImageNet transfer to other datasets [129, 207].

Unlabeled dataset. We obtain unlabeled images from the JFT [42, 98], which has around 300M images. Although the images in the dataset have labels, we ignore the labels and treat them as unlabeled data. We filter the ImageNet validation set images from the dataset (see [177]).

We then perform data filtering and balancing on this corpus. First, we run an EfficientNet-B0 trained on ImageNet [250] over the JFT [42, 98] to predict a label for each image. We then select images that have confidence of the label higher than 0.3. For each class, we select at most 130K images that have the highest confidence. Finally, for classes that have less than 130K images, we duplicate some images at random so that each class can have 130K images. Hence the total number of images that we use for training a student model is 130M (with some duplicated images). Due to duplications, there are only 81M unique images among these 130M images. We do not tune these hyperparameters extensively since our method is highly robust to them.

To enable fair comparisons with our results, we also experiment with a public dataset YFCC100M [**?**].

Architecture. We use EfficientNets [250] as our baseline models because they provide better capacity for more data. In our experiments, we also further scale up EfficientNet-B7 and obtain EfficientNet-L2. EfficientNet-L2 is wider and deeper than EfficientNet-B7 but uses a lower resolution, which gives it more parameters to fit a large number of unlabeled images. Due to the large model size, the training time of EfficientNet-L2 is approximately five times the training time of EfficientNet-B7.

The architecture specifications of EfficientNet-L2 are listed in Table 3.2. We also list EfficientNet-B7 as a reference. Scaling width and resolution by c leads to an increase factor of c^2 in training time and scaling depth by c leads to an increase factor of c. The training time of EfficientNet-L2 is around 5 times the training time of EfficientNet-B7.

Architecture Name	w	d	Train Res.	Test Res.	# Params
EfficientNet-B7	2.0	3.1	600	600	66M
EfficientNet-L2	4.3	5.3	475	800	480M

Table 3.2: Architecture specifications for EfficientNets used in the work. The width w and depth d are the scaling factors that need to be contextualized in EfficientNet [250]. Train Res. and Test Res. denote training and testing resolutions respectively.

Training details. For labeled images, we use a batch size of 2048 by default and reduce the batch size when we could not fit the model into the memory. We find that using a batch size of 512, 1024, and 2048 leads to the same performance. We determine the number of training steps and the learning rate schedule by the batch size for labeled images. Specifically, we train the student model for 350 epochs for models larger than EfficientNet-B4, including EfficientNet-L2 and train smaller student models for 700 epochs. The learning rate starts at 0.128 for labeled batch size 2048 and decays by 0.97 every 2.4 epochs if trained for 350 epochs or every 4.8 epochs if trained for 700 epochs.

We use a large batch size for unlabeled images, especially for large models, to make full use of large quantities of unlabeled images. Labeled images and unlabeled images are concatenated together to compute the average cross entropy loss. We apply the recently proposed technique to fix train-test resolution discrepancy [257] for EfficientNet-L2. We first perform normal training with a smaller resolution for 350 epochs. Then we finetune the model with a larger resolution for 1.5 epochs on unaugmented labeled images. Similar to [257], we fix the shallow layers during finetuning.

Our largest model, EfficientNet-L2, needs to be trained for 6 days on a Cloud TPU v3 Pod, which has 2048 cores, if the unlabeled batch size is 14x the labeled batch size.

Method	# Params	Extra Data	Top-1 Acc.	Top-5 Acc.
ResNet-50 [88]	26M	-	76.0%	93.0%
ResNet-152 [88]	60M	-	77.8%	93.8%
DenseNet-264 [105]	34M	-	77.9%	93.9%
Inception-v3 [248]	24M	-	78.8%	94.4%
Xception [42]	23M	-	79.0%	94.5%
Inception-v4 [249]	48M	-	80.0%	95.0%
Inception-resnet-v2 [249]	56M	-	80.1%	95.1%
ResNeXt-101 [279]	84M	-	80.9%	95.6%
PolyNet [307]	92M	-	81.3%	95.8%
SENet [102]	146M	-	82.7%	96.2%
NASNet-A [322]	89M	-	82.7%	96.2%
AmoebaNet-A [206]	87M	-	82.8%	96.1%
PNASNet [150]	86M	-	82.9%	96.2%
AmoebaNet-C [48]	155M	-	83.5%	96.5%
GPipe [107]	557M	-	84.3%	97.0%
EfficientNet-B7 [250]	66M	-	85.0%	97.2%
EfficientNet-L2 [250]	480M	-	85.5%	97.5%
ResNet-50 Billion-scale [284]	26M	3.5B images labeled with tags	81.2%	96.0%
ResNeXt-101 Billion-scale [284]	193M		84.8%	-
ResNeXt-101 WSL [161]	829M		85.4%	97.6%
FixRes ResNeXt-101 WSL [257]	829M		86.4%	98.0%
Big Transfer (BiT-L) [?]†	928M	300M weakly labeled images from JFT	87.5%	98.5%
Noisy Student Training (EfficientNet-L2)	480M	300M unlabeled images from JFT	**88.4%**	**98.7%**

Table 3.3: Top-1 and Top-5 Accuracy of Noisy Student Training and previous state-of-the-art methods on ImageNet. EfficientNet-L2 with Noisy Student Training is the result of iterative training for multiple iterations by putting back the student model as the new teacher. It has better tradeoff in terms of accuracy and model size compared to previous state-of-the-art models. †: Big Transfer is a concurrent work that performs transfer learning from the JFT dataset.

Noise. We use stochastic depth [104], dropout [241], and RandAugment [49] to noise the student. The hyperparameters for these noise functions are the same for EfficientNet-B7 and L2. In particular, we set the survival probability in stochastic depth to 0.8 for the final layer and follow the linear decay rule for other layers. We apply dropout to the final layer with a dropout rate of 0.5. For RandAugment, we apply two random operations with magnitude set to 27.

Iterative training. The best model in our experiments is a result of three iterations of putting back the student as the new teacher. We first trained an EfficientNet-B7 on ImageNet as the teacher model. Then by using the B7 model as the teacher, we trained an EfficientNet-L2 model with the unlabeled batch size set to 14 times the labeled batch size. Then, we trained a new EfficientNet-L2 model with the EfficientNet-L2 model as the teacher. Lastly, we iterated again and used an unlabeled batch size of 28 times the labeled batch size. The detailed results of the three iterations are available in Section 3.4.2.

Robustness Benchmarks Metrics. For completeness, we provide brief descriptions of metrics used in robustness benchmarks ImageNet-A, ImageNet-C and ImageNet-P.

- **ImageNet-A.** The top-1 and top-5 accuracy are measured on the 200 classes that ImageNet-A includes. The mapping from the 200 classes to the original ImageNet classes are available online.[2]
- **ImageNet-C.** mCE (mean corruption error) is the weighted average of error rate on different corruptions, with AlexNet's error rate as a baseline. The score is normalized by AlexNet's error rate so that corruptions with different difficulties lead to scores of a similar scale. Please refer to [93] for details about mCE and AlexNet's error rate. The top-1 accuracy is simply the average top-1 accuracy for all corruptions and all severity degrees. The top-1 accuracy of prior methods are computed from their reported corruption error on each corruption.
- **ImageNet-P.** Flip probability is the probability that the model changes top-1 prediction for different perturbations. mFR (mean flip rate) is the weighted average of flip probability on different perturbations, with AlexNet's flip probability as a baseline. Please refer to [93] for details about mFR and AlexNet's flip probability. The top-1 accuracy reported in this paper is the average accuracy for all images included in ImageNet-P.

On Using RandAugment for ImageNet-C and ImageNet-P. Since Noisy Student Training leads to significant improvements on ImageNet-C and ImageNet-P, we briefly discuss the influence of RandAugment on robustness results. First, note that our supervised baseline EfficientNet-L2 also uses RandAugment. Noisy Student Training leads to significant improvements when compared to the supervised baseline as shown in Table 3.5 and Table 3.6.

Second, the overlap between transformations of RandAugment and ImageNet-C, P is small. For completeness, we list transformations in RandAugment and corruptions and perturbations in ImageNet-C and ImageNet-P here:

[2] https://github.com/hendrycks/natural-adv-examples/blob/master/eval.py

- RandAugment transformations: AutoContrast, Equalize, Invert, Rotate, Posterize, Solarize, Color, Contrast, Brightness, Sharpness, ShearX, ShearY, TranslateX and TranslateY.
- Corruptions in ImageNet-C: Gaussian Noise, Shot Noise, Impulse Noise, Defocus Blur, Frosted Glass Blur, Motion Blur, Zoom Blur, Snow, Frost, Fog, Brightness, Contrast, Elastic, Pixelate, JPEG.
- Perturbations in ImageNet-P: Gaussian Noise, Shot Noise, Motion Blur, Zoom Blur, Snow, Brightness, Translate, Rotate, Tilt, Scale.

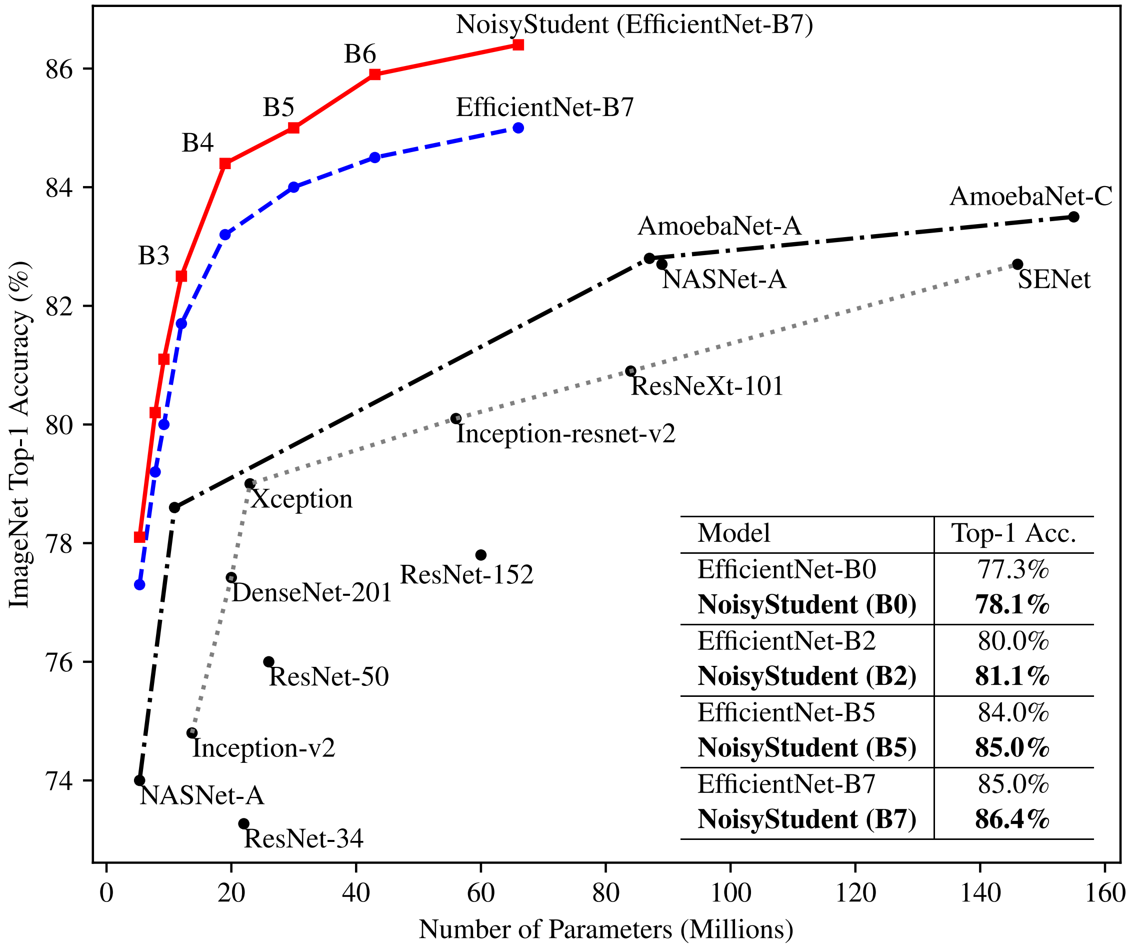
The main overlap between RandAugment and ImageNet-C are Contrast, Brightness and Sharpness. Among them, augmentation Contrast and Brightness are also used in ResNeXt-101 WSL [161] and in vision models that uses the Inception preprocessing [100, 247]. The overlap between RandAugment and ImageNet-P includes Brightness, Translate and Rotate.

3.3.2 ImageNet Results

We first report the validation set accuracy on the ImageNet 2012 ILSVRC challenge prediction task as commonly done in literature [88, 131, 247, 250] (see also [207]). As shown in Table 3.3, EfficientNet-L2 with Noisy Student Training achieves 88.4% top-1 accuracy which is significantly better than the best reported accuracy on EfficientNet of 85.0%. The total gain of 3.4% comes from two sources: by making the model larger (+0.5%) and by Noisy Student Training (+2.9%). In other words, Noisy Student Training makes a much larger impact on the accuracy than changing the architecture.

Further, Noisy Student Training outperforms the state-of-the-art accuracy of 86.4% by FixRes ResNeXt-101 WSL [161, 257] that requires 3.5 Billion Instagram images labeled with tags. As a comparison, our method only requires 300M unlabeled images, which is perhaps more easy to collect. Our model is also approximately twice as small in the number of parameters compared to FixRes ResNeXt-101 WSL.

Model size study: Noisy Student Training for EfficientNet B0-B7 without Iterative Training. In addition to improving state-of-the-art results, we conduct experiments to verify if Noisy Student Training can benefit other EfficienetNet models. In previous experiments, iterative training was used to optimize the accuracy of EfficientNet-L2 but here we skip it as it is difficult to use iterative training for many experiments. We vary the model size from EfficientNet-B0 to EfficientNet-B7 [250] and use the same model as both the teacher and the student. We apply RandAugment to all EfficientNet baselines, leading to more competitive baselines. We set the unlabeled batch size to be three times the batch size of labeled images for all model sizes except for EfficientNet-B0. For EfficientNet-B0, we set the unlabeled batch size to be the same as the batch size of labeled images. As shown in Figure 3.2, Noisy Student Training leads to a consistent improvement of around 0.8% for all model sizes. Overall, EfficientNets with Noisy Student Training provide a much better tradeoff between model size and accuracy than prior works. The results also confirm that vision models can benefit from Noisy Student Training even without iterative training.



Model	Top-1 Acc.
EfficientNet-B0	77.3%
NoisyStudent (B0)	**78.1%**
EfficientNet-B2	80.0%
NoisyStudent (B2)	**81.1%**
EfficientNet-B5	84.0%
NoisyStudent (B5)	**85.0%**
EfficientNet-B7	85.0%
NoisyStudent (B7)	**86.4%**

Figure 3.2: Noisy Student Training leads to significant improvements across all model sizes. We use the same architecture for the teacher and the student and do not perform iterative training.

3.3.3 Robustness Results on ImageNet-A, ImageNet-C and ImageNet-P

We evaluate the best model, that achieves 88.4% top-1 accuracy, on three robustness test sets: ImageNet-A, ImageNet-C and ImageNet-P. ImageNet-C and P test sets [93] include images with common corruptions and perturbations such as blurring, fogging, rotation and scaling. ImageNet-A test set [94] consists of difficult images that cause significant drops in accuracy to state-of-the-art models. These test sets are considered as "robustness" benchmarks because the test images are either much harder, for ImageNet-A, or the test images are different from the training images, for ImageNet-C and P.

For ImageNet-C and ImageNet-P, we evaluate models on two released versions with resolution 224x224 and 299x299 and resize images to the resolution EfficientNet trained on. As shown in Table 3.4, 3.5 and 3.6, Noisy Student Training yields substantial gains on robustness datasets compared to the previous state-of-the-art model ResNeXt-101 WSL [161, 186] trained on 3.5B weakly labeled images. On ImageNet-A, it improves the top-1 accuracy from 61.0% to 83.7%.

Method	Top-1 Acc.	Top-5 Acc.
ResNet-101 [94]	4.7%	-
ResNeXt-101 [94] (32x4d)	5.9%	-
ResNet-152 [94]	6.1%	-
ResNeXt-101 [94] (64x4d)	7.3%	-
DPN-98 [94]	9.4%	-
ResNeXt-101+SE [94] (32x4d)	14.2%	-
ResNeXt-101 WSL [161, 186]	61.0%	-
EfficientNet-L2	49.6%	78.6%
Noisy Student Training (L2)	**83.7%**	**95.2%**

Table 3.4: Robustness results on ImageNet-A.

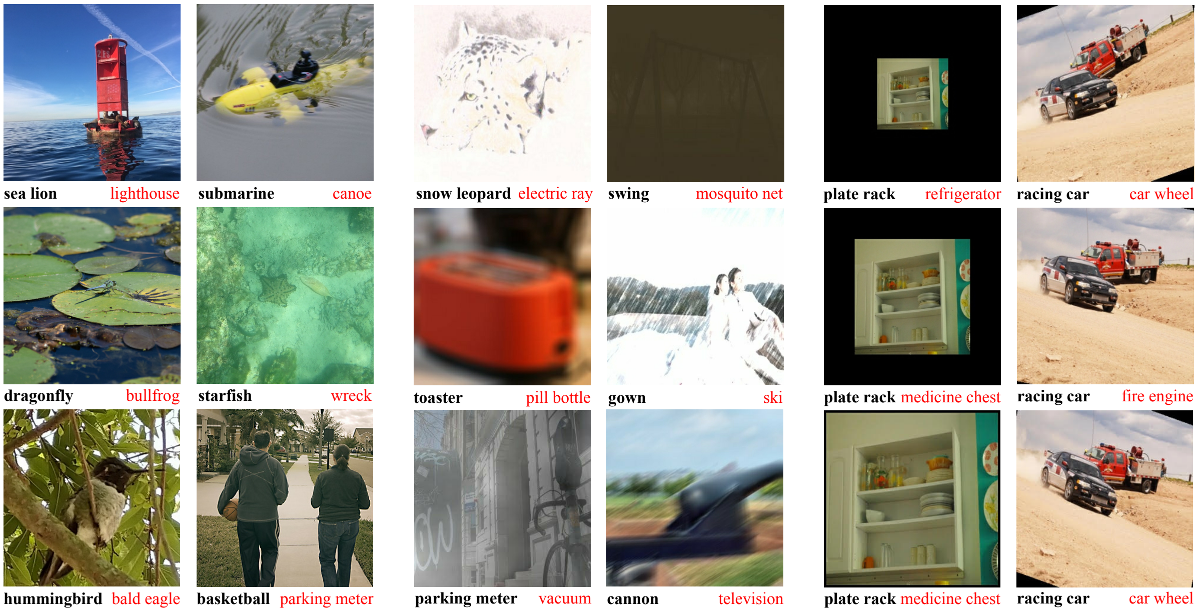
Method	Res.	Top-1 Acc.	mCE
ResNet-50 [93]	224	39.0%	76.7
SIN [71]	224	45.2%	69.3
Patch Gaussian [151]	299	52.3%	60.4
ResNeXt-101 WSL [161, 186]	224	-	45.7
EfficientNet-L2	224	62.6%	47.5
Noisy Student Training (L2)	224	76.5%	30.0
EfficientNet-L2	299	66.6%	42.5
Noisy Student Training (L2)	299	**77.8%**	**28.3**

Table 3.5: Robustness results on ImageNet-C. mCE is the weighted average of error rate on different corruptions, with AlexNet's error rate as a baseline (lower is better).

Method	Res.	Top-1 Acc.	mFR
ResNet-50 [93]	224	-	58.0
Low Pass Filter Pooling [305]	224	-	51.2
ResNeXt-101 WSL [161, 186]	224	-	27.8
EfficientNet-L2	224	80.4%	27.2
Noisy Student Training (L2)	224	85.2%	14.2
EfficientNet-L2	299	81.6%	23.7
Noisy Student Training (L2)	299	**86.4%**	**12.2**

Table 3.6: Robustness results on ImageNet-P, where images are generated with a sequence of perturbations. mFR measures the model's probability of flipping predictions under perturbations with AlexNet as a baseline (lower is better).

On ImageNet-C, it reduces mean corruption error (mCE) from 45.7 to 28.3. On ImageNet-P, it leads to a mean flip rate (mFR) of 14.2 if we use a resolution of 224x224 (direct comparison) and



(a) ImageNet-A (b) ImageNet-C (c) ImageNet-P

Figure 3.3: Selected images from robustness benchmarks ImageNet-A, C and P. Test images from ImageNet-C underwent artificial transformations (also known as common corruptions) that cannot be found on the ImageNet training set. Test images on ImageNet-P underwent different scales of perturbations. On ImageNet-A, C, EfficientNet with Noisy Student Tranining produces correct top-1 predictions (shown in **bold black** texts) and EfficientNet without Noisy Student Training produces incorrect top-1 predictions (shown in red texts). On ImageNet-P, EfficientNet without Noisy Student Training flips predictions frequently.

12.2 if we use a resolution of 299x299.[3] These significant gains in robustness in ImageNet-C and ImageNet-P are surprising because our method was not deliberately optimized for robustness.[4]

Qualitative Analysis. To intuitively understand the significant improvements on the three robustness benchmarks, we show several images in Figure 3.3 where the predictions of the standard model are incorrect while the predictions of the model with Noisy Student Training are correct.

Figure 3.3a shows example images from ImageNet-A and the predictions of our models. The model with Noisy Student Training can successfully predict the correct labels of these highly difficult images. For example, without Noisy Student Training, the model predicts *bullfrog* for the image shown on the left of the second row, which might be resulted from the black lotus leaf on the water. With Noisy Student Training, the model correctly predicts *dragonfly* for the

[3] For EfficientNet-L2, we use the model without finetuning with a larger test time resolution, since a larger resolution results in a discrepancy with the resolution of data and leads to degraded performance on ImageNet-C and ImageNet-P.

[4] Note that both our model and ResNeXt-101 WSL use augmentations that have a small overlap with corruptions in ImageNet-C, which might result in better performance. Specifically, RandAugment includes augmentation Brightness, Contrast and Sharpness. ResNeXt-101 WSL uses augmentation of Brightness and Contrast.

image. At the top-left image, the model without Noisy Student Training ignores the *sea lion*s and mistakenly recognizes a buoy as a lighthouse, while the model with Noisy Student Training can recognize the *sea lion*s.

Figure 3.3b shows images from ImageNet-C and the corresponding predictions. As can be seen from the figure, our model with Noisy Student Training makes correct predictions for images under severe corruptions and perturbations such as snow, motion blur and fog, while the model without Noisy Student Training suffers greatly under these conditions. The most interesting image is shown on the right of the first row. The *swing* in the picture is barely recognizable by human while the model with Noisy Student Training still makes the correct prediction.

Figure 3.3c shows images from ImageNet-P and the corresponding predictions. As can be seen, our model with Noisy Student Training makes correct and consistent predictions as images undergone different perturbations while the model without Noisy Student Training flips predictions frequently.

3.3.4 Adversarial Robustness Results

After testing our model's robustness to common corruptions and perturbations, we also study its performance on adversarial perturbations. We evaluate our EfficientNet-L2 models with and without Noisy Student Training against an FGSM attack. This attack performs one gradient descent step on the input image [76] with the update on each pixel set to ϵ. As shown in Figure 3.4, Noisy Student Training leads to very significant improvements in accuracy even though the model is not optimized for adversarial robustness. Under a stronger attack PGD with 10 iterations [160], at $\epsilon = 16$, Noisy Student Training improves EfficientNet-L2's accuracy from 1.1% to 4.4%.

Note that these adversarial robustness results are not directly comparable to prior work since we use a large input resolution of 800x800 and adversarial vulnerability can scale with the input dimension [66, 74, 76, 235].

3.4 Ablation Study

In this section, we study the importance of noise and iterative training and summarize the ablations for other components of our method.

3.4.1 The Importance of Noise in Self-training

Since we use soft pseudo labels generated from the teacher model, when the student is trained to be exactly the same as the teacher model, the cross entropy loss on unlabeled data would be zero and the training signal would vanish. Hence, a question that naturally arises is why the student can outperform the teacher with soft pseudo labels. As stated earlier, we hypothesize that noising the student is needed so that it does not merely learn the teacher's knowledge. We investigate the importance of noising in two scenarios with different amounts of unlabeled data and different teacher model accuracies. In both cases, we gradually remove augmentation, stochastic depth and dropout for unlabeled images when training the student model, while keeping them for labeled

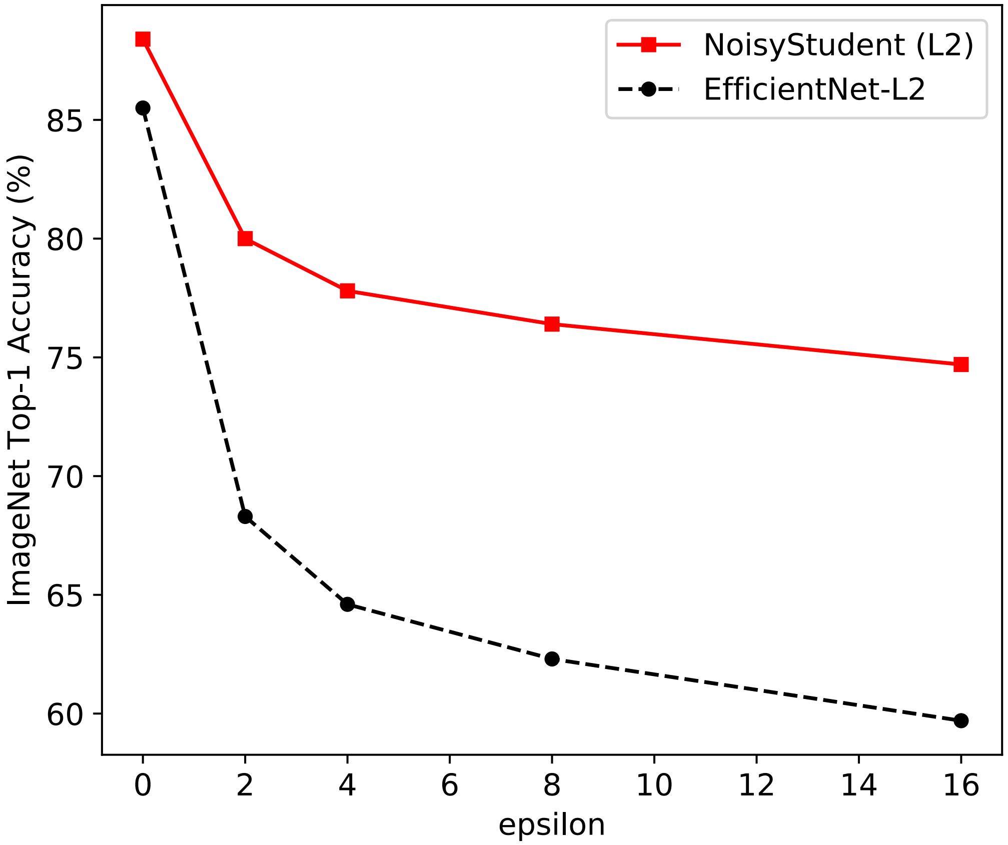


Figure 3.4: Noisy Student Training improves adversarial robustness against an FGSM attack though the model is not optimized for adversarial robustness. The accuracy is improved by 11% at $\epsilon = 2$ and gets better as ϵ gets larger.

images. This way, we can isolate the influence of noising on unlabeled images from the influence of preventing overfitting for labeled images. In addition, we compare using a noised teacher and an unnoised teacher to study if it is necessary to disable noise when generating pseudo labels.

Here, we show the evidence in Table 3.7, noise such as stochastic depth, dropout and data augmentation plays an important role in enabling the student model to perform better than the teacher. The performance consistently drops with noise function removed. However, in the case with 130M unlabeled images, when compared to the supervised baseline, the performance is still improved to 84.3% from 84.0% with noise function removed. We hypothesize that the improvement can be attributed to SGD, which introduces stochasticity into the training process.

One might argue that the improvements from using noise can be resulted from preventing overfitting the pseudo labels on the unlabeled images. We verify that this is not the case when we use 130M unlabeled images since the model does not overfit the unlabeled set from the training loss. While removing noise leads to a much lower training loss for labeled images, we observe that, for unlabeled images, removing noise leads to a smaller drop in training loss. This is probably because it is harder to overfit the large unlabeled dataset.

Lastly, adding noise to the teacher model that generates pseudo labels leads to lower accuracy,

Model / Unlabeled Set Size	1.3M	130M
EfficientNet-B5	83.3%	84.0%
Noisy Student Training (B5)	**83.9%**	**85.1%**
student w/o Aug	83.6%	84.6%
student w/o Aug, SD, Dropout	83.2%	84.3%
teacher w. Aug, SD, Dropout	83.7%	84.4%

Table 3.7: Ablation study of noising. We use EfficientNet-B5 as the teacher model and study two cases with different numbers of unlabeled images and different augmentations. For the experiment with 1.3M unlabeled images, we use the standard augmentation including random translation and flipping for both the teacher and the student. For the experiment with 130M unlabeled images, we use RandAugment. Aug and SD denote data augmentation and stochastic depth respectively. We remove the noise for unlabeled images while keeping them for labeled images. Here, iterative training is not used and unlabeled batch size is set to be the same as the labeled batch size to save training time.

which shows the importance of having a powerful unnoised teacher model.

3.4.2 A Study of Iterative Training

Here, we show the detailed effects of iterative training. As mentioned in Section 3.3.1, we first train an EfficientNet-B7 model on labeled data and then use it as the teacher to train an EfficientNet-L2 student model. Then, we iterate this process by putting back the new student model as the teacher model.

As shown in Table 3.8, the model performance improves to 87.6% in the first iteration and then to 88.1% in the second iteration with the same hyperparameters (except using a teacher model with better performance). These results indicate that iterative training is effective in producing increasingly better models. For the last iteration, we make use of a larger ratio between unlabeled batch size and labeled batch size to boost the final performance to 88.4%.

Iteration	Model	Batch Size Ratio	Top-1 Acc.
1	EfficientNet-L2	14:1	87.6%
2	EfficientNet-L2	14:1	88.1%
3	EfficientNet-L2	28:1	88.4%

Table 3.8: Iterative training improves the accuracy, where batch size ratio denotes the ratio between unlabeled data and labeled data.

3.4.3 Additional Ablation Study Summarization

In this section, we provide comprehensive studies of various components of our method. Since iterative training results in longer training time, we conduct ablation without it. To further save training time, we reduce the training epochs for small models from 700 to 350, starting from Study #4. We also set the unlabeled batch size to be the same as the labeled batch size for models smaller than EfficientNet-B7 starting from Study #2.

Study #1: Teacher Model's Capacity. Here, we study if using a larger and better teacher model would lead to better results. We use our best model Noisy Student Training with EfficientNet-L2, that achieves a top-1 accuracy of 88.4%, to teach student models with sizes ranging from EfficientNet-B0 to EfficientNet-B7. We use the standard augmentation instead of RandAugment on unlabeled data in this experiment to give the student model more capacity. This setting is in principle similar to distillation on unlabeled data.

The comparison is shown in Table 3.9. Using Noisy Student Training (EfficientNet-L2) as the teacher leads to another 0.5% to 1.6% improvement on top of the improved results by using the same model as the teacher. For example, we can train a medium-sized model EfficientNet-B4, which has fewer parameters than ResNet-50, to an accuracy of 85.3%. Therefore, *using a large teacher model with better performance leads to better results.*

Study #2: Unlabeled Data Size. Next, we conduct experiments to understand the effects of using different amounts of unlabeled data. We start with the 130M unlabeled images and gradually reduce the unlabeled set. We experiment with using $\frac{1}{128}, \frac{1}{64}, \frac{1}{32}, \frac{1}{16}, \frac{1}{4}$ of the whole data by uniformly sampling images from the the unlabeled set for simplicity, though taking images with highest confidence may lead to better results. We use EfficientNet-B4 as both the teacher and the student.

As can be seen from Table 3.10, the performance stays similar when we reduce the data to $\frac{1}{16}$ of the whole data,[5] which amounts to 8.1M images after duplicating. The performance drops when we further reduce it. Hence, *using a large amount of unlabeled data leads to better performance*.

Study #3: Hard Pseudo-Label vs. Soft Pseudo-Label on Out-of-domain Data. Unlike previous studies in semi-supervised learning that use in-domain unlabeled data (e.g., CIFAR-10 images as unlabeled data for a small CIFAR-10 training set), to improve ImageNet, we must use out-of-domain unlabeled data. Here we compare hard pseudo-label and soft pseudo-label for out-of-domain data. Since a teacher model's confidence on an image can be a good indicator of whether it is an out-of-domain image, we consider the high-confidence images as in-domain images and the low-confidence images as out-of-domain images. We sample 1.3M images in each confidence interval $[0.0, 0.1], [0.1, 0.2], \cdots, [0.9, 1.0]$.

We use EfficientNet-B0 as both the teacher model and the student model and compare using Noisy Student Training with soft pseudo labels and hard pseudo labels. The results are shown in Figure 3.5 with the following observations: *(1) Soft pseudo labels and hard pseudo labels can both lead to significant improvements with in-domain unlabeled images, i.e., high-confidence images. (2) With out-of-domain unlabeled images, hard pseudo labels can hurt the performance while soft pseudo labels lead to robust performance.*

[5] A larger model might benefit from more data while a small model with limited capacity can easily saturate.

Model	# Params	Top-1 Acc.	Top-5 Acc.
EfficientNet-B0	5.3M	77.3%	93.4%
Noisy Student Training (B0)		78.1%	94.2%
Noisy Student Training (B0, L2)		**78.8%**	**94.5%**
EfficientNet-B1	7.8M	79.2%	94.4%
Noisy Student Training (B1)		80.2%	95.2%
Noisy Student Training (B1, L2)		**81.5%**	**95.8%**
EfficientNet-B2	9.2M	80.0%	94.9%
Noisy Student Training (B2)		81.1%	95.5%
Noisy Student Training (B2, L2)		**82.4%**	**96.3%**
EfficientNet-B3	12M	81.7%	95.7%
Noisy Student Training (B3)		82.5%	96.4%
Noisy Student Training (B3, L2)		**84.1%**	**96.9%**
EfficientNet-B4	19M	83.2%	96.4%
Noisy Student Training (B4)		84.4%	97.0%
Noisy Student Training (B4, L2)		**85.3%**	**97.5%**
EfficientNet-B5	30M	84.0%	96.8%
Noisy Student Training (B5)		85.1%	97.3%
Noisy Student Training (B5, L2)		**86.1%**	**97.8%**
EfficientNet-B6	43M	84.5%	97.0%
Noisy Student Training (B6)		85.9%	97.6%
Noisy Student Training (B6, L2)		**86.4%**	**97.9%**
EfficientNet-B7	66M	85.0%	97.2%
Noisy Student Training (B7)		86.4%	97.9%
Noisy Student Training (B7, L2)		**86.9%**	**98.1%**

Table 3.9: Using our best model with 88.4% accuracy as the teacher (denoted as Noisy Student Training (X, L2)) leads to more improvements than using the same model as the teacher (denoted as Noisy Student Training (X)). Models smaller than EfficientNet-B5 are trained for 700 epochs (better than training for 350 epochs as used in Study #4 to Study #8). Models other than EfficientNet-B0 uses an unlabeled batch size of three times the labeled batch size, while other ablation studies set the unlabeled batch size to be the same as labeled batch size by default for models smaller than B7.

Data	1/128	1/64	1/32	1/16	1/4	1
Top-1 Acc.	83.4%	83.3%	83.7%	83.9%	83.8%	**84.0%**

Table 3.10: Noisy Student Training's performance improves with more unlabeled data. Models are trained for 700 epochs without iterative training. The baseline model achieves an accuracy of 83.2%.

Note that we have also observed that using hard pseudo labels can achieve as good results or slightly better results when a larger teacher is employed. Hence, whether soft pseudo labels or hard pseudo labels work better might need to be determined on a case-by-case basis.

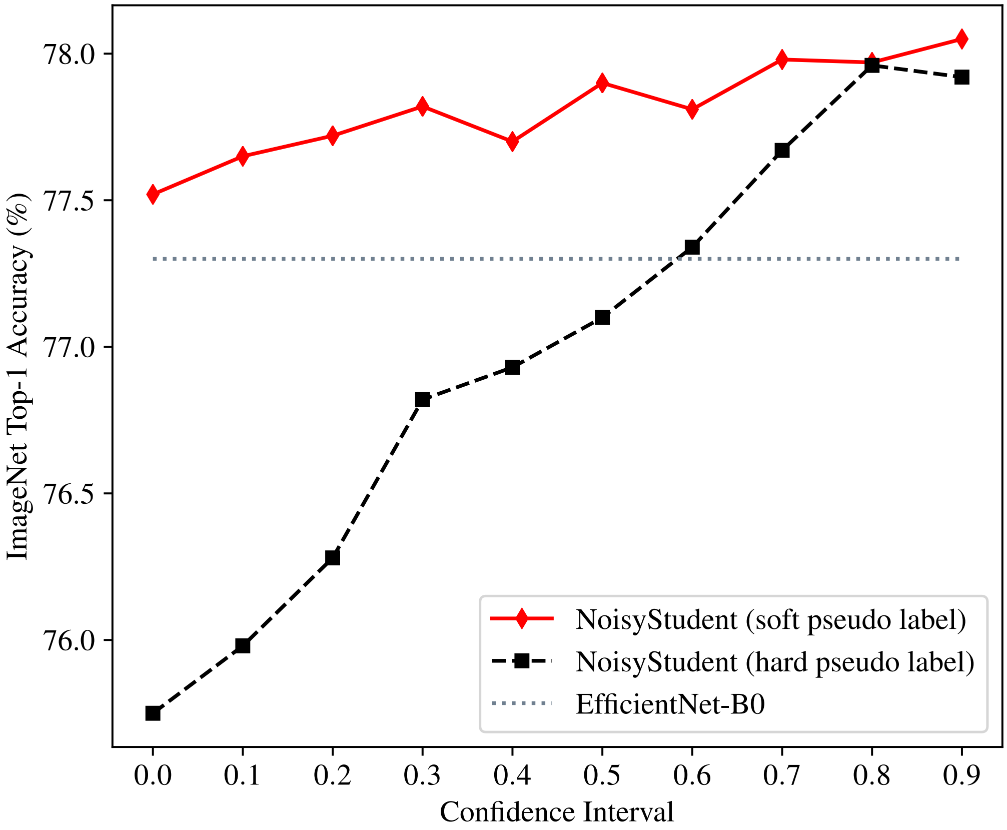


Figure 3.5: Soft pseudo labels lead to better performance for low confidence data (out-of-domain data). Each dot at p represents a Noisy Student Training model trained with 1.3M ImageNet labeled images and 1.3M unlabeled images with confidence scores in $[p, p + 0.1]$.

Study #4: Student Model's Capacity. Then, we investigate the effects of student models with different capacities. For teacher models, we use EfficientNet-B0, B2 and B4 trained on labeled data and EfficientNet-B7 trained using Noisy Student Training. We compare using a student model with the same size or with a larger size. The comparison is shown in Table 3.11. With the same teacher, using a larger student model leads to consistently better performance, showing that *using a large student model is important to enable the student to learn a more powerful model.*

Study #5: Data Balancing. Here, we study the necessity of keeping the unlabeled data balanced across categories. As a comparison, we use all unlabeled data that has a confidence score higher than 0.3. We present results with EfficientNet-B0 to B3 as the backbone models in Table 3.12. Using data balancing leads to better performance for small models EfficientNet-B0 and B1. Interestingly, the gap becomes smaller for larger models such as EfficientNet-B2 and B3, which shows that more powerful models can learn from unbalanced data effectively. *To enable Noisy Student Training to work well for all model sizes, we use data balancing by default.*

Teacher	Teacher Acc.	Student	Student Acc.
B0	77.3%	B0	77.9%
		B1	**79.5%**
B2	80.0%	B2	80.7%
		B3	**82.0%**
B4	83.2%	B4	84.0%
		B5	**84.7%**
B7	86.9%	B7	86.9%
		L2	**87.2%**

Table 3.11: Using a larger student model leads to better performance. Student models are trained for 350 epochs instead of 700 epochs without iterative training. The B7 teacher with an accuracy of 86.9% is trained by Noisy Student Training with multiple iterations using B7. The comparison between B7 and L2 as student models is not completely fair for L2, since we use an unlabeled batch size of 3x the labeled batch size for training L2, which is not as good as using an unlabeled batch size of 7x the labeled batch size when training B7 (See Study #7 for more details).

Model	B0	B1	B2	B3
Supervised Learning	77.3%	79.2%	80.0%	81.7%
Noisy Student Training	**77.9%**	**79.9%**	**80.7%**	82.1%
w/o Data Balancing	77.6%	79.6%	80.6%	82.1%

Table 3.12: Data balancing leads to better results for small models. Models are trained for 350 epochs instead of 700 epochs without iterative training.

Study #6: Joint Training. In our algorithm, we train the model with labeled images and pseudo-labeled images jointly. Here, we also compare with an alternative approach used by Yalniz et al. [284], which first pretrains the model on pseudo-labeled images and then finetunes it on labeled images. For finetuning, we experiment with different steps and take the best results. The comparison is shown in Table 3.13.

It is clear that joint training significantly outperforms pretraining + finetuning. Note that pretraining only on pseudo-labeled images leads to a much lower accuracy than supervised learning only on labeled data, which suggests that the distribution of unlabeled data is very different from that of labeled data. *In this case, joint training leads to a better solution that fits both types of data.*

Study #7: Ratio between Unlabeled Batch Size and Labeled Batch Size. Since we use 130M unlabeled images and 1.3M labeled images, if the batch sizes for unlabeled data and labeled data are the same, the model is trained on unlabeled data only for one epoch every time it is trained on labeled data for a hundred epochs. Ideally, we would also like the model to be trained on unlabeled data for more epochs by using a larger unlabeled batch size so that it can fit the unlabeled data better. Hence we study the importance of the ratio between unlabeled batch size and labeled batch size.

Model	B0	B1	B2	B3
Supervised Learning	77.3%	79.2%	80.0%	81.7%
Pretraining	72.6%	75.1%	75.9%	76.5%
Pretraining + Finetuning	77.5%	79.4%	80.3%	81.7%
Joint Training	**77.9%**	**79.9%**	**80.7%**	**82.1%**

Table 3.13: Joint training work better than pretraining and finetuning. We vary the finetuning steps and report the best results. Models are trained for 350 epochs instead of 700 epochs without iterative training.

Teacher (Acc.)	Batch Size Ratio	Top-1 Acc.
B4 (83.2)	1:1	84.0%
	3:1	84.0%
L2 (87.0)	1:1	86.7%
	3:1	**87.4%**
L2 (87.4)	3:1	87.4%
	6:1	**87.9%**

Table 3.14: With a fixed labeled batch size, a larger unlabeled batch size leads to better performance for EfficientNet-L2. The Batch Size Ratio denotes the ratio between unlabeled batch size and labeled batch size.

In this study, we try a medium-sized model EfficientNet-B4 as well as a larger model EfficientNet-L2. We use models of the same size as both the teacher and the student. As shown in Table 3.14, the larger model EfficientNet-L2 benefits from a large ratio while the smaller model EfficientNet-B4 does not. *Using a larger ratio between unlabeled batch size and labeled batch size, leads to substantially better performance for a large model.*

Study #8: Warm-starting the Student Model. Lastly, one might wonder if we should train the student model from scratch when it can be initialized with a converged teacher model with good accuracy. In this ablation, we first train an EfficientNet-B0 model on ImageNet and use it to initialize the student model. We vary the number of epochs for training the student and use the same exponential decay learning rate schedule. Training starts at different learning rates so that the learning rate is decayed to the same value in all experiments. As shown in Table 3.15, the accuracy drops significantly when we reduce the training epoch from 350 to 70 and drops slightly when reduced to 280 or 140. Hence, the student still needs to be trained for a large number of epochs even with warm-starting.

Further, we also observe that a student initialized with the teacher can sometimes be stuck in a local optimal. For example, when we use EfficientNet-B7 with an accuracy of 86.4% as the teacher, the student model initialized with the teacher achieves an accuracy of 86.4% halfway through the training but gets stuck there when trained for 210 epochs, while a model trained from scratch achieves an accuracy of 86.9%. Hence, though we can save training time by warm-staring, *we train our model from scratch to ensure the best performance.*

Warm-start	Initializing student with teacher				No Init
Epoch	35	70	140	280	350
Top-1 Acc.	77.4%	77.5%	77.7%	77.8%	**77.9%**

Table 3.15: A student initialized with the teacher still requires at least 140 epochs to perform well. The baseline model, trained with labeled data only, has an accuracy of 77.3%.

3.4.4 Results with a Different Architecture and Dataset

Results with ResNet-50. To study whether other architectures can benefit from Noisy Student Training, we conduct experiments with ResNet-50 [88]. We use the full ImageNet as the labeled data and the 130M images from JFT as the unlabeled data. We train a ResNet-50 model on ImageNet and use it as our teacher model. We use RandAugment with the magnitude set to 9 as the noise.

The results are shown in Table 3.16. Noisy Student Training leads to an improvement of 1.3% on the baseline model, which shows that Noisy Student Training is effective for architectures other than EfficientNet.

Method	Top-1 Acc.	Top-5 Acc.
ResNet-50	77.6%	93.8%
Noisy Student Training (ResNet-50)	**78.9%**	**94.3%**

Table 3.16: Experiments on ResNet-50.

Results on SVHN. We also evaluate Noisy Student Training on a smaller dataset SVHN. We use the core set with 73K images as the training set and the validation set. The extra set with 531K images are used as the unlabeled set. We use EfficientNet-B0 with strides of the second and the third blocks set to 1 so that the final feature map is 4x4 when the input image size is 32x32.

As shown in Table 3.17, Noisy Student Training improves the baseline accuracy from 98.1% to 98.6% and outperforms the previous state-of-the-art results achieved by RandAugment with Wide-ResNet-28-10.

Method	Accuracy
RandAugment (WRN)	98.3%
RandAugment (EfficientNet-B0)	98.1%
Noisy Student Training (B0)	**98.6%**

Table 3.17: Results on SVHN.

Model	Dataset	Top-1 Acc.	Top-5 Acc.
EfficientNet-B0	-	77.3%	93.4%
Noisy Student Training (B0)	YFCC	79.9%	95.0%
Noisy Student Training (B0)	JFT	**78.1%**	**94.2%**
EfficientNet-B1	-	79.2%	94.4%
Noisy Student Training (B1)	YFCC	79.9%	95.0%
Noisy Student Training (B1)	JFT	**80.2%**	**95.2%**
EfficientNet-B2	-	80.0%	94.9%
Noisy Student Training (B2)	YFCC	81.0%	**95.6%**
Noisy Student Training (B2)	JFT	**81.1%**	95.5%
EfficientNet-B3	-	81.7%	95.7%
Noisy Student Training (B3)	YFCC	82.3%	96.2%
Noisy Student Training (B3)	JFT	**82.5%**	**96.4%**
EfficientNet-B4	-	83.2%	96.4%
Noisy Student Training (B4)	YFCC	84.2%	96.9%
Noisy Student Training (B4)	JFT	**84.4%**	**97.0%**
EfficientNet-B5	-	84.0%	96.8%
Noisy Student Training (B5)	YFCC	85.0%	97.2%
Noisy Student Training (B5)	JFT	**85.1%**	**97.3%**
EfficientNet-B6	-	84.5%	97.0%
Noisy Student Training (B6)	YFCC	85.4%	97.5%
Noisy Student Training (B6)	JFT	**85.6%**	**97.6%**
EfficientNet-B7	-	85.0%	97.2%
Noisy Student Training (B7)	YFCC	86.2%	**97.9%**
Noisy Student Training (B7)	JFT	**86.4%**	**97.9%**

Table 3.18: Results using YFCC100M and JFT as the unlabeled dataset.

3.5 Results on YFCC100M

Since JFT is not a public dataset, we also experiment with a public unlabeled dataset YFCC100M [**?**], so that researchers can make fair comparisons with our results. Similar to the setting in Section 3.3.2, we experiment with different model sizes without iterative training. We use the same model for both the teacher and the student. We also use the same hyperparamters when using JFT and YFCC100M. Similar to the case for JFT, we first filter images from ImageNet validation set. We then filter low confidence images according to B0's prediction and only keep the top 130K images for each class according to the top-1 predicted class. The resulting set has 34M images since there are not enough images for most classes. We then balance the dataset and increase it to 130M images. As a comparison, before the data balancing stage, there are 81M images in JFT.

As shown in Table 3.18, Noisy Student Training also leads to significant improvements using YFCC100M though it achieves better performance using JFT. The performance difference is probably due to the dataset size difference.

3.6 Related works

Self-training. Our work is based on self-training (e.g., [211, 224, 291**?**]). Self-training first uses labeled data to train a good teacher model, then use the teacher model to label unlabeled data and finally use the labeled data and unlabeled data to jointly train a student model. In typical self-training with the teacher-student framework, noise injection to the student is not used by default, or the role of noise is not fully understood or justified. The main difference between our work and prior work is that we identify the importance of noise, and aggressively inject noise to make the student better.

Self-training was previously used to improve ResNet-50 from 76.4% to 81.2% top-1 accuracy [284] which is still far from the state-of-the-art accuracy. Yalniz et al. [284] also did not show significant improvements in terms of robustness on ImageNet-A, C and P as we did. In terms of methodology, they proposed to first only train on unlabeled images and then finetune their model on labeled images as the final stage. In Noisy Student Training, we combine these two steps into one because it simplifies the algorithm and leads to better performance in our experiments.

Data Distillation [202], which ensembles predictions for an image with different transformations to strengthen the teacher, is the opposite of our approach of weakening the student. Parthasarathi et al. [193] find a small and fast speech recognition model for deployment via knowledge distillation on unlabeled data. As noise is not used and the student is also small, it is difficult to make the student better than teacher. The domain adaptation framework in [214] is related but highly optimized for videos, e.g., prediction on which frame to use in a video. The method in [314] ensembles predictions from multiple teacher models, which is more expensive than our method.

Co-training [22] divides features into two disjoint partitions and trains two models with the two sets of features using labeled data. Their source of “noise" is the feature partitioning such that two models do not always agree on unlabeled data. Our method of injecting noise to the student model also enables the teacher and the student to make different predictions and is more suitable for ImageNet than partitioning features.

Self-training / co-training has also been shown to work well for a variety of other tasks including leveraging noisy data [264], semantic segmentation [7], text classification [120, 244]. Back translation and self-training have led to significant improvements in machine translation [41, 60, 86, 87, 225, 277].

Semi-supervised Learning. Apart from self-training, another important line of work in semi-supervised learning [33, 317] is based on consistency training [4, 8, 18, 20, 40, 43, 134, 135, 142, 154, 171, 192, 200, 205, 251, 266, 278, 302]. They constrain model predictions to be invariant to noise injected to the input, hidden states or model parameters. As discussed in Section 9.3, consistency regularization work less well on ImageNet because consistency regularization uses a model being trained to generate the pseudo-labels. In the early phase of training, they regularize the model towards high entropy predictions, and prevents it from achieving good accuracy.

Works based on pseudo label [3, 109, 138, 232] are similar to self-training, but also suffer the same problem with consistency training, since they rely on a model being trained instead of a converged model with high accuracy to generate pseudo labels. Finally, frameworks in semi-supervised learning also include graph-based methods [126, 274, 287, 316], methods that make use of latent variables as target variables [125, 157, 288] and methods based on low-density separation [52, 77, 221], which might provide complementary benefits to our method.

Knowledge Distillation. Our work is also related to methods in Knowledge Distillation [6, 13, 29, 65, 98] via the use of soft targets. The main use of knowledge distillation is model compression by making the student model smaller. The main difference between our method and knowledge distillation is that knowledge distillation does not consider unlabeled data and does not aim to improve the student model.

Robustness. A number of studies, e.g. [78, 93, 207, 246], have shown that vision models lack robustness. Addressing the lack of robustness has become an important research direction in machine learning and computer vision in recent years. Our study shows that using unlabeled data improves accuracy and general robustness. Our finding is consistent with arguments that using unlabeled data can improve *adversarial* robustness [30, 175, 242, 301]. The main difference between our work and these work is that they directly optimize adversarial robustness on unlabeled data, whereas we show that Noisy Student Training improves robustness greatly even without directly optimizing robustness.

3.7 Discussion

Prior work on weakly-supervised learning required billions of weakly labeled data to improve state-of-the-art ImageNet models. We showed that it is possible to use unlabeled images to significantly advance both accuracy and robustness of state-of-the-art ImageNet models. We found that self-training is a simple and effective algorithm to leverage unlabeled data at scale. We improved it by adding noise to the student, hence the name Noisy Student Training, to learn beyond the teacher's knowledge.

Our experiments showed that Noisy Student Training and EfficientNet can achieve an accuracy of 88.4% which is 2.9% higher than without Noisy Student Training. This result is also a new state-of-the-art and 2.0% better than the previous best method that used an order of magnitude more weakly labeled data [161, 257].

Since the method leads to great performance and is easy to apply, using this method on different tasks is recommended whenever unlabeled data is available.

Chapter 4

Semi-supervised Learning for Reading Comprehension Dataset RACE

After studying semi-supervised learning's effectiveness on classification tasks. We are interested in whether semi-supervised learning can lead to performance improvements on more complex tasks such as reasoning. We first present a reading comprehension dataset that contains questions used to evaluate human's reasoning abilities and that requires significantly more reasoning than existing reading comprehension datasets. Then we evaluate the performance of Noisy Student Training on this task. We find that Noisy Student Training works well for this task that requires a lot of reasoning.

4.1 Introduction

Constructing an intelligence agent capable of understanding text as people is the major challenge of NLP research. With recent advances in deep learning techniques, it seems possible to achieve human-level performance in certain language understanding tasks, and a surge of effort has been devoted to the machine comprehension task where people aim to construct a system with the ability to answer questions related to a document that it has to comprehend [35, 58, 119, 288].

Towards this goal, several large-scale datasets [95, 97, 185, 203, 260] have been proposed, which allow researchers to train deep learning systems and obtain results comparable to the human performance. While having a suitable dataset is crucial for evaluating the system's true ability in reading comprehension, the existing datasets suffer several critical limitations. Firstly, in all datasets, the candidate options are directly extracted from the context (as a single entity or a text span), which leads to the fact that lots of questions can be solved trivially via word-based search and context-matching without deeper reasoning; this constrains the types of questions as well. Secondly, answers and questions of most datasets are either crowd-sourced or automatically-generated, bringing a significant amount of noises in the datasets and limits the ceiling performance by domain experts, such as 82% for Children's Book Test and 84% for Who-did-What. Yet another issue in existing datasets is that the topic coverages are often biased due to the specific ways that the data were initially collected, making it hard to evaluate the ability of systems in text comprehension over a broader range of topics.

To address the aforementioned limitations, we constructed a new dataset by collecting a large set of questions, answers and associated passages in the English exams for middle-school and high-school Chinese students within the 12–18 age range. Those exams were designed by domain experts (instructors) for evaluating the reading comprehension ability of students, with ensured quality and broad topic coverage. Furthermore, the answers by machines or by humans can be objectively graded for evaluation and comparison using the same evaluation metrics. Although efforts have been made with a similar motivation, including the MCTest dataset created by [210] (containing 500 passages and 2000 questions) and several others [122, 195, 212, 233], the usefulness of those datasets is significantly restricted due to their small sizes, especially not suitable for training powerful deep neural networks whose success relies on the availability of relatively large training sets.

Our new dataset, namely RACE, consists of 27,933 passages and 97,687 questions. After reading each passage, each student is asked to answer several questions where each question is provided with four candidate answers – only one of them is correct . Unlike existing datasets, both the questions and candidate answers in RACE are not restricted to be the text spans in the original passage; instead, they can be described in any words. A sample from our dataset is presented in Table 7.1.

Our latter analysis shows that correctly answering a large portion of questions in RACE requires the ability of reasoning, the most important feature as a machine comprehension dataset [35]. RACE also offers two important subdivisions of the reasoning types in its questions, namely passage summarization and attitude analysis, which have not been introduced by the any of the existing large-scale datasets to our knowledge.

In addition, compared to other existing datasets where passages are either domain-specific or of a single fixed style (namely news stories for CNN/Daily Mail, NEWSQA and Who-did-What, fiction stories for Children's Book Test and Book Test, and Wikipedia articles for SQUAD), passages in RACE almost cover all types of human articles, such as news, stories, ads, biography, philosophy, etc., in a variety of styles. This comprehensiveness of topic/style coverage makes RACE a desirable resource for evaluating the reading comprehension ability of machine learning systems in general.

The advantages of our dataset over existing large datasets in machine reading comprehension can be summarized as follows:

- All questions and candidate options are generated by human experts, which are intentionally designed to test human agent's ability in reading comprehension. This makes RACE a relatively accurate indicator for reflecting the text comprehension ability of machine learning systems under human judge.
- The questions are substantially more difficult than those in existing datasets, in terms of the large portion of questions involving reasoning. At the meantime, it is also sufficiently large to support the training of deep learning models.
- Unlike existing large-scale datasets, candidate options in RACE are human generated sentences which may not appear in the original passage. This makes the task more challenging and allows a rich type of questions such as passage summarization and attitude analysis.
- Broad coverage in various domains and writing styles: a desirable property for evaluating generic (in contrast to domain/style-specific) comprehension ability of learning models.

Passage:

In a small village in England about 150 years ago, a mail coach was standing on the street. It didn’t come to that village often. People had to pay a lot to get a letter. The person who sent the letter didn’t have to pay the postage, while the receiver had to.
“Here’s a letter for Miss Alice Brown," said the mailman.
“ I’m Alice Brown," a girl of about 18 said in a low voice.
Alice looked at the envelope for a minute, and then handed it back to the mailman.
“I’m sorry I can’t take it, I don’t have enough money to pay it", she said.
A gentleman standing around were very sorry for her. Then he came up and paid the postage for her.
When the gentleman gave the letter to her, she said with a smile, “ Thank you very much, This letter is from Tom. I’m going to marry him. He went to London to look for work. I’ve waited a long time for this letter, but now I don’t need it, there is nothing in it."
“Really? How do you know that?" the gentleman said in surprise.
“He told me that he would put some signs on the envelope. Look, sir, this cross in the corner means that he is well and this circle means he has found work. That’s good news."
The gentleman was Sir Rowland Hill. He didn’t forgot Alice and her letter.
“The postage to be paid by the receiver has to be changed," he said to himself and had a good plan.
“The postage has to be much lower, what about a penny? And the person who sends the letter pays the postage. He has to buy a stamp and put it on the envelope." he said . The government accepted his plan. Then the first stamp was put out in 1840. It was called the “Penny Black". It had a picture of the Queen on it.

Questions:

1): The first postage stamp was made _.
A. in England B. in America C. by Alice D. in 1910

2): The girl handed the letter back to the mailman because _ .
A. she didn’t know whose letter it was
B. she had no money to pay the postage
C. she received the letter but she didn’t want to open it
D. she had already known what was written in the letter

3): We can know from Alice’s words that _ .
A. Tom had told her what the signs meant before leaving
B. Alice was clever and could guess the meaning of the signs
C. Alice had put the signs on the envelope herself
D. Tom had put the signs as Alice had told him to

4): The idea of using stamps was thought of by _ .
A. the government
B. Sir Rowland Hill
C. Alice Brown
D. Tom

5): From the passage we know the high postage made _ .
A. people never send each other letters
B. lovers almost lose every touch with each other
C. people try their best to avoid paying it
D. receivers refuse to pay the coming letters

Answer: ADABC

Table 4.1: Sample reading comprehension problems from our dataset.

4.2 Related Work

In this section, we briefly outline existing datasets for the machine reading comprehension task, including their strengths and weaknesses.

4.2.1 MCTest

MCTest [210] is a popular dataset for question answering in the same format as RACE, where each question is associated with four candidate answers with a single correct answer. Although questions in MCTest are of high-quality ensured by careful examinations through crowdsourcing, it contains only 500 stores and 2000 questions, which substantially restricts its usage in training advanced machine comprehension models. Moreover, while MCTest is designed for 7 years old children, RACE is constructed for middle and high school students at 12–18 years old hence is more complicated and requires stronger reasoning skills. In other words, RACE can be viewed as a larger and more difficult version of the MCTest dataset.

4.2.2 Cloze-style datasets

The past few years have witnessed several large-scale cloze-style datasets [12, 95, 97, 185], whose questions are formulated by obliterating a word or an entity in a sentence.

CNN/Daily Mail [95] are the largest machine comprehension datasets with 1.4M questions. However, both require limited reasoning ability [35]. In fact, the best machine performance obtained by researchers [35, 58] is close to human's performance on CNN/Daily Mail.

Children's Book Test (CBT) [97] and Book Test (BT) [12] are constructed in a similar manner. Each passage in CBT consist of 20 contiguous sentences extracted from children's books and the next (21st) sentence is used to make the question. The main difference between the two datasets is the size of BT being 60 times larger. Machine comprehension models have also matched human performance on CBT [12].

Who Did What (WDW) [185] is yet another cloze-style dataset constructed from the LDC English Gigaword newswire corpus. The authors generate passages and questions by picking two news articles describing the same event, using one as the passage and the other as the question.

High noise is inevitable in cloze-style datasets due to their automatic generation process, which is reflected in the human performance on these datasets: 82% for CBT and 84% for WDW.

4.2.3 Datasets with Span-based Answers

In datasets such as SQUAD [203], NEWSQA [260] MS MARCO [180] and recently proposed TriviaQA [117]. the answer to each question is in the form of a text span in the article. Articles of SQUAD, NEWSQA and MS MARCO come from Wikipedia, CNN news and the Bing search engine respectively. The answer to a certain question may not be unique and could be multiple spans. Instead of evaluating the accuracy, researchers need to use F1 score, BLEU [189] or ROUGE [146] as metrics, which measure the overlap between the prediction and ground truth answers since the questions come without candidate spans.

Datasets with span-based answers are challenging as the space of possible spans is usually large. However, restricting answers to be text spans in the context passage may be unrealistic and more importantly, may not be intuitive even for humans, indicated by the suffered human performance of 80.3% on SQUAD (or 65% claimed by Trischler et al. [260]) and 46.5% on NEWSQA. In other words, the format of span-based answers may not necessarily be a good examination of reading comprehension of machines whose aim is to approach the comprehension ability of *humans*.

4.2.4 Datasets from Examinations

There have been several datasets extracted from examinations, aiming at evaluating systems under the same conditions as how humans are evaluated in schools. E.g., the AI2 Elementary School Science Questions dataset [122] contains 1080 questions for students in elementary schools; NTCIR QA Lab [233] evaluates systems by the task of solving real-world university entrance exam questions; The Entrance Exams task at CLEF QA Track [195, 212] evaluates the system's reading comprehension ability. However, data provided in these existing tasks are far from sufficient for the training of advanced data-driven machine reading models, partially due to the expensive data generation process by human experts.

To the best of our knowledge, RACE is the first *large-scale* dataset of this type, where questions are created based on exams designed to evaluate human performance in reading comprehension.

4.3 Data Analysis

Dataset	RACE-M			RACE-H			RACE			
Subset	Train	Dev	Test	Train	Dev	Test	Train	Dev	Test	All
# passages	6,409	368	362	18,728	1,021	1,045	25,137	1,389	1,407	27,933
# questions	25,421	1,436	1,436	62,445	3,451	3,498	87,866	4,887	4,934	97,687

Table 4.2: The separation of the training, development and test sets of RACE-M,RACE-H and RACE

Dataset	RACE-M	RACE-H	RACE
Passage Len	231.1	353.1	321.9
Question Len	9.0	10.4	10.0
Option Len	3.9	5.8	5.3
Vocab size	32,811	125,120	136,629

Table 4.3: Statistics of RACE where Len denotes length and Vocab denotes Vocabulary.

In this section, we study the nature of questions covered in RACE at a detailed level. Specifically, we present the dataset statistics in Section 4.3.1, and then analyze different reasoning/question types in RACE in the remaining subsections.

4.3.1 Dataset Statistics

As mentioned in section 9.1, RACE is collected from English examinations designed for 12–15 year-old middle school students, and 15–18 year-old high school students in China. To distin-

Dataset	RACE-M	RACE-H	RACE	CNN	SQuAD	NewsQA
Word Matching	29.4%	11.3%	15.8%	13.0%†	39.8%*	32.7%*
Paraphrasing	14.8%	20.6%	19.2%	41.0%†	34.3%*	27.0%*
Single-Sentence Reasoning	31.3%	34.1%	33.4%	19.0%†	8.6%*	13.2%*
Multi-Sentence Reasoning	22.6%	26.9%	25.8%	2.0%†	11.9%*	20.7%*
Ambiguous/Insufficient	1.8%	7.1%	5.8%	25.0%†	5.4%*	6.4%*

Table 4.4: Statistic information about Reasoning type in different datasets. * denotes the numbers coming from [260] based on 1000 samples per dataset, and numbers with † come from [35].

guish the two subgroups with drastic difficulty gap, RACE-M denotes the middle school examinations and RACE-H denotes high school examinations. We split 5% data as the development set and 5% as the test set for RACE-M and RACE-H respectively. The number of samples in each set is shown in Table 4.2. The statistics for RACE-M and RACE-H is summarized in Table 4.3. We can find that the length of the passages and the vocabulary size in the RACE-H are much larger than that of the RACE-M, an evidence of the higher difficulty of high school examinations.

However, notice that since the articles and questions are selected and designed to test Chinese students learning English as a foreign language, the vocabulary size and the complexity of the language constructs are simpler than news articles and Wikipedia articles in other QA datasets.

4.3.2 Reasoning Types of the Questions

To get a comprehensive picture about the reasoning difficulty requirement of RACE, we conduct human annotations of questions types. Following Chen et al. [35], Trischler et al. [260], we stratify the questions into five classes as follows with ascending order of difficulty:

- Word matching: The question exactly matches a span in the article. The answer is self-evident.
- Paraphrasing: The question is entailed or paraphrased by exactly one sentence in the passage. The answer can be extracted within the sentence.
- Single-sentence reasoning: The answer could be inferred from a single sentence of the article by recognizing incomplete information or conceptual overlap.
- Multi-sentence reasoning: The answer must be inferred from synthesizing information distributed across multiple sentences.
- Insufficient/Ambiguous: The question has no answer or the answer is not unique based on the given passage.

We refer readers to [35, 260] for examples of each category.

To obtain the proportion of different question types, we sample 100 passages from RACE (50 from RACE-M and 50 from RACE-H), all of which have 5 questions hence there are 500 questions in total. We put the passages on Amazon Mechanical Turk[1], and a Hit is generated by a passage with 5 questions. Each question is labeled by two crowdworkers. We require the

[1] https://www.mturk.com/mturk/welcome

turkers to both answer the questions and label the reasoning type. We pay $0.70 and $1.00 per passage in RACE-M and RACE-H respectively, and restrict the access to master turkers only. Finally, we get 1000 labels for the 500 questions.

The statistics about the reasoning type is summarized in Table 4.4. The higher difficulty level of RACE is justified by its higher ratio of reasoning questions in comparison to CNN, SQUAD and NEWSQA. Specifically, 59.2% questions of RACE are either in the category of single-sentence reasoning or in the category of multi-sentence reasoning, while the ratio is 21%, 20.5% and 33.9% for CNN, SQUAD and NEWSQA respectively. Also notice that the ratio of word matching questions on RACE is only 15.8%, the lowest among several categories. In addition, questions in RACE-H are more complex than questions in RACE-M since RACE-M has more word matching questions and fewer reasoning questions.

4.3.3 Subdividing Reasoning Types

To better understand our dataset and facilitate future research, we list the subdivisions of questions under the reasoning category. We find the most frequent reasoning subdivisions include: detail reasoning, whole-picture understanding, passage summarization, attitude analysis and world knowledge. One question may fall into multiple divisions. Definition of these subdivisions and their associated examples are as follows:

1. Detail reasoning: to answer the question, the agent should be clear about the details of the passage. The answer appears in the passage but it cannot be found by simply matching the question with the passage. For example, Question 1 in the sample passage falls into this category.

2. Whole-picture reasoning: the agent needs to understand the whole picture of the story to obtain the correct answer. For example, to answer the Question 2 in the sample passage, the agent is required to comprehend the entire story.

3. Passage summarization: The question requires the agent to select the best summarization of the passage among four candidate summarizations. A typical question of this type is "The main idea of this passage is __.". An example question can be found in Table 4.5.

4. Attitude analysis: The question asks about the opinions/attitudes of the author or a character in the story towards somebody or something, e.g.,

- *Evidence*: "... Many people optimistically thought industry awards for better equipment would stimulate the production of quieter appliances. It was even suggested that noise from building sites could be alleviated ..."
- *Question*: What was the author's attitude towards the industry awards for quieter?
- *Options*: A.suspicious B.positive C.enthusiastic D.indifferent

5. World knowledge: Certain external knowledge is needed. Most frequent questions under this category involve simple arithmetic.

- *Evidence*: “The park is open from 8 am to 5 pm.”
- *Question*: The park is open for __ hours a day.
- *Options*: A.eight B.nine C.ten D.eleven

To the best of our knowledge, questions like passage summarization and attitude analysis have not been introduced by any of the existing large-scale machine comprehension datasets. Both are crucial components in evaluating humans’ reading comprehension abilities.

Passage: Do you love holidays but hate gaining weight? You are not alone. Holidays are times for celebrating. Many people are worried about their weight. With proper planning, though, it is possible to keep normal weight during the holidays. The idea is to enjoy the holidays but not to eat too much. You don’t have to turn away from the foods that you enjoy.

Here are some tips for preventing weight gain and maintaining physical fitness:

Don’t skip meals. Before you leave home, have a small, low-fat meal or snack. This may help to avoid getting too excited before delicious foods.

Control the amount of food. Use a small plate that may encourage you to "load up". You should be most comfortable eating an amount of food about the size of your fist.

Begin with soup and fruit or vegetables. Fill up beforehand on water-based soup and raw fruit or vegetables, or drink a large glass of water before you eat to help you to feel full.

Avoid high-fat foods. Dishes that look oily or creamy may have large amount of fat. Choose lean meat . Fill your plate with salad and green vegetables. Use lemon juice instead of creamy food.

Stick to physical activity. Don’t let exercise take a break during the holidays. A 20-minute walk helps to burn off extra calories.

Questions:

1): What is the best title of the passage?
A. How to avoid holiday feasting
B. Do’s and don’ts for keeping slim and fit.
C. How to avoid weight gain over holidays.
D. Wonderful holidays, boring experiences.

Answer: C

Table 4.5: A sample reading comprehension problem passage summarization.

4.4 Collection Methodology

We collected the raw data from three large free public websites in China[2], where the reading comprehension problems are extracted from English examinations designed by teachers in China. The data before cleaning contains 137,918 passages and 519,878 questions in total, where there are 38,159 passages with 156,782 questions in the middle school group, and 99,759 passages with 363,096 questions in the high school group.

[2] We checked that our dataset does not include example questions of exams with copyright, such as SSAT, SAT, TOEFL and GRE.

The following filtering steps are conducted to clean the raw data. Firstly, we remove all problems and questions that do not have the same format as our problem setting, e.g., a question would be removed if the number of its options is not four. Secondly, we filter all articles and questions that are not self-contained based on the text information, i.e. we remove the articles and questions containing images or tables. We also remove all questions containing keywords “underlined" or “paragraph", since it is difficult to reproduce the effect of underlines and the paragraph segment information. Thirdly, we remove all duplicated articles.

On one of the websites (xkw.com), the answers are stored as images. We used two standard OCR programs tesseract [3] and ABBYY FineReader [4] to process the images. We remove all the answers that two software disagree. The OCR task is easy since we only need to recognize printed alphabet A, B, C, D with a standard font. Finally, we get the cleaned dataset RACE, with 27,933 passages and 97,687 questions.

4.5 Experiments

	RACE-M	RACE-H	RACE	MCTest	CNN	DM	CBT-N	CBT-C	WDW
Random	24.6	25.0	24.9	24.8	0.06	0.06	10.6	10.2	32.0†
Sliding Window	37.3	30.4	32.2	51.5†	24.8	30.8	16.8†	19.6†	48.0†
Stanford AR	44.2	43.0	43.3	–	73.6†	76.6†	–	–	64.0†
GA	43.7	44.2	44.1	–	77.9†	80.9†	70.1†	67.3†	71.2†
Turkers	85.1	69.4	73.3	–	–	–	–	–	–
Ceiling Performance	95.4	94.2	94.5	–	–	–	81.6†	81.6†	84†

Table 4.6: Accuracy of models and human on the each dataset, where † denotes the results coming from previous publications. DM denotes Daily Mail and WDW denotes Who-Did-What
.

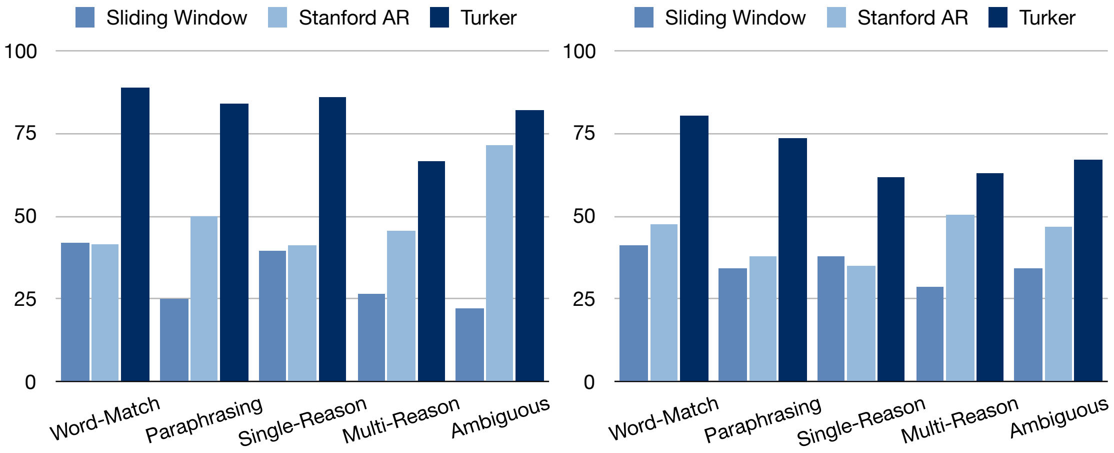
In this section, we compare the performance of several state-of-the-art reading comprehension models with human performance. We use accuracy as the metric to evaluate different models.

4.5.1 Methods for Comparison

Sliding Window Algorithm Firstly, we build the rule-based baseline introduced by Richardson et al. [210]. It chooses the answer having the highest matching score. Specifically, it first concatenates the question and the answer and then calculates the TF-IDF style matching score between the concatenated sentence with every window (a span of text) of the article. The window size is decided by the model performance in the training and dev sets.

[3] https://github.com/tesseract-ocr

[4] https://www.abbyy.com/FineReader



(a) RACE-M

(b) RACE-H

Figure 4.1: Test accuracy of different baselines on each question type category introduced in Section 4.3.2, where Word-Match, Single-Reason, Multi-Reason and Ambiguous are the abbreviations for Word matching, Single-sentence Reasoning, Multi-sentence Reasoning and Insufficient/Ambiguous respectively.

Stanford Attentive Reader Stanford Attentive Reader (Stanford AR) [35] is a strong model that achieves state-of-the-art results on CNN/Daily Mail. Moreover, the authors claim that their model has nearly reached the ceiling performance on these two datasets.

Suppose that the triple of passage, question and options is denoted by $(p, q, o_{1,\cdots,4})$. We first employ bidirectional GRUs to encode p and q respectively into $h^p_1, h^p_2, \ldots, h^p_n$ and h^q. Then we summarize the most relevant part of the passage into s^p with an attention model. Following Chen et al. [35], we adopt a bilinear attention form. Specifically,

$$\begin{aligned} \alpha_i &= \text{Softmax}_i((h^p_i)^T W_1 h^q) \\ s^p &= \sum_i \alpha_i h^p_i \end{aligned} \tag{4.1}$$

Similarly, we use bidirectional GRUs to encode option o_i into a vector h^{o_i}. Finally, we compute the matching score between the i-th option $(i = 1, \cdots, 4)$ and the summarized passage using a bilinear attention. We pass the scores through softmax to get a probability distribution. Specifically, the probability of option i being the right answer is calculated as

$$p_i = \text{Softmax}_i(h^{o_i} W_2 s^d) \tag{4.2}$$

Gated-Attention Reader Gated AR [58] is the state-of-the-art model on multiple datasets. To build query-specific representations of tokens in the document, it employs an attention mechanism to model multiplicative interactions between the query embedding and the document representation. With a multi-hop architecture, GA also enables a model to scan the document and the question iteratively for multiple passes. In other words, the multi-hop structure makes it possible for the reader to refine token representations iteratively and the attention mechanism find the most relevant part of the document. We refer readers to [58] for more details.

After obtaining a query specific document representation s^d, we use the same method as bilinear operation listed in Equation 4.2 to get the output.

Note that our implementation slightly differs from the original GA reader. Specifically, the Attention Sum layer is not applied at the final layer and no character-level embeddings are used.

Implementation Details We follow Chen et al. [35] in our experiment settings. The vocabulary size is set to $50k$. We choose word embedding size $d = 100$ and use the 100-dimensional Glove word embedding [197] as embedding initialization. GRU weights are initialized from Gaussian distribution $\mathcal{N}(0, 0.1)$. Other parameters are initialized from a uniform distribution on $(-0.01, 0.01)$. The hidden dimensionality is set to 128 and the number of layers is set to one for both Stanford AR and GA. We use vanilla stochastic gradient descent (SGD) to train our models. We apply dropout on word embeddings and the gradient is clipped when the norm of the gradient is larger than 10. We use a grid search on validation set to choose the learning rate within $\{0.05, 0.1, 0.3, 0.5\}$ and dropout rate within $\{0.2, 0.5, 0.7\}$. The highest accuracy on validation set is obtained by setting learning rate to 0.1 for Stanford AR and 0.3 for GA and dropout rate to 0.5. The data of RACE-M and RACE-H is used together to train our model and testing is performed separately.

4.5.2 Human Evaluation

As described in section 4.3.2, a randomly sampled subset of test set has been labeled by Amazon Turkers, which contains 500 questions with half from RACE-H and with the other half from RACE-M. The turkers' performance is 85% for RACE-M and 70% for RACE-H. However, it is hard to guarantee that every turker performs the survey carefully, given the difficult and long passages of high school problems. Therefore, to obtain the ceiling human performance on RACE, we manually labeled the proportion of valid questions. A question is valid if it is unambiguous and has a correct answer. We found that 94.5% of the data is valid, which sets the ceiling human performance. Similarly, the ceiling performance on RACE-M and RACE-H is 95.4% and 94.2% respectively.

4.5.3 Main Results

We compare models' and human ceiling performance on datasets which have the same evaluation metric with RACE. The compared datasets include RACE, MCTest, CNN/Daily Mail (CNN and DM), CBT and WDW. On CBT, we report performance on two subsets where the missing token is either a common noun (CBT-C) or name entity (CBT-N) since the language models have already reached human-level performance on other types [97]. The comparison is shown in Table 8.3.

Performance of Sliding Window We first compare MCTest with RACE using Sliding Window, where it is unable to train Stanford AR and Gated AR on MCTest's limited training data. Sliding Window achieves an accuracy of 51.5% on MCTest while only 37.3% on RACE, meaning that to answer the questions of RACE requires more reasoning than MCTest.

The performance of sliding window on RACE is not directly comparable with CBT and WDW since CBT has ten candidate answers for each question and WDW has an average of three. Instead, we evaluate the performance improvement of sliding window on the random baseline. Larger improvement indicates more questions solvable by simple matching. On RACE, Sliding Window is 28.6% better than the random baseline, while the improvement is 58.5%, 92.2% and 50% for CBT-N, CBT-C and WDW.

The accuracy on RACE-M (37.3%) and RACE-H (30.4%) indicates that the middle school questions are simpler based on the matching algorithm.

Performance of Neural Models We further compare the difficulty of different datasets by state-of-the-art neural models' performance. A lower performance means that more problems are unsolvable by machines. The Stanford AR and Gated AR achieve an accuracy of only 43.3% and 44.1% on RACE while their accuracy is much higher on CNN/Daily Mail, Children's Book Test and Who-Did-What. It justifies the fact that, among current large-scale machine comprehension datasets, RACE is the most challenging one.

Human Ceiling Performance The human performance is 94.5% which shows our data is quite clean compared to other large-scale machine comprehension datasets. Since we cannot enforce every turker do the test cautiously, the result shows a gap between turkers' performance and human performance. Reasonably, problems in the high school group with longer passages and more complex questions lead to more significant divergence. Nevertheless, the start-of-the-art models still have a large room to be improved to reach turkers' performance. The performance gap is 41% for the middle school problems and 25% for the high school problems. What's more, The performance of Stanford AR and GA is only less than a half of the ceiling human performance, which indicates that to match the humans' reading comprehension ability, we still have a long way to go.

4.5.4 Reason Types Analysis

We evaluate human and models on different types of questions, shown in Figure 4.1. Turkers do the best on word matching problems while doing the worst on reasoning problems. Sliding window performs better on word matching than problems needing reasoning or paraphrasing. Surprisingly, Stanford AR does not have a stronger performance on the word matching category than reasoning categories. A possible reason is that the proportion of data in reasoning categories is larger than that of data. Also, the candidate answers of simple matching questions may share similar word embeddings. For example, if the question is about color, it is difficult to distinguish candidate answers, “green", “red", “blue" and “yellow", in the embedding vector space. The similar performance on different categories also explains the reason that the performance of the neural models is close in the middle and high school groups in Table 8.3.

4.6 Noisy Student Training for RACE

Given the difficulties of the RACE dataset shown in previous sections, we are interested about whether semi-supervised learning can help with such difficult problems. Here we show the performance of the Noisy Student Training with Funnel-Transformer [53], which is a state-of-the-art pretraining method, as the backbone model on RACE given different amounts of labeled data. We use the B8-8-8H1024 model. As shown in Table 4.7, Noisy Student Training leads to consistent improvements across all data sizes. For example, with noisy student training, we can achieve similar performance with methods that use 2x labeled data. For example, Noisy Student Training can use 20% labeled data to achieve the performance of using 50% labeled data on RACE and RACE-M. Hence, Noisy Student Training is still effective for this problem. However, even with Noisy Student Training, we still need a large amount of labeled data to do well on this task.

Note that the noise used here only includes model noise such as dropout and attention dropout. We hypothesize that investigating more advanced noise such as data augmentation might lead to even better the performance.

	Method	2%	5%	10%	20%	50%
RACE	Supervised	68.2	74.6	77.4	80.2	81.7
	Noisy Student Training	**71.1**	**77.0**	**79.7**	**82.0**	**83.7**
RACE-M	Supervised	73.6	78.4	81.1	84.2	85.0
	Noisy Student Training	**77.1**	**81.5**	**83.8**	**85.9**	**87.1**
RACE-H	Supervised	66.7	73.4	76.4	79.1	80.7
	Noisy Student Training	**68.6**	**75.1**	**78.0**	**80.4**	**82.3**

Table 4.7: Accuracy on RACE with different amount of labeled data. The model achieves an accuracy of 84.5/87.3/83.4 on RACE/RACE-M/RACE-H.

4.7 Discussion

We present a large, high-quality dataset for reading comprehension that is carefully designed to examine human ability on this task. Some desirable properties of RACE include the broad coverage of domains/styles and the richness in the question format. Most importantly, it requires substantially more reasoning to do well on RACE than on other datasets, as there is a significant gap between the performance of state-of-the-art machine comprehension models and that of the human.

In addition, we show that semi-supervised learning bring significant gains even for this difficult reading comprehension task as evidenced by the improved performance with different amounts of labeled data.

Part II

Data-Efficient Learning by Transfer Learning

Chapter 5

Transfer Learning by Parameter Sharing

In this chapter, we present ITransF, a method that learns a parameter sharing mechanism for transferring knowledge between different sub-tasks for knowledge base completion. Specifically, different relations can be treated as different sub-tasks and many relations share common statistical regularities. At the core of ITransF is a sparse attention mechanism, which learns to compose shared concept parameters into relation-specific parameters, leading to a better generalization property. ITransF improves mean rank and Hits@10 on two benchmark datasets on knowledge base completion, over all previous approaches of the same kind. In addition, the parameter sharing is clearly indicated by the learned sparse attention vectors, enabling us to interpret how knowledge transfer is carried out.

5.1 Introduction

Knowledge bases (KB), such as WordNet [62], Freebase [23], YAGO [243] and DBpedia [139], are useful resources for many applications such as question answering [16, 51, 293] and information extraction [168]. However, knowledge bases suffer from incompleteness despite their formidable sizes [239, 273], leading to a number of studies on automatic knowledge base completion (KBC) [182] or link prediction.

The fundamental motivation behind these studies is that there exist some statistical regularities under the intertwined facts stored in the multi-relational knowledge base. By discovering generalizable regularities in known facts, missing ones may be recovered in a faithful way. Due to its excellent generalization capability, distributed representations, a.k.a. embeddings, have been popularized to address the KBC task [25, 26, 27, 80, 178, 181, 239, 271].

As a seminal work, Bordes et al. [26] proposes the TransE, which models the statistical regularities with linear translations between entity embeddings operated by a relation embedding. Implicitly, TransE assumes both entity embeddings and relation embeddings dwell in the same vector space, posing an unnecessarily strong prior. To relax this requirement, a variety of models first project the entity embeddings to a relation-dependent space [27, 111, 149, 178], and then model the translation property in the projected space. Typically, these relation-dependent spaces are characterized by the projection matrices unique to each relation. As a benefit, different aspects of the same entity can be temporarily emphasized or depressed as an effect of the projection. For instance, STransE [178] utilizes two projection matrices per relation, one for the head entity and the other for the tail entity.

Despite the superior performance of STransE compared to TransE, it is more prone to the data sparsity problem. Concretely, since the projection spaces are unique to each relation, projection matrices associated with rare relations can only be exposed to very few facts during training, resulting in poor generalization. For common relations, a similar issue exists. Without any restrictions on the number of projection matrices, logically related or conceptually similar relations may have distinct projection spaces, hindering the discovery, sharing, and generalization of statistical regularities.

Previously, a line of research makes use of external information such as textual relations from web-scale corpus or node features [179, 255, 256], alleviating the sparsity problem. In parallel, recent work has proposed to model regularities beyond local facts by considering multi-relation paths [69, 148, 231]. Since the number of paths grows exponentially with its length, as a side effect, path-based models enjoy much more training cases, suffering less from the problem.

In this work, we present an interpretable knowledge transfer model (ITransF), which encourages the sharing of statistic regularities between the projection matrices of relations and alleviates the data sparsity problem. At the core of ITransF is a sparse attention mechanism, which learns to compose shared concept matrices into relation-specific projection matrices, leading to a better generalization property. Without any external resources, ITransF improves mean rank and Hits@10 on two benchmark datasets, over all previous approaches of the same kind. In addition, the parameter sharing is clearly indicated by the learned sparse attention vectors, enabling us to interpret how knowledge transfer is carried out. To induce the desired sparsity during optimization, we further introduce a block iterative optimization algorithm.

In summary, the contributions of this work are: (i) proposing a novel knowledge embedding model which enables knowledge transfer by learning to discover shared regularities; (ii) introducing a learning algorithm to directly optimize a sparse representation from which the knowledge transferring procedure is interpretable; (iii) showing the effectiveness of our model by outperforming baselines on two benchmark datasets for knowledge base completion task.

Notation and Previous Models Let E denote the set of entities and R denote the set of relations. In knowledge base completion, given a training set P of triples (h, r, t) where $h, t \in E$ are the head and tail entities having a relation $r \in R$, e.g., (*Steve Jobs*, `FounderOf`, *Apple*), we want to predict missing facts such as (*Steve Jobs*, `Profession`, *Businessperson*).

Most of the embedding models for knowledge base completion define an energy function $f_r(h, t)$ according to the fact's plausibility [25, 26, 27, 80, 178, 239, 271, 286]. The models are learned to minimize energy $f_r(h, t)$ of a plausible triple (h, r, t) and to maximize energy $f_r(h', t')$ of an implausible triple (h', r, t').

Motivated by the linear translation phenomenon observed in well trained word embeddings [166], TransE [26] represents the head entity h, the relation r and the tail entity t with vectors $\mathbf{h}, \mathbf{r}$ and $\mathbf{t} \in \mathbb{R}^n$ respectively, which were trained so that $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$. They define the energy function as

$$f_r(h, t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_\ell$$

where $\ell = 1$ or 2, which means either the ℓ_1 or the ℓ_2 norm of the vector $\mathbf{h} + \mathbf{r} - \mathbf{t}$ will be used depending on the performance on the validation set.

To better model relation-specific aspects of the same entity, TransR [149] uses projection matrices and projects the head entity and the tail entity to a relation-dependent space. STransE [178] extends TransR by employing different matrices for mapping the head and the tail entity. The energy function is

$$f_r(h,t) = \|\mathbf{W}_{r,1}\mathbf{h} + \mathbf{r} - \mathbf{W}_{r,2}\mathbf{t}\|_\ell$$

However, not all relations have abundant data to estimate the relation specific matrices as most of the training samples are associated with only a few relations, leading to the data sparsity problem for rare relations.

5.2 Interpretable Knowledge Transfer

5.2.1 Model

As discussed above, a fundamental weakness in TransR and STransE is that they equip each relation with a set of unique projection matrices, which not only introduces more parameters but also hinders knowledge sharing. Intuitively, many relations share some concepts with each other, although they are stored as independent symbols in KB. For example, the relation "(somebody) won award for (some work)" and "(somebody) was nominated for (some work)" both describe a person's high-quality work which wins an award or a nomination respectively. This phenomenon suggests that one relation actually represents a collection of real-world concepts, and one concept can be shared by several relations. Inspired by the existence of such lower-level concepts, instead of defining a unique set of projection matrices for every relation, we can alternatively define a small set of concept projection matrices and then compose them into customized projection matrices. Effectively, the relation-dependent translation space is then reduced to the smaller concept spaces.

However, in general, we do not have prior knowledge about what concepts exist out there and how they are composed to form relations. Therefore, in ITransF, we aim to learn this information simultaneously from data, together with all knowledge embeddings. Following this idea, we first present the model details, then discuss the optimization techniques for training.

Energy function Specifically, we stack all the concept projection matrices to a 3-dimensional tensor $\mathbf{D} \in \mathbb{R}^{m \times n \times n}$, where m is the pre-specified number of concept projection matrices and n is the dimensionality of entity embeddings and relation embeddings. We let each relation select the most useful projection matrices from the tensor, where the selection is represented by an attention vector. The energy function of ITransF is defined as:

$$f_r(h,t) = \|\boldsymbol{\alpha}_r^H \cdot \mathbf{D} \cdot \mathbf{h} + \mathbf{r} - \boldsymbol{\alpha}_r^T \cdot \mathbf{D} \cdot \mathbf{t}\|_\ell \tag{5.1}$$

where $\boldsymbol{\alpha}_r^H, \boldsymbol{\alpha}_r^T \in [0,1]^m$, satisfying $\sum_i \boldsymbol{\alpha}_{r,i}^H = \sum_i \boldsymbol{\alpha}_{r,i}^T = 1$, are normalized attention vectors used to compose all concept projection matrices in $\mathbf{D}$ by a convex combination. It is obvious that STransE can be expressed as a special case of our model when we use $m = 2|R|$ concept matrices and set attention vectors to disjoint one-hot vectors. Hence our model space is a generalization of STransE. Note that we can safely use fewer concept matrices in ITransF and obtain better performance (see section 5.3.3), though STransE always requires $2|R|$ projection matrices.

We follow previous work to minimize the following hinge loss function:

$$\mathcal{L} = \sum_{\substack{(h,r,t)\sim P, \\ (h',r,t')\sim N}} [\gamma + f_r(h,t) - f_r(h',t')]_+ \tag{5.2}$$

where P is the training set consisting of correct triples, N is the distribution of corrupted triples defined in section 5.2.3, and $[\cdot]_+ = \max(\cdot, 0)$. Note that we have omitted the dependence of N on (h, r, t) to avoid clutter. We normalize the entity vectors $\mathbf{h}, \mathbf{t}$, and the projected entity vectors $\boldsymbol{\alpha}_r^H \cdot \mathbf{D} \cdot \mathbf{h}$ and $\boldsymbol{\alpha}_r^T \cdot \mathbf{D} \cdot \mathbf{t}$ to have unit length after each update, which is an effective regularization method that benefits all models.

Sparse attention vectors In Eq. (5.1), we have defined $\boldsymbol{\alpha}_r^H, \boldsymbol{\alpha}_r^T$ to be some normalized vectors used for composition. With a dense attention vector, it is computationally expensive to perform the convex combination of m matrices in each iteration. Moreover, a relation usually does not consist of all existing concepts in practice. Furthermore, when the attention vectors are sparse, it is often easier to interpret their behaviors and understand how concepts are shared by different relations.

Motivated by these potential benefits, we further hope to learn sparse attention vectors in ITransF. However, directly posing ℓ_1 regularization [254] on the attention vectors fails to produce sparse representations in our preliminary experiment, which motivates us to enforce ℓ_0 constraints on $\boldsymbol{\alpha}_r^T, \boldsymbol{\alpha}_r^H$.

In order to satisfy both the normalization condition and the ℓ_0 constraints, we reparameterize the attention vectors in the following way:

$$\begin{aligned}\boldsymbol{\alpha}_r^H &= \mathrm{SparseSoftmax}(\mathbf{v}_r^H, \mathbf{I}_r^H) \\ \boldsymbol{\alpha}_r^T &= \mathrm{SparseSoftmax}(\mathbf{v}_r^T, \mathbf{I}_r^T)\end{aligned}$$

where $\mathbf{v}_r^H, \mathbf{v}_r^T \in \mathbb{R}^m$ are the pre-softmax scores, $\mathbf{I}_r^H, \mathbf{I}_r^T \in \{0,1\}^m$ are the sparse assignment vectors, indicating the non-zero entries of attention vectors, and the $\mathrm{SparseSoftmax}$ is defined as

$$\mathrm{SparseSoftmax}(\mathbf{v}, \mathbf{I})_i = \frac{\exp(\mathbf{v}_i/\tau)\mathbf{I}_i}{\sum_j \exp(\mathbf{v}_j/\tau)\mathbf{I}_j}$$

with τ being the temperature of Softmax.

With this reparameterization, $\mathbf{v}_r^H, \mathbf{v}_r^T$ and $\mathbf{I}_r^H, \mathbf{I}_r^T$ replace $\boldsymbol{\alpha}_r^T, \boldsymbol{\alpha}_r^H$ to become the real parameters of the model. Also, note that it is equivalent to pose the ℓ_0 constraints on $\mathbf{I}_r^H, \mathbf{I}_r^T$ instead of $\boldsymbol{\alpha}_r^T, \boldsymbol{\alpha}_r^H$. Putting these modifications together, we can rewrite the optimization problem as

$$\begin{aligned}&\text{minimize} && \mathcal{L} \\ &\text{subject to} && \|\mathbf{I}_r^H\|_0 \leq k, \|\mathbf{I}_r^T\|_0 \leq k\end{aligned} \tag{5.3}$$

where $\mathcal{L}$ is the loss function defined in Eq. (5.2).

5.2.2 Block Iterative Optimization

Though sparseness is favorable in practice, it is generally NP-hard to find the optimal solution under ℓ_0 constraints. Thus, we resort to an approximated algorithm in this work.

For convenience, we refer to the parameters with and without the sparse constraints as the *sparse* partition and the *dense* partition, respectively. Based on this notion, the high-level idea of the approximated algorithm is to iteratively optimize one of the two partitions while holding the other one fixed. Since all parameters in the dense partition, including the embeddings, the projection matrices, and the pre-softmax scores, are fully differentiable with the sparse partition fixed, we can simply utilize SGD to optimize the dense partition. Then, the core difficulty lies in the step of optimizing the sparse partition (i.e. the sparse assignment vectors), during which we want the following two properties to hold

1. the sparsity required by the ℓ_0 constaint is maintained, and
2. the cost define by Eq. (5.2) is decreased.

Satisfying the two criterion seems to highly resemble the original problem defined in Eq. (5.3). However, the dramatic difference here is that with parameters in the dense partition regarded as constant, the cost function is decoupled w.r.t. each relation r. In other words, the optimal choice of $\mathbf{I}_r^H, \mathbf{I}_r^T$ is independent of $\mathbf{I}_{r'}^H, \mathbf{I}_{r'}^T$ for any $r' \neq r$. Therefore, we only need to consider the optimization for a single relation r, which is essentially an assignment problem. Note that, however, $\mathbf{I}_r^H$ and $\mathbf{I}_r^T$ are still coupled, without which we basically reach the situation in a backpack problem. In principle, one can explore combinatorial optimization techniques to optimize $\mathbf{I}_{r'}^H, \mathbf{I}_{r'}^T$ jointly, which usually involve some iterative procedure. To avoid adding another inner loop to our algorithm, we turn to a simple but fast approximation method based on the following single-matrix cost.

Specifically, for each relation r, we consider the induced cost $\mathcal{L}_{r,i}^H$ where only a single projection matrix i is used for the head entity:

$$\mathcal{L}_{r,i}^H = \sum_{\substack{(h,r,t)\sim P_r,\\ (h',r,t')\sim N_r}} \left[\gamma + f_{r,i}^H(h,t) - f_{r,i}^H(h',t')\right]_+$$

where $f_{r,i}^H(h,t) = \|\mathbf{D}_i \cdot \mathbf{h} + \mathbf{r} - \boldsymbol{\alpha}_r^T \cdot \mathbf{D} \cdot \mathbf{t}\|$ is the corresponding energy function, and the subscript in P_r and N_r denotes the subsets with relation r. Intuitively, $\mathcal{L}_{r,i}^H$ measures, given the current tail attention vector $\boldsymbol{\alpha}_r^T$, if only one project matrix could be chosen for the head entity, how implausible D_i would be. Hence, $i^* = \arg\min_i \mathcal{L}_{r,i}^H$ gives us the best single projection matrix on the head side given $\boldsymbol{\alpha}_r^T$.

Now, in order to choose the best k matrices, we basically ignore the interaction among projection matrices, and update $\mathbf{I}_r^H$ in the following way:

$$\mathbf{I}_{r,i}^H \leftarrow \begin{cases} 1, & i \in \operatorname{argpartition}_i(\mathcal{L}_{r,i}^H, k) \\ 0, & \text{otherwise} \end{cases}$$

where the function $\operatorname{argpartition}_i(x_i, k)$ produces the index set of the lowest-k values of x_i.

Analogously, we can define the single-matrix cost $\mathcal{L}_{r,i}^T$ and the energy function $f_{r,i}^T(h,t)$ on the tail side in a symmetric way. Then, the update rule for $\mathbf{I}_r^H$ follows the same derivation.

Admittedly, the approximation described here is relatively crude. But as we will show in section **??**, the algorithm yields good performance empirically. We leave the further improvement of the optimization method as future work.

5.2.3 Corrupted Sample Generating Method

Recall that we need to sample a negative triple (h', r, t') to compute hinge loss shown in Eq. 5.2, given a positive triple $(h, r, t) \in P$. The distribution of negative triple is denoted by $N(h, r, t)$. Previous work [26, 149, 178, 286] generally constructs a set of corrupted triples by replacing the head entity or tail entity with a random entity uniformly sampled from the KB.

However, uniformly sampling corrupted entities may not be optimal. Often, the head and tail entities associated a relation can only belong to a specific domain. When the corrupted entity comes from other domains, it is very easy for the model to induce a large energy gap between true triple and corrupted one. As the energy gap exceeds γ, there will be no training signal from this corrupted triple. In comparison, if the corrupted entity comes from the same domain, the task becomes harder for the model, leading to more consistent training signal.

Motivated by this observation, we sample corrupted head or tail from entities in the same domain with a probability p_r and from the whole entity set with probability $1 - p_r$.

We define the probability p_r to generate a negative sample from the same domain mentioned in Section 5.2.3. The probability cannot be too high to avoid generating negative samples that are actually correct, since there are generally a lot of facts missing in KBs.

Specifically, let $\mathrm{M}_r^H = \{h \mid \exists t(h, r, t) \in P\}$ and $\mathrm{M}_r^T = \{t \mid \exists h(h, r, t) \in P\}$ denote the head or tail domain of relation r. Suppose $N_r = \{(h, r, t) \in P\}$ is the induced set of edges with relation r. We define the probability p_r as

$$p_r = min(\frac{\lambda|\mathrm{M}_r^T||\mathrm{M}_r^H|}{|N_r|}, 0.5) \tag{5.4}$$

Our motivation of such a formulation is as follows: Suppose O_r is the set that contains all truthful fact triples on relation r, i.e., all triples in training set and all other missing correct triples. If we assume all fact triples within the domain has uniform probability of being true, the probability of a random triple being correct is $Pr((h, r, t) \in O_r \mid h \in \mathrm{M}_r^H, t \in \mathrm{M}_r^T) = \frac{|O_r|}{|\mathrm{M}_r^H||\mathrm{M}_r^T|}$

Assume that all facts are missing with a probability λ, then $|N_r| = \lambda|O_r|$ and the above probability can be approximated by $\frac{|N_r|}{\lambda|\mathrm{M}_r^H||\mathrm{M}_r^T|}$. We want the probability of generating a negative sample from the domain to be inversely proportional to the probability of the sample being true, so we define the probability as Eq. 5.4. The results in section **??** are obtained with λ set to 0.001.

In the rest of the paper, we refer to the new sampling method as "domain sampling".

5.3 Experiments

5.3.1 Setup

To evaluate link prediction, we conduct experiments on the WN18 (WordNet) and FB15k (Freebase) introduced by Bordes et al. [26] and use the same training/validation/test split as in [26].

The information of the two datasets is given in Table 5.1.

Dataset	**#E**	**#R**	**#Train**	**#Valid**	**#Test**
WN18	40,943	18	141,442	5,000	5,000
FB15k	14,951	1,345	483,142	50,000	59,071

Table 5.1: Statistics of FB15k and WN18 used in experiments. #E, #R denote the number of entities and relation types respectively. #Train, #Valid and #Test are the numbers of triples in the training, validation and test sets respectively.

In knowledge base completion task, we evaluate model's performance of predicting the head entity or the tail entity given the relation and the other entity. For example, to predict head given relation r and tail t in triple (h, r, t), we compute the energy function $f_r(h', t)$ for each entity h' in the knowledge base and rank all the entities according to the energy. We follow Bordes et al. [26] to report the *filter* results, i.e., removing all other correct candidates h' in ranking. The rank of the correct entity is then obtained and we report the mean rank (mean of the predicted ranks) and Hits@10 (top 10 accuracy). Lower mean rank or higher Hits@10 mean better performance.

5.3.2 Implementation Details

We initialize the projection matrices with identity matrices added with a small noise sampled from normal distribution $\mathcal{N}(0,\ 0.005^2)$. The entity and relation vectors of ITransF are initialized by TransE [26], following García-Durán et al. [69, 70], Ji et al. [111], Lin et al. [148, 149]. We ran mini-batch SGD until convergence. We employ the "*Bernoulli*" sampling method to generate incorrect triples as used in Wang et al. [271], Lin et al. [149], He et al. [90], Ji et al. [111] and Lin et al. [148].

STransE [178] is the most similar knowledge embedding model to ours except that they use distinct projection matrices for each relation. We use the same hyperparameters as used in STransE and no significant improvement is observed when we alter hyperparameters. We set the margin γ to 5 and dimension of embedding n to 50 for WN18, and $\gamma = 1, n = 100$ for FB15k. We set the batch size to 20 for WN18 and 1000 for FB15k. The learning rate is 0.01 on WN18 and 0.1 on FB15k. We use 30 matrices on WN18 and 300 matrices on FB15k. All the models are implemented with Theano [17]. The Softmax temperature is set to $1/4$.

5.3.3 Results and Analysis

The overall link prediction results[1] are reported in Table 5.2. Our model consistently outperforms previous models without external information on both the metrics of WN18 and FB15k. On WN18, we even achieve a much better mean rank with comparable Hits@10 than current state-of-the-art model IRN employing external information.

[1] Note that although IRN [231] does not explicitly exploit path information, it performs multi-step inference through the multiple usages of external memory. When IRN is allowed to access memory once for each prediction, its Hits@10 is 80.7, similar to models without path information.

Model	Additional Information	WN18		FB15k	
		Mean Rank	Hits@10	Mean Rank	Hits@10
SE [25]	No	985	80.5	162	39.8
Unstructured [27]	No	304	38.2	979	6.3
TransE [26]	No	251	89.2	125	47.1
TransH [271]	No	303	86.7	87	64.4
TransR [149]	No	225	92.0	77	68.7
CTransR [149]	No	218	92.3	75	70.2
KG2E [90]	No	348	93.2	59	74.0
TransD [111]	No	212	92.2	91	77.3
TATEC [70]	No	-	-	**58**	76.7
NTN [239]	No	-	66.1	-	41.4
DISTMULT [286]	No	-	94.2	-	57.7
STransE [178]	No	206 (244)	93.4 (94.7)	69	79.7
ITransF	No	**205**	94.2	65	81.0
ITransF (domain sampling)	No	223	**95.2**	77	**81.4**
RTransE [69]	Path	-	-	50	76.2
PTransE [148]	Path	-	-	58	84.6
NLFeat [255]	Node + Link Features	-	94.3	-	87.0
Random Walk [272]	Path	-	94.8	-	74.7
IRN [231]	External Memory	249	*95.3*	*38*	*92.7*

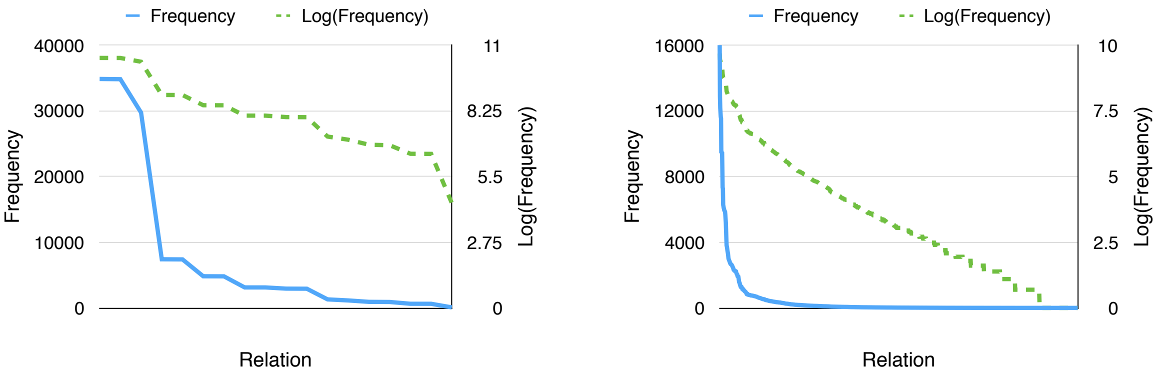
Table 5.2: Link prediction results on two datasets. Higher Hits@10 or lower Mean Rank indicates better performance. Following Nguyen et al. [178] and Shen et al. [231], we divide the models into two groups. The first group contains intrinsic models without using extra information. The second group make use of additional information. Results in the brackets are another set of results STransE reported.

We can see that path information is very helpful on FB15k and models taking advantage of path information outperform intrinsic models by a significant margin. Indeed, a lot of facts are easier to recover with the help of multi-step inference. For example, if we know Barack Obama is born in Honolulu, a city in the United States, then we easily know the nationality of Obama is the United States. An straightforward way of extending our model to k-step path $P = \{r_i\}_{i=1}^k$ is to define a path energy function $\|\boldsymbol{\alpha}_P^H \cdot \mathbf{D} \cdot \mathbf{h} + \sum_{r_i \in P} \mathbf{r}_i - \boldsymbol{\alpha}_P^T \cdot \mathbf{D} \cdot \mathbf{t}\|_\ell$, $\boldsymbol{\alpha}_P^H$ is a concept association related to the path. We plan to extend our model to multi-step path in the future.

To provide a detailed understanding why the model achieves better performance, we present some further analysis in the sequel.

Performance on Rare Relations In ITransF, we design an attention mechanism to encourage knowledge sharing across different relations. Naturally, facts associated with rare relations should benefit most from such sharing, boosting the overall performance. To verify this hypothesis, we investigate our model's performance on relations with different frequency.

The overall distribution of relation frequencies resembles that of word frequencies, subject to the zipf's law. Since the frequencies of relations approximately follow a power distribution, their log frequencies are linear. The statistics of relations on FB15k and WN18 are shown in Figure 5.1. We can clearly see that the distributions exhibit long tails, just like the Zipf's law for word frequency.



(a) WN18

(b) FB15k

Figure 5.1: Frequencies and log frequencies of relations on two datasets. The X-axis are relations sorted by frequency.

In order to study the performance of relations with different frequencies, we sort all relations by their frequency in the training set, and split them into 3 buckets evenly so that each bucket has a similar interval length of log frequency.

Within each bucket, we compare our model with STransE, as shown in Figure 5.3.[2] As we can see, on WN18, ITransF outperforms STransE by a significant margin on rare relations. In particular, in the last bin (rarest relations), the average Hits@10 increases from 74.4 to 92.0, showing the great benefits of transferring statistical strength from common relations to rare ones. We plot the performance of ITransF and STransE on each relation on WN18. We see that the improvement is greater on rare relations.

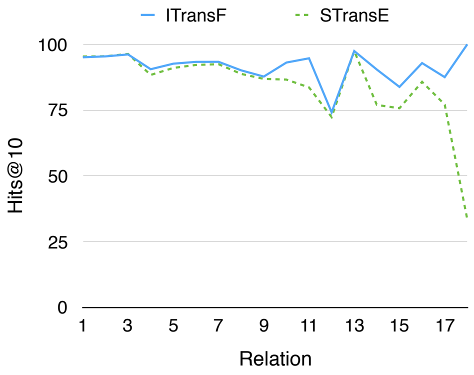
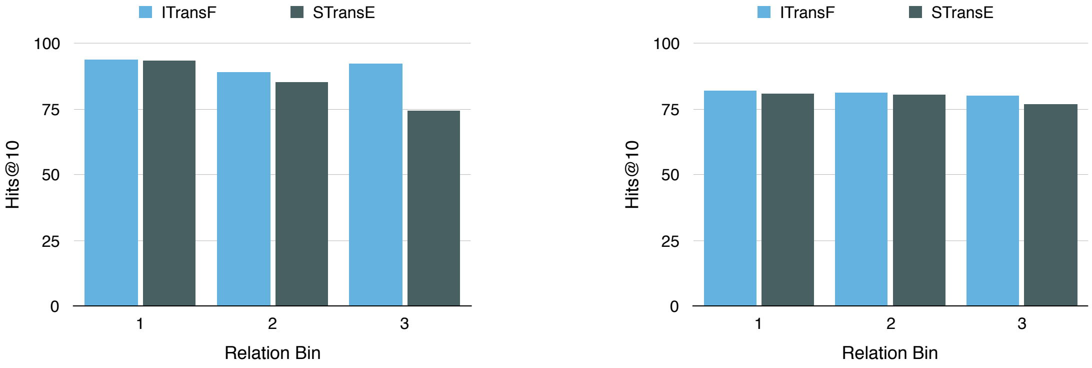


Figure 5.2: Hits@10 on each relation in WN18. The relations are sorted according to their frequency.

On FB15k, we can also observe a similar pattern, although the degree of improvement is less significant. We conjecture the difference roots in the fact that many rare relations on FB15k have disjoint domains, knowledge transfer through common concepts is harder.

[2] Domain sampling is not employed.



(a) WN18 (b) FB15k

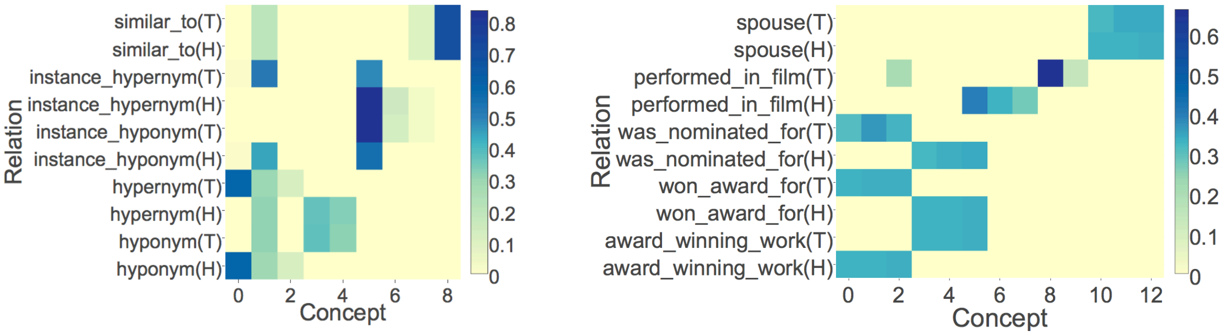
Figure 5.3: Hits@10 on relations with different amount of data. We give each relation the equal weight and report the average Hits@10 of each relation in a bin instead of reporting the average Hits@10 of each sample in a bin. Bins with smaller index corresponding to high-frequency relations.

Interpretability In addition to the quantitative evidence supporting the effectiveness of knowledge sharing, we provide some intuitive examples to show how knowledge is shared in our model. As we mentioned earlier, the sparse attention vectors fully capture the association between relations and concepts and hence the knowledge transfer among relations. Thus, we visualize the attention vectors for several relations on both WN18 and FB15K in Figure 5.4.

For WN18, the words "hyponym" and "hypernym" refer to words with more specific or general meaning respectively. For example, PhD is a hyponym of student and student is a hypernym of PhD. As we can see, concepts associated with the head entities in one relation are also associated with the tail entities in its reverse relation. Further, "instance_hypernym" is a special hypernym with the head entity being an instance, and the tail entity being an abstract notion. A typical example is $(\textit{New York}, \texttt{instance\_hypernym}, \textit{city})$. This connection has also been discovered by our model, indicated by the fact that "instance_hypernym(T)" and "hypernym(T)" share a common concept matrix. Finally, for symmetric relations like "similar_to", we see the head attention is identical to the tail attention, which well matches our intuition.

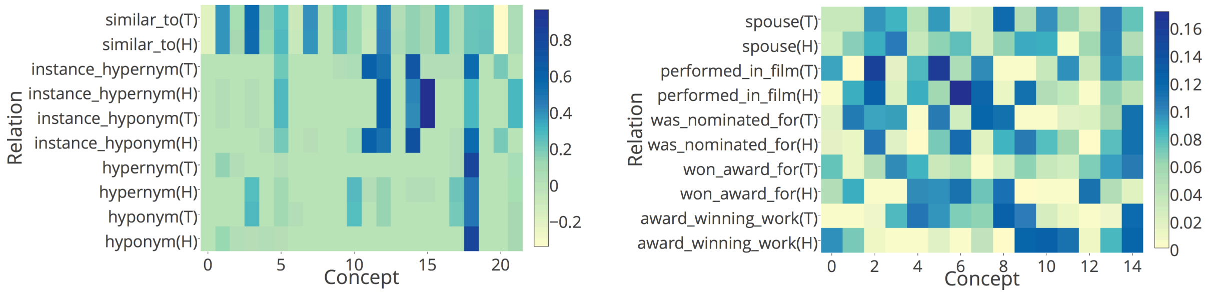
On FB15k, we also see the sharing between reverse relations, as in "(somebody) won_award_for (some work)" and "(some work) award_winning_work (somebody)". What's more, although relation "won_award_for" and "was_nominated_for" share the same concepts, their attention distributions are different, suggesting distinct emphasis. Finally, symmetric relations like spouse behave similarly as mentioned before.

Model Compression A byproduct of parameter sharing mechanism employed by ITransF is a much more compact model with equal performance. Figure 5.6 plots the average performance of ITransF against the number of projection matrices m, together with two baseline models. On FB15k, when we reduce the number of matrices from 2200 to 30 ($\sim 90\times$ compression), our model performance decreases by only 0.09% on Hits@10, still outperforming STransE. Simi-



(a) WN18 (b) FB15k

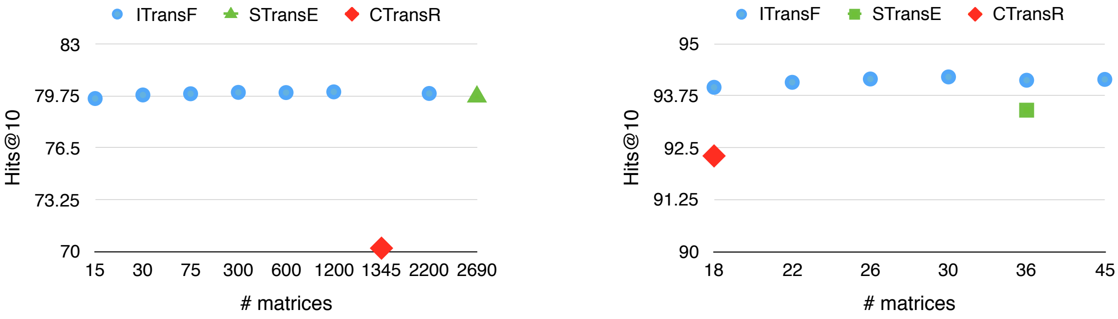
Figure 5.4: Heatmap visualization of attention vectors for ITransF on WN18 and FB15k. Each row is an attention vector $\boldsymbol{\alpha}_r^H$ or $\boldsymbol{\alpha}_r^T$ for a relation's head or tail concepts.



(a) WN18 (b) FB15k

Figure 5.5: Heatmap visualization of ℓ_1 regularized dense attention vectors, which are not sparse. Note that the colorscale is not from 0 to 1 since Softmax is not applied.

larly, on WN18, ITransF continues to achieve the best performance when we reduce the number of concept project matrices to 18.



(a) FB15k (b) WN18

Figure 5.6: Performance with different number of projection matrices. Note that the X-axis denoting the number of matrices is not linearly scaled.

5.3.4 Analysis on Sparseness

Sparseness is desirable since it contribute to interpretability and computational efficiency of our model. We investigate whether enforcing sparseness would deteriorate the model performance and compare our method with another sparse encoding methods in this section.

Dense Attention w/o ℓ_1 regularization Although ℓ_0 constrained model usually enjoys many practical advantages, it may deteriorate the model performance when applied improperly. Here, we show that our model employing sparse attention can achieve similar results with dense attention with a significantly less computational burden. We also compare dense attention with ℓ_1 regularization. We set the ℓ_1 coefficient to 0.001 in our experiments and does not apply Softmax since the ℓ_1 of a vector after Softmax is always 1. We compare models in a setting where the computation time of dense attention model is acceptable[3]. We use 22 weight matrices on WN18 and 15 weight matrices on FB15k and train both the models for 2000 epochs.

The results are reported in Table 5.3. Generally, ITransF with sparse attention has slightly better or comparable performance comparing to dense attention. Further, we show the attention vectors of model with ℓ_1 regularized dense attention in Figure 5.5. We see that ℓ_1 regularization does not produce a sparse attention, especially on FB15k.

Method	**WN18**			**FB15k**		
	MR	H10	Time	MR	H10	Time
Dense	**199**	94.0	4m34s	69	79.4	4m30s
Dense + ℓ_1	228	**94.2**	4m25s	131	78.9	5m47s
Sparse	207	94.1	**2m32s**	**67**	**79.6**	**1m52s**

Table 5.3: Performance of model with dense attention vectors or sparse attention vectors. MR, H10 and Time denotes mean rank, Hits@10 and training time per epoch respectively

Nonnegative Sparse Encoding We induce the sparsity by a carefully designed iterative optimization procedure. Apart from this approach, one may utilize sparse encoding techniques to obtain sparseness based on the pretrained projection matrices from STransE. Concretely, stacking $|2R|$ pretrained projection matrices into a 3-dimensional tensor $X \in \mathbb{R}^{2|R| \times n \times n}$, similar sparsity can be induced by solving an ℓ_1-regularized tensor completion problem $\min_{\mathbf{A},\mathbf{D}} ||\mathbf{X} - \mathbf{DA}||_2^2 + \lambda ||\mathbf{A}||_{\ell_1}$. Basically, $\mathbf{A}$ plays the same role as the attention vectors in our model. For more details, we refer readers to [61].

For completeness, we compare our model with the aforementioned approach[4]. The comparison is summarized in table 5.4. On both benchmarks, ITransF achieves significant improvement against sparse encoding on pretrained model. This performance gap should be expected since the objective function of sparse encoding methods is to minimize the reconstruction loss rather than optimize the criterion for link prediction.

[3] With 300 projection matrices, it takes $1h1m$ to run one epoch for a model with dense attention.

[4] We use the toolkit provided by [61].

Method	**WN18**		**FB15k**	
	MR	H10	MR	H10
Sparse Encoding	211	86.6	66	79.1
ITransF	**205**	**94.2**	**65**	**81.0**

Table 5.4: Different methods to obtain sparse representations

5.4 Related Work

In KBC, CTransR [149] enables relation embedding sharing across similar relations, but they cluster relations before training rather than learning it in a principled way. Further, they do not solve the data sparsity problem because there is no sharing of projection matrices which have a lot more parameters. Learning the association between semantic relations has been used in related problems such as relational similarity measurement [261] and relation adaptation [24].

Data sparsity is a common problem in many fields. Transfer learning [187] has been shown to be promising to transfer knowledge and statistical strengths across similar models or languages. For example, Bharadwaj et al. [21] transfers models on resource-rich languages to low resource languages by parameter sharing through common phonological features in name entity recognition. Zoph et al. [321] initialize from models trained by resource-rich languages to translate low-resource languages.

Several work on obtaining a sparse attention [162, 163, 228] share a similar idea of sorting the values before softmax and only keeping the K largest values. However, the sorting operation in these work is not GPU-friendly.

The block iterative optimization algorithm in our work is inspired by LightRNN [141]. They allocate every word in the vocabulary in a table. A word is represented by a row vector and a column vector depending on its position in the table. They iteratively optimize embeddings and allocation of words in tables.

5.5 Discussion

In summary, we present a knowledge embedding model which can discover shared hidden concepts, and design a learning algorithm to induce the interpretable sparse representation. Empirically, we show our model can improve the performance on two benchmark datasets without external resources, over all previous models of the same kind.

Such a transfer learning mechanism leads to knowledge sharing and better performance. One drawback of the algorithm is that we need to design special optimization procedures since the model is not fully differentiable due to the sparseness constraints, which makes it difficult to apply for new tasks.

Chapter 6

Transfer Learning through Invariant Feature Learning

In this chapter, we present a method that uses adversarial training to learn domain-invariant representation to perform transfer learning between similar domains. The representation learning process is formulated as an adversarial minimax game. We analyze the optimal equilibrium of such a game and find that it amounts to maximizing the uncertainty of inferring the detrimental factor given the representation while maximizing the certainty of making task-specific predictions. On three benchmark tasks, we show that the proposed framework induces an invariant representation, and leads to better generalization evidenced by the improved performance.

6.1 Introduction

How to produce a data representation that maintains meaningful variations of data while eliminating noisy signals is a consistent theme of machine learning research. In the last few years, the dominant paradigm for finding such a representation has shifted from manual feature engineering based on specific domain knowledge to representation learning that is fully data-driven, and often powered by deep neural networks [15]. Being universal function approximators [81], deep neural networks can easily uncover the complicated variations in data [303], leading to powerful representations. However, how to systematically incorporate a desired invariance into the learned representation in a controllable way remains an open problem.

A possible avenue towards the solution is to devise a dedicated neural architecture that by construction has the desired invariance property. As a typical example, the parameter sharing scheme and pooling mechanism in modern deep convolutional neural networks (CNN) [137] take advantage of the spatial structure of image processing problems, allowing them to induce more generic feature representations than fully connected networks. Since the invariance we care about can vary greatly across tasks, this approach requires us to design a new architecture each time a new invariance desideratum shows up, which is time-consuming and inflexible.

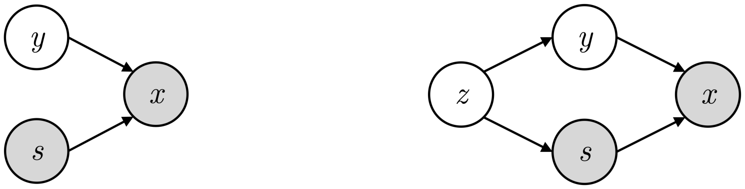
When our belief of invariance is specific to some attribute of the input data, an alternative approach is to build a probabilistic model with a random variable corresponding to the attribute, and explicitly reason about the invariance. For instance, the variational fair auto-encoder

(VFAE) [152] employs the maximum mean discrepancy (MMD) to eliminate the negative influence of specific "nuisance variables", such as removing the lighting conditions of images to predict the person's identity. Similarly, under the setting of domain adaptation, standard binary adversarial cost [67, 68] and central moment discrepancy (CMD) [298] have been utilized to learn features that are domain invariant. However, all these invariance inducing criteria suffer from a similar drawback, which is they are defined to measure the divergence between a *pair* of distributions. Consequently, they can only express the invariance belief w.r.t. a pair of values of the random variable at a time. When the attribute is a multinomial variable that takes more than two values, combinatorial number of pairs (specifically, $O(n^2)$) have to be added to express the belief that the representation should be invariant to the attribute. The problem is even more dramatic when the attribute represents a structure that has exponentially many possible values (e.g. the parse tree of a sentence) or when the attribute is simply a continuous variable.

Motivated by the aforementioned drawbacks and difficulties, in this work, we consider the problem of learning a feature representation with the desired invariance. We aim at creating a unified framework that is (1) generic enough such that it can be easily plugged into different models, and (2) more flexible to express an invariance belief in quantities beyond discrete variables with limited value choices. Specifically, inspired by the recent advancement of adversarial learning [75], we formulate the representation learning as a minimax game among three players: an *encoder* which maps the observed data deterministically into a feature space, a *discriminator* which looks at the representation and tries to identify a specific type of variation we hope to eliminate from the feature, and a *predictor* which makes use of the invariant representation to make predictions as in typical discriminative models. We provide theoretical analysis of the equilibrium condition of the minimax game, and give an intuitive interpretation. On three benchmark tasks from different domains, we show that the approach not only improves upon vanilla discriminative approaches that do not encourage invariance, but also outperforms existing approaches that enforce invariant features.

6.2 Adversarial Invariant Feature Learning

In this section, we formulate our problem and then present the framework of learning invariant features.



(a) y and s are marginally independent (b) y and s are not marginally independent

Figure 6.1: Dependencies between x, s, y, where x is the observation and y is the target to be predicted. s is the attribute to which the prediction should be invariant.

Given observation/input x, we are interested in the task of predicting the target y based on the value of x using a discriminative approach. In addition, we have access to some intrinsic attribute s of x as well as a prior belief that the prediction result should be invariant to s.

There are two possible dependency scenarios of x, s and y here: (1) s and y can be marginally independent. For example, in image classifications, lighting conditions s and identities of persons y are independent. The data generation process is $s \sim p(s), y \sim p(y), x \sim p(x \mid s, y)$. (2) In some cases, s and y are not marginally independent. For example, in fairness classifications, s are the sensitive factors such as age and gender. y can be the saving, credit and health condition of a person. s and y are related due to the inherent bias within the data. Using a latent variable z to model the dependency between s and y, the data generation process is $z \sim p(z), s \sim p(s \mid z), y \sim p(y \mid z), x \sim p(x \mid s, y)$. We show the corresponding dependency graphs in Figure 6.1.

Unlike vanilla discriminative models that outputs the conditional distribution $p(y \mid x)$, we model $p(y \mid x, s)$ to make predictions invariant to s. Our intuition is that, due to the explaining away effect, y and s are not independent when conditioned on x although they can be marginally independent. Consequently, $p(y \mid x, s)$ is a more accurate estimation of y than $p(y \mid x)$. Intuitively, this can inform and guide the model to remove information about undesired variations. For example, if we want to learn a representation of image x that is invariant to the lighting condition s, the model can learn to "brighten" the input if it knows the original picture is dark, and vice versa. Also, in multi-lingual machine translation, a word with the same surface form may have different meanings in different languages. For instance, "gift" means "present" in English but means "poison" in German. Hence knowing the language of a source sentence helps inferring the meaning of the sentence and conducting translation.

As the input x can have highly complicated structure, we employ a dedicated model or algorithm to extract an expressive representation h from x. Thus, when we extract the representation h from x, we want the representation h to preserve variations that are necessary to predict y while eliminating information of s. To achieve the aforementioned goal, we employ a deterministic encoder E to obtain the representation by encoding x and s into h, namely, $h = E(x, s)$. It should be noted here that we are using s as an additional input. Given the obtained representation h, the target y is predicted by a predictor M, which effectively models the distribution $q_M(y \mid h)$. By construction, instead of modeling $p(y \mid x)$ directly, the discriminative model we formulate captures the conditional distribution $p(y \mid x, s)$ with additional information coming from s.

Surely, feeding s into the encoder by no means guarantees the induced feature h will be invariant to s. Thus, in order to enforce the desired invariance and eliminate variations of factor s from h, we set up an adversarial game by introducing a discriminator D which inspects the representation h and ensure that it is invariant to s. Concretely, the discriminator D is trained to predict s based on the encoded representation h, which effectively maximizes the likelihood $q_D(s \mid h)$. Simultaneously, the encoder fights to minimize the same likelihood of inferring the correct s by the discriminator. Intuitively, the discriminator and the encoder form an adversarial game where the discriminator tries to detect an attribute of the data while the encoder learns to conceal it.

Note that under our framework, in theory, s can be any type of data as long as it represents an attribute of x. For example, s can be a real value scalar/vector, which may take many possible values, or a complex sub-structure such as the parse tree of a natural language sentence. But in this work, we focus mainly on instances where s is a discrete label with multiple choices. We plan to extend our framework to deal with continuous s and structured s in the future.

Formally, E, M and D jointly play the following minimax game:

$$\min_{E,M} \max_{D} J(E, M, D)$$

where

$$J(E, M, D) = \mathop{\mathbb{E}}_{x,s,y \sim p(x,s,y)} [\gamma \log q_D(s \mid h = E(x, s)) - \log q_M(y \mid h = E(x, s))] \tag{6.1}$$

where γ is a hyper-parameter to adjust the strength of the invariant constraint, and $p(x, s, y)$ is the true underlying distribution that the empirical observations are drawn from.

Note that the problem of domain adaption can be seen as a special case of our problem, where s is a Bernoulli variable representing the domain and the model only has access to the target y when $s =$ "source domain" during training.

6.3 Theoretical Analysis

In this section, we theoretically analyze, given enough capacity and training time, whether such a minimax game will converge to an equilibrium where variations of y are preserved and variations of s are removed. The theoretical analysis is done in a non-parametric limit, i.e., we assume a model with infinite capacity. In addition, we discuss the equilibriums of the minimax game when s is independent/dependent to y.

Since both the discriminator and the predictor only use h which is transformed deterministically from x and s, we can substitute x with h and define a joint distribution $\tilde{p}(h, s, y)$ of h, s and y as follows

$$\tilde{p}(h, s, y) = \int_x \tilde{p}(x, s, h, y)dx = \int_x p(x, s, y)p_E(h \mid x, s)dx = \int_x p(x, s, y)\delta(E(x, s) = h)dx$$

Here, we have used the fact that the encoder is a deterministic transformation and thus the distribution $p_E(h \mid x, s)$ is merely a delta function denoted by $\delta(\cdot)$. Intuitively, h absorbs the randomness in x and has an implicit distribution of its own. Also, note that the joint distribution $\tilde{p}(h, s, y)$ depends on the transformation defined by the encoder.

Thus, we can equivalently rewrite objective (6.1) as

$$J(E, M, D) = \mathop{\mathbb{E}}_{h,s,y \sim \tilde{p}(h,s,y)} [\gamma \log q_D(s \mid h) - \log q_M(y \mid h)] \tag{6.2}$$

To analyze the equilibrium condition of the new objective (6.2), we first deduce the optimal discriminator D and the optimal predictor M for a given encoder E and then prove the global optimality of the minimax game.

Claim 1. *Given a fixed encoder E, the optimal discriminator outputs $q_D^*(s \mid h) = \tilde{p}(s \mid h)$ and the optimal predictor corresponds to $q_M^*(y \mid h) = \tilde{p}(y \mid h)$.*

Proof. We first prove the optimal solution of the discriminator. With a fixed encoder, we have the following optimization problem

$$
\begin{aligned}
\min_{q_D} \quad & -J(E, M, D) \\
\text{s.t.} \quad & \sum_s q_D(s \mid h) = 1, \forall h
\end{aligned}
$$

Then $L = J(E, M, D) - \sum_h \lambda(h)(\sum_s q_D(s \mid h) - 1)$ is the Lagrangian dual function of the above optimization problem where $\lambda(h)$ are the dual variables introduced for equality constraints.

The optimal D satisfies the following equation

$$
\begin{aligned}
& 0 = \frac{\partial L}{\partial q_D^*(s \mid h)} \\
\iff \quad & 0 = -\frac{\partial J}{\partial q_D^*(s \mid h)} - \lambda(h) \\
\iff \quad & \lambda(h) = -\frac{\sum_y \tilde{q}(h, s, y)}{q_D^*(s \mid h)} \\
\iff \quad & \lambda(h) q_D^*(s \mid h) = -\tilde{q}(s, h)
\end{aligned}
\tag{6.3}
$$

Summing w.r.t. s on both sides of the last line of Eqn. (6.3) and using the fact that $\sum_s q_D^*(s \mid h) = 1$, we get

$$\lambda(h) = -\tilde{q}(h) \tag{6.4}$$

Substituting Eqn. 6.4 back into Eqn. 6.3, we can prove the optimal discriminator is

$$q_D^*(s \mid h) = \tilde{q}(s \mid h)$$

Similarly, taking derivation w.r.t. $q_M(y \mid h)$ and setting it to 0, we can prove $q_M^*(y \mid h) = \tilde{q}(y \mid h)$. □

Note that the optimal $q_D^*(s \mid h)$ and $q_M^*(y \mid h)$ given in Claim 1 are both functions of the encoder E. Thus, by plugging q_D^* and q_M^* into the original minimax objective (6.2), it can be simplified as a minimization problem only w.r.t. the encoder E with the following form:

$$
\begin{aligned}
\min_E J(E) &= \min_E \mathop{\mathbb{E}}_{h,s,y \sim \tilde{q}(h,s,y)} [\gamma \log \tilde{q}(s \mid h) - \log \tilde{q}(y \mid h)] \\
&= \min_E -\gamma H(\tilde{q}(s \mid h)) + H(\tilde{q}(y \mid h))
\end{aligned}
\tag{6.5}
$$

where $H(\tilde{q}(s \mid h))$ is the conditional entropy of the distribution $\tilde{q}(s \mid h)$.

Equilibrium Analysis As we can see, the objective (6.5) consists of two conditional entropies with different signs. Optimizing the first term amounts to maximizing the uncertainty of inferring s based on h, which is essentially filtering out any information of s from the representation. On the contrary, optimizing the second term leads to increasing the certainty of predicting y based on h. Implicitly, the objective defines the equilibrium of the minimax game.

- **Win-win equilibrium:** Firstly, for cases where the attribute s is entirely irrelevant to the prediction task (corresponding to the dependency graph shown in Figure 6.1a), the two terms can reach the optimum at the same time, leading to a win-win equilibrium. For example, with the lighting condition of an image removed, we can still/better classify the identity of the people in that image. With enough model capacity, the optimal equilibrium solution would be the same regardless of the value of γ.
- **Competing equilibrium:** However, there are cases where these two optimization objectives are competing. For example, in fair classifications, sensitive factors such as gender and age may help the overall prediction accuracies due to inherent biases within the data. In other words, knowing s may help in predicting y since s and y are not marginally independent (corresponding to the dependency graph shown in Figure 6.1b). Learning a fair/invariant representation is harmful to predictions. In this case, the optimality of these two entropies cannot be achieved simultaneously, and γ defines the relative strengths of the two objectives in the final equilibrium.

6.4 Parametric Instantiation of the Proposed Framework

6.4.1 Models

To show the general applicability of our framework, we experiment on three different tasks including sentence generation, image classification and fair classifications. Due to the different natures of data of x and y, here we present the specific model instantiations we use.

Sentence Generation We use multi-lingual machine translation as the testbed for sentence generation. Concretely, we have translation pairs between several source languages and a target language. x is the source sentence to be translated and s is a scalar denoting which source language x belongs to. y is the translated sentence for the target language.

Recall that s is used as an input of E to obtain a language-invariant representation. To make full use of s, we employ separate encoders Enc_s for sentences in each language s. In other words, $h = E(s, x) = \mathrm{Enc}_s(x)$ where each Enc_s is a different encoder. The representation of a sentence is captured by the hidden states of an LSTM encoder [99] at each time step.

We employ a single LSTM predictor for different encoders. As often used in language generation, the probability q_M output by the predictor is parametrized by an autoregressive process, i.e.,

$$q_M(y_{1:T} \mid h) = \prod_{t=1}^{T} q_M(y_t | y_{<t}, h)$$

where we use an LSTM with attention model [9] to compute $q_M(y_t | y_{<t}, h)$.

The discriminator is also parameterized as an LSTM which gives it enough capacity to deal with input of multiple timesteps. $q_D(s \mid h)$ is instantiated with the multinomial distribution computed by a softmax layer on the last hidden state of the discriminator LSTM.

Classification For our classification experiments, the input is either a picture or a feature vector. All of the three players in the minimax game are constructed by feedforward neural networks. We feed s to the encoder as an embedding vector.

6.4.2 Optimization

There are two possible approaches to optimize our framework in an adversarial setting. The first one is similar to the alternating approach used in Generative Adversarial Nets (GANs) [75]. We can alternately train the two adversarial components while freezing the third one. This approach has more control in balancing the encoder and the discriminator, which effectively avoids saturation. Another method is to train all three components together with a gradient reversal layer [67]. In particular, the encoder admits gradients from both the discriminator and the predictor, with the gradient from the discriminator negated to push the encoder in the opposite direction desired by the discriminator. Chen et al. [39] found the second approach easier to optimize since the discriminator and the encoder are fully in sync being optimized altogether. Hence we adopt the latter approach. In all of our experiments, we use Adam [123] with a learning rate of 0.001.

6.5 Experiments

In this section, we perform empirical experiments to evaluate the effectiveness of the framework. We first introduce the tasks and corresponding datasets we consider. Then, we present the quantitative results showing the superior performance of our framework, and discuss some qualitative analysis which verifies the learned representations have the desired invariance property.

6.5.1 Datasets

Our experiments include three tasks in different domains: (1) fair classification, in which predictions should be unaffected by nuisance factors; (2) language-independent generation which is conducted on the multi-lingual machine translation problem; (3) lighting-independent image classification.

Fair Classification For fair classification, we use three datasets to predict the savings, credit ratings and health conditions of individuals with variables such as gender or age specified as "nuisance variable" that we would like to not consider in our decisions [152, 299]. The German dataset [64] is a small dataset with $1,000$ samples describing whether a person has a good credit rating. The sensitive nuisance variable to be factored out is gender. The Adult income dataset [64] has $45,222$ data points and the objective is to predict whether a person has savings of over $50,000$ dollars with the sensitive factor being age. The task of the health dataset[1] is to predict whether a person will spend any days in the hospital in the following year. The sensitive variable is also the age and the dataset contains $147,473$ entries. We follow the same 5-fold train/validation/test splits and feature preprocessing used in [152, 299].

[1] www.heritagehealthprize.com

Both the encoder and the predictor are parameterized by single-layer neural networks. A three-layer neural network with batch normalization [108] is employed for the discriminator. We use a batch size of 16 and the number of hidden units is set to 64. γ is set to 1 in our experiments.

Multi-lingual Machine Translation For the multi-lingual machine translation task we use French to English (fr-en) and German to English (de-en) pairs from IWSLT 2015 dataset [31]. There are $198,435$ pairs of fr-en sentences and $188,661$ pairs of de-en sentences in the training set. In the test set, there are $4,632$ pairs of fr-en sentences and $7,054$ pairs of de-en sentences. We evaluate BLEU scores [189] using the standard Moses `multi-bleu.perl` script. Here, s indicates the language of the source sentence.

We use the OpenNMT [127] in our multi-lingual MT experiments[2]. The encoder is a two-layer bidirectional LSTM with 256 units for each direction. The discriminator is a one-layer single-directional LSTM with 256 units. The predictor is a two-layer LSTM with 512 units and attention mechanism [9]. We follow Johnson et al. [114] and use Byte Pair Encoding (BPE) subword units [226] as the cross-lingual input. Every model is run for 20 epochs. γ is set to 8 and the batch size is set to 64.

Image Classification We use the Extended Yale B dataset [72] for our image classification task. It comprises face images of 38 people under 5 different lighting conditions: upper right, lower right, lower left, upper left, or the front. The variable s to be purged is the lighting condition. The label y is the identity of the person. We follow Li et al. [143], Louizos et al. [152]'s train/test split and no validation is used: $38 \times 5 = 190$ samples are used for training and all other $1,096$ data points are used for testing.

We use a one-layer neural network for the encoder and a one-layer neural network for prediction. γ is set to 2. The discriminator is a two-layer neural network with batch normalization. The batch size is set to 16 and the hidden size is set to 100.

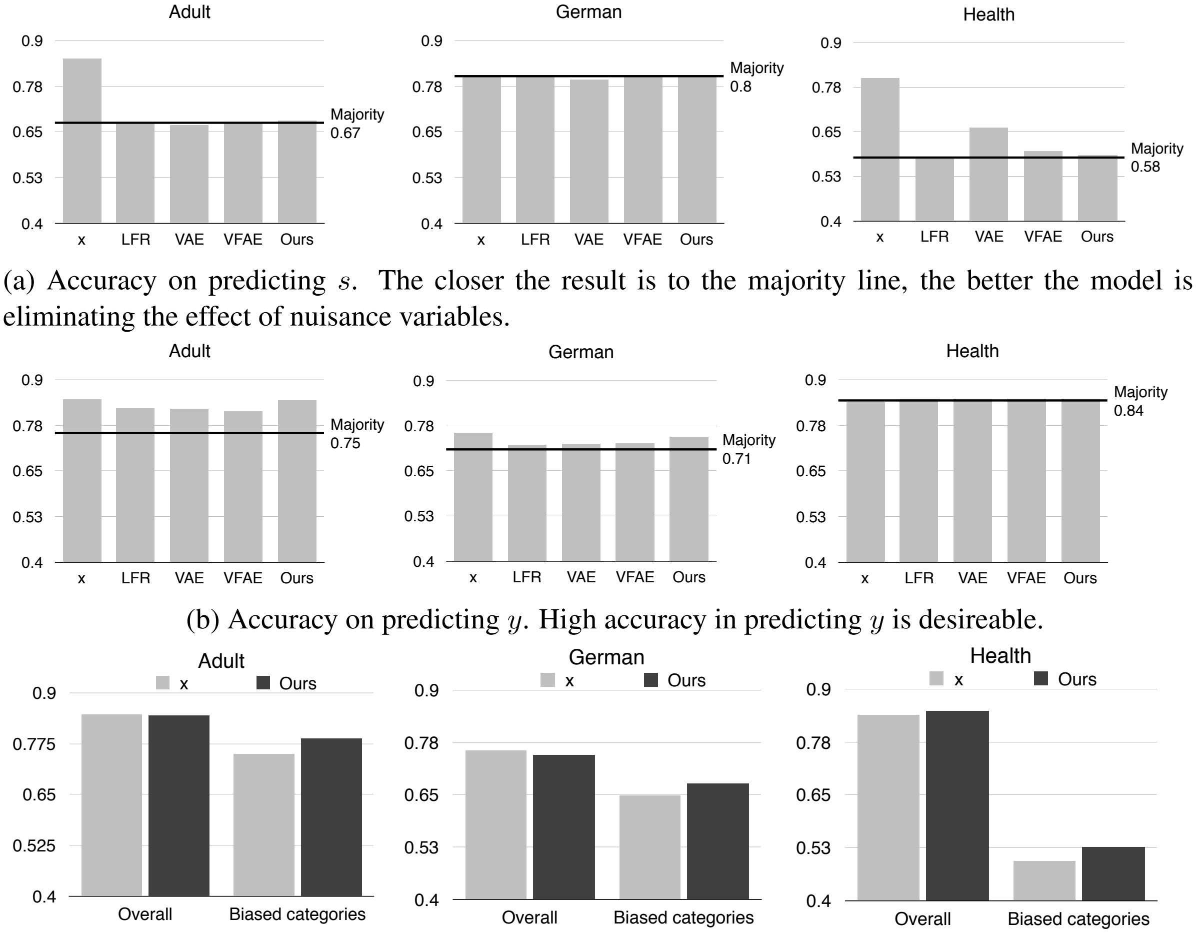
6.5.2 Results

Fair Classification The results on three fairness tasks are shown in Figure 6.2. We compare our model with two prior work on learning fair representations: Learning Fair Representations (LFR) [299] and Variational Fair Autoencoder (VFAE) [152]. Results of VAE and directly using x as the representation are also shown.

We first study how much information about s is retained in the learned representation h by using a logistic regression to predict factor s. In the top row, we see that s cannot be recognized from the representations learned by three models targeting at fair representations. The accuracy of classifying s is similar to the trivial baseline predicting the majority label shown by the black line.

The performance on predicting label y is shown in the second row. We see that LFR and VFAE suffer on Adult and German datasets after removing information of s. In comparison, our model's performance does not suffer even when making fair predictions. Specifically, on German, our model's accuracy is 0.744 compared to 0.727 and 0.723 achieved by VFAE and

[2] Our MT code is available at https://github.com/qizhex/Controllable-Invariance



(a) Accuracy on predicting s. The closer the result is to the majority line, the better the model is in eliminating the effect of nuisance variables.

(b) Accuracy on predicting y. High accuracy in predicting y is desireable.

(c) Overall performance and performance on biased categories. Fair representations lead to high accuracy on baised categories.

Figure 6.2: Fair classification results on different representations. x denotes directly using the observation x as the representation. The black lines in the first and the second row show the performance of predicting the majority label.

LFR. On Adult, our model's accuracy is 0.844 while VFAE and LFR have accuracies of 0.813 and 0.823 respectively. On the health dataset, all models' performances are barely better than the majority baseline. The unsatisfactory performances of all models may be due to the extreme imbalance of the dataset, in which 85% of the data has the same label.

We also investigate how fair representations would alleviate biases of machine learning models. We measure the unbiasedness by evaluating models' performances on identifying minority groups. For instance, suppose the task is to predict savings with the nuisance factor being age, with savings above a threshold of $\$50,000$ being adequate, otherwise being insufficient. If people of advanced age generally have fewer savings, then a biased model would tend to predict insufficient savings for those with an advanced age. In contrast, an unbiased model can better factor out age information and recognize people that do not fit into these stereotypes.

Concretely, for groups pooled by each possible value of y, we seek for the minority s in

Model	test (fr-en)	test (de-en)
Bilingual Enc-Dec [9]	35.2	27.3
Multi-lingual Enc-Dec [114]	35.5	27.7
Our model	**36.1**	**28.1**
w.o. discriminator	35.3	27.6
w.o. separate encoders	35.4	27.7

Table 6.1: Results on multi-lingual machine translation.

each of these groups and define the minority s as the biased category for the group. Then we first calculate the accuracy on each biased category and report the average performance for all categories. We do not compute the instance-level average performance since one category may hold the dominant amount of data among all categories.

As shown in the third row of Figure 6.2, on German and Adult, we achieve higher accuracy on the biased categories, even though our overall accuracy is similar to or lower than the baseline which does not employ fairness constraints. Specifically, on Adult, our performance on the biased categories is 0.788 while the baseline's accuracy is 0.748. On German, our accuracy on biased categories is 0.676 while the baseline achieves 0.648. The results show that our model is able to learn a more unbiased representation.

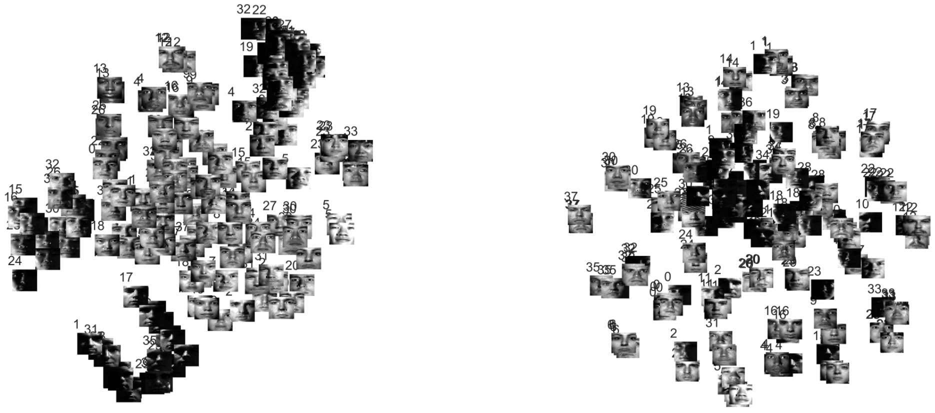
Multi-lingual Machine Translation The results of systems on multi-lingual machine translation are shown in Table 6.1. We compare our model with attention based encoder-decoder trained on bilingual data [9] and multi-lingual data [114]. The encoder-decoder trained on multi-lingual data employs a single encoder for both source languages. Firstly, both multi-lingual systems outperform the bilingual encoder-decoder even though multi-lingual systems use similar number of parameters to translate two languages, which shows that learning invariant representation leads to better generalization in this case. The better generalization may be due to transferring statistical strength between data in two languages.

Comparing two multi-lingual systems, our model outperforms the baseline multi-lingual system on both languages, where the improvement on French-to-English is 0.6 BLEU score. We also verify the design decisions in our framework by ablation studies. Firstly, without the discriminator, the model's performance is worse than the standard multi-lingual system, which rules out the possibility that the gain of our model comes from more parameters of separating encoders. Secondly, when we do not employ separate encoders, the model's performance deteriorates and it is more difficult to learn a cross-lingual representation, which

- verifies the theoretical advantage of modeling $p(y \mid x, s)$ instead of $p(y \mid x)$ as mentioned in Section 6.2. Intuitively, German and French have different grammars and vocabulary, so it is hard to obtain a unified semantic representation by performing the same operations.
- means that the encoder needs to have enough capacity to reach the equilibrium in the minimax game. We also observe that the discriminator needs enough capacity to provide faithful gradients towards the equilibrium. Specifically, instantiating the discriminator with feedforward neural network w./w.o. attention mechanism [9] does not work in our experiments.

Method	Accuracy of classifying s	Accuracy of classifying y
Logistic regression	0.96	0.78
NN + MMD [143]	-	0.82
VFAE [152]	**0.57**	0.85
Ours	**0.57**	**0.89**

Table 6.2: Results on Extended Yale B dataset. A better representation has lower accuracy of classifying factor s and higher accuracy of classifying label y



(a) Using the original image x as the representation

(b) Representation learned by our model

Figure 6.3: t-SNE visualizations of images in the Extended Yale B. The original pictures are clustered by the lighting conditions, while the representation learned by our model is clustered by identities of individuals

Image Classification We report the results in Table 6.2 with two baselines [143, 152] that use MMD regularizations to remove lighting conditions. The advantage of factoring out lighting conditions is shown by the improved accuracy 89% for classifying identities, while the best baseline achieves an accuracy of 85%.

In terms of removing s, our framework can filter the lighting conditions since the accuracy of classifying s drops from 0.96 to 0.57, as shown in Table 6.2. We also visualize the learned representation by t-SNE [159] in comparison to the visualization of original pictures in Figure 6.3. We see that, without removing lighting conditions, the images are clustered based on the lighting conditions. After removing information of lighting conditions, images are clustered according to the identity of each person.

6.6 Related Work

As a specific case of our problem where s takes two values, domain adaption has attracted a large amount of research interest. Domain adaptation aims to learn domain-invariant representations that are transferable to other domains. For example, in image classification, adversarial training has been shown to able to learn an invariant representation across domains [28, 67, 68, 262] and enables classifiers trained on the source domain to be applicable to the target domain. Moment discrepancy regularizations can also effectively remove domain specific information [28, 298] for the same purpose. By learning language-invariant representations, classifiers trained on the source language can be applied to the target language [39, 281].

Works targeting the development of fair, bias-free classifiers also aim to learn representations invariant to "nuisance variables" that could induce bias and hence makes the predictions fair, as data-driven models trained using historical data easily inherit the bias exhibited in the data. Zemel et al. [299] proposes to regularize the ℓ_1 distance between representation distributions for data with different nuisance variables to enforce fairness. The Variational Fair Autoencoder [152] targets the problem with a Variational Autoencoder [124, 209] approach with maximum mean discrepancy regularization.

Our work is also related to learning disentangled representations, where the aim is to separate different influencing factors of the input data into different parts of the representation. Ideally, each part of the learned representation can be marginally independent to the other. An early work by Tenenbaum and Freeman [253] propose a bilinear model to learn a representation with the style and content disentangled. From information theory perspective, [38] augments standard generative adversarial networks with an inference network, whose objective is to infer part of the latent code that leads to the generated sample. This way, the information carried by the chosen part of the latent code can be retained in the generative sample, leading to disentangled representation.

As we have discussed in Section 9.1, these methods bear the same drawback that the cost used to regularize the representation is pairwise, which does not scale well as the number of values that the attribute can take could be large. Louppe et al. [153] propose an adversarial training framework to learn representations independent to a categorical or continuous variable. A basic assumption in their theoretical analysis is that the attribute is irrelevant to the prediction, which limits its capabilities in analyzing the fairness classifications.

6.7 Discussion

In sum, we show a generic framework to learn representations invariant to a specified factor or trait. We cast the representation learning problem as an adversarial game among an encoder, a discriminator, and a predictor. We theoretically analyze the optimal equilibrium of the minimax game and evaluate the performance of our framework on three tasks from different domains empirically. We show that an invariant representation is learned, resulting in better generalization and improvements on the three tasks.

The invariance inducing framework is applied to learn domain-invariant features for transfer learning. It results in better generalization for two machine translation tasks though the algorithm requires additional effort in optimization.

Chapter 7

Transfer Learning by Pretraining for Cloze Test

In this chapter, we first present a cloze test dataset CLOTH collected from exams. We then show that it is possible to achieve great performance on this dataset by transfer learning from LM-1B, a language model pretrained on a large corpus. This shows that pretraining can leverage a large amount of unlabeled data to learn general knowledge about natural language.

7.1 Introduction

Being a classic language exercise, the cloze test [252] is an accurate assessment of language proficiency [63, 116, 258] and has been widely employed in language examinations. Under a typical setting, a cloze test requires examinees to fill in missing words (or sentences) to best fit the surrounding context. To facilitate natural language understanding, automatically-generated cloze datasets are introduced to measure the ability of machines in reading comprehension [95, 97, 185]. In these datasets, each cloze question typically consists of a context paragraph and a question sentence. By randomly replacing a particular word in the question sentence with a blank symbol, a single test case is created. For instance, CNN/Daily Mail datasets [95] use news articles as contexts and summary bullet points as the question sentence. Only named entities are removed when creating the blanks. Similarly, in Children's Books test (CBT) [97], cloze questions are obtained by removing a word in the last sentence of every consecutive 21 sentences, with the first 20 sentences being the context. Different from CNN/Daily Mail datasets, CBT also provides each question with a candidate answer set, consisting of randomly sampled words with the same part-of-speech tag from the context as that of the correct answer.

Thanks to the automatic generation process, these datasets can be very large in size, leading to significant research progresses. However, compared to how humans would create cloze questions and evaluate reading comprehension ability, the automatic generation process bears some inevitable issues. Firstly, blanks are chosen uniformly without considering which aspect of the language phenomenon that questions will test. Hence, quite a portion of automatically-generated questions can be purposeless or even trivial to answer. Another issue involves the ambiguity of answers. Given a context and a sentence with a blank, there can be multiple words that fit almost equally well into the blank. A possible solution is to include a candidate option set, as done by CBT, to get rid of the ambiguity. However, automatically generating the candidate option set can be problematic since it cannot guarantee the ambiguity is removed. More importantly, automatically-generated candidates can be totally irrelevant or simply grammatically unsuitable for the blank, resulting in again purposeless or trivial questions. Probably due to these unsatisfactory issues, neural models have achieved comparable results to the human-level performance within a very short time [35, 58, 227]. While there have been work trying to incorporate human design into cloze question generation [188, 324], due to the expensive labeling process, the MSR Sentence Completion Challenge created by this effort has $1,040$ questions and the LAMBADA [188] dataset has $10,022$ questions, limiting the possibility of developing powerful neural models on it. As a result of the small size, human-created questions are only used to compose development sets and test sets. Motivated by the aforementioned drawbacks, we collect a large-scale cloze test dataset CLOTH from English exams. Questions in the dataset are designed by middle-school and high-school teachers to prepare Chinese students for entrance exams. To design a cloze test, teachers firstly determine the words that can test students' knowledge of vocabulary, reasoning or grammar; then replace those words with blanks and provide other three candidate options for each blank. If a question does not specifically test grammar usage, all of the candidate options would complete the sentence with correct grammar, leading to highly nuanced questions. As a result, human-created questions are usually harder and are a better assessment of language proficiency. A general cloze test evaluates several aspects of language proficiency including vocabulary, reasoning and grammar, which are key components of comprehending natural language.

To verify if human-created cloze questions are difficult for current models, we train and evaluate the state-of-the-art language model (LM) and machine comprehension models on this dataset, including a language model trained on the One Billion Word Corpus. We find that the state-of-the-art model lags behind human performance even if the model is trained on a large external corpus. We analyze where the model fails compared to humans who perform well. After conducting error analysis, we assume the performance gap results from the model's inability to use a long-term context. To examine this assumption, we evaluate human-level performance when the human subjects are only allowed to see one sentence as the context. Our assumption is confirmed by the matched performances of the models and human when given only one sentence. In addition, we demonstrate that human-created data is more difficult than automatically-generated data. Specifically, it is much easier for the same model to perform well on automatically-generated data.

Our motivation is to provide a valuable testbed for both the language modeling community and the machine comprehension community. Specifically, the language modeling community can use CLOTH to evaluate their models' abilities in modeling long contexts, while the machine comprehension community can use CLOTH to test machine's understanding of language phenomena.

7.2 Related Work

Large-scale automatically-generated cloze tests [95, 97, 185] lead to significant research advancements. However, generated questions do not consider language phenomenon to be tested and are relatively easy to solve. Recently proposed reading comprehension datasets are all labeled by humans to ensure a high quality [117, 180, 203, 260].

Perhaps the closet work to CLOTH is the LAMBADA dataset [188]. LAMBADA also targets at finding challenging words to test LM's ability in comprehending a longer context. However, LAMBADA does not provide a candidate set for each question, which can cause ambiguities when multiple words can fit in. Furthermore, only test set and development set are labeled manually. The provided training set is the unlabeled Book Corpus [318]. Such unlabeled data do not emphasize long-dependency questions and have a mismatched distribution with the test set, as showed in Section 7.5. Further, the Book Corpus is too large to allow rapid algorithm development for researchers who do not have access to a huge amount of computational power.

Aiming to evaluate machines under the same conditions that the humans are evaluated, there is a growing interest in obtaining data from examinations. NTCIR QA Lab [233] contains a set of real-world college entrance exam questions. The Entrance Exams task at CLEF QA Track [195, 212] evaluates machine's reading comprehension ability. The AI2 Reasoning Challenge [44, 222] contains approximately eight thousand scientific questions used in middle school. Lai et al. [133] proposes the first large-scale machine comprehension dataset obtained from exams. They show that questions designed by teachers have a significantly larger proportion of reasoning questions. Our dataset focuses on evaluating both language proficiency and reasoning abilities.

7.3 CLOTH Dataset

In this section, we introduce the CLOTH dataset that is collected from English examinations, and study its abilities of assessment.

7.3.1 Data Collection and Statistics

We collect the raw data from three free and public websites in China that gather exams created by English teachers to prepare students for college/high school entrance exams[1]. Before cleaning, there are $20,605$ passages and $332,755$ questions. We perform the following processes to ensure the validity of data: Firstly, we remove questions with an inconsistent format such as questions with more than four options. Then we filter all questions whose validity relies on external information such as pictures or tables. Further, we find that half of the total passages are duplicates and we delete those passages. Lastly, on one of the websites, the answers are stored as images. We use two OCR software programs[2] to extract the answers from images. We discard the questions when results from the two software are different. After the cleaning process, we obtain a clean dataset of $7,131$ passages and $99,433$ questions.

[1] The three websites include http://www.21cnjy.com/; http://5utk.ks5u.com/; http://zujuan.xkw.com/. We checked that CLOTH does not contain sentence completion example questions from GRE, SAT and PSAT.

[2] tesseract: https://github.com/tesseract-ocr; ABBYY FineReader: https://www.abbyy.com/en-us/finereader/

Since high school questions are more difficult than middle school questions, we divide the datasets into CLOTH-M and CLOTH-H, which stand for the middle school part and the high school part. We split 11% of the data for both the test set and the development set. The detailed statistics of the whole dataset and two subsets are presented in Table **??**. Note that the questions were created to test non-native speakers, hence the vocabulary size is not very large.

Dataset	CLOTH-M			CLOTH-H			CLOTH (Total)		
	Train	Dev	Test	Train	Dev	Test	Train	Dev	Test
# passages	2,341	355	335	3,172	450	478	5,513	805	813
# questions	22,056	3,273	3,198	54,794	7,794	8,318	76,850	11,067	11,516
Vocab. size		15,096			32,212			37,235	
Avg. # sentence		16.26			18.92			17.79	
Avg. # words		242.88			365.1			313.16	

7.3.2 Question Type Analysis

In order to evaluate students' mastery of a language, teachers usually design tests in a way that questions cover different aspects of a language. Specifically, they first identify words in the passage that can examine students' knowledge in vocabulary, logic, or grammar. Then, they replace the words with blanks and prepare three incorrect but nuanced candidate options to make the test non-trivial. A sample passage is presented in Table 7.1.

To understand the abilities of assessment on this dataset, we divide questions into several types and label the proportion of each type. According to English teachers who regularly create cloze test questions for English exams in China, there are largely three types: grammar, vocabulary and reasoning. Grammar questions are easily differentiated from other two categories. However, the teachers themselves cannot specify a clear distinction between reasoning questions and vocabulary questions since all questions require comprehending the words within the context and conducting some level of reasoning by recognizing incomplete information or conceptual overlap.

Hence, we divided the questions except grammar questions based on the difficulty level for a machine to answer the question, following work on analyzing machine comprehension datasets [35, 260]. In particular, we divide them in terms of their dependency ranges, since questions that only involve a single sentence are easier to answer than questions involving evidence distributed in multiple sentences. Further, we divided questions involving long-term dependency into matching/paraphrasing questions and reasoning questions since matching questions are easier. The four types include:

- Grammar: The question is about grammar usage, involving tense, preposition usage, active/-passive voices, subjunctive mood and so on.
- Short-term-reasoning: The question is about content words and can be answered based on the information within the same sentence. Note that the content words can evaluate knowledge of both vocabulary and reasoning.

Passage: Nancy had just got a job as a secretary in a company. Monday was the first day she went to work, so she was very _1_ and arrived early. She _2_ the door open and found nobody there. "I am the _3_ to arrive." She thought and came to her desk. She was surprised to find a bunch of _4_ on it. They were fresh. She _5_ them and they were sweet. She looked around for a _6_ to put them in. "Somebody has sent me flowers the very first day!" she thought _7_ . " But who could it be?" she began to _8_ . The day passed quickly and Nancy did everything with _9_ interest. For the following days of the _10_ , the first thing Nancy did was to change water for the followers and then set about her work.
Then came another Monday. _11_ she came near her desk she was overjoyed to see a(n) _12_ bunch of flowers there. She quickly put them in the vase, _13_ the old ones. The same thing happened again the next Monday. Nancy began to think of ways to find out the _14_ . On Tuesday afternoon, she was sent to hand in a plan to the _15_ . She waited for his directives at his secretary's _16_ . She happened to see on the desk a half-opened notebook, which _17_ : "In order to keep the secretaries in high spirits, the company has decided that every Monday morning a bunch of fresh flowers should be put on each secretaryâĂŹs desk." Later, she was told that their general manager was a business management psychologist.

Questions:

1.	A. depressed	B. encouraged	**C. excited**	D. surprised
2.	A. turned	**B. pushed**	C. knocked	D. forced
3.	A. last	B. second	C. third	**D. first**
4.	A. keys	B. grapes	**C. flowers**	D. bananas
5.	**A. smelled**	B. ate	C. took	D. held
6.	**A. vase**	B. room	C. glass	D. bottle
7.	A. angrily	B. quietly	C. strangely	**D. happily**
8.	A. seek	**B. wonder**	C. work	D. ask
9.	A. low	B. little	**C. great**	D. general
10.	A. month	B. period	C. year	**D. week**
11.	A. Unless	**B. When**	C. Since	D. Before
12.	A. old	B. red	C. blue	**D. new**
13.	A. covering	B. demanding	**C. replacing**	D. forbidding
14.	**A. sender**	B. receiver	C. secretary	D. waiter
15.	A. assistant	B. colleague	C. employee	**D. manager**
16.	A. notebook	**B. desk**	C. office	D. house
17.	**A. said**	B. written	C. printed	D. signed

Table 7.1: A Sample passage from our dataset. Bold faces highlight the correct answers. There is only one best answer among four candidates, although several candidates may seem correct.

- Matching/paraphrasing: The question is answered by copying/paraphrasing a word in the context.
- Long-term-reasoning: The answer must be inferred from synthesizing information distributed across multiple sentences.

We sample 100 passages in the high school category and the middle school category respectively with totally $3,000$ questions. The types of these questions are labeled on Amazon Turk. We pay \$1 and \$0.5 for high school passages and middle school passages respectively. To label the questions, we provided the definition and an example for each question category to the Amazon Mechanical Turkers. To ensure quality, we limited the workers to master Turkers who are experienced and maintain a high acceptance rate. However, we did not restrict the backgrounds of the Turkers since master Turkers should have a reasonable amount of knowledge about English to conduct previous tasks. In addition, the vocabulary used in CLOTH are usually not difficult since they are constructed to test non-native speakers in middle school or high school. To get a concrete idea of the nature of question types, please refer to examples shown in Tab. 7.2.

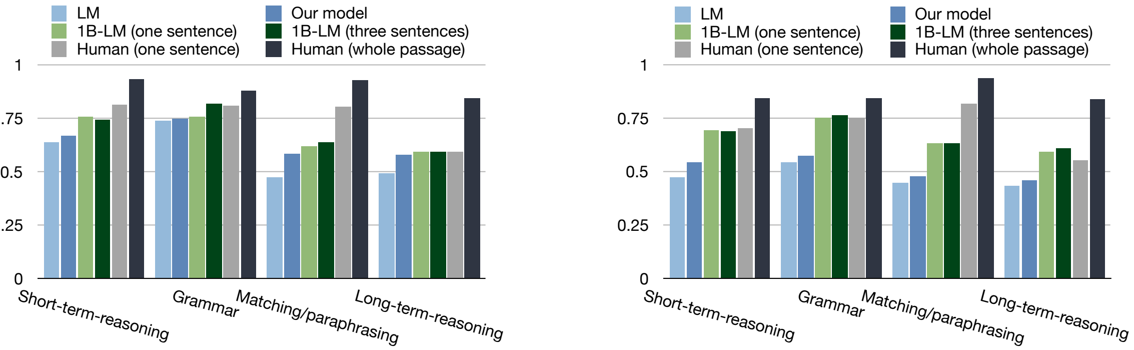
Passage: Nancy had just got a job as a secretary in a company. Monday was the first day she went to work, so she was very _1_ and arrived early. She _2_ the door open and found nobody there. "I am the _3_ to arrive." She thought and came to her desk. She was surprised to find a bunch of _4_ on it. They were fresh. She _5_ them and they were sweet. She looked around for a _6_ to put them in. "Somebody has sent me flowers the very first day!" she thought _7_ . " But who could it be?" she began to _8_ . The day passed quickly and Nancy did everything with _9_ interest. For the following days of the _10_ , the first thing Nancy did was to change water for the followers and then set about her work.
Then came another Monday. _11_ she came near her desk she was overjoyed to see a(n) _12_ bunch of flowers there. She quickly put them in the vase, _13_ the old ones. The same thing happened again the next Monday. Nancy began to think of ways to find out the _14_ . On Tuesday afternoon, she was sent to hand in a plan to the _15_ . She waited for his directives at his secretary's _16_ . She happened to see on the desk a half-opened notebook, which _17_ : "In order to keep the secretaries in high spirits, the company has decided that every Monday morning a bunch of fresh flowers should be put on each secretary's desk." Later, she was told that their general manager was a business management psychologist.

		Questions			**Question type**
1.	A. depressed	B. encouraged	**C. excited**	D. surprised	short-term reasoning
2.	A. turned	**B. pushed**	C. knocked	D. forced	short-term reasoning
3.	A. last	B. second	C. third	**D. first**	long-term reasoning
4.	A. keys	B. grapes	**C. flowers**	D. bananas	matching
5.	**A. smelled**	B. ate	C. took	D. held	short-term reasoning
6.	**A. vase**	B. room	C. glass	D. bottle	long-term reasoning
7.	A. angrily	B. quietly	C. strangely	**D. happily**	short-term reasoning
8.	A. seek	**B. wonder**	C. work	D. ask	long-term reasoning
9.	A. low	B. little	**C. great**	D. general	long-term reasoning
10.	A. month	B. period	C. year	**D. week**	long-term reasoning
11.	A. Unless	**B. When**	C. Since	D. Before	grammar
12.	A. old	B. red	C. blue	**D. new**	long-term reasoning
13.	A. covering	B. demanding	**C. replacing**	D. forbidding	long-term reasoning
14.	**A. sender**	B. receiver	C. secretary	D. waiter	long-term reasoning
15.	A. assistant	B. colleague	C. employee	**D. manager**	matching
16.	A. notebook	**B. desk**	C. office	D. house	matching
17.	**A. said**	B. written	C. printed	D. signed	grammar

Table 7.2: An Amazon Turker's label for the sample passage

7.3.3 Type-specific Performance Analysis

We can also further verify the strengths and weaknesses of the 1B-LM by studying the performance of models and human on different question categories. Note that the performance presented here may be subject to a high variance due to the limited number of samples in each category. From the comparison shown in Figure 7.1, we see that 1B-LM is indeed good at short-term questions. Specifically, when the human only has access to the context of one sentence,



(a) Middle school group (CLOTH-M)

(b) High school group (CLOTH-H)

Figure 7.1: Model and human's performance on questions with different types. Our model will be introduced in Sec. 7.6.

1B-LM is close to human's performance on almost all categories. Further, comparing LM and 1B-LM, we find that training on the large corpus leads to improvements on all categories, showing that training on a large amount of data leads to a substantial improvement in learning complex language regularities.

The proportion of different questions is shown in Table 7.3. The majority of questions are short-term-reasoning questions while approximately 22.4% of the data needs long-term information, in which the long-term-reasoning questions constitute a large proportion.

	Short-term		Long-term		
Dataset	GM	STR	MP	LTR	O
CLOTH	0.265	0.503	0.044	0.180	0.007
CLOTH-M	0.330	0.413	0.068	0.174	0.014
CLOTH-H	0.240	0.539	0.035	0.183	0.004

Table 7.3: The question type statistics of 3000 sampled questions where GM, STR, MP, LTR and O denotes grammar, short-term-reasoning, matching/paraphrasing, long-term-reasoning and others respectively.

7.4 Exploring Models' Limits

In this section, we investigate if human-created cloze test is a challenging problem for state-of-the-art models. We find that LM trained on the One Billion Word Corpus can achieve a remarkable score but cannot solve the cloze test. After conducting an error analysis, we hypothesize that the model is not able to deal with long-term dependencies. We verify the hypothesis by comparing the model's performance with the human performance when the information humans obtain is limited to one sentence.

7.4.1 Human and Model Performance

LSTM To test the performance of RNN-based supervised models, we train a bidirectional LSTM [99] to predict the missing word given the context with only labeled data.

Attentive Readers To enable the model to gather information from a longer context, we augment the supervised LSTM model with the attention mechanism [9], so that the representation at the blank is used as a query to find the relevant context in the document and a blank-specific representation of the document is used to score each candidate answer. Specifically, we adapt the Stanford Attentive Reader [35] and the position-aware attention model [308] to the cloze test problem. With the position-aware attention model, the attention scores are based on both the context match and the distance from a context to the blank. Both attention models are trained only with human-created blanks just as the LSTM model.

LM In cloze test, the context on both sides may be enough to determine the correct answer. Suppose x_i is the missing word and $x_1, \cdots, x_{i-1}$, $x_{i+1}, \cdots, x_n$ are the context, we choose x_i that maximizes the joint probability $p(x_1, \cdots, x_n)$, which essentially maximizes the conditional likelihood $p(x_i \mid x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n)$. Therefore, LM can be naturally adapted to cloze test.

In essence, LM treats each word as a possible blank and learns to predict it. As a result, it receives more supervision than the LSTM trained on human-labeled questions. Besides training a neural LM on our dataset, interested in whether the state-of-the-art LM can solve cloze test, we also test the LM trained on the One Billion Word Benchmark [34] (referred as 1B-LM) that achieves a perplexity of 30.0 [118][3]. To make the evaluation time tractable, we limit the context length to one sentence or three sentences. Note that the One Billion Word Corpus does not overlap with the CLOTH corpus.

Human performance We measure the performance of Amazon Mechanical Turkers on $3,000$ sampled questions when the whole passage is given.

Implementation Details We implement our models using PyTorch [194]. We train our model on all questions in CLOTH and test it on CLOTH-M and CLOTH-H separately. For our final model, we use Adam [123] with the learning rate of 0.001. The hidden dimension is set to 650 and we initialize the word embedding by 300-dimensional Glove word vector [197]. The temperature α is set to 2. We tried to increase the dimensionality of the model but do not observe performance improvement.

When we train the small LM on CLOTH, we largely follow the recommended hyperparameters in the Pytorch LM example[4]. Specifically, we employ a 2-layer LSTM with hidden dimension as 1024. The input embedding and output weight matrix are tied. We set the dropout rate to 0.5. The initial learning rate is set to 10 and divided by 4 whenever the PPL stops improving on the dev set.

[3] The pre-trained model is obtained from https://github.com/tensorflow/models/tree/master/research/ lm_1b

[4] https://github.com/pytorch/examples/tree/master/word_language_model

We predict the answer for each blank independently for all of the models mentioned in this work, since we do not observe significant performance improvements in our preliminary experiments when an auto-regressive approach is employed, i.e., when we fill all previous blanks with predicted answers. We hypothesize that, regardless of whether there exist inter-blank dependencies, since blanks are usually distributed far away from each other, LSTM is not able to capture such long dependencies. When testing language models, we use the longest text spans that do not contain blanks.

Model	CLOTH	CLOTH-M	CLOTH-H
LSTM	0.484	0.518	0.471
Stanford AR	0.487	0.529	0.471
Position-aware AR	0.485	0.523	0.471
LM	0.548	0.646	0.506
1B-LM (one sent.)	0.695	0.723	0.685
1B-LM (three sent.)	0.707	0.745	0.693
Human performance	0.859	0.897	0.845

Table 7.4: Models' performance and human-level performance on CLOTH. LSTM, Stanford Attentive Reader and Attentive Reader with position-aware attention shown in the top part only use supervised data labelled by human. LM outperforms LSTM since it receives more supervisions in learning to predict each word. Training on large external corpus further significantly enhances LM's accuracy.

Results The comparison is shown in Table 7.4. Both attentive readers achieve similar accuracy to the LSTM. We hypothesize that the reason of the attention model's unsatisfactory performance is that the evidence of a question cannot be simply found by matching the context. Similarly, on reading comprehension, though attention-based models [58, 227, 269] have reached human performance on the SQuAD dataset [203], their performance is still not comparable to human performance on datasets that focus more on reasoning where the evidence cannot be simply found by a matching behavior [133, 283]. Since the focus of this work is to analyze CLOTH, we leave the design of reasoning oriented attention models for future work.

The LM achieves much better performance than LSTM. The gap is larger when the LM is trained on the 1 Billion Word Corpus, indicating that more training data results in a better generalization. Specifically, the accuracy of 1B-LM is 0.695 when one sentence is used as the context. It indicates that LM can learn sophisticated language regularities when given sufficient data. The same conclusion can also be drawn from the success of a concurrent work ELMo which uses LM representations as word vectors and achieves state-of-the-art results on six language tasks [198]. However, if we increase the context length to three sentences, the accuracy of 1B-LM only has a marginal improvement. In contrast, humans outperform 1B-LM by a significant margin, which demonstrates that deliberately designed questions in CLOTH are not completely solved even for state-of-the-art models.

Context	Options			
She pushed the door open and found nobody there. "I am the __ to arrive." She thought and came to her desk.	*A. last*	B. second	C. third	**D. first**
They were fresh. She __ them and they were sweet. She looked around for a vase to put them in.	**A. smelled**	*B. ate*	C. took	D. held
She smelled them and they were sweet. She looked around for a __ to put them in. "Somebody has sent me flowers the very first day!"	**A. vase**	*B. room*	C. glass	D. bottle
"But who could it be?" she began to __ . The day passed quickly and Nancy did everything with great interest.	A. seek	**B. wonder**	C. work	*D. ask*

Table 7.5: Error analysis of 1-billion-language-model with three sentences as the context. The questions are sampled from the sample passage shown in Table 7.1. The correct answer is in bold text. The incorrectly selected options are in italics.

7.4.2 Analyzing 1B-LM's Strengths and Weaknesses

In this section, we would like to understand why 1B-LM lags behind human performance. We find that most of the errors involve long-term reasoning. Additionally, in a lot of cases, the dependency is within the context of three sentences. We show several errors made by the 1B-LM in Table 7.5. In the first example, the model does not know that Nancy found nobody in the company means that Nancy was the first one to arrive at the company. In the second and third example, the model fails probably because of not recognizing “they" referred to “flowers". The dependency in the last case is longer. It depends on the fact that Nancy was alone in the company.

Based on the case study, we hypothesize that the LM is not able to take long-term information into account, although it achieves a surprisingly good overall performance. Additionally, the 1B-LM is trained on the sentence level, which might also result in the inability to track paragraph level information. However, to investigate the differences between training on sentence level and on paragraph level, a prohibitive amount of computational resource is required to train a large model on the 1 Billion Word Corpus.

On the other hand, a practical comparison is to test the model's performance on different types of questions. We find that the model's accuracy is 0.591 on long-term-reasoning questions of CLOTH-H while it achieves 0.693 on short-term-reasoning, which partially confirms that long-term-reasoning is harder. However, we could not completely rely on the performance on specific questions types, partly due to a large variance caused by the small sample size. Another reason is that the reliability of question type labels depends on whether turkers are careful enough. For example, in the error analysis shown in Table 7.5, a careless turker would label the second example as short-term-reasoning without noticing that the meaning of “they" relies on a long context.

To objectively verify if the LM's strengths lie in dealing with short-term information, we obtain the ceiling performance of only utilizing short-term information. Showing only one sentence as the context, we ask the Turkers to select an option based on their best guesses given the insufficient information. By limiting the context span manually, the ceiling performance with the access to only a short context is estimated accurately.

As shown in Table 7.6, The performance of 1B-LM using one sentence as the context can almost match the human ceiling performance of only using short-term information. Hence we

	Model	CLOTH	CLOTH-M	CLOTH-H
Short context	1B-LM	0.695	0.723	0.685
	Human	0.713	0.771	0.691
Long context	1B-LM	0.707	0.745	0.693
	Human	0.859	0.897	0.845

Table 7.6: Humans' performance compared with 1-billion-language-model. In the short context part, both 1B-LM and humans only use information of one sentence. In the long context part, humans have the whole passage as the context, while 1B-LM uses contexts of three sentences.

conclude that the LM can almost perfectly solve all short-term cloze questions. However, the performance of LM is not improved significantly when a long-term context is given, indicating that the performance gap is due to the inability of long-term reasoning.

7.5 Comparing Human-created Data and Automatically-generated Data

In this section, we demonstrate that human-created data is a better testbed than automatically-generated cloze test since it results in a larger gap between model's performance and human performance.

A casual observation is that a cloze test can be created by randomly deleting words and randomly sampling candidate options. In fact, to generate large-scale data, similar generation processes have been introduced and widely used in machine comprehension [95, 97, 185]. However, research on cloze test design [217] shows that tests created by deliberately deleting words are more reliable than tests created by randomly or periodically deleting words. To design accurate language proficiency assessment, teachers usually deliberately select words in order to examine students' proficiency in grammar, vocabulary and reasoning. Moreover, in order to make the question non-trivial, three incorrect options provided by teachers are usually grammatically correct and relevant to the context. For instance, in the fourth problem of the sample passage shown in Table 7.1, "grapes", "flowers" and "bananas" all fit the description of being fresh.

Hence we naturally hypothesize that human-generated data has distinct characteristics when compared with automatically-generated data. To verify this assumption, we compare the LSTM model's performance when given different proportions of the two types of data. Specifically, to train a model with α percent of automatically-generated data, we randomly replace a percent blanks with blanks at random positions, while keeping the remaining $1 - \alpha$ percent questions the same. The candidate options for the generated blanks are random words sampled from the unigram distribution. We test models obtained with varying α on human-created data and automatically-generated data respectively.

From the comparison in Table 7.7, we have the following observations: (1) human-created data leads to a larger gap between model's performance and the ceiling/human performance. The model's performance and human's performance on the human-created data are 0.484 and 0.859 respectively, as shown in Tab. 7.4, leading to a gap of 0.376. In comparison, the performance gap

Test \ $\alpha\%$	0%	25%	50%	75%	100%
human-created	0.484	0.475	0.469	0.423	0.381
Generated	0.422	0.699	0.757	0.785	0.815

Table 7.7: The model's performance when trained on α percent of automatically-generated data and $100 - \alpha$ percent of human-created data

on the automatically-generated data is at most 0.185 since the model's performance reaches an accuracy of 0.815 when fully trained on generated data. (2) Although human-created data may provide more information in distinguishing similar words, the distributional mismatch between two types of data makes it non-trivial to transfer the knowledge gained from human-created data to tackle automatically-generated data. Specifically, the model's performance on automatically-generated data monotonically decreases when given a higher ratio of human-created data.

7.6 Combining Human-created Data with Automatically-generated Data

In Section 7.4.1, we show that LM is able to take advantage of more supervision since it predicts each word based on the context. At the same time, we also show that human-created data and the automatically-generated data are quite different in Section 7.5. In this section, we investigate a model that takes advantage of both sources.

7.6.1 Representative-based Model

Specifically, for each question, regardless of being human-created or automatically-generated, we can compute the negative log likelihood of the correct answer as the loss function. Suppose J_{H} is the average negative log likelihood loss for human-created questions and J_{R} is the loss function on generated questions, we combine losses on human-created questions and generated questions by simply adding them together, i.e., $J_{\mathrm{R}} + J_{\mathrm{H}}$ is used as the final loss function. We will introduce the definition of J_{R} in the following paragraphs.

Although automatically-generated data has a large quantity and is valuable to the model training, as shown in the previous Section, automatically-generated questions are quite different from human-created questions. Ideally, a large amount of human-created questions is more desirable than a large amount of automatically-generated questions. A possible avenue towards having large-scale human-created data is to automatically pick out a large number of generated questions which are *representative* of or similar to human-created questions. In other words, we train a network to predict whether a question is a generated question or a human-created question. A generated question is representative of human-created questions if it has a high probability of being a human-created question. Then we can give higher weights to questions that resemble human-created question.

We first introduce our method to obtain the representativeness information. Let x denote the passage and z denote whether a word is selected as a question by human, i.e., z is 1 if this word is selected to be filled in the original passage or 0 otherwise. Suppose h_i is the representation of i-th word given by a bidirectional LSTM. The network computes the probability p_i of x_i being a human-created question as follows:

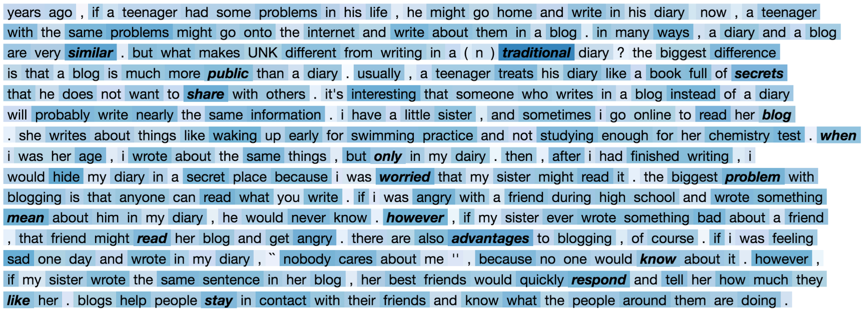
$$l_i = h_i^T w_{x_i}; \quad p_i = \text{Sigmoid}(l_i)$$

where l_i is the logit which will be used as in the final model and w_{x_i} is the the word embedding. We train the network to minimize the binary cross entropy between p and ground-truth labels at each token.

After obtaining the representativeness information, we define the representativeness weighted loss function as

$$J_{\text{R}} = \sum_{i \notin H} \text{Softmax}_i(\frac{l_1}{\alpha}, \cdots, \frac{l_n}{\alpha}) J_i$$

where J_i denotes the negative log likelihood loss for the $i-$th question and let l_i be the output representativeness of the i-th question and H is the set of all human-generated questions and α is the temperature of the Softmax function. The model degenerates into assigning a uniform weight to all questions when the temperature is $+\infty$. We set α to 2 based on the performance on the dev set. [5].



years ago , if a teenager had some problems in his life , he might go home and write in his diary now , a teenager
with the same problems might go onto the internet and write about them in a blog . in many ways , a diary and a blog
are very ***similar*** . but what makes UNK different from writing in a (n) ***traditional*** diary ? the biggest difference
is that a blog is much more ***public*** than a diary . usually , a teenager treats his diary like a book full of ***secrets***
that he does not want to ***share*** with others . it's interesting that someone who writes in a blog instead of a diary
will probably write nearly the same information . i have a little sister , and sometimes i go online to read her ***blog***
. she writes about things like waking up early for swimming practice and not studying enough for her chemistry test . ***when***
i was her age , i wrote about the same things , but ***only*** in my dairy . then , after i had finished writing , i
would hide my diary in a secret place because i was ***worried*** that my sister might read it . the biggest ***problem*** with
blogging is that anyone can read what you write . if i was angry with a friend during high school and wrote something
mean about him in my diary , he would never know . ***however*** , if my sister ever wrote something bad about a friend
, that friend might ***read*** her blog and get angry . there are also ***advantages*** to blogging , of course . if i was feeling
sad one day and wrote in my diary , `` nobody cares about me '' , because no one would ***know*** about it . however ,
if my sister wrote the same sentence in her blog , her best friends would quickly ***respond*** and tell her how much they
like her . blogs help people ***stay*** in contact with their friends and know what the people around them are doing .

Figure 7.2: Representativeness prediction for each word. Lighter color means less representative. The words deleted by human as blanks are in bold text.

Model	Ex.	CLOTH	CLOTH-M	CLOTH-H
Our model	No	**0.583**	**0.673**	**0.549**
LM		0.548	0.646	0.506
LSTM		0.484	0.518	0.471
Stanford AR		0.487	0.529	0.471
1B-LM	Yes	0.707	0.745	0.693
Human		0.859	0.897	0.845

Table 7.8: Overall results on CLOTH. Ex. denotes external data.

[5] The code is available at https://github.com/qizhex/Large-scale-Cloze-Test-Dataset-Created-by-Teachers

Model	CLOTH	CLOTH-M	CLOTH-H
Our model	**0.583**	**0.673**	**0.549**
w.o. rep.	0.566	0.662	0.528
w.o. hum.	0.565	0.665	0.526
w.o. rep. or hum.	0.543	0.643	0.505

Table 7.9: Ablation study on using the representativeness information (denoted as rep.) and the human-created data (denoted as hum.)

7.6.2 Results

We summarize performances of all models in Table 7.8. Our representativeness model outperforms all other models that do not use external data on CLOTH, CLOTH-H and CLOTH-M.

7.6.3 Analysis

In this section, we verify the effectiveness of the representativeness-based averaging by ablation studies. When we remove the representativeness information by setting α to infinity, the accuracy drops from 0.583 to 0.566. When we further remove the human-created data so that only generated data is employed, the accuracy drops to 0.543, similar to the performance of LM. The results further confirm that it is beneficial to incorporate human-created questions into training.

A sample of the predicted representativeness is shown in Figure 7.2[6]. Clearly, words that are too obvious have low scores, such as punctuation marks, simple words “a" and “the". In contrast, content words whose semantics are directly related to the context have a higher score, e.g., “same", “similar", “difference" have a high score when the difference between two objects is discussed and “secrets" has a high score since it is related to the subsequent sentence “does not want to share with others". Our prediction model achieves an F1 score of 36.5 on the test set, which is understandable since there are many plausible questions within a passage.

It has been shown that features such as morphology information and readability are beneficial in cloze test prediction [46, 47, 132, 237]. We leave investigating the advanced approaches of automatically designing cloze test to future work.

7.7 Discussion

In this work, we collect a large-scale cloze test dataset CLOTH that is designed by teachers. With missing blanks and candidate options carefully created by teachers to test different aspects of language phenomena, CLOTH requires a deep language understanding and better captures the complexity of human language. We find that LM-1B achieves great performance on CLOTH though human outperforms LM-1B by a significant margin.

In addition, the performance difference between transfer learning from a large out-of-domain dataset 1-Billion-Word and transfer learning from a small in-domain dataset shows that it is nec-

[6] The script to generate the Figure is obtained at `https://gist.github.com/ihsgnef/f13c35cd46624c8f458a4d23589ac768`

essary to perform pretraining on a large corpus so that the model can learn a general knowledge of text.

Part III

Data-Efficient Learning by Using External Knowledge

Chapter 8

Making Use of Inductive Biases as External Knowledge for Text Generation

In this chapter, with the prior knowledge that token-level training signals provides better credit assignments than sentence-level training signals, we present methods that lead to improved performance for text generation by breaking down the sentence-level training signals into token-level signals. Specifically, we use the sentence-level training signal provided by RAML [183] and establish a theoretical equivalence between the token-level counterpart of RAML and the entropy regularized reinforcement learning. Motivated by this connection, we present two sequence prediction algorithms with improved performance.

8.1 Introduction

Modeling and predicting discrete sequences is the central problem to many natural language processing tasks. In the last few years, the adaption of recurrent neural networks (RNNs) and the sequence-to-sequence model (seq2seq) [9, 245] has led to a wide range of successes in conditional sequence prediction, including machine translation [9, 245], automatic summarization [215], image captioning [121, 267, 280] and speech recognition [32].

Despite the distinct evaluation metrics for the aforementioned tasks, the standard training algorithm has been the same for all of them. Specifically, the algorithm is based on maximum likelihood estimation (MLE), which maximizes the log-likelihood of the "ground-truth" sequences empirically observed.[1]

While largely effective, the MLE algorithm has two obvious weaknesses. Firstly, the MLE training ignores the information of the task specific metric. As a result, the potentially large discrepancy between the log-likelihood during training and the task evaluation metric at test time can lead to a suboptimal solution. Secondly, MLE can suffer from the exposure bias, which refers to the phenomenon that the model is never exposed to its own failures during training, and thus cannot recover from an error at test time. Fundamentally, this issue roots from the difficulty

[1] In this work, we use the terms "ground-truth" and "reference" to refer to the empirical observations interchangeably.

in statistically modeling the exponentially large space of sequences, where most combinations cannot be covered by the observed data.

To tackle these two weaknesses, there have been various efforts recently, which we summarize into two broad categories:

- A widely explored idea is to directly optimize the task metric for sequences produced by the model, with the specific approaches ranging from minimum risk training (MRT) [229] and learning as search optimization (LaSO) [54, 276] to reinforcement learning (RL) [10, 204]. In spite of the technical differences, the key component to make these training algorithms *practically efficient* is often a delicate credit assignment scheme, which transforms the sequence-level signal into dedicated smaller units (e.g., token-level or chunk-level), and allocates them to specific decisions, allowing for efficient optimization with a much lower variance. For instance, the beam search optimization (BSO) [276] utilizes the position of margin violations to produce signals to the specific chunks, while the actor-critic (AC) algorithm [10] trains a critic to enable token-level signals.
- Another alternative idea is to construct a task metric dependent target distribution, and train the model to match this task-specific target instead of the empirical data distribution. As a typical example, the reward augmented maximum likelihood (RAML) [183] defines the target distribution as the exponentiated pay-off (sequence-level reward) distribution. This way, RAML not only can incorporate the task metric information into training, but it can also alleviate the exposure bias by exposing imperfect outputs to the model. However, RAML only work on the sequence-level training signal.

In this work, we are intrigued by the question whether it is possible to incorporate the idea of fine-grained credit assignment into RAML. More specifically, inspired by the token-level signal used in AC, we aim to find the token-level counterpart of the sequence-level RAML, i.e., defining a token-level target distribution for each auto-regressive conditional factor to match. Motived by the question, we first formally define the desiderata the token-level counterpart needs to satisfy and derive the corresponding solution (§8.2). Then, we establish a theoretical connection between the derived token-level RAML and entropy regularized RL (§8.3). Motivated by this connection, we present two algorithms for neural sequence prediction, where one is the token-level extension to RAML, and the other a RAML-inspired improvement to the AC (§8.4). We empirically evaluate the two presented algorithms, and show different levels of improvement over the corresponding baseline. We further study the importance of various techniques used in our experiments, providing practical suggestions to readers (§8.7).

8.2 Token-level Equivalence of RAML

We first introduce the notations used throughout the chapter. Firstly, capital letters will denote random variables and lower-case letters are the values to take. As we mainly focus on conditional sequence prediction, we use $\mathbf{x}$ for the conditional input, and $\mathbf{y}$ for the target sequence. With $\mathbf{y}$ denoting a sequence, $\mathbf{y}_i^j$ then denotes the subsequence from position i to j inclusively, while y_t denotes the single value at position t. Also, we use $|\mathbf{y}|$ to indicate the length of the sequence. To emphasize the ground-truth data used for training, we add superscript * to the input and target, i.e., $\mathbf{x}^*$ and $\mathbf{y}^*$. In addition, we use $\mathcal{Y}$ to denote the set of all possible sequences with one and only one `eos` symbol at the end, and $\mathcal{W}$ to denote the set of all possible symbols in a position. Finally, we assume length of sequences in $\mathcal{Y}$ is bounded by T.

8.2.1 Background: RAML

As discussed in §9.1, given a ground-truth pair $(\mathbf{x}^*, \mathbf{y}^*)$, RAML defines the target distribution using the exponentiated pay-off of sequences, i.e.,

$$P_R(\mathbf{y} \mid \mathbf{x}^*, \mathbf{y}^*) = \frac{\exp\left(R(\mathbf{y}; \mathbf{y}^*)/\tau\right)}{\sum_{\mathbf{y}' \in \mathcal{Y}} \exp\left(R(\mathbf{y}'; \mathbf{y}^*)/\tau\right)}, \tag{8.1}$$

where $R(\mathbf{y}; \mathbf{y}^*)$ is the sequence-level reward, such as BLEU score, and τ is the temperature hyper-parameter controlling the sharpness. With the definition, the RAML algorithm simply minimizes the cross entropy (CE) between the target distribution and the model distribution $P_\theta(\mathbf{Y} \mid \mathbf{x}^*)$, i.e.,

$$\min_\theta \mathrm{CE}\left(P_R(\mathbf{Y} \mid \mathbf{x}^*, \mathbf{y}^*) \| P_\theta(\mathbf{Y} \mid \mathbf{x}^*)\right). \tag{8.2}$$

Note that, this is quite similar to the MLE training, except that the target distribution is different. With the particular choice of target distribution, RAML not only makes sure the ground-truth reference remains the mode, but also allows the model to explore sequences that are not exactly the same as the reference but have relatively high rewards.

Compared to algorithms trying to directly optimize task metric, RAML avoids the difficulty of tracking and sampling from the model distribution that is consistently changing. Hence, RAML enjoys a much more stable optimization without the need of pretraining. However, in order to optimize the RAML objective (Eqn. (8.2)), one needs to sample from the exponentiated pay-off distribution, which is quite challenging in practice. Thus, importance sampling is often used [156, 183].

8.2.2 Token-level Target Distribution

Despite the appealing properties, RAML only operates on the sequence-level reward. As a result, the reward gap between any two sequences cannot be attributed to the responsible decisions precisely, which often leads to a low sample efficiency. Ideally, since we rely on the auto-regressive factorization $P_\theta(\mathbf{y} \mid \mathbf{x}^*) = \prod_{t=1}^{|\mathbf{y}|} P_\theta(y_t \mid \mathbf{y}_1^{t-1}, \mathbf{x}^*)$, the optimization would be much more efficient if we have the target distribution for each token-level factor $P_\theta(Y_t \mid \mathbf{y}_1^{t-1}, \mathbf{x}^*)$ to match. Conceptually, this is exactly how the AC algorithm improves upon the vanilla sequence-level REINFORCE algorithm [204].

With this idea in mind, we set out to find such a token-level target. Firstly, we assume the token-level target shares the form of a Boltzmann distribution but parameterized by some unknown negative energy function Q_R, i.e.,[2]

$$P_{Q_R}(y_t \mid \mathbf{y}_1^{t-1}, \mathbf{y}^*) = \frac{\exp\left(Q_R(\mathbf{y}_1^{t-1}, y_t; \mathbf{y}^*)/\tau\right)}{\sum_{w \in \mathcal{W}} \exp\left(Q_R(\mathbf{y}_1^{t-1}, w; \mathbf{y}^*)/\tau\right)}. \tag{8.3}$$

[2] To avoid clutter, the conditioning on $\mathbf{x}^*$ will be omitted in the sequel, assuming it's clear from the context.

Intuitively, $Q_R(\mathbf{y}_1^{t-1}, w; \mathbf{y}^*)$ measures how much *future* pay-off one can expect if w is generated, given the current status $\mathbf{y}_1^{t-1}$ and the reference $\mathbf{y}^*$. This quantity highly resembles the action-value function (Q-function) in reinforcement learning. As we will show later, it is indeed the case.

Before we state the desiderata for Q_R, we need to extend the definition of R in order to evaluate the goodness of an unfinished partial prediction, i.e., sequences without an $\texttt{eos}$ suffix. Let $\mathcal{Y}^-$ be the set of unfinished sequences, following Bahdanau et al. [10], we define the pay-off function R for a partial sequence $\hat{\mathbf{y}} \in \mathcal{Y}^-$, $|\hat{\mathbf{y}}| < T$ as

$$R(\hat{\mathbf{y}}; \mathbf{y}^*) = R(\hat{\mathbf{y}} + \texttt{eos}; \mathbf{y}^*), \tag{8.4}$$

where the $+$ indicates string concatenation.

With the extension, we are ready to state two requirements for Q_R:

1. **Marginal match**: For P_{Q_R} to be the token-level equivalence of P_R, the sequence-level marginal distribution induced by P_{Q_R} must match P_R, i.e., for any $\mathbf{y} \in \mathcal{Y}$,

$$\prod_{t=1}^{|\mathbf{y}|} P_{Q_R}(y_t \mid \mathbf{y}_1^{t-1}) = P_R(\mathbf{y}). \tag{8.5}$$

   Note that there are infinitely many Q_R's satisfying Eqn. (8.5), because adding any constant value to Q_R does not change the Boltzmann distribution, known as shift-invariance w.r.t. the energy.

2. **Terminal condition**: Secondly, let's consider the value of Q_R when emitting an $\texttt{eos}$ symbol to immediately terminate the generation. As mentioned earlier, Q_R measures the expected future pay-off. Since the emission of $\texttt{eos}$ ends the generation, the future pay-off can only come from the immediate increase of the pay-off. Thus, we require Q_R to be the incremental pay-off when producing $\texttt{eos}$, i.e.

$$Q_R(\hat{\mathbf{y}}, \texttt{eos}; \mathbf{y}^*) = R(\hat{\mathbf{y}} + \texttt{eos}; \mathbf{y}^*) - R(\hat{\mathbf{y}}; \mathbf{y}^*), \tag{8.6}$$

   for any $\hat{\mathbf{y}} \in \mathcal{Y}^-$. Since Eqn. (8.6) enforces the absolute of Q_R at a point, it also solves the ambiguity caused by the shift-invariance property.

Based on the two requirements, we can derive the form Q_R, which is summarized by Proposition 1.

Proposition 1. *P_{Q_R} and Q_R satisfy requirements* (8.5) *and* (8.6) *if and only if for any ground-truth pair* $(\mathbf{x}^*, \mathbf{y}^*)$ *and any sequence prediction* $\mathbf{y} \in \mathcal{Y}$*,*

$$Q_R(\mathbf{y}_1^{t-1}, y_t; \mathbf{y}^*) = \begin{cases} R(\mathbf{y}_1^t; \mathbf{y}^*) - R(\mathbf{y}_1^{t-1}; \mathbf{y}^*) + \tau \log \sum_{w \in \mathcal{W}} \exp\left(Q_R(\mathbf{y}_1^t, w; \mathbf{y}^*)/\tau\right), & t < |\mathbf{y}| \\ R(\mathbf{y}_1^t; \mathbf{y}^*) - R(\mathbf{y}_1^{t-1}; \mathbf{y}^*), & t = |\mathbf{y}| \end{cases} \tag{8.7}$$

Proof. To avoid clutter, we drop the dependency on $\mathbf{x}^*$ and $\mathbf{y}^*$. The following proof holds for each possible pair of $(\mathbf{x}^*, \mathbf{y}^*)$.

Firstly, it is easy to see that the terminal condition in Eqn. (**??**) exactly corresponds to the $t = |\mathbf{y}|$ case of Eqn. (8.7), since $y_t = \texttt{eos}$ for $y \in \mathcal{Y}$. So, we will focus on the non-terminal case next.

Sufficiency For convenience, define $V_R(\mathbf{y}_1^t) = \tau \log \sum_{w \in \mathcal{W}} \exp\left(Q_R(\mathbf{y}_1^t, w)/\tau\right)$. Suppose Eqn. (8.7) is true. Then for any $\mathbf{y} \in \mathcal{Y}$,

$$\begin{aligned}
P_{Q_R}(\mathbf{y}) &= \prod_{t=1}^{|\mathbf{y}|} P_{Q_R}(y_t \mid \mathbf{y}_1^{t-1}) \\
&= \exp\left(\frac{\sum_{t=1}^{|\mathbf{y}|} Q_R(\mathbf{y}_1^{t-1}, y_t) - V_R(\mathbf{y}_1^{t-1})}{\tau}\right) \\
&= \exp\left(\frac{\sum_{t=1}^{|\mathbf{y}|} \left[R(\mathbf{y}_1^t) - R(\mathbf{y}_1^{t-1})\right] + \sum_{t=1}^{|\mathbf{y}|-1} V_R(\mathbf{y}_1^t) - \sum_{t=1}^{|\mathbf{y}|} V_R(\mathbf{y}_1^{t-1})}{\tau}\right) \\
&= \exp\left(\frac{R(\mathbf{y}) - V_R(\emptyset)}{\tau}\right)
\end{aligned}$$

where $V_R(\emptyset)$ denotes $V_R(\mathbf{y}_1^t)$ when $t = 0$ and $\mathbf{y}_1^t$ is an empty set. Since $P_{Q_R}(\mathbf{y})$ is a valid distribution by construction, we have

$$V_R(\emptyset) = \sum_{\mathbf{y} \in \mathcal{Y}} \exp\left(\frac{R(\mathbf{y})}{\tau}\right)$$

Hence,

$$P_{Q_R}(\mathbf{y}) = \frac{R(\mathbf{y})/\tau}{\sum_{\mathbf{y}' \in \mathcal{Y}} R(\mathbf{y}')/\tau} = P_R(\mathbf{y}),$$

which satisfies the marginal match requirement.

Necessity Now, we show that the specific formulation of Q_R (Eqn. (8.7)) is also a necessary condition of the marginal match condition (Eqn. (8.5)).

The token-level target distribution can be simplified as

$$P_{Q_R}(y_t \mid \mathbf{y}_1^{t-1}) = \frac{\exp\left(Q_R(\mathbf{y}_1^{t-1}, y_t)/\tau\right)}{\sum_{w \in \mathcal{W}} \exp\left(Q_R(\mathbf{y}_1^{t-1}, w)/\tau\right)} = \exp\left(\frac{Q_R(\mathbf{y}_1^{t-1}, y_t) - V_R(\mathbf{y}_1^{t-1})}{\tau}\right).$$

Suppose Eqn. (8.5) is true. For any $\mathbf{y} \in \mathcal{Y}^-$ and $t \leq |\mathbf{y}|$ and define $\mathbf{y}' = \mathbf{y}_1^t + \texttt{eos}$ and $\mathbf{y}'' = \mathbf{y}_1^{t-1} + \texttt{eos}$. Obviously, it follows $\mathbf{y}', \mathbf{y}'' \in \mathcal{Y}$. Also, by definition,

$$\begin{aligned}
P_R(\mathbf{y}') &= P_R(\texttt{eos} \mid \mathbf{y}_1^t) \times P_R(y_t \mid \mathbf{y}_1^{t-1}) \times P_R(\mathbf{y}_1^{t-1}) \\
P_R(\mathbf{y}'') &= P_R(\texttt{eos} \mid \mathbf{y}_1^{t-1}) \times P_R(\mathbf{y}_1^{t-1})
\end{aligned}$$

Then, consider the ratio

$$\begin{aligned}
\frac{P_R(\mathbf{y}')}{P_R(\mathbf{y}'')} &= \frac{P_R(\texttt{eos} \mid \mathbf{y}_1^t) \times P_R(y_t \mid \mathbf{y}_1^{t-1}) \times \cancel{P_R(\mathbf{y}_1^{t-1})}}{P_R(\texttt{eos} \mid \mathbf{y}_1^{t-1}) \times \cancel{P_R(\mathbf{y}_1^{t-1})}} \\
\exp\left(\frac{R(\mathbf{y}') - R(\mathbf{y}'')}{\tau}\right) &= \exp\left(\frac{Q_R(\mathbf{y}_1^t, \texttt{eos}) - V_R(\mathbf{y}_1^t)}{\tau}\right) \times \exp\left(\frac{Q_R(\mathbf{y}_1^{t-1}, y_t) - \cancel{V_R(\mathbf{y}_1^{t-1})}}{\tau}\right) \\
&\Big/ \exp\left(\frac{Q_R(\mathbf{y}_1^{t-1}, \texttt{eos}) - \cancel{V_R(\mathbf{y}_1^{t-1})}}{\tau}\right) \\
R(\mathbf{y}') - R(\mathbf{y}'') &= Q_R(\mathbf{y}_1^t, \texttt{eos}) - Q_R(\mathbf{y}_1^{t-1}, \texttt{eos}) - V_R(\mathbf{y}_1^t) + Q_R(\mathbf{y}_1^{t-1}, y_t).
\end{aligned}$$

Now, by the terminal condition (Eqn. (8.6)), we essentially have

$$Q_R(\mathbf{y}_1^t, \texttt{eos}) = R(\mathbf{y}_1^t + \texttt{eos}) - R(\mathbf{y}_1^t) = 0$$
$$Q_R(\mathbf{y}_1^{t-1}, \texttt{eos}) = R(\mathbf{y}_1^{t-1} + \texttt{eos}) - R(\mathbf{y}_1^{t-1}) = 0$$

Thus, it follows

$$\begin{aligned} R(\mathbf{y}') - R(\mathbf{y}'') &= Q_R(\mathbf{y}_1^{t-1}, y_t) - V_R(\mathbf{y}_1^t) \\ \iff Q_R(\mathbf{y}_1^{t-1}, y_t) &= R(\mathbf{y}_1^t) - R(\mathbf{y}_1^{t-1}) + \tau \log \sum_{w \in \mathcal{W}} \exp\left(Q_R(\mathbf{y}_1^t, w)/\tau\right), \end{aligned}$$

which completes the proof.

□

Note that, instead of giving an explicit form for the token-level target distribution, Proposition 1 only provides an equivalent condition in the form of an implicit recursion. Thus, we haven't obtained a practical algorithm yet. However, as we will discuss next, the recursion has a deep connection to entropy regularized RL, which ultimately inspires our presented algorithms.

8.3 Connection to Entropy-regularized RL

Before we dive into the connection, we first give a brief review of the entropy-regularized RL. For an in-depth treatment, we refer readers to [223, 319].

8.3.1 Background: Entropy-regularized RL

Following the standard convention of RL, we denote a Markov decision process (MDP) by a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p_s, r, \gamma)$, where $\mathcal{S}, \mathcal{A}, p_s, r, \gamma$ are the state space, action space, transition probability, reward function and discounting factor respectively.[3]

Based on the notation, the goal of entropy-regularized RL augments is to learn a policy $\pi(a_t \mid s_t)$ which maximizes the discounted expected future return and causal entropy [319], i.e.,

$$\max_{\pi} \sum_{t} \mathop{\mathbb{E}}_{s_t \sim \rho_s, a_t \sim \pi(\cdot|s_t)} \gamma^{t-1}[r(s_t, a_t) + \alpha \mathcal{H}(\pi(\cdot \mid s_t))],$$

where $\mathcal{H}$ denotes the entropy and α is a hyper-parameter controlling the relative importance between the reward and the entropy. Intuitively, compared to standard RL, the extra entropy term encourages exploration and promotes multi-modal behaviors. Such properties are highly favorable in a complex environment.

Given an entropy-regularized MDP, for any fixed policy π, the state-value function $V^\pi(s)$ and the action-value function Q^π can be defined as

$$\begin{aligned} V^\pi(s) &= \mathop{\mathbb{E}}_{a \sim \pi(\cdot|s)} [Q^\pi(s, a)] + \alpha \mathcal{H}(\pi(\cdot \mid s)), \\ Q^\pi(s, a) &= r(s, a) + \mathop{\mathbb{E}}_{s' \sim \rho_s} [\gamma V^\pi(s')]. \end{aligned} \tag{8.8}$$

[3] In sequence prediction, we are only interested in the periodic (finite horizon) case.

With the definitions above, it can further be proved [223, 319] that the optimal state-value function V^*, the action-value function Q^* and the corresponding optimal policy π^* satisfy the following equations

$$V^*(s) = \alpha \log \sum_{a \in \mathcal{A}} \exp\left(Q^*(s, a)/\alpha\right), \tag{8.9}$$

$$Q^*(s, a) = r(s, a) + \gamma \mathop{\mathbb{E}}_{s' \sim \rho_s} [V^*(s')], \tag{8.10}$$

$$\pi^*(a \mid s) = \frac{\exp\left(Q^*(s, a)/\alpha\right)}{\sum_{a' \in \mathcal{A}} \exp\left(Q^*(s, a')/\alpha\right)}. \tag{8.11}$$

Here, Eqn. (8.9) and (8.10) are essentially the entropy-regularized counterparts of the optimal Bellman equations in standard RL. Following previous literature, we will refer to Eqn. (8.9) and (8.10) as the optimal *soft* Bellman equations, and the V^* and Q^* as optimal *soft* value functions.

8.3.2 An RL Equivalence of the Token-level RAML

To reveal the connection, it is convenient to define the incremental pay-off

$$r(\mathbf{y}_1^{t-1}, y_t; \mathbf{y}^*) = R(\mathbf{y}_1^t; \mathbf{y}^*) - R(\mathbf{y}_1^{t-1}; \mathbf{y}^*), \tag{8.12}$$

and the last term of Eqn. (**??**) as

$$V_R(\mathbf{y}_1^t; \mathbf{y}^*) = \tau \log \sum_{w \in \mathcal{W}} \exp\left(Q_R(\mathbf{y}_1^t, w; \mathbf{y}^*)/\tau\right) \tag{8.13}$$

Substituting the two definitions into Eqn. (**??**), the recursion simplifies as

$$Q_R(\mathbf{y}_1^{t-1}, y_t; \mathbf{y}^*) = r(\mathbf{y}_1^{t-1}, y_t; \mathbf{y}^*) + V_R(\mathbf{y}_1^t; \mathbf{y}^*). \tag{8.14}$$

Now, it is easy to see that the Eqn. (8.13) and (8.14), which are derived from the token-level RAML, highly resemble the optimal soft Bellman equations (8.9) and (8.10) in entropy-regularized RL. The following Corollary formalizes the connection.

Corollary 1. *For any ground-truth pair* $(\mathbf{x}^*, \mathbf{y}^*)$*, the recursion specified by Eqn.* (8.12)*,* (8.13) *and* (8.14) *is equivalent to the optimal soft Bellman equation of a "deterministic" MDP in entropy-regularized reinforcement learning, denoted as* $\mathcal{M}_R$*, where*

- *the state space* $\mathcal{S}$ *corresponds to* $\mathcal{Y}^-$*,*
- *the action space* $\mathcal{A}$ *corresponds to* $\mathcal{W}$*,*
- *the transition probability* ρ_s *is a deterministic process defined by string concatenation*
- *the reward function* r *corresponds to the incremental pay-off defined in Eqn.* (8.12)*,*
- *the discounting factor* $\gamma = 1$*,*
- *the entropy hyper-parameter* $\alpha = \tau$*,*
- *and a period terminates either when* `eos` *is emitted or when its length reaches* T *and we enforce the generation of* `eos`*.*

Moreover, the optimal soft value functions V^* and Q^* of the MDP exactly match the V_R and Q_R defined by Eqn. (8.13) *and* (8.14) *respectively. The optimal policy π^* is hence equivalent to the token-level target distribution P_{Q_R}.*

Proof. Similarly, we drop the dependency on $\mathbf{x}^*$ and $\mathbf{y}^*$ to avoid clutter. We first prove the equivalence of $Q^*(\mathbf{y}_1^{t-1}, y_t)$ with $Q_R(\mathbf{y}_1^{t-1}, y_t)$ by induction.

- **Base case**: When $t = T$, for any $\mathbf{y} \in \mathcal{Y}$, y_T can only be $\texttt{eos}$. So, by definition, we have

$$\begin{aligned} & V^*(\mathbf{y}_1^{T-1}) = Q^*(\mathbf{y}_1^{T-1}, \texttt{eos}) \\ \iff & \tau \log \sum_{a \in \mathcal{W}} \exp\big(Q^*(\mathbf{y}_1^{T-1}, a)/\tau\big) = Q^*(\mathbf{y}_1^{T-1}, \texttt{eos}) \\ \implies & Q^*(\mathbf{y}_1^{T-1}, a) = -\infty, \forall a \neq \texttt{eos}. \end{aligned}$$

  Hence,

$$Q^*(\mathbf{y}_1^{T-1}, y_T) = \begin{cases} r(\mathbf{y}_1^{T-1}, \texttt{eos}), & \text{if } y_T = \texttt{eos} \\ -\infty, & \text{otherwise} \end{cases}$$

  For the first case, it directly follows

$$Q^*(\mathbf{y}_1^{T-1}, \texttt{eos}) = r(\mathbf{y}_1^{T-1}, \texttt{eos}) = R(\mathbf{y}_1^{T-1} + \texttt{eos}) - R(\mathbf{y}_1^{T-1}) = Q_R(\mathbf{y}_1^{T-1}, \texttt{eos}).$$

  For the second case, since only $\texttt{eos}$ is allowed to be generated, the target distribution P_{Q_R} should be a single-point distribution at $\texttt{eos}$. This is equivalent to define

$$Q_R(\mathbf{y}_1^{T-1}, a) = -\infty, \forall a \neq \texttt{eos},$$

  which proves the second case. Combining the two cases, it concludes

$$Q^*(\mathbf{y}_1^{T-1}, a) = Q_R(\mathbf{y}_1^{T-1}, a), \forall \mathbf{y} \in \mathcal{Y}, a \in \mathcal{W}.$$

- **Induction step**: When $0 < t < T$, assume the equivalence holds when $k > t$, i.e.,

$$Q^*(\mathbf{y}_1^{k-1}, w) = Q_R(\mathbf{y}_1^{k-1}, w), \forall k > t, w \in \mathcal{W}.$$

  Then,

$$\begin{aligned} Q^*(\mathbf{y}_1^{t-1}, y_t) &= r(\mathbf{y}_1^{t-1}, y_t) + \gamma \mathop{\mathbb{E}}_{s' \sim \rho_s}[\alpha \log \sum_{a \in \mathcal{A}} \exp\left(Q^*(s', a)/\alpha\right)] \\ &= r(\mathbf{y}_1^{t-1}, y_t) + \tau \log \sum_{a \in \mathcal{W}} \exp\big(Q^*(\mathbf{y}_1^t, a)/\tau\big) && (\alpha = \tau, \mathcal{A} = \mathcal{W}) \\ &= r(\mathbf{y}_1^{t-1}, y_t) + \tau \log \sum_{a \in \mathcal{W}} \exp\big(Q_R(\mathbf{y}_1^t, a)/\tau\big) && (Q^*(\mathbf{y}_1^k, a) = Q_R(\mathbf{y}_1^k, a) \text{ for } k \geq t) \\ &= Q_R(\mathbf{y}_1^{t-1}, y_t). \end{aligned}$$

Thus, $Q^*(\mathbf{y}_1^{t-1}, y_t) = Q_R(\mathbf{y}_1^{t-1}, y_t)$ holds for $t \in [1, T]$.

With the equivalence between Q_R and Q^*, we can easily prove $V^* = V_R$ and $\pi^* = P_{Q_R}$,

$$\begin{aligned}
V^*(\mathbf{y}_1^{t-1}) &= \alpha \log \sum_{a \in \mathcal{A}} \exp\left(Q^*(\mathbf{y}_1^{t-1}, a)/\alpha\right) \\
&= \tau \log \sum_{a \in \mathcal{W}} \exp\left(Q^*(\mathbf{y}_1^{t-1}, a)/\tau\right) && (\alpha = \tau, \mathcal{A} = \mathcal{W}) \\
&= V_R(\mathbf{y}_1^{t-1}) \\
\pi^*(y_t \mid \mathbf{y}_1^{t-1}) &= \frac{\exp\left(Q^*(\mathbf{y}_1^{t-1}, y_t)/\tau\right)}{\sum_{w \in \mathcal{W}} \exp\left(Q^*(\mathbf{y}_1^{t-1}, y_t)/\tau\right)} \\
&= \frac{\exp\left(Q_R(\mathbf{y}_1^{t-1}, y_t)/\tau\right)}{\sum_{w \in \mathcal{W}} \exp\left(Q_R(\mathbf{y}_1^{t-1}, y_t)/\tau\right)} \\
&= P_{Q_R}(y_t \mid \mathbf{y}_1^{t-1})
\end{aligned}$$

□

The connection established by Corollary 1 is quite inspiring:

- Firstly, it provides a rigorous and generalized view of the connection between RAML and entropy-regularized RL. In the original work, Norouzi et al. [183] point out RAML can be seen as reversing the direction of $\mathrm{KL}\,(P_\theta \| P_R)$, which is a sequence-level view of the connection. Now, with the equivalence between the token-level target P_{Q_R} and the optimal Q^*, it generalizes to matching the future action values consisting of both the reward and the entropy.
- Secondly, due to the equivalence, if we solve the optimal soft Q-function of the corresponding MDP, we directly obtain the token-level target distribution. This hints at a practical algorithm with token-level credit assignment.
- Moreover, since RAML is able to improve upon MLE by injecting entropy, the entropy-regularized RL counterpart of the standard AC algorithm should also lead to an improvement in a similar manner.

8.4 Proposed Algorithms

In this section, we explore the insights gained from Corollary 1 and present two new algorithms for sequence prediction.

8.4.1 Value Augmented Maximum Likelihood

The first algorithm we consider is the token-level extension of RAML, which we have been discussing since §8.2. As mentioned at the end of §8.2.2, Proposition 1 only gives an implicit form of Q_R, and so is the token-level target distribution P_{Q_R} (Eqn. (8.3)). However, thanks to Corollary 1, we now know that Q_R is the same as the optimal soft action-value function Q^* of the entropy-regularized MDP $\mathcal{M}_R$. Hence, by finding the Q^*, we will have access to P_{Q_R}.

At the first sight, it seems recovering Q^* is as difficult as solving the original sequence prediction problem, because solving Q^* from the MDP is essentially the same as learning the optimal policy for sequence prediction. However, it is not true because Q_R (i.e., P_{Q_R}) can condition on the correct reference $\mathbf{y}^*$. In contrast, the model distribution P_θ can only depend on $\mathbf{x}^*$. Therefore, the function approximator trained to recover Q^* can take $\mathbf{y}^*$ as input, making the estimation task much easier. Intuitively, when recovering Q^*, we are trying to train an ideal "oracle", which has access to the ground-truth reference output, to decide the best behavior (policy) given any arbitrary (good or not) state.

Thus, following the reasoning above, we first train a parametric function approximator Q_ϕ to search the optimal soft action value. In this work, for simplicity, we employ the Soft Q-learning algorithm [223] to perform the policy *optimization*. In a nutshell, Soft Q-Learning is the entropy-regularized version of Q-Learning, an off-policy algorithm which minimizes the mean squared soft Bellman residual according to Eqn. (8.10). Specifically, given ground-truth pair $(\mathbf{x}^*, \mathbf{y}^*)$, for any trajectory $\mathbf{y} \in \mathcal{Y}$, the training objective is

$$\min_\phi \sum_{t=1}^{|\mathbf{y}|} \left[Q_\phi(\mathbf{y}_1^{t-1}, y_t; \mathbf{y}^*) - \hat{Q}_\phi(\mathbf{y}_1^{t-1}, y_t; \mathbf{y}^*) \right]^2, \tag{8.15}$$

where $\hat{Q}_\phi(\mathbf{y}_1^{t-1}, y_t; \mathbf{y}^*) = r(\mathbf{y}_1^{t-1}, y_t; \mathbf{y}^*) + V_\phi(\mathbf{y}_1^t; \mathbf{y}^*)$ is the one-step look-ahead target Q-value, and $V_\phi(\mathbf{y}_1^t; \mathbf{y}^*) = \tau \log \sum_{w \in \mathcal{W}} \exp\left(Q_\phi(\mathbf{y}_1^t, w; \mathbf{y}^*)/\tau\right)$ as defined in Eqn. (8.9). In the recent instantiation of Q-Learning [172], to stabilize training, the target Q-value is often estimated by a separate slowly updated target network. In our case, as we have access to a significant amount of reference sequences, we find the target network not necessary. Thus, we directly optimize Eqn. (8.15) using gradient descent, and let the gradient flow through both $Q_\phi(\mathbf{y}_1^{t-1}, y_t; \mathbf{y}^*)$ and $V_\phi(\mathbf{y}_1^t; \mathbf{y}^*)$ [11].

After the training of Q_ϕ converges, we fix the parameters of Q_ϕ, and optimize the cross entropy $\mathrm{CE}\left(P_{Q_\phi} \| P_\theta\right)$ w.r.t. the model parameters θ, which is equivalent to

$$\min_\theta \mathop{\mathbb{E}}_{\mathbf{y} \sim P_{Q_\phi}} \left[\sum_{t=1}^{|\mathbf{y}|} \mathrm{CE}\left(P_{Q_\phi}(Y_t \mid \mathbf{y}_1^{t-1}) \| P_\theta(Y_t \mid \mathbf{y}_1^{t-1})\right) \right]. \tag{8.16}$$

Here, we derive the equivalence between the VAML's objective (Eqn. (8.16)) and the RAML's objective (Eqn. (8.2)).

$$
\begin{aligned}
&\mathrm{CE}\left(P_{Q_\phi} \| P_\theta\right) \\
={}& -\mathop{\mathbb{E}}_{\mathbf{y} \sim P_{Q_\phi}} \log P_\theta(\mathbf{y}) \\
={}& -\mathop{\mathbb{E}}_{\mathbf{y} \sim P_{Q_\phi}} \sum_{t=1}^{|\mathbf{y}|} \log P_\theta(y_t \mid \mathbf{y}_1^{t-1}) \\
={}& -\sum_{t=1}^{T} \mathop{\mathbb{E}}_{\mathbf{y}_1^t \sim P_{Q_\phi}(Y_1^t)} \log P_\theta(y_t \mid \mathbf{y}_1^{t-1}) && (T \text{ is longest possible length}) \\
={}& \sum_{t=1}^{T} \mathop{\mathbb{E}}_{\mathbf{y}_1^{t-1} \sim P_{Q_\phi}(\mathbf{Y}_1^{t-1})} \left[-\mathop{\mathbb{E}}_{y_t \sim P_{Q_\phi}(Y_t | \mathbf{y}_1^{t-1})} \log P_\theta(y_t \mid \mathbf{y}_1^{t-1}) \right] \\
={}& \sum_{t=1}^{T} \mathop{\mathbb{E}}_{\mathbf{y}_1^{t-1} \sim P_{Q_\phi}(\mathbf{Y}_1^{t-1})} \mathrm{CE}\left(P_{Q_\phi}(Y_t \mid \mathbf{y}_1^{t-1}) \| P_\theta(Y_t \mid \mathbf{y}_1^{t-1})\right) \\
={}& \sum_{t=1}^{T} \mathop{\mathbb{E}}_{\mathbf{y}_1^{t-1} \sim P_{Q_\phi}(\mathbf{Y}_1^{t-1})} \sum_{y_t \in \mathcal{W}} P_{Q_\phi}(y_t \mid \mathbf{y}_1^{t-1}) \underbrace{\mathrm{CE}\left(P_{Q_\phi}(Y_t \mid \mathbf{y}_1^{t-1}) \| P_\theta(Y_t \mid \mathbf{y}_1^{t-1})\right)}_{\text{const. w.r.t. } y_t} \\
={}& \sum_{t=1}^{T} \underbrace{\mathop{\mathbb{E}}_{\mathbf{y}_1^{t-1} \sim P_{Q_\phi}(\mathbf{Y}_1^{t-1})} \mathop{\mathbb{E}}_{y_t \in P_{Q_\phi}(W | \mathbf{y}_1^{t-1})}}_{\mathbb{E}_{\mathbf{y}_1^t \sim P_{Q_\phi}(\mathbf{Y}_1^t)}} \left[\mathrm{CE}\left(P_{Q_\phi}(Y_t \mid \mathbf{y}_1^{t-1}) \| P_\theta(Y_t \mid \mathbf{y}_1^{t-1})\right)\right] \\
={}& \sum_{t=1}^{T} \mathop{\mathbb{E}}_{\mathbf{y}_1^t \sim P_{Q_\phi}(\mathbf{Y}_1^t)} \left[\mathrm{CE}\left(P_{Q_\phi}(Y_t \mid \mathbf{y}_1^{t-1}) \| P_\theta(Y_t \mid \mathbf{y}_1^{t-1})\right)\right] \\
={}& \mathop{\mathbb{E}}_{\mathbf{y} \sim P_{Q_\phi}(\mathbf{Y})} \sum_{t=1}^{|\mathbf{y}|} \mathrm{CE}\left(P_{Q_\phi}(Y_t \mid \mathbf{y}_1^{t-1}) \| P_\theta(Y_t \mid \mathbf{y}_1^{t-1})\right)
\end{aligned}
$$

Compared to the of objective of RAML in Eqn. (8.2), having access to $P_{Q_\phi}(Y_t \mid \mathbf{y}_1^{t-1})$ allows us to provide a distinct token-level target for each conditional factor $P_\theta(Y_t \mid \mathbf{y}_1^{t-1})$ of the model. While directly sampling from P_R is practically infeasible (§8.2.1), having a parametric target distribution P_{Q_ϕ} makes it theoretically possible to sample from P_{Q_ϕ} and perform the optimization. However, empirically, we find the samples from P_{Q_ϕ} are not diverse enough (§8.7). Hence, we fall back to the same importance sampling approach (see Appendix 8.6.2) as used in RAML.

Finally, since the algorithm utilizes the optimal soft action-value function to construct the token-level target, we will refer to it as value augmented maximum likelihood (VAML) in the sequel.

8.4.2 Entropy-regularized Actor Critic

The second algorithm follows the discussion at the end of §8.3.2, which is essentially an actor-critic algorithm based on the entropy-regularized MDP in Corollary 1. For this reason, we name the algorithm entropy-regularized actor critic (ERAC). As with standard AC algorithm, the training process interleaves the evaluation of current policy using the parametric critic Q_ϕ and the optimization of the actor policy π_θ given the current critic.

Critic Training. The critic is trained to perform policy *evaluation* using the temporal difference learning (TD), which minimizes the TD error

$$\min_{\phi} \mathop{\mathbb{E}}_{\mathbf{y} \sim \pi_\theta} \sum_{t=1}^{|\mathbf{y}|} \Big[Q_\phi(\mathbf{y}_1^{t-1}, y_t; \mathbf{y}^*) - \hat{Q}_{\bar{\phi}}(\mathbf{y}_1^{t-1}, y_t; \mathbf{y}^*) \Big]^2 \tag{8.17}$$

where the TD target $\hat{Q}_{\bar{\phi}}$ is constructed based on fixed policy iteration in Eqn. (8.8), i.e.,

$$\begin{aligned} \hat{Q}_{\bar{\phi}}(\mathbf{y}_1^{t-1}, y_t; \mathbf{y}^*) &= r(\mathbf{y}_1^{t-1}, y_t) + \tau\, \mathcal{H}(\pi_\theta(\cdot \mid \mathbf{y}_1^t)) \\ &\quad + \sum_{w \in \mathcal{W}} \pi_\theta(w \mid \mathbf{y}_1^t) Q_{\bar{\phi}}(\mathbf{y}_1^t, w; \mathbf{y}^*). \end{aligned} \tag{8.18}$$

It is worthwhile to emphasize that the objective (8.17) trains the critic Q_ϕ to evaluate the current policy. Hence, it is entirely different from the objective (8.15), which is performing policy optimization by Soft Q-Learning. Also, the trajectories $\mathbf{y}$ used in (8.17) are sequences drawn from the actor policy π_θ, while objective (8.15) theoretically accepts any trajectory since Soft Q-Learning can be fully off-policy.[4] Finally, following Bahdanau et al. [10], the TD target $\hat{Q}_{\bar{\phi}}$ in Eqn. (8.8) is evaluated using a target network, which is indicated by the bar sign above the parameters, i.e., $\bar{\phi}$. The target network is slowly updated by linearly interpolating with the up-to-date network, i.e., the update is $\bar{\phi} \leftarrow \beta\phi + (1-\beta)\bar{\phi}$ for β in $(0, 1)$ [145].

We also adapt another technique proposed by Bahdanau et al. [10], which smooths the critic by minimizing the "variance" of Q-values, i.e.,

$$\min_{\phi} \lambda_{\text{var}} \mathop{\mathbb{E}}_{\mathbf{y} \sim \pi_\theta} \sum_{t=1}^{|\mathbf{y}|} \sum_{w \in \mathcal{W}} \left[Q_\phi(\mathbf{y}_1^t, w; \mathbf{y}^*) - \bar{Q}_\phi(\mathbf{y}_1^t; \mathbf{y}^*) \right]^2$$

where $\bar{Q}_\phi(\mathbf{y}_1^t; \mathbf{y}^*) = \frac{1}{|\mathcal{W}|} \sum_{w' \in \mathcal{W}} Q_\phi(\mathbf{y}_1^t, w'; \mathbf{y}^*)$ is the mean Q-value, and λ_{var} is a hyper-parameter controlling the relative weight between the TD loss and the smooth loss.

Actor Training. Given the critic Q_ϕ, the actor gradient (to maximize the expected return) is given by the policy gradient theorem of the entropy-regularized RL [223], which has the form

$$\begin{aligned} \mathop{\mathbb{E}}_{\mathbf{y} \sim \pi_\theta} \sum_{t=1}^{|\mathbf{y}|} \sum_{w \in \mathcal{W}} & \nabla_\theta \pi_\theta(w \mid \mathbf{y}_1^{t-1}) Q_\phi(\mathbf{y}_1^{t-1}, w; \mathbf{y}^*) \\ & + \tau \nabla_\theta \mathcal{H}(\pi_\theta(\cdot \mid \mathbf{y}_1^{t-1})). \end{aligned} \tag{8.19}$$

Here, for each step t, we follow Bahdanau et al. [10] to sum over the entire symbol set $\mathcal{W}$, instead of using the single sample estimation often seen in RL. Hence, no baseline is employed. It is worth mentioning that Eqn. (8.19) is *not* simply adding an entropy term to the standard

[4] Different from Bahdanau et al. [10], we don't use a delayed actor network to collect trajectories for critic training.

policy gradient as in A3C [173]. The difference lies in that the critic Q_ϕ trained by Eqn. (8.17) additionally captures the *entropy from future steps*, while the $\nabla_\theta \mathcal{H}$ term only captures the entropy of the current step.

Finally, similar to [10], we find it necessary to first pretrain the actor using MLE and then pretrain the critic before the actor-critic training. Also, to prevent divergence during actor-critic training, it is helpful to continue performing MLE training along with Eqn. (8.19), though using a smaller weight λ_{mle}.

8.5 Related Work

Language Generation The sequence-to-sequence model (seq2seq) [9, 245] has results in successes of many conditional sequence prediction problems, including machine translation [9, 245], automatic summarization [215], image captioning [121, 267, 280] and speech recognition [32].

Task Loss Optimization and Exposure Bias Apart from the previously introduced RAML, BSO, Actor-Critic (§9.1), MIXER [204] also utilizes chunk-level signals where the length of chunk grows as training proceeds. In contrast, minimum risk training [229] directly optimizes sentence-level BLEU. As a result, it requires a large number (100) of samples per data to work well. To solve the exposure bias, scheduled sampling [14] adopts a curriculum learning strategy to bridge the training and the inference. Professor forcing [136] introduces an adversarial training mechanism to encourage the dynamics of the model to be the same at training time and inference time. For image caption, self-critic sequence training (SCST) [208] extends the MIXER algorithm with an improved baseline based on the current model performance.

Entropy-regularized RL Entropy regularization been explored by early work in RL and inverse RL [275, 320]. Lately, Schulman et al. [223] establish the equivalence between policy gradients and Soft Q-Learning under entropy-regularized RL. Motivated by the multi-modal behavior induced by entropy-regularized RL, Haarnoja et al. [82] apply energy-based policy and Soft Q-Learning to continuous domain. Later, Nachum et al. [174] proposes path consistency learning, which can be seen as a multi-step extension to Soft Q-Learning. More recently, in the domain of simulated control, Haarnoja et al. [83] also consider the actor critic algorithm under the framework of entropy regularized reinforcement learning. Despite the conceptual similarity to ERAC presented here, Haarnoja et al. [83] focuses on continuous control and employs the advantage actor critic variant as in [173], while ERAC follows the Q actor critic as in [10].

8.6 Implementation

8.6.1 RAML

In RAML, we want to optimize the cross entropy $\mathrm{CE}\left(P_R(\mathbf{Y} \mid \mathbf{x}^*, \mathbf{y}^*) \| P_\theta(\mathbf{Y} \mid \mathbf{x}^*)\right)$. As discussed in §8.2.1, directly sampling from the exponentiated pay-off distribution $P_R(Y \mid x^*)$ is impractical. Hence, normalized importance sampling has been exploited in previous work [156, 183]. Define the proposal distribution to be $P_S(\mathbf{Y} \mid \mathbf{x}^*, \mathbf{y}^*)$. Then, the objective can be rewritten as

$$\begin{aligned}
\mathrm{CE}\left(P_R(\mathbf{Y} \mid \mathbf{x}^*, \mathbf{y}^*) \| P_\theta(\mathbf{Y} \mid \mathbf{x}^*)\right) &= - \mathop{\mathbb{E}}_{\mathbf{y} \sim P_S(\mathbf{Y} \mid \mathbf{x}^*, \mathbf{y}^*)} \frac{P_R(\mathbf{y} \mid \mathbf{x}^*, \mathbf{y}^*)}{P_S(\mathbf{y} \mid \mathbf{x}^*, \mathbf{y}^*)} \log P_\theta(\mathbf{y} \mid \mathbf{x}^*) \\
&= - \mathop{\mathbb{E}}_{\mathbf{y} \sim P_S(\mathbf{Y} \mid \mathbf{x}^*, \mathbf{y}^*)} \frac{\frac{\exp(R(\mathbf{y}, \mathbf{y}^*)/\tau)}{\tilde{P}_S(\mathbf{y} \mid \mathbf{x}^*, \mathbf{y}^*)}}{\mathbb{E}_{\mathbf{y}' \sim P_S(\mathbf{Y} \mid \mathbf{x}^*, \mathbf{y}^*)} \frac{\exp(R(\mathbf{y}', \mathbf{y}^*)/\tau)}{\tilde{P}_S(\mathbf{y}' \mid \mathbf{x}^*, \mathbf{y}^*)}} \log P_\theta(\mathbf{y} \mid \mathbf{x}^*) \\
&= - \mathop{\mathbb{E}}_{\mathbf{y} \sim P_S(\mathbf{Y} \mid \mathbf{x}^*, \mathbf{y}^*)} \frac{w(\mathbf{y}, \mathbf{y}^*)}{\mathbb{E}_{\mathbf{y}' \sim P_S(\mathbf{Y} \mid \mathbf{x}^*, \mathbf{y}^*)} w(\mathbf{y}', \mathbf{y}^*)} \log P_\theta(\mathbf{y} \mid \mathbf{x}^*) \\
&\approx - \sum_{i=1}^{M} \frac{w(\mathbf{y}^{(i)}, \mathbf{y}^*)}{\sum_{i=1}^{M} w(\mathbf{y}^{(i)}, \mathbf{y}^*)} \log P_\theta(\mathbf{y}^{(i)} \mid \mathbf{x}^*),
\end{aligned}$$

where $w(\mathbf{y}, \mathbf{y}^*) = \frac{\exp(R(\mathbf{y}, \mathbf{y}^*)/\tau)}{\tilde{P}_S(\mathbf{y} \mid \mathbf{x}^*, \mathbf{y}^*)}$ is the unnormalized importance weight, $\tilde{P}_S$ denotes the unnormalized probability of $P_S = \frac{\tilde{P}_S}{Z}$, M is the number of samples used, and $\mathbf{y}^{(i)}$ is the i-th sample drawn from the proposal distribution $P_S(\mathbf{Y} \mid \mathbf{x}^*, \mathbf{y}^*)$.

With importance sampling, the problem turns to what proposal distribution we should use. In the original work [183], the proposal distribution is defined by the hamming distance as used. Ma et al. [156] find that it suffices to perform N-gram replacement of the reference sentence. Specifically, $P_S(\mathbf{Y} \mid \mathbf{x}^*, \mathbf{y}^*)$ can be a uniform distribution defined on set $\mathcal{Y}_{\text{ngram}}$ where $\mathcal{Y}_{\text{ngram}}$ is obtained by randomly replacing an n-gram of $\mathbf{y}^*$ ($n \leq 4$).

In this work, we adapt the simple n-gram replacement distribution, denoted as $P_{\text{ngram}}(\mathbf{Y} \mid \mathbf{x}^*, \mathbf{y}^*)$, which simplifies the RAML objective into

$$\min_\theta - \sum_{i=1}^{M} \frac{\exp\left(R(\mathbf{y}^{(i)}, \mathbf{y}^*)/\tau\right)}{\sum_{i=1}^{M} \exp\left(R(\mathbf{y}^{(i)}, \mathbf{y}^*)/\tau\right)} \log P_\theta(\mathbf{y}^{(i)} \mid \mathbf{x}^*)$$

Following Ma et al. [156], we make sure the reference sequence is always among the M samples used.

8.6.2 VAML

As discussed in §8.4, the VAML training consists of two phases:

- In the first phase, Soft Q-Learning is used to train Q_ϕ based on Eqn. (8.15). Since Soft Q-Learning accepts off-policy trajectories, in this work, we use two types of off-policy sequences:
    1. The first type is simply the ground-truth sequence, which provides strong learning signals.
    2. The second type of sequences is actually drawn from the same n-gram replacement distribution discussed above. The reason is that in the second training phase, such n-gram replaced trajectories will be used. Since the learned Q_ϕ won't be perfect, we hope the exposing Q_ϕ with these trajectories can improve its accuracy on them, making the second phase of training easier.

Algorithm 2 summarizes the first phase.

Algorithm 2 VAML Phase 1: Soft Q-Learning to approximate Q^*

```
Require: A Q-function approximator $Q_\phi$ with parameter $\phi$, and the hyper-parameters $\tau$, $M$.
while Not Converged do
    Receive a random example $(\mathbf{x}^*, \mathbf{y}^*)$.
    Sample $M-1$ sequences $\{\mathbf{y}^{(i)}\}_{i=1}^{M-1}$ from $P_{\text{ngram}}(\mathbf{Y} \mid \mathbf{x}^*, \mathbf{y}^*)$ and let $\mathbf{y}^{(M)} = \mathbf{y}^*$.
    Compute all the rewards $r(\mathbf{y}_1^{t-1}, y_t; \mathbf{y}^*)$ for each $\mathbf{y} \in \{\mathbf{y}^{(i)}\}_{i=1}^{M}$ and $t = 1, \dots, |\mathbf{y}|$.
    Compute the target Q-values for each $\mathbf{y} \in \{\mathbf{y}^{(i)}\}_{i=1}^{M}$ and $t = 1, \dots, |\mathbf{y}|$

        $\hat{Q}_\phi(\mathbf{y}_1^{t-1}, y_t; \mathbf{y}^*) = r(\mathbf{y}_1^{t-1}, y_t; \mathbf{y}^*) + \tau \log \sum_{w \in \mathcal{W}} \exp\big(Q_\phi(\mathbf{y}_1^{t}, w; \mathbf{y}^*)/\tau\big).$

    Compute the Soft-Q Learning loss

        $\mathcal{L}_{\text{SoftQ}} = \frac{1}{M} \sum_{i=1}^{M} \sum_{t=1}^{|\mathbf{y}^{(i)}|} \left\| Q_\phi({\mathbf{y}^{(i)}}_1^{t-1}, y_t^{(i)}; \mathbf{y}^*) - \hat{Q}_\phi({\mathbf{y}^{(i)}}_1^{t-1}, y_t^{(i)}; \mathbf{y}^*) \right\|_2^2.$

    Update $Q_\phi$ according to the loss $\mathcal{L}_{\text{SoftQ}}$.
end while
```

- Once the Q_ϕ is well trained in the first phase, the second phase is to minimize the cross entropy $\text{CE}\left(P_{Q_\phi}(\mathbf{Y} \mid \mathbf{x}^*, \mathbf{y}^*) \| P_\theta(\mathbf{Y} \mid \mathbf{x}^*)\right)$ based on Eqn. (8.16), i.e.,

$$\min_\theta \mathop{\mathbb{E}}_{\mathbf{y} \sim P_{Q_\phi}} \left[\sum_{t=1}^{|\mathbf{y}|} \text{CE}\left(P_{Q_\phi}(Y_t \mid \mathbf{y}_1^{t-1}) \| P_\theta(Y_t \mid \mathbf{y}_1^{t-1})\right) \right].$$

Ideally, we would like to directly sample from P_{Q_ϕ}, and perform the optimization. However, we find samples from P_{Q_ϕ} are quite similar to each other. We conjecture this results from both the imperfect training in the first phase, and the intrinsic difficulty of getting diverse samples from an exponentially large space when the distribution is high concentrated.

Nevertheless, for this work, we fall back to the same importance sampling method as used in RAML and use the n-gram replacement distribution as the proposal. Hence, the objective becomes

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{\mathbf{y}\sim P_{Q_\phi}}\left[\sum_{t=1}^{|\mathbf{y}|}\mathrm{CE}\left(P_{Q_\phi}(Y_t \mid \mathbf{y}_1^{t-1})\|P_\theta(Y_t \mid \mathbf{y}_1^{t-1})\right)\right] \\
=&\mathop{\mathbb{E}}_{\mathbf{y}\sim P_{\text{ngram}}}\left[\frac{w(\mathbf{y},\mathbf{y}^*)}{\mathbb{E}_{\mathbf{y}'\sim P_{\text{ngram}}(\mathbf{Y}\mid\mathbf{x}^*,\mathbf{y}^*)}\,w(\mathbf{y}',\mathbf{y}^*)}\sum_{t=1}^{|\mathbf{y}|}\mathrm{CE}\left(P_{Q_\phi}(Y_t \mid \mathbf{y}_1^{t-1})\|P_\theta(Y_t \mid \mathbf{y}_1^{t-1})\right)\right] \\
\approx&\sum_{i=1}^{M}\frac{\exp\left(R(\mathbf{y}^{(i)},\mathbf{y}^*)/\tau\right)}{\sum_{i=1}^{M}\exp\left(R(\mathbf{y}^{(i)},\mathbf{y}^*)/\tau\right)}\left[\sum_{t=1}^{|\mathbf{y}^{(i)}|}\mathrm{CE}\left(P_{Q_\phi}(Y_t \mid {\mathbf{y}^{(i)}}_1^{t-1})\|P_\theta(Y_t \mid {\mathbf{y}^{(i)}}_1^{t-1})\right)\right].
\end{aligned}
$$

However, we found directly using this objective does not yield improved performance compared to RAML, mostly likely due to some erratic estimations of Q_ϕ. Thus, we only use this objective for some step with certain probability $\kappa \in (0,1)$, leaving others trained by MLE. Formally, define

$$
\mathcal{J}_\kappa(\mathbf{y}_1^t) = \mathop{\mathbb{E}}_{z\sim\text{Bernoulli}(\kappa)}\left[z\mathrm{CE}\left(P_{Q_\phi}(Y_t \mid \mathbf{y}_1^{t-1})\|P_\theta(Y_t \mid \mathbf{y}_1^{t-1})\right) - (1-z)\log P_\theta(y_t \mid \mathbf{y}_1^{t-1})\right],
$$

the VAML objective practically used is

$$
\min_\theta\sum_{i=1}^{M}\frac{\exp\left(R(\mathbf{y}^{(i)},\mathbf{y}^*)/\tau\right)}{\sum_{i=1}^{M}\exp\left(R(\mathbf{y}^{(i)},\mathbf{y}^*)/\tau\right)}\left[\sum_{t=1}^{|\mathbf{y}^{(i)}|}\mathcal{J}_\kappa({\mathbf{y}^{(i)}}_1^t)\right].
$$

Algorithm 3 summarizes the second phase.

Algorithm 3 VAML Phase 2: Sequence model training with token-level target

```
Require: A sequence prediction model $P_\theta$ with parameter $\theta$, a pre-trained Q-function approximator $Q_\phi$, and hyper-parameters $\tau$, $M$, $\kappa$
while Not Converged do
    Receive a random example $(\mathbf{x}^*, \mathbf{y}^*)$.
    Sample $M-1$ sequences $\{\mathbf{y}^{(i)}\}_{i=1}^{M-1}$ from $P_{\text{ngram}}(\mathbf{Y} \mid \mathbf{x}^*, \mathbf{y}^*)$ and let $\mathbf{y}^{(M)} = \mathbf{y}^*$.
    Compute the VAML loss using
        $\mathcal{L}_{\text{VAML}} = \sum_{i=1}^{M}\frac{\exp\left(R(\mathbf{y}^{(i)},\mathbf{y}^*)/\tau\right)}{\sum_{i=1}^{M}\exp\left(R(\mathbf{y}^{(i)},\mathbf{y}^*)/\tau\right)}\left[\sum_{t=1}^{|\mathbf{y}^{(i)}|}\mathcal{J}_\kappa({\mathbf{y}^{(i)}}_1^t)\right].$
    Update $P_\theta$ according to the loss $\mathcal{L}_{\text{VAML}}$.
end while
```

8.6.3 ERAC

Following Bahdanau et al. [10], we first pre-train the actor, then train the critic with the fixed actor and finally fine-tune them together. The specific procedure for training ERAC is

- Pre-training the actor using maximum likelihood training
- Pre-training the critic using Algorithm 4 with the actor fixed
- Fine-tuning both the actor and critic with Algorithm 4

8.6.4 Hyper-parameters

RAML & VAML The hyper-parameters for RAML and VAML training are summarized in Tab. 8.1. We set the gradient clipping value to 5.0 for both the Q-function approximator Q_ϕ and the sequence prediction model P_θ, except for the sequence prediction model in the captioning task where the gradient clipping value is set to 1.0.

	Machine Translation			**Image Captioning**		
Hyper-parameters	VAML-1	VAML-2	RAML	VAML-1	VAML-2	RAML
optimizer	Adam	SGD	SGD	Adam	SGD	SGD
learning rate	0.001	0.6	0.6	0.001	0.5	0.5
batch size	50	42	42	32×5	32×5	32×5
M	5	5	5	2	6	6
τ	0.4	0.4	0.4	0.7	0.7	0.7
κ	N.A.	0.2	N.A.	N.A.	0.1	N.A.

Table 8.1: Optimization related hyper-parameters of RAML and VAML for two tasks. “VAML-1” and “VAML-2” indicate the phase 1 and phase 2 of VAML training respectively. “N.A.” means not applicable. “32×5” indicates using 32 images each with 5 reference captions.

AC & ERAC As described in §8.6.3, the training using AC and ERAC involves three phases. The hyper-parameters used for ERAC training in each phase are summarized in Table 8.2. In all phases, the learning rate is halved when there is no improvement on the validation set. We use the same hyper-parameters for AC training, except that the entropy regularization coefficient τ is 0. Similar to the VAML case, the gradient clipping value is set to 5.0 for both the actor and the critic, except that we set the gradient clipping value to 1.0 for the actor in the captioning task.

8.7 Experiments

8.7.1 Experiment Settings

In this work, we focus on two sequence prediction tasks: machine translation and image captioning. Due to the space limit, we only present the information necessary to compare the empirical

Hyper-parameters	**MT w/ input feeding**	**MT w/o input feeding**	**Image Captioning**
Pre-train Actor			
optimizer	SGD	SGD	SGD
learning rate	0.6	0.6	0.5
batch size	50	50	32 × 5
Pre-train Critic			
optimizer	Adam	Adam	Adam
learning rate	0.001	0.001	0.001
batch size	50	50	32 × 5
τ (entropy regularization)	0.045	0.04	0.01
β (target net speed)	0.001	0.001	0.001
λ_{var} (smoothness)	0.001	0.001	0.001
Joint Training			
optimizer	Adam	Adam	Adam
learning rate	0.0001	0.0001	0.0001
batch size	50	50	32 × 5
τ (entropy regularization)	0.045	0.04	0.01
β (target net speed)	0.001	0.001	0.001
λ_{var} (smoothness)	0.001	0.001	0.001
λ_{MLE}	0.1	0.1	0.1

Table 8.2: Hyper-parameters for ERAC training

results at this moment.

Machine Translation Following Ranzato et al. [204], we evaluate on IWSLT 2014 German-to-English dataset [164]. The corpus contains approximately $153K$ sentence pairs in the training set. We follow the pre-processing procedure used in [204].

Architecture wise, we employ a seq2seq model with dot-product attention [9, 155], where the encoder is a bidirectional LSTM [99] with each direction being size 128, and the decoder is another LSTM of size 256. Moreover, we consider two variants of the decoder, one using the input feeding technique [155] and the other not.

For all algorithms, the sequence-level BLEU score is employed as the pay-off function R, while the corpus-level BLEU score [189] is used for the final evaluation. The sequence-level BLEU score is scaled up by the sentence length so that the scale of the immediate reward at each step is invariant to the length.

Image Captioning For image captioning, we consider the MSCOCO dataset [147]. We adapt the same preprocessing procedure and the train/dev/test split used by Karpathy and Fei-Fei [121].

The NIC [267] is employed as the baseline model, where a feature vector of the image is

	MT (w/o input feeding)			**MT (w/ input feeding)**			**Image Captioning**		
Algorithm	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max
MLE	27.01 ± 0.20	26.72	27.27	28.06 ± 0.15	27.84	28.22	29.54 ± 0.21	29.27	29.89
RAML	27.74 ± 0.15	27.47	27.93	28.56 ± 0.15	28.35	28.80	29.84 ± 0.21	29.50	30.17
VAML	**28.16 ± 0.11**	**28.00**	**28.26**	**28.84 ± 0.10**	**28.62**	**28.94**	**29.93 ± 0.22**	**29.51**	**30.24**
AC	28.04 ± 0.05	27.97	28.10	29.05 ± 0.06	28.95	29.16	30.90 ± 0.20	30.49	31.16
ERAC	**28.30 ± 0.06**	**28.25**	**28.42**	**29.31 ± 0.04**	**29.26**	**29.36**	**31.44 ± 0.22**	**31.07**	**31.82**

Table 8.3: Test results on two benchmark tasks. Bold faces highlight the best in the corresponding category.

extracted by a pre-trained CNN and then used to initialize the LSTM decoder. Different from the original NIC model, we employ a pre-trained 101-layer ResNet [88] rather than a GoogLeNet as the CNN encoder.

For training, each image-caption pair is treated as an i.i.d. sample, and sequence-level BLEU score is used as the pay-off. For testing, the standard multi-reference BLEU4 is used.

8.7.2 Comparison with the Direct Baseline

Firstly, we compare ERAC and VAML with their corresponding direct baselines, namely AC [10] and RAML [183] respectively. As a reference, the performance of MLE is also provided.

Due to non-neglected performance variance observed across different runs, we run each algorithm for 9 times with different random seeds,[5] and report the average performance, the standard deviation and the performance range (min, max).

Machine Translation The results on MT are summarized in the left half of Tab. 8.3. Firstly, all four advanced algorithms significantly outperform the MLE baseline. More importantly, both VAML and ERAC improve upon their direct baselines, RAML and AC, by a clear margin on average. The result suggests the two algorithms both well combine the benefits of a delicate credit assignment scheme and the entropy regularization, achieving improved performance.

Image Captioning The results on image captioning are shown in the right half of Tab. 8.3. Despite the similar overall trend, the improvement of VAML over RAML is smaller compared to that in MT. Meanwhile, the improvement from AC to ERAC becomes larger in comparison. We suspect this is due to the multi-reference nature of the MSCOCO dataset, where a larger entropy is preferred. As a result, the explicit entropy regularization in ERAC becomes immediately fruitful. On the other hand, with multiple references, it can be more difficult to learn a good oracle Q^* (Eqn. (8.14)). Hence, the token-level target can be less accurate, resulting in smaller improvement.

[5] For AC, ERAC and VAML, 3 different critics are trained first, and each critic is then used to train 3 actors.

8.7.3 Comparison with Existing Work

To further evaluate the algorithms, we compare ERAC and VAML with the large body of existing algorithms evaluated on IWSTL 2014. As a note of caution, previous work don't employ the exactly same architectures (e.g. number of layers, hidden size, attention type, etc.). Despite that, for VAML and ERAC, we use an architecture that is most similar to the majority of previous works, which is the one described in §8.7.1 with input feeding.

Based on the setting, the comparison is summarized in Table 8.4. As we can see, both VAML and ERAC outperform previous methods, with ERAC leading the comparison with a significant margin. This further verifies the effectiveness of the two algorithms.

Algorithm	**BLEU**
MIXER [204]	20.73
BSO [276]	27.9
Q(BLEU) [140]	28.3
AC [10]	28.53
RAML [156]	28.77
VAML	28.94
ERAC	**29.36**

Table 8.4: Comparison with existing algorithms on IWSTL 2014 dataset for MT. All numbers of previous algorithms are from the original work.

8.7.4 Ablation Study

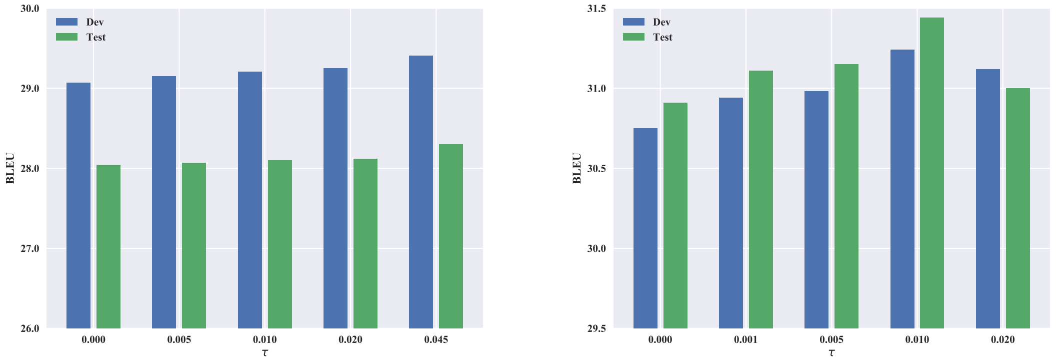
Due to the overall excellence of ERAC, we study the importance of various components of it, hopefully offering a practical guide for readers. As the input feeding technique largely slows down the training, we conduct the ablation based on the model variant *without* input feeding.

λ_{var} \ β	0.001	0.01	0.1	1
0	27.91	26.27†	28.88	27.38†
0.001	**29.41**	29.26	29.32	27.44

Table 8.5: Average validation BLEU of ERAC. As a reference, the average BLEU is 28.1 for MLE. $\lambda_{\text{var}} = 0$ means not using the smoothing technique. $\beta = 1$ means not using a target network. † indicates excluding extreme values due to divergence.

Firstly, we study the importance of two techniques aimed for training stability, namely the target network and the smoothing technique (§8.4.2). Based on the MT task, we vary the update speed β of the target critic, and the λ_{var}, which controls the strength of the smoothness regularization. The average *validation* performances of different hyper-parameter values are summarized in Tab. 8.5.

- Comparing the two rows of Tab. 8.5, the smoothing technique consistently leads to performance improvement across all values of τ. In fact, removing the smoothing objective often causes the training to diverge, especially when $\beta = 0.01$ and 1. But interestingly, we find the divergence does not happen if we update the target network a little bit faster ($\beta = 0.1$) or quite slowly ($\beta = 0.001$).
- In addition, even with the smoothing technique, the target network is still necessary. When the target network is not used ($\beta = 1$), the performance drops below the MLE baseline. However, as long as a target network is employed to ensure the training stability, the specific choice of target network update rate does not matter as much. Empirically, it seems using a slower ($\beta = 0.001$) update rate yields the best result.



(a) Machine translation

(b) Image captioning

Figure 8.1: ERAC's average performance over multiple runs on two tasks when varying τ.

Next, we investigate the effect of enforcing different levels of entropy by varying the entropy hyper-parameter τ. As shown in Fig. 8.1, it seems there is always a sweet spot for the level of entropy. On the one hand, posing an over strong entropy regularization can easily cause the actor to diverge. Specifically, the model diverges when τ reaches 0.03 on the image captioning task or 0.06 on the machine translation task. On the other hand, as we decrease τ from the best value to 0, the performance monotonically decreases as well. This observation further verifies the effectiveness of entropy regularization in ERAC, which well matches our theoretical analysis.

Finally, as discussed in §8.4.2, ERAC takes the effect of future entropy into consideration, and thus is different from simply adding an entropy term to the standard policy gradient as in A3C [173]. To verify the importance of explicitly modeling the entropy from future steps, we compared ERAC with the variant that only applies the entropy regularization to the actor but not to the critic. In other words, the τ is set to 0 when performing policy evaluating according to Eqn. (8.4.2), while the τ for the entropy gradient in Eqn. (8.19) remains. The comparison result based on 9 runs on test set of IWSTL 2014 is shown in Table 8.6. As we can see, simply adding a local entropy gradient does not even improve upon the AC. This further verifies the difference between ERAC and A3C, and shows the importance of taking future entropy into consideration.

Algorithm	Mean	Max
ERAC	**28.30 ± 0.06**	**28.42**
ERAC w/o Future Ent.	28.06 ± 0.05	28.11
AC	28.04 ± 0.05	28.10

Table 8.6: Comparing ERAC with the variant without considering future entropy.

8.7.5 Comparison with Previous Work

The detailed comparison with previous work in shown in Table 8.7. Under different comparable architectures (1 layer or 2 layers), ERAC outperforms previous algorithms with a clear margin.

Algorithm	Encoder		Decoder				BLEU
	NN Type	Size	NN Type	Size	Attention	Input Feed	
MIXER [204]	1-layer CNN	256	1-layer LSTM	256	Dot-Prod	N	20.73
BSO [276]	1-layer BiLSTM	128 × 2	1-layer LSTM	256	Dot-Prod	Y	27.9
Q(BLEU) [140]	1-layer BiLSTM	128 × 2	1-layer LSTM	256	Dot-Prod	Y	28.3
AC [10]	1-layer BiGRU	256 × 2	1-layer GRU	256	MLP	Y	28.53
RAML [156]	1-layer BiLSTM	256 × 2	1-layer LSTM	256	Dot-Prod	Y	28.77
VAML	1-layer BiLSTM	128 × 2	1-layer LSTM	256	Dot-Prod	Y	28.94
ERAC	1-layer BiLSTM	128 × 2	1-layer LSTM	256	Dot-Prod	Y	**29.36**
NPMT [106]	2-layer BiGRU	256 × 2	2-layer LSTM	512	N.A.	N.A.	29.92
NPMT+LM [106]	2-layer BiGRU	256 × 2	2-layer LSTM	512	N.A.	N.A.	30.08
ERAC	2-layer BiLSTM	256 × 2	2-layer LSTM	512	Dot-Prod	Y	**30.85**

Table 8.7: Comparison of algorithms with detailed architecture information on the IWSTL 2014 dataset for MT.

8.8 Discussion

In this work, motivated by the intriguing connection between the token-level RAML and the entropy-regularized RL, we present two algorithms for neural sequence prediction. Despite the distinct training procedures, both algorithms combine the idea of fine-grained credit assignment and the entropy regularization, leading to positive empirical results.

However, many problems remain widely open. In particular, the oracle Q-function Q_ϕ we obtain is far from perfect. We believe the ground-truth reference contains sufficient information for such an oracle, and the current bottleneck lies in the RL algorithm. Given the numerous potential applications of such an oracle, we believe improving its accuracy will be a promising future direction.

Though the method has strong theoretical and leads to significant performance gains, the inductive bias is designed specifically for the language generation task. For a new problem, it would require significant efforts to find out the desired inductive biases.

Algorithm 4 ERAC Algorithm

```
Require: A critic $Q_\phi(\mathbf{y}_1^{t-1}, y_t; \mathbf{y}^*)$ and an actor $\pi_\theta(w \mid \mathbf{y}_1^t)$ with weights $\phi$ and $\theta$ respectively,
    and hyper-parameters $\tau$, $\beta$, $\lambda_{\text{var}}$, $\lambda_{\text{mle}}$
Initialize delayed target critic $Q_{\bar{\phi}}$ with the same weights: $\bar{\phi} = \phi$.
while Not Converged do
    Receive a random example $(\mathbf{x}^*, \mathbf{y}^*)$.
    Generate a sequence $\mathbf{y}$ from $\pi_\theta$.
    Compute the rewards $r(\mathbf{y}_1^{t-1}, y_t; \mathbf{y}^*)$ for $t = 1, \dots, |\mathbf{y}|$.
    Compute targets for the critic

        $\hat{Q}_{\bar{\phi}}(\mathbf{y}_1^{t-1}, y_t; \mathbf{y}^*) = r(\mathbf{y}_1^{t-1}, y_t) + \tau\,\mathcal{H}(\pi_\theta(\cdot \mid \mathbf{y}_1^t)) + \sum_{w \in \mathcal{W}} \pi_\theta(w \mid \mathbf{y}_1^t) Q_{\bar{\phi}}(\mathbf{y}_1^t, w; \mathbf{y}^*).$

    Compute loss for critic

        $\mathcal{L}_{\text{critic}} = \sum_{t=1}^{|\mathbf{y}|} \left[Q_\phi(\mathbf{y}_1^{t-1}, y_t; \mathbf{y}^*) - \hat{Q}_{\bar{\phi}}(\mathbf{y}_1^{t-1}, y_t; \mathbf{y}^*)\right]^2 + \lambda_{\text{var}} \sum_{w \in \mathcal{W}} \left[Q_\phi(\mathbf{y}_1^{t-1}, w; \mathbf{y}^*) - \bar{Q}_\phi(\mathbf{y}_1^{t-1}; \mathbf{y}^*)\right]^2,$
            where $\bar{Q}_\phi(\mathbf{y}_1^{t-1}; \mathbf{y}^*) = \frac{1}{|\mathcal{W}|} \sum_{w' \in \mathcal{W}} Q_\phi(\mathbf{y}_1^{t-1}, w'; \mathbf{y}^*)$

    Compute loss for actor

        $\mathcal{L}_{\text{actor}} = -\left[\sum_{t=1}^{|\mathbf{y}|} \sum_{w \in \mathcal{W}} \pi_\theta(w \mid \mathbf{y}_1^{t-1}) Q_\phi(\mathbf{y}_1^{t-1}, w; \mathbf{y}^*) + \tau\mathcal{H}(\pi_\theta(\cdot \mid \mathbf{y}_1^{t-1})) + \lambda_{\text{mle}} \sum_{t=1}^{|\mathbf{y}^*|} \log \pi_\theta(y_t^* \mid {\mathbf{y}^*}_1^{t-1})\right]$

    Update critic according to the loss $\mathcal{L}_{\text{critic}}$.
    If actor is not fixed, update actor according to the loss $\mathcal{L}_{\text{actor}}$
    Update delayed target critic: $\bar{\phi} = \beta\phi + (1 - \beta)\bar{\phi}$
end while
```

Chapter 9

Making Use of Knowledge Bases as External Knowledge for Relation Extraction

In this chapter, we present a method to incorporate structured knowledge information from knowledge bases to enable the model to understand entities that are usually not well covered in raw text. We show that using external knowledge bases is particularly useful for tasks related to entities and relations and leads to significant performance gains.

9.1 Introduction

Relation Extraction (RE) has long been a core task in natural language processing and information extraction, which aims to extract structured knowledge from unstructured text. Figure 9.1 shows a simple case, where the relation between `Adolfo NicolÃąs PachÃşn` and `Roman Catholic Church` can be extracted based on the first sentence. The extracted relation triplets can further be used in downstream tasks, such as knowledge base population [112] and question answering [296].

Traditional methods focus on utilizing rich information from lexical and syntactic features to help relation extraction [282, 295, 310]. More recently, unsupervised representation learning models such as BERT [57] have been shown to lead to significant improvements [238]. In essence, representation learning's effectiveness can be attributed to enabling model to understand natural language by making use of large amounts of external unstructured text corpora.

Given the success of using large-scale external text corpora, researchers have proposed to incorporate structured knowledge information to enable the model to understand entities that are usually not well covered in raw text. Two recent work integrate pretrained knowledge embeddings into unsupervised representation learning models. Specifically, Zhang et al. [312] train a representation learning model with entity mentions linked to their corresponding entities in knowledge bases to use their knowledge embeddings. Peters et al. [199] propose to finetune BERT by dynamically retrieving relevant entity embeddings from knowledge bases and updating contextual word representations via word-to-entity attention. These models implicitly encode ex-

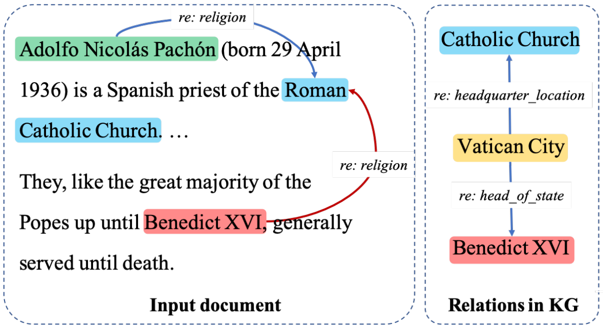

re: religion
Adolfo Nicolás Pachón (born 29 April 1936) is a Spanish priest of the Roman Catholic Church. …
re: religion
They, like the great majority of the Popes up until Benedict XVI, generally served until death.
Input document
Catholic Church
re: headquarter_location
Vatican City
re: head_of_state
Benedict XVI
Relations in KG

Figure 9.1: An example adapted from the DocRED development set, where the blue arrow refers to an intra-sentence, and the red arrow refers to an inter-sentence relation. From the input document, we know that `Benedict XVI`'s occupation is a pope, but it is difficult to directly extract the relation with the `Roman Catholic Church`. With the help of high-order relations retrieved from external KG, we can easily deduce that the religion of the popes should be the church they serve. Here, the `Vatican City` serves as a bridge entity and does not appear in the document.

ternal relational knowledge and achieve significant improvements on various downstream tasks. However, considering the rapid growth of existing knowledge bases, it would take a considerable cost to retrain the knowledge base embedding and finetuning the unsupervised representation learning model after the corresponding knowledge base is updated. In addition, different downstream tasks usually use different knowledge bases and switching tasks would also require retraining the knowledge base embedding and finetuning representation learning model.

Motivated by these difficulties, in this paper, we propose a relation extraction model, Knowledge-Enhanced Relation Extractor (KERE), which incorporates document-specific high-order entity graphs from the knowledge bases without needing to use pretrained entity embeddings. Instead of using knowledge embeddings pretrained on the whole knowledge bases, we believe that focusing on entities relevant to the document / sentence at hand would be sufficient to obtain background knowledge. Specifically, for each input documents, we recognize a set of *seed entities* and link them to the items in a knowledge base. Based on these seed entities, we construct a multi-hop entity graph containing high-order relational information and then prune it to a reasonable size. We then encode this entity graph by multi-layer graph convolutional networks to get knowledge aware entity representations. Lastly, we synthesis external structured representations and contextual representations for multi-class relation classification. We evaluate our model on a document-level relation extraction dataset DocRED and two sentence-level relation extraction datasets TACRED and CoNLL03. Our model leads to an improvement of ranging from 1.9% to 4.5% in F1 on these datasets.

9.2 Related Work

9.2.1 Sentence-level Relation Extraction

Sentence-level relation extraction is a widely studied task in the NLP community. Various existing methods mainly fall into two classes: dependency-based and sequence-based. For dependency-based models, Xu et al. [282] and Miwa and Bansal [169] introduce shortest dependency paths between relation mentions into tree-LSTM to capture dependency information. Zhang et al. [311] present a path-centric pruning technique to help dependency-based models maximally remove irrelevant information. Guo et al. [79] improve this by proposing a soft-pruning approach that automatically learns to select the most relevant sub-structures.

Despite the great success of the dependency-based model, some researchers also explored the feasibility of performing relation classification directly from input sequences without complicated pre-processing (such as dependency parsing, and pos tagging). Zeng et al. [300] and dos Santos et al. [59] encode the sentences with convolutional neural network (CNN) and word embeddings for relation extraction. Zhou et al. [315] and Zhang et al. [308] apply the attention mechanism over Recurrent Neural Networks (RNN) which enables much better relation extraction performance. Zheng et al. [313] transform the relation extraction as a sequential tagging problem (NovelTagging) and extract entities and their relations in an end-to-end style. Recently, Soares et al. [238] propose to pretrain task-agnostic relation representations from large-scale distant-supervised sentence pairs by matching the blanks, which significantly improves the performance on intra-sentence RE. They also explore variants of input schema and relation representation with deep Transformers network.

9.2.2 Document-level Relation Extraction

Document-level relation extraction was originally studied in the context of precision medicine and biomedical text analysis. Peng et al. [196] first build a distant-supervised dataset from biomedical literature for cross-sentence n-ary relation extraction. They employ Graph-LSTMs to extract *drug-gene-mutation* interactions within 3-consecutive sentences. Song et al. [240] further improve the Graph-LSTM framework by taking edge labels as part of the input to the gated network. Verga et al. [265] extend the sentence-level relation extraction into document-level. They use a Transformer Block to encode the documents and aggregated over mentions to form entity pair representations, which allows the model to predict relationships between all mention pairs in one pass. Sahu et al. [218] further replace Transformer with a GCNN model for full-abstract encoding using non-local dependencies such as entity co-reference.

However, all these work above applied on biomedical or biochemical datasets, making it unsuitable for developing general-purpose document-level relation extraction framework. In 2019, Yao et al. [290] build a large-scale human-annotated document-level relation extraction dataset from Wikipedia and Wikidata, named DocRED, which accelerates the research on inter-sentence document-level relation extraction.

9.2.3 Incorporation of External Knowledge

With the population of existing knowledge bases, large amounts of structured and relational knowledge become available, including manually annotated lexical database like WordNet [167] and editable multilingual knowledge bases like Wikidata[268] and DBPedia[5].

The early work focuses on directly integrate knowledge base embeddings into task-specific models, and optimize under task supervision. Yang and Mitchell [285] employ an attention mechanism with a sentinel to adaptively attend to the most relevant background knowledge and surpass previous methods on entity extraction and event extraction tasks. Chen et al. [36] incorporate lexical semantic relations from WordNet [167] into premise and hypothesis sentences, and further improves the state-of-the-art of natural language inference.

Recently, some pretrained language models like ERNIE [312] and KnowBERT [199] explore to enhance language representation with external knowledge. They propose to pretrain entity-aware language models by retrieving entity embeddings from KB and combine them with contextual representations. These knowledge enhanced language encoder achieved significant improvements on various knowledge-driven tasks (such as relationship extraction, and entity typing).

In this work, instead of using knowledge embedding, we proved that using the document-specific graph structures from knowledge base can also bring great improvements on relation extraction tasks.

9.3 Method

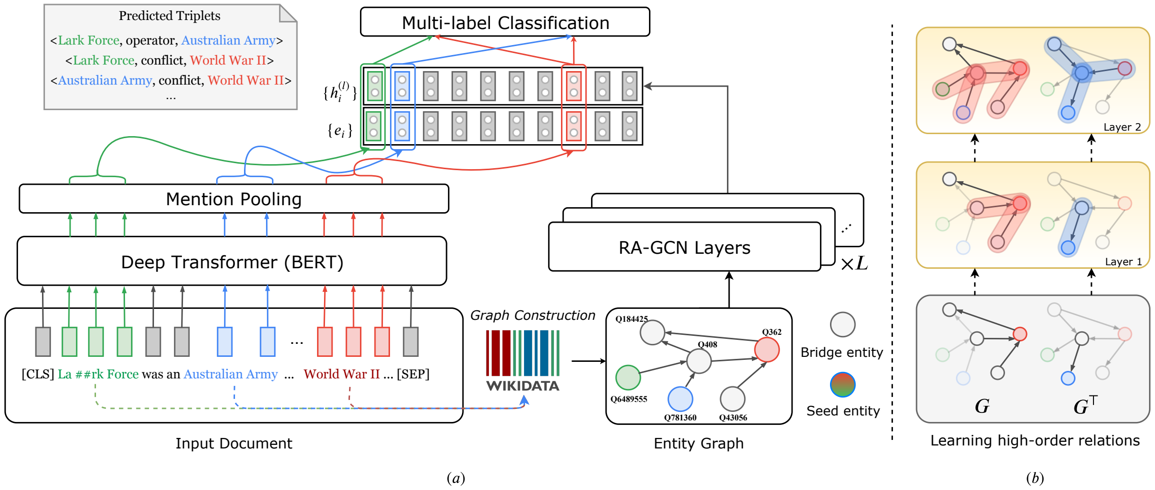


Figure 9.2: Overview of the KERE architecture illustrated with an example document and its simplified entity graph. (a) The model is composed of a sequence encoder and L identical RA-GCN layers. Every RA-GCN layer takes node embeddings and adjacency matrices that represent entity graph and its transpose. The contextual entity states $\{e_i\}$ are obtained by pooling over mention representations of BERT encoder. (b) Illustration of learning high-order relations with RA-GCN layers.

In this section, we first describe the entity graph construction algorithm over grounded knowledge bases, and then we introduce the architecture of our Knowledge-Enhanced Relation Extractor (KERE) for document-level relation extraction.

9.3.1 Constructing Entity Graphs for Input Documents

We first introduce the notations in this paper. Given a relational triplet $\langle h, r, t\rangle$ while $h, t \in \mathbf{E}$ stand for head entity and tail entity respectively, and $r \in \mathbf{R}$ stands for relation. $\mathbf{E}$ is the set of entities and $\mathbf{R}$ is the set of relations.

Algorithm 5 Entity Graph Construction

```
**Input:** Set of seed entities $E_S$; Set of all relation triplets in wikidata $KG = \{\langle h, r, t\rangle\}$
**Output:** Constructed graph $G$
Initialize the bridge entities with empty set $E_B \leftarrow \{\}$
For e **in** $E_S$
    Extract $G_e = \{\langle h, r, t\rangle \mid h = e\}$ from $KG$
    $E_t = \{t \mid \langle *, *, t\rangle \in G_e\}$
    $G = G \cup G_e, E_B = E_B \cup E_t$
For e **in** $E_B$
    Extract $G_e = \{\langle h, r, t\rangle \mid h = e\}$ from $KG$.
    For $\langle e, r, t\rangle$ **in** $G_e$
        If $t \in E_S \cup E_B$
            $G = \{\langle e, r, t\rangle\} \cup G$
For e **in** $E_B$
    If **outdegree**(e) $< 1$ or **indegree**(e) $\leq 1$
        Remove all triplets containing $e$ in $G$
Remove all *weak-links* in $G$.
```

For each input document D, a set of named entities was recognized for supervised relation extraction. However, limited by their surface form, we cannot obtain more properties of these entities mentions beyond their context. Knowledge bases (KBs) provide a rich source of high quality, human-curated knowledge that can be used to ground the entities. We investigate the utility of knowledge bases, which breaks down the independent interaction assumption in intra-sentence relation extraction and excavate high-order relations which is missing in sentence-level relation extraction.

Let E_S denotes the set of entities recognized in document D. We first link the entities in E_S to item ids in Wikidata and set them as the *seeds*. We denote the edge that links two seed entities as a *weak-link*. We use Breadth-First Search to extract relevant 2 or 3-hop high-order relations from a knowledge base. Here we choose the 97 wikidata relation types provided in [290] as the set of relations R. Notice that there could be hubs with a large number of links that greatly exceeds the average in the knowledge base. We believe that most of their neighboring entities are useless for the document-level relation extraction, and even obfuscate the core structure that contains the most important information. Therefore, we prune the graph by eliminating nodes with few connections, which significantly reduces the scale of the graphs and improves its quality.

Furthermore, since we aim to extract the relationships between all entity pairs in the document, however, some of these relationships may already exist in the knowledge base. To avoid introducing gold answers to the model input and mask these relations, we eliminate all the edges between seed entities, i.e. the *weak-links*. An overview of our approach is given in Algorithm 5.

9.3.2 Relation-aware Graph Convolution Network

Traditional graph convolutional networks (GCN) can be treated as a special case of differentiable message-passing framework[73]. It aggregates the incoming messages from neighboring nodes and effectively encodes the local structures in graphs with a homogeneous edge type. The original GCN was designed for undirected graphs. To consider both incoming and outgoing entity features and encode high-order relational structure, we proposed Relation-aware Graph Convolution Network (RA-GCN) to better exploit relational directed entity graphs extracted from knowledge bases.

Entity Embeddings Initialization

Denote $h_i^{(l)}$ as the hidden state of entity i in the l-th layer of the RA-GCN. At the first layer, we encode the initial representations of the entities through a pretrained BERT model, since it can provide high quality language features and can better align with the entity representations from input documents. Instead of just using the final output of BERT, we choose to integrate hidden states from multiple Transformer layers. Recent study [263] has shown that the representations from lower layers of BERT might be more applicable to certain language understanding tasks (e.g. named entity recognition, coreference resolution) than others, which means the layer depth should be chosen individually depending on the task at hand.

To this end, we initialize node embeddings by the weighted sum of multi-layer representations: $h_i^{(0)} = \sum_k \lambda_k T_i^{(k)}$, where $T_i^{(k)}$ is the hidden states of the i-th entity from the k-th layer, λ_k are trainable weight parameters. It enables our model to learn the best initial combination by itself.

Message Passing over Relational Entity Graphs

As described in Sec. 9.3.1, we construct a directed relational graph G for each input document. By flipping the direction of each edge, we can obtain the transpose graph $G^\top$. These document-specific graphs act as the backbone upon which the graph networks are constructed. Then the model aggregate the message of neighboring entities from "source to target" and "target to source" directions on G and $G^\top$. Here we also add self-connection to each entity, such that the old representation vector of the entity itself is taken into consideration when updating each representation:

$$\overrightarrow{h}_i^{(l+1)} = \frac{1}{c_i} \sum_{j \in \overrightarrow{\mathbb{N}}(i)} \overrightarrow{W}_e^{(l)} h_j^{(l)} \odot \overrightarrow{W}_r^{(l)} r_{ij} + h_i^{(l)} \tag{9.1}$$

$$\overleftarrow{h}_i^{(l+1)} = \frac{1}{c_i} \sum_{j \in \overleftarrow{\mathbb{N}}(i)} \overleftarrow{W}_e^{(l)} h_j^{(l)} \odot \overleftarrow{W}_r^{(l)} r_{ij} + h_i^{(l)} \tag{9.2}$$

, $\overrightarrow{\mathbb{N}}(i)$ and $\overleftarrow{\mathbb{N}}(i)$ denotes the neighbors of entity i in graph G and $G^\top$, and r_{ij} refers to the embedding of relation between entity i and j, which is initialized by the BERT representation of its surface form. The normalization constant c_i equals to the number of neighbors of entity i. Notice that here we choose to bind the representations of entity and relation together by element-wise product rather than concatenation and linear projections, since the latter one is equivalent to a mean pooling operation over all the projected relation and entity representations.

After information aggregation, the representations from both directions are concatenated and updated by passing through a linear layer with ReLu activation:

$$h_i^{(l)} = \mathrm{ReLU}(W^{(l)}[\overrightarrow{h}_i^{(l)}; \overleftarrow{h}_i^{(l)}] + b^{(l)}) \tag{9.3}$$

9.3.3 Integrating structured information with textual semantics

The input document is first tokenized into a sequence of word pieces $\{w_i\}_{i=1}^n$, and then encoded into hidden state sequence $\{h_i\}$ by a pretrained BERT-base model [57]. We encode the entity information by integrating the context information of the entire document to generate a context-aware representation. Inspired by [265], for each entity mention ranging from s-th and t-th word, we represent it as $m_k = \frac{1}{t-s+1}\sum_{i=s}^{t} h_i$. The model compute the entity representation by pooling over all word pieces in a mention span $e_i = \frac{1}{K}\sum_k m_k$.

Since multi-layer RA-GCN solely encoded the high-order relational information of each entity over existing knowledge base, we try to combine it with the context-aware representations through a linear layer with activation as equation 9.4. Then each entity pairs (i, j) together with their relative distance embedding are concatenated to compute the probability for each relation type.

$$\hat{e}_i = \mathrm{ReLu}(\mathbf{W}[e_i; h_i^{(l)}] + \mathbf{b}) \tag{9.4}$$

$$p(r|i,j) = \mathrm{sigmoid}(\mathbf{W}_r[\hat{e}_i; \hat{e}_j; d_{ij}] + b_r) \tag{9.5}$$

where d_{ij} is the distance embedding vector corresponding to the relative distance of the first mention of the two entities, $\mathbf{W}$, $\mathbf{b}$, $\mathbf{W}_r$ and b_r are trainable parameters.

Following [113] and [290], we formalize the document-level relation extraction as a multi-label classification task. We maximize the log-likelihood of the correct relation triplets in the training set $\mathcal{D}$:

$$\begin{aligned}\mathcal{L} = -\frac{1}{N}\sum_r \sum_{i \neq j} &[y \log p(r|i,j) \\ &- (1-y)\log(1 - p(r|i,j))]\end{aligned} \tag{9.6}$$

where $y \in \{0, 1\}$ is a binary label which indicates whether the triplet $\langle i, r, j\rangle$ is in the gold set, and N is the total number of possible triplets of the input documents.

9.4 Experiments

In this section, we present the experimental results of the proposed KERE model. We first describe implementation details, the datasets, and the baselines to compare. Then we show the quantitative results for an document-level relation extraction dataset with ablation studies. We also conduct experiments on two sentence-level relation extraction datasets and compared with several baseline models. Finally, we demonstrate the improved effect via a case study.

9.4.1 Implementation Detail

In our experiments, we use the cased version of BERT Tokenizer [57] to tokenize all the documents and entity mentions. All input is encoded into 768-dimensional vectors with a pretrained BERT base model. We tune the hyper-parameters according to results on the development sets. We use $d = 768$ as the feature size in all layers, and set the dropout rate to 0.5, the learning rate as $3e^{-5}$. We choose an Adam optimizer to optimize the KERE model. Our framework is implemented with PyTorch, and the RA-GCN layers are built with PyG[1]. For entity graph con-

[1] pytorch-geometric(PyG): `https://pytorch-geometric.readthedocs.io/en/latest/`

Model	Dev			Test		
	Ingore F1	F1	AUC	Ingore F1	F1	AUC
CNN†	37.99	43.45	39.41	36.44	42.33	38.98
LSTM†	44.41	50.66	49.48	43.60	50.12	49.31
Bi-LSTM†	45.12	50.95	50.27	44.73	51.06	50.43
Context-Aware†	44.84	51.10	50.20	43.93	50.64	49.70
BERT baseline‡	51.74	53.66	50.92	51.16	53.18	-
KERE	**56.14**	**57.69**	**60.83**	**55.70**	**57.28**	-
KERE + *weak-links*	*56.62*	*58.27*	*59.84*	*55.86*	*57.43*	-

Table 9.1: Performance comparison on the public dev set and private test set of DocRED. We submit our prediction and evaluated on its official competition site. † are baseline models implemented by [290]. ‡ The BERT baseline is implemented in this paper, by using BERT encoder followed by a multi-label classification layer.

struction, we use the is constructed from the English Wikidata dump [2] and link the entities with MediaWiki API.

During training, we also introduced a pretraining technique to alleviate the cold-start problem. In early experiments, we observed that if we directly optimize all the parameters in KERE from the very beginning, the model was hard to converge. Therefore, we first freeze the parameters in the BERT encoder and update the remaining components in the first 20 epochs, then unfreeze them and train all parameters together. This pretraining technique shows a 1.5% gain in the F1-score shown in Table 9.2.

9.4.2 Datasets

We use the DocRED [290] dataset to evaluate the performance on document-level relation extraction. Cases in DocRED requires the model to infer entity relations by synthesizing all information scattered among multiple sentences in the document. It provides $5,053$ human-annotated documents and $101,873$ distant-supervised documents for training. In this work, we only chose the human-annotated split for model training. Also, the authors proposed a new task to predict the supporting evidences for relation instances. Since our focus is on relation extraction, we have not designed modules specifically for this task. We also evaluate our model on TACRED [309] and CoNLL 2004 [213], which are two human-annotated sentence-level relation extraction datasets.

9.4.3 Experimental Results and Analysis

Main Results As is shown in Table 9.1, we report our experimental results on DocRED dataset. Our model, KERE, outperforms all the baseline models and achieved state-of-the-art F1 score. Specifically, compared with the best model (Context-Aware) reported in [290], our model obtained about 4% improvements in ignore F1 compared with BERT baseline. Also, without the

[2] We use the 2019-09-10 version dumps of wikidata.

weakly supervised links in our entity graph, KERE also achieves comparable performance, which validates the effectiveness of high-ordered external knowledge introduced by RA-GCN. We also implemented a BERT baseline to exclude the improvements induces by pretrained LM, and the KERE still outperforms it by 4% in F1. We consider that it is because our model is capable of incorporate external relevant knowledge and capture high-order relations. We will discuss this further in the case study section.

Setting	F1	Precision	Recall
Best Model	57.69	61.84	54.07
- 1 RA-GCN layer	57.37	61.92	53.44
- 2 RA-GCN layer	56.71	65.55	49.97
- 3 RA-GCN layer	53.66	55.07	52.32
- dist embedding	55.95	61.00	51.68
- Pretraining	55.21	58.25	52.47
- BERT encoder	27.52	58.28	18.02
w/ *weak-links*	58.27	61.67	55.23
w/ GCN layer	56.62	63.15	51.31

Table 9.2: Ablation study of document-level relation extraction in the development set of DocRED. We set the best model with 3 RA-GCN layers as default.

Ablation Study To evaluate the performance of different components in our model, we perform ablation study on model components, training procedure, and graph construction.

Table 9.2 illustrates the experiment result of ablation study. With fewer layers of RA-GCN, the F1 score dropped 3% to 4%, which highlighted the effectiveness of aggregating the information from multi-hop neighboring entities. The distance embedding and pretraining technique (described in 9.4.1) also contribute to a 2 percent performance gain. In addition, by replacing RA-GCN as vanilla GCN layers caused performance degrade, we thereby validate the importance of modeling relation types in the entity graph. By removing BERT encoder and use the entity graphs as the only model input, the performance drastically decreased, which means it is insufficient to infer relations with graph structure alone. It also indicates the decisive role of jointly encoding input documents and entity graphs.

The ablation results shows that all the components play an important role in our model.

Effect on Number of Evidence The DocRED dataset provides the supporting evidence for each relation triplet. i.e., the relevant sentences that are required to infer the relation. We also conduct experiments to evaluate the model recall over different number of evidences. The number equals to one means the triplet is an intra-sentence relation, and the number greater to one indicates an inter-sentence relation. We group the target relation triplets are by the number of evidence and evaluate the recall score on these subsets. As is shown in Figure 9.3, our model consistently improves the classification performance on both inter-sentence and intra-sentence relation triplets. We also noticed the performance degradation when the number of RA-GCN

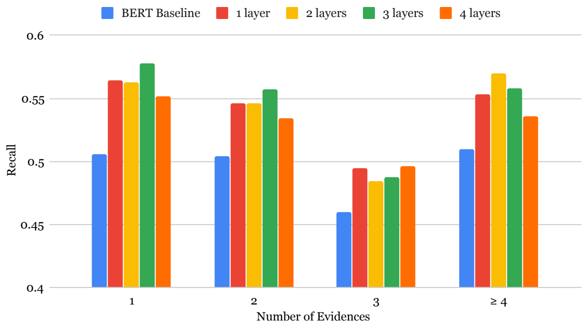


Figure 9.3: Recall on the triplets with different number of evidences on DocRED development set.

layers increase to 4, which is compliant with the graph construction algorithm because the entity graph we build contains at most 3-hop neighbors.

9.4.4 Results on Sentence-level RE

Since most of the current relation extraction datasets focus on sentence-level relation extraction, we also test our model on those datasets in this section. Table 9.3 shows the experiment results.

Model	TACRED	CoNLL04
GCN ([311])	64.0	-
AGGCN ([79])	65.1	-
TRE ([2])	67.4	-
ERNIE ([312])	68.0	-
MTB ([238])	**71.5**	-
BERT baseline	66.8	68.6
KERE (Ours)	68.7	**72.6**

Table 9.3: Performace on two intra-sentence relation extraction dataset: TACRED [308] and CoNLL04 [213]. We compare the F1-score on the test sets.

We compared with several published models. GCN [311] and AGGCN [79] are two dependency-based model encoded by graph networks. TRE [2], ERNIE [312] and MTB [238] are sequence-based model encode with pretrained LM, where ERNIE and MTB incorporate external knowledge embeddings.

Following the input schema of [238], we modified our model by using special entity tokens `[E1],[/E1],[E2],[/E2]` to bound the mention spans of the head and tail entities, then select the contextual word representations for `[E1]` and `[E2]` as entity representations e_1 and e_2. We also report the performance of BERT baseline implemented by ourselves for comparison.

The results in Table 9.3 shows that our model achieved comparable performance with the state-of-the-art models. Compared with the document-level setting, the improvement is not very significant, because most of these sentence-level relationships can be inferred by pattern matching from plain text and are not highly dependent on high-order relationships.

Sentences	Predictions	Supportive Links
[1] Although Orzabal had effectively made two solo albums under the Tears for Fears moniker in the 1990s (following the departure of bandmate Curt Smith) , this was the first recording to be released …	Retrieved by BERT: <Tears for Fears, has_part, Roland Orzabal> … Retrieved by our method: <Tears for Fears, has_part, Roland Orzabal> **<Tears for Fears, has_part, Curt Smith>** …	<Tears for Fears, record_label, Mercury Records> <Curt Smith, record_label, Mercury Records> head → bridge ← tail
[1] The Foix is a river in Catalonia, northeastern Spain. … [3] Then it goes through the Comarca of Garraf from north to the south where it ends into the Mediterranean Sea.	Retrieved by BERT <Catalonia, country, Spain> … Retrieved by our method: <Catalonia, country, Spain> **<Catalonia, located in or next to body of water, Mediterranean Sea>** …	<Iberian Peninsula, located in or next to body of water, Mediterranean Sea> <Catalonia, located on terrain feature, Iberian Peninsula> head → bridge → tail
[1] Adolfo Nicolás Pachón is a Spanish priest of the Roman Catholic Church. … [4] They, like the great majority of the Popes up until Benedict XVI, generally served until death.	Retrieved by BERT: <Adolfo Nicolás Pachón, religion, Roman Catholic Church> … Retrieved by our method: <Adolfo Nicolás Pachón, religion, Roman Catholic Church> **<Benedict XVI, religion, Roman Catholic Church>** …	<Vatican City, head_of_state, Benedict XVI> <Catholic Church, headquarters_location, Vatican City> head ← bridge → tail

Figure 9.4: Case study: the examples are adapted from DocRED development set. Bold triples are predictable by our method but missed by the BERT baseline. Supportive links were selected from the document-specific entity graphs, and we believe they can help identify the missing relationships.

9.4.5 Case Study

In this section, we select three cases from the development set of DocRED to show the benefit of incorporating high-order external knowledge. These cases correspond to three typical high-order relational patterns, as illustrated in Figure 9.4.

The first case shows an intra-sentence relation between a rock band named `Tear for Fears` and a songwriter `Curt Smith`. The `has_part` relation cannot be inferred by the two supportive links and was ignored by the BERT model. From the knowledge base, we know the rock band and the songwriter both signed with `Mercury Records`. So the songwriter was probably a member of this band. Together with the input document, our methods further validates this hypothesis and retrieved the relation triplet, <`Tear for Fears`, `has part`, `Curt Smith`>.

In the second case, it requires commonsense reasoning to infer the relation that `Catalonia` is located in or next to `the Mediterranean Sea` based on the evidence that the Foix river is in Catalonia, and it flows into the Mediterranean Sea. But for the BERT model, it cannot infer this relation from the input document alone due to the lack of reasoning capability. Also, We cannot rashly extract the target triplet from the knowledge base, because the `Iberian Peninsula` locates next to `the Mediterranean` does not mean `Catalonia` also located next to it. By integrating structural information from knowledge bases into plain text, our method successfully identified the target relation.

The last example was the one shown in the introduction. The original descriptions about the relation between `Benedict XVI` and `Catholic Church` is too vague to be identified by the baseline model. The supportive links tell us that the man is the head of Vatican City, which is also the "headquarter" of the Catholic Church. Combine with the fact that he is also a Pope. Our model successfully extracts the relation that the religion of `Benedict XVI` is the `Roman Catholic Church`.

9.5 Discussion

We present Knowledge-Enhanced Relation Extractor (KERE) for document-level relation extraction. Experimental results show that KERE achieves state-of-the-art results on both inter-sentence and intra-sentence relation extraction tasks. Unlike previous approaches, KERE dynamically construct document-specific entity graph from knowledge bases and operate bi-directionally on the entity graph to distill the high-order information into contextual representations.

However, it is not clear that whether such significant improvements can generalize to other tasks that do not heavily involve the understanding of entities and relations. In addition, to apply this algorithm, we need to change the underlying architecture to integrate information from knowledge bases, which introduce some engineering difficulties.

Chapter 10

Conclusion

10.1 Contribution

In this thesis, we present data-efficient algorithms that uses (1) unlabeled data (2) data from another domain and (3) external knowledge. Given a new task, whether an algorithm should be applied depends on the effectiveness, the applicability and the engineering difficulty of the algorithm. To provide practical suggestions to readers, , and discuss the engineering efforts required for each algorithm.

Effectiveness We evaluate different algorithms on different problems and have the following observations:

- **Semi-supervised learning:** For natural language processing tasks and computer vision tasks, Semi-supervised learning leads to significant improvements in both high-data regime and low-data regime. In low-data regime, on image classification tasks, UDA is very effective. It leads to an error rate of 5.43 on CIFAR-10 with only 250 labeled examples as presented in Section 2.5.2. On natural language processing tasks, UDA reduces the error rate from 43.27 to 25.23 for IMDb with 20 labeled examples and from 50.80 to 41.35 for Yelp-5 with 2,500 labeled examples as shown in Section 2.5.3. Noisy Student Training achieves 88.4 top-1 accuracy on ImageNet, 2.0 percent better than the state-of-the-art model that uses 3.5B weakly labeled Instagram images, as shown in Section 3.3. Noisy Student Training also leads to very significant improvements on robustness. It improves ImageNet-A top-1 accuracy from 61.0 to 83.7.
- **Transfer learning:** Transfer learning using a pretrained model leads to significant improvements on many NLP tasks. For example, on text classifications, as shown in Section 2.5.3, it lowers the error rate from 43.27 to 11.72 for IMDb with 20 labeled examples and from 50.80 to 38.90 for Yelp-5 with 2,500 labeled examples. Transfer learning from a language model pretrained on 1-Billion-Word corpus leads to an accuracy of 70.7, significantly outperforming a model trained on the CLOTH dataset achieving an accuracy of 48.7 as shown in Section 7.4.1. In comparison, transfer learning between similar tasks do not achieve such big improvements though the improvements are still respectable: as shown in Section 6.5.2, it improves BLEU score from 35.2 to 36.1 on French-to-English translation

and 27.3 to 28.1 on German-to-English translation. It also improves the Hits@10 from 79.7 to 81.4 for knowledge base completion in Section 5.3.3.

- **External knowledge:** The improvements brought by external knowledge vary for different tasks. External knowledge bases are useful for tasks that require modeling entities and relations. Specifically, using external knowledge bases improves the F-1 score from 53.18 to 57.28 on relation extraction as shown in Section 9.4.3. Using prior knowledge to guide the model design improves the BLEU score from 30.90 to 31.44 on image captioning and from 28.04 to 28.30 on German-to-English translation as shown in Section 8.7.3.
- **Complementariness:** Luckily, algorithms in different categories are usually complementary. Hence, it is possible to combine different algorithms for a better performance. For example, semi-supervised learning is complementary to transfer learning. Transfer learning from BERT achieves an error rate of 11.72 for IMDb with 20 examples and 38.90 for Yelp-5 with 2,500 examples. Semi-supervised learning further reduce the error rate from 11.72 to 4.78 for IMDb and from 38.90 to 33.54 for Yelp-5 as shown in Section 2.5.3. Using external knowledge bases also improve the the F-1 score of transfer learning using BERT from 53.18 to 57.28, as shown in Section 8.7.3.

Applicability We are also interested in whether a presented algorithm can be applied to a variety of tasks or is restricted to a certain task.

- **Semi-supervised learning:** Semi-supervised learning works well on many different tasks including text classification, image classification and machine comprehension. Semi-supervised learning methods UDA and Noisy Student Training are applied to 7 language datasets and 3 computer vision datasets in Section 2.5, Section 3.3 and Section 4.6. Moreover, only unlabeled data is required for semi-supervised learning and unlabeled data is usually easy to obtain.
- **Transfer learning:** Transfer learning from pretraining are applicable for many NLP tasks and have become the standard practice. It is used for 10 natural NLP datasets as shown in Section 2.5, Section 7.4.1 and Section 9.4. It does not require extra task-specific data that is similar to the task at hand and hence it can be easily used for any tasks. On the other hand, transfer learning from similar tasks is harder since it uses task-specific data from similar tasks.
- **External knowledge:** Whether external knowledge is helpful for a task is not straightforward since different tasks require different external knowledge. For example, tasks of the medical domain and legal domain relies on very different knowledge bases. In addition, it requires a deep understanding to discover prior knowledge useful for a certain task.

Engineering Difficulty Lastly, we discuss the engineering difficulty so that readers know how much efforts are required to apply an algorithm.

- **Semi-supervised learning:** Noisy Student Training is very easy to implement and since only the loss function and the data loader need to be changed.
- **Transfer learning:** Transfer learning from pretraining has become the standard practice due to its effectiveness and easy-to-use nature. In comparison, transfer learning from sim-

ilar tasks requires more efforts since significant changes to the architecture or learning algorithm are needed to achieve a great performance.

- **External knowledge:** Using external knowledge also requires more efforts for the model to work well.

In summary, we have the following recommendations for a new task: (1) Whenever unlabeled data is available, semi-supervised learning should be used by default because it requires little engineering leads while bringing large gains to both NLP tasks and computer vision tasks; (2) Transfer learning from pretraining models should be used by default for NLP tasks since it brings significant improvements to many tasks and is easy to use; (3) Whether external knowledge should be used can be determined case-by-case since each task may require different external knowledge and it is significantly more costly in engineering efforts than using transfer learning or semi-supervised learning.

10.2 Future Directions

In the thesis, we have taken several steps in exploring methods for data-efficient machine learning. We think that there are several interesting directions worth exploring in the future:

Algorithms: Further Advancing Data-Efficient Learning In this thesis, we show that semi-supervised learning leads to consistent improvements in high-data regime and low-data regime. We think that there are many problems worth investigations in the future. Advanced data augmentation for supervised learning is an essential component for semi-supervised learning to work well since it produces valid and diverse noised examples. As the noising function is an essential component in semi-supervised learning, a question that naturally arises is: what are the best noising functions and how should we find them? In this thesis, we choose three noising functions for different tasks at hand. Is it possible to automatically learn a noising function given a new task? If it is possible to do so, does the qualities of the noising functions depend on the amount of labeled data and unlabeled data? Additionally, what unlabeled data is required for semi-supervised learning to work well? We have shown that out-of-domain unlabeled data can serve well in the case of image recognition. Going further, is unlabeled data generated by a model helpful? For example, could we use images generated by Generative Adversarial Networks as unlabeled data? Lastly, is it helpful to use an explicitly constructed graph for unlabeled data instead of the implicit graph constructed by data augmentation?

Transfer learning from pretraining works well for NLP. It would also be interesting to explore pretraining for computer vision tasks. Recent studies [37, 89] show that using advanced augmentations is essential for pretraining on computer vision. One interesting question is: What are the characteristics of effective augmentations for pretraining? Are best augmentations for supervised learning and semi-supervised learning also the best augmentations for pretraining? In addition, would augmentation also be effective for pretraining for NLP? Given the pretrained models, what knowledge bases is the most helpful for NLP tasks? Is there a knowledge base that is universally helpful for all tasks?

Theories: Bridging the Gap Between Applications and Theories A lot of deep learning research is based on intuitions and empirical results. However, it is also very important to rigorously prove why an algorithm works or not. However, at the current moment, there are fundamental difficulties in characterizing the capacity, the generalization power and the optimization of deep learning models. Without solving these difficulties, analyzing a specific data-efficient algorithm would also be less effective. For example, many theoretical analysis makes simplistic assumptions that deep models have infinite capacity. However, in Noisy Student Training, we find that a large model size plays an important role in the final performance. Hence, we might ask ourselves how to quantify the correlation between the number of model parameters, the capacity of the model and the final performance. In addition, it would be interesting to theoretically characterize data augmentation methods. Specifically, what characterizes good data augmentations in the high-dimensional space of text and image? What is the sweet spot in the tradeoff between validity and diversity? And is the sweet spot the same for all tasks with different amounts of data?

Applications: Applying Data-Efficient Algorithms on More Tasks In this thesis, we studied various applications including text classification, image classification, reading comprehension, machine translation and knowledge base completion. There are still many tasks that are worth investigating such as speech recognition, chatbot, object detection and segmentation. Many algorithms we presented may generalize and be easily applied to these new tasks. For example, it has been that noisy student training can lead to significant improvements on speech recognition and object detection and segmentation in recent works [191, 323].

In addition, there are more interesting applications involving an interactive environment, e.g., self-driving and stock prediction. Semi-supervised learning is particularly useful here since collecting labeled data for these tasks is harder than labeling images or text as it requires interactions with the environment over a long period of time. Semi-supervised learning provides the ability to explore an exponential space resulted from the interactive sequential environment, which is impossible for labeled data to cover. For the task of self-driving, apart from using generic model noise and data augmentation noise, one can inject noise that have physical meanings, e.g., fogging, blurring and snowing to make the model robust to these conditions. With snowing as a noise, one can quickly adapt a self-driving model trained in California to be used in Alaska.

Similarly, we can use semi-supervised learning for the problem of stock prediction. For high-frequency trading, one can generate features needed for stock price prediction to explore the exponential feature space. For medium to low frequency trading that relies more on raw text data, one can train a model to generate financial statements and social media trends as the additional text data.

It is worth noting that there might be a risk that a system can compound its mistake in a sequence of actions in an interactive environment. To minimize the effect of this risk, one needs to have enough labeled data so that the model has a high accuracy. When we have enough labeled data, the improvements of semi-supervised model on top of the supervised model might not be huge, but any improvements in self-driving and stock price prediction result in a large number of reduced accidents and much more money earned given the scale of those tasks. Moreover, the improvements usually comes from better robustness and solving hard cases, which leads to more trustworthy models.

Transfer learning from pretraining also lead to consistent improvements for many NLP tasks. Hence, it may require little effort to achieve significant improvements on these tasks. In addition to directly applying the algorithms, it is more interesting to investigate what inductive biases can lead to further gains on a specific task and whether these inductive biases can generalize to other tasks. For example, better augmentation methods might need to be invented to make semi-supervised learning even more effective on NLP tasks, since data augmentation is shown to be effective in computer vision while data augmentation is less well-studied in NLP.

Bibliography

[1] Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*, 2020.

[2] Christoph Alt, Marc Hübner, and Leonhard Hennig. Improving relation extraction by pre-trained language representations. *arXiv preprint arXiv:1906.03088*, 2019.

[3] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. *arXiv preprint arXiv:1908.02983*, 2019.

[4] Ben Athiwaratkun, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. There are many consistent explanations of unlabeled data: Why you should average. 2018.

[5] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.

[6] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *Advances in Neural Information Processing Systems*, pages 2654–2662, 2014.

[7] Yauhen Babakhin, Artsiom Sanakoyeu, and Hirotoshi Kitamura. Semi-supervised segmentation of salt bodies in seismic images using an ensemble of convolutional neural networks. *arXiv preprint arXiv:1904.04445*, 2019.

[8] Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. In *Advances in Neural Information Processing Systems*, pages 3365–3373, 2014.

[9] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 90, 92, 94, 104, 115, 127, 132

[10] Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. An actor-critic algorithm for sequence prediction. *arXiv preprint arXiv:1607.07086*, 2016. 116, 118, 126, 127, 131, 133, 134, 136

[11] Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In *Machine Learning Proceedings 1995*, pages 30–37. Elsevier, 1995. 124

[12] Ondrej Bajgar, Rudolf Kadlec, and Jan Kleindienst. Embracing data abundance: Booktest dataset for reading comprehension. *arXiv preprint arXiv:1610.00956*, 2016. 58

[13] Anoop Korattikara Balan, Vivek Rathod, Kevin P Murphy, and Max Welling. Bayesian dark knowledge. In *Advances in Neural Information Processing Systems*, pages 3438–3446, 2015.

[14] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 1171–1179, 2015. 127

[15] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013. 85

[16] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. 71

[17] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a cpu and gpu math expression compiler. In *Proceedings of the Python for scientific computing conference (SciPy)*, volume 4, page 3. Austin, TX, 2010. 77

[18] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*, 2019.

[19] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. *arXiv preprint arXiv:1905.02249*, 2019.

[20] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, 2019.

[21] Akash Bharadwaj, David Mortensen, Chris Dyer, and Jaime Carbonell. Phonologically aware neural model for named entity recognition in low resource transfer settings. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1462–1472. Association for Computational Linguistics, 2016.

[22] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM, 1998.

[23] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pages 1247–1250, 2008. 71

[24] Danushka Bollegala, Takanori Maehara, and Ken-ichi Kawarabayashi. Embedding semantic relations into word representations. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[25] Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. Learning Structured Embeddings of Knowledge Bases. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, pages 301–306, 2011. 71, 72, 78

[26] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating Embeddings for Modeling Multi-relational Data. In *Advances in Neural Information Processing Systems 26*, pages 2787–2795. 2013. 71, 72, 76, 77, 78

[27] Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. A Semantic Matching Energy Function for Learning with Multi-relational Data. *Machine Learning*, 94(2):233–259, 2014. 71, 72, 78

[28] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In *NIPS*, 2016.

[29] Cristian BuciluǢŐ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541. ACM, 2006.

[30] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, Percy Liang, and John C Duchi. Unlabeled data improves adversarial robustness. *arXiv preprint arXiv:1905.13736*, 2019.

[31] Mauro Cettolo, Christian Girardi, and Marcello Federico. Wit3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, volume 261, page 268, 2012. 92

[32] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 4960–4964. IEEE, 2016. 115, 127

[33] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20 (3):542–542, 2009.

[34] Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. One billion word benchmark for measuring progress in statistical lming. *arXiv preprint arXiv:1312.3005*, 2013. 104

[35] Danqi Chen, Jason Bolton, and Christopher D Manning. A thorough examination of the cnn/daily mail reading comprehension task. *arXiv preprint arXiv:1606.02858*, 2016. xxi, 55, 56, 58, 60, 64, 65, 98, 100, 104

[36] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Diana Inkpen, and Si Wei. Neural natural language inference models enhanced with external knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2406–2417, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1224. URL `https://www.aclweb.org/anthology/P18-1224`.

[37] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*,

2020.

[38] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*, 2016.

[39] Xilun Chen, Yu Sun, Ben Athiwaratkun, Claire Cardie, and Kilian Weinberger. Adversarial deep averaging networks for cross-lingual sentiment classification. *arXiv preprint arXiv:1606.01614*, 2016. 4, 91, 96

[40] Yanbei Chen, Xiatian Zhu, and Shaogang Gong. Semi-supervised deep learning with memory. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 268–283, 2018.

[41] Yong Cheng, Wei Xu, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. Semi-supervised learning for neural machine translation. *arXiv preprint arXiv:1606.04596*, 2016.

[42] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[43] Kevin Clark, Minh-Thang Luong, Christopher D Manning, and Quoc V Le. Semi-supervised sequence modeling with cross-view training. *arXiv preprint arXiv:1809.08370*, 2018.

[44] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018. 99

[45] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

[46] Rui Correia, Jorge Baptista, Nuno Mamede, Isabel Trancoso, and Maxine Eskenazi. Automatic generation of cloze question distractors. In *Proceedings of the Interspeech 2010 Satellite Workshop on Second Language Studies: Acquisition, Learning, Education and Technology, Waseda University, Tokyo, Japan*, 2010. 110

[47] Rui Correia, Jorge Baptista, Maxine Eskenazi, and Nuno J Mamede. Automatic generation of cloze question stems. In *PROPOR*, pages 168–178. Springer, 2012. 110

[48] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

[49] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical data augmentation with no separate search. *arXiv preprint arXiv:1909.13719*, 2019.

[50] Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. In *Advances in neural information processing systems*, pages 3079–3087, 2015.

[51] Zihang Dai, Lei Li, and Wei Xu. Cfo: Conditional focused neural question answering with large-scale knowledge bases. In *Proceedings of the 54th Annual Meeting of the As-*

sociation for Computational Linguistics (Volume 1: Long Papers), pages 800–810, Berlin, Germany, August 2016. Association for Computational Linguistics. 71

[52] Zihang Dai, Zhilin Yang, Fan Yang, William W Cohen, and Ruslan R Salakhutdinov. Good semi-supervised learning that requires a bad gan. In *Advances in Neural Information Processing Systems*, pages 6510–6520, 2017.

[53] Zihang Dai, Guokun Lai, Yiming Yang, and Quoc V Le. Funnel-transformer: Filtering out sequential redundancy for efficient language processing. *arXiv preprint arXiv:2006.03236*, 2020.

[54] Hal Daumé III and Daniel Marcu. Learning as search optimization: Approximate large margin methods for structured prediction. In *Proceedings of the 22nd international conference on Machine learning*, pages 169–176. ACM, 2005. 116

[55] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[56] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[57] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019. URL `https://www.aclweb.org/anthology/N19-1423/`.

[58] Bhuwan Dhingra, Hanxiao Liu, Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. Gated-attention readers for text comprehension. *arXiv preprint arXiv:1606.01549*, 2016. 55, 58, 64, 98, 105

[59] Cícero dos Santos, Bing Xiang, and Bowen Zhou. Classifying relations by ranking with convolutional neural networks. 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)*, pages 626–634, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1061. URL `https://www.aclweb.org/anthology/P15-1061`.

[60] Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*, 2018.

[61] Manaal Faruqui, Yulia Tsvetkov, Dani Yogatama, Chris Dyer, and Noah A. Smith. Sparse overcomplete word vector representations. 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)*, pages 1491–1500, Beijing, China, July 2015. Association for Computational Linguistics. 82

[62] Christiane D. Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, 1998. 71

[63] Sandra S Fotos. The cloze test as an integrative measure of efl proficiency: A substitute for essays on college entrance examinations? *Language Learning*, 41(3):313–336, 1991. 97

[64] Andrew Frank, Arthur Asuncion, et al. Uci machine learning repository, 2010. URL `http://archive.ics.uci.edu/ml`. 91

[65] Tommaso Furlanello, Zachary C Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. *arXiv preprint arXiv:1805.04770*, 2018.

[66] Angus Galloway, Anna Golubeva, Thomas Tanay, Medhat Moussa, and Graham W Taylor. Batch normalization is a cause of adversarial vulnerability. *arXiv preprint arXiv:1905.02161*, 2019.

[67] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. *ICML*, 2015. 3, 4, 86, 91, 96

[68] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016. 3, 4, 86, 96

[69] Alberto García-Durán, Antoine Bordes, and Nicolas Usunier. Composing Relationships with Translations. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 286–290, 2015. 72, 77, 78

[70] Alberto García-Durán, Antoine Bordes, Nicolas Usunier, and Yves Grandvalet. Combining Two and Three-Way Embedding Models for Link Prediction in Knowledge Bases. *Journal of Artificial Intelligence Research*, 55:715–742, 2016. 77, 78

[71] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019.

[72] Athinodoros S. Georghiades, Peter N. Belhumeur, and David J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE transactions on pattern analysis and machine intelligence*, 23(6):643–660, 2001. 92

[73] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1263–1272. JMLR. org, 2017.

[74] Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian Goodfellow. Adversarial spheres. *arXiv preprint arXiv:1801.02774*, 2018.

[75] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 86, 91

[76] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing

adversarial examples. In *International Conference on Learning Representations*, 2015.

[77] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pages 529–536, 2005.

[78] Keren Gu, Brandon Yang, Jiquan Ngiam, Quoc Le, and Jonathan Shlens. Using videos to evaluate image model robustness. In *ICLR Workshop*, 2019.

[79] Zhijiang Guo, Yan Zhang, and Wei Lu. Attention guided graph convolutional networks for relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 241–251, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1024. URL `https://www.aclweb.org/anthology/P19-1024`.

[80] Kelvin Guu, John Miller, and Percy Liang. Traversing Knowledge Graphs in Vector Space. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 318–327, 2015. 71, 72

[81] G Gybenko. Approximation by superposition of sigmoidal functions. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989. 85

[82] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. *arXiv preprint arXiv:1702.08165*, 2017. 127

[83] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018. 127

[84] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

[85] Ryuichiro Hataya and Hideki Nakayama. Unifying semi-supervised and robust learning by mixup. *ICLR The 2nd Learning from Limited Labeled Data (LLD) Workshop*, 2019.

[86] Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. Dual learning for machine translation. In *Advances in Neural Information Processing Systems*, pages 820–828, 2016.

[87] Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. Revisiting self-training for neural sequence generation. *arXiv preprint arXiv:1909.13788*, 2019.

[88] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 22, 31, 36, 38, 50, 133

[89] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[90] Shizhu He, Kang Liu, Guoliang Ji, and Jun Zhao. Learning to Represent Knowledge Graphs with Gaussian Embedding. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 623–632, 2015. 77, 78

[91] Xuanli He, Gholamreza Haffari, and Mohammad Norouzi. Sequence to sequence mixture model for diverse machine translation. *arXiv preprint arXiv:1810.07391*, 2018.

[92] Olivier J Hénaff, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019.

[93] Dan Hendrycks and Thomas G Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.

[94] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *arXiv preprint arXiv:1907.07174*, 2019.

[95] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *NIPS*, 2015. 55, 58, 97, 99, 107

[96] Alex Hernández-García and Peter König. Data augmentation instead of explicit regularization. *arXiv preprint arXiv:1806.03852*, 2018.

[97] Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. The goldilocks principle: Reading children's books with explicit memory representations. *ICLR*, 2016. 55, 58, 65, 97, 99, 107

[98] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[99] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 90, 104, 132

[100] Andrew G Howard. Some improvements on deep convolutional neural network based image classification. *arXiv preprint arXiv:1312.5402*, 2013.

[101] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 328–339, 2018.

[102] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[103] Weihua Hu, Takeru Miyato, Seiya Tokui, Eiichi Matsumoto, and Masashi Sugiyama. Learning discrete representations via information maximizing self-augmented training. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1558–1567. JMLR. org, 2017.

[104] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European conference on computer vision*, pages 646–661. Springer, 2016.

[105] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[106] Po-Sen Huang, Chong Wang, Dengyong Zhou, and Li Deng. Toward neural phrasebased machine translation. 2017. 136

[107] Yanping Huang, Yonglong Cheng, Dehao Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V Le, and Zhifeng Chen. GPipe: Efficient training of giant neural networks using pipeline parallelism. In *Advances in Neural Information Processing Systems*, 2019.

[108] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 92

[109] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5070–5079, 2019.

[110] Jacob Jackson and John Schulman. Semi-supervised learning by label gradient alignment. *arXiv preprint arXiv:1902.02336*, 2019.

[111] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. Knowledge Graph Embedding via Dynamic Mapping Matrix. 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)*, pages 687–696, 2015. 71, 77, 78

[112] Heng Ji and Ralph Grishman. Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pages 1148–1158. Association for Computational Linguistics, 2011.

[113] Xiaotian Jiang, Quan Wang, Peng Li, and Bin Wang. Relation extraction with multi-instance multi-label convolutional neural networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1471–1480, 2016.

[114] Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. Google's multilingual neural machine translation system: Enabling zero-shot translation. *arXiv preprint arXiv:1611.04558*, 2016. 92, 94

[115] Rie Johnson and Tong Zhang. Deep pyramid convolutional neural networks for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 562–570, 2017.

[116] Jon Jonz. Cloze item types and second language comprehension. *Language testing*, 8(1): 1–22, 1991. 97

[117] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *ACL*, 2017. 58, 99

[118] Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. Exploring the limits of lming. *arXiv preprint arXiv:1602.02410*, 2016. 104

[119] Rudolf Kadlec, Martin Schmid, Ondrej Bajgar, and Jan Kleindienst. Text understanding with the attention sum reader network. *arXiv preprint arXiv:1603.01547*, 2016. 55

[120] Giannis Karamanolakis, Daniel Hsu, and Luis Gravano. Leveraging just a few keywords

for fine-grained aspect detection through weakly supervised co-training. *Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

[121] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015. 115, 127, 132

[122] Daniel Khashabi, Tushar Khot, Ashish Sabharwal, Peter Clark, Oren Etzioni, and Dan Roth. Question answering via integer programming over semi-structured knowledge. *arXiv preprint arXiv:1604.06076*, 2016. 56, 59

[123] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 91, 104

[124] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *ICLR*, 2014.

[125] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pages 3581–3589, 2014.

[126] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[127] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush. OpenNMT: Open-Source Toolkit for Neural Machine Translation. *ArXiv e-prints*, 2017. 92

[128] Wouter Kool, Herke van Hoof, and Max Welling. Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement. *arXiv preprint arXiv:1903.06059*, 2019.

[129] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2661–2671, 2019.

[130] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[131] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[132] Andrey Kurtasov. A system for generating cloze test items from russian-language text. In *Proceedings of the Student Research Workshop associated with RANLP 2013*, pages 107–112, 2013. 110

[133] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations. *EMNLP*, 2017. 99, 105

[134] Guokun Lai, Barlas Oguz, and Veselin Stoyanov. Bridging the domain gap in cross-lingual document classification. *arXiv preprint arXiv:1909.07009*, 2019.

[135] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

[136] Alex M Lamb, Anirudh Goyal ALIAS PARTH GOYAL, Ying Zhang, Saizheng Zhang,

Aaron C Courville, and Yoshua Bengio. Professor forcing: A new algorithm for training recurrent networks. In *Advances In Neural Information Processing Systems*, pages 4601–4609, 2016. 127

[137] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 85

[138] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, volume 3, page 2, 2013.

[139] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web*, 6(2):167–195, 2015. 71

[140] Jiwei Li, Will Monroe, and Dan Jurafsky. Learning to decode for future success. *arXiv preprint arXiv:1701.06549*, 2017. 134, 136

[141] Xiang Li, Tao Qin, Jian Yang, and Tieyan Liu. LightRNN: Memory and Computation-Efficient Recurrent Neural Networks. In *Advances in Neural Information Processing Systems 29*. 2016.

[142] Yingting Li, Lu Liu, and Robby T Tan. Certainty-driven consistency loss for semi-supervised learning. *arXiv preprint arXiv:1901.05657*, 2019.

[143] Yujia Li, Kevin Swersky, and Richard Zemel. Learning unbiased features. *arXiv preprint arXiv:1412.5244*, 2014. 92, 95

[144] Davis Liang, Zhiheng Huang, and Zachary C Lipton. Learning noise-invariant representations for robust speech recognition. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 56–63. IEEE, 2018.

[145] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015. 126

[146] Chin-Yew Lin and Eduard Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 71–78. Association for Computational Linguistics, 2003. 58

[147] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 132

[148] Yankai Lin, Zhiyuan Liu, Huanbo Luan, Maosong Sun, Siwei Rao, and Song Liu. Modeling Relation Paths for Representation Learning of Knowledge Bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 705–714, 2015. 72, 77, 78

[149] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In *Proceedings of the Twenty-*

Ninth AAAI Conference on Artificial Intelligence Learning, pages 2181–2187. 2015. 71, 73, 76, 77, 78, 83

[150] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 19–34, 2018.

[151] Raphael Gontijo Lopes, Dong Yin, Ben Poole, Justin Gilmer, and Ekin D Cubuk. Improving robustness without sacrificing accuracy with patch gaussian augmentation. *arXiv preprint arXiv:1906.02611*, 2019.

[152] Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard Zemel. The variational fair autoencoder. *ICLR*, 2016. 86, 91, 92, 95, 96

[153] Gilles Louppe, Michael Kagan, and Kyle Cranmer. Learning to pivot with adversarial networks. *arXiv preprint arXiv:1611.01046*, 2016.

[154] Yucen Luo, Jun Zhu, Mengxi Li, Yong Ren, and Bo Zhang. Smooth neighbors on teacher graphs for semi-supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8896–8905, 2018.

[155] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015. 132

[156] Xuezhe Ma, Pengcheng Yin, Jingzhou Liu, Graham Neubig, and Eduard Hovy. Softmax q-distribution estimation for structured prediction: A theoretical interpretation for raml. *arXiv preprint arXiv:1705.07136*, 2017. 117, 127, 128, 134, 136

[157] Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. Auxiliary deep generative models. *arXiv preprint arXiv:1602.05473*, 2016.

[158] Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pages 142–150. Association for Computational Linguistics, 2011.

[159] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 2008. 95

[160] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *International Conference on Learning Representations*, 2018.

[161] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 181–196, 2018.

[162] Alireza Makhzani and Brendan Frey. K-sparse autoencoders. In *Proceedings of the International Conference on Learning Representations*, 2014.

[163] André FT Martins and Ramón Fernandez Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *Proceedings of the 33th International*

Conference on Machine Learning, 2016.

[164] Cettolo Mauro, Girardi Christian, and Federico Marcello. Wit3: Web inventory of transcribed and translated talks. In *Conference of European Association for Machine Translation*, pages 261–268, 2012. 132

[165] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 43–52. ACM, 2015.

[166] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013. 4, 29, 72

[167] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[168] Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore, August 2009. Association for Computational Linguistics. 71

[169] Makoto Miwa and Mohit Bansal. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1105. URL `https://www.aclweb.org/anthology/P16-1105`.

[170] Takeru Miyato, Andrew M Dai, and Ian Goodfellow. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*, 2016.

[171] Takeru Miyato, Shin-ichi Maeda, Shin Ishii, and Masanori Koyama. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[172] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015. 124

[173] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1928–1937, 2016. 127, 135

[174] Ofir Nachum, Mohammad Norouzi, Kelvin Xu, and Dale Schuurmans. Bridging the gap between value and policy based reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2772–2782, 2017. 127

[175] Amir Najafi, Shin-ichi Maeda, Masanori Koyama, and Takeru Miyato. Robustness to

adversarial perturbations in learning from incomplete data. In *Advances in Neural Information Processing Systems*, 2019.

[176] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[177] Jiquan Ngiam, Daiyi Peng, Vijay Vasudevan, Simon Kornblith, Quoc V Le, and Ruoming Pang. Domain adaptive transfer learning with specialist models. *arXiv preprint arXiv:1811.07056*, 2018.

[178] Dat Quoc Nguyen, Kairit Sirts, Lizhen Qu, and Mark Johnson. STransE: a novel embedding model of entities and relationships in knowledge bases. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 460–466, 2016. xxi, 71, 72, 73, 76, 77, 78

[179] Dat Quoc Nguyen, Kairit Sirts, Lizhen Qu, and Mark Johnson. Neighborhood mixture model for knowledge base completion. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, page 40âĂŞ50. Association for Computational Linguistics, 2016.

[180] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*, 2016. 58, 99

[181] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A Three-Way Model for Collective Learning on Multi-Relational Data. In *Proceedings of the 28th International Conference on Machine Learning*, pages 809–816, 2011. 71

[182] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A Review of Relational Machine Learning for Knowledge Graphs. *Proceedings of the IEEE, to appear*, 2015. 71

[183] Mohammad Norouzi, Samy Bengio, Navdeep Jaitly, Mike Schuster, Yonghui Wu, Dale Schuurmans, et al. Reward augmented maximum likelihood for neural structured prediction. In *Advances In Neural Information Processing Systems*, pages 1723–1731, 2016. 5, 115, 116, 117, 123, 127, 128, 133

[184] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems*, pages 3235–3246, 2018.

[185] Takeshi Onishi, Hai Wang, Mohit Bansal, Kevin Gimpel, and David McAllester. Who did what: A large-scale person-centered cloze dataset. *arXiv preprint arXiv:1608.05457*, 2016. 55, 58, 97, 99, 107

[186] A Emin Orhan. Robustness properties of facebook's resnext wsl models. *arXiv preprint arXiv:1907.07640*, 2019.

[187] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[188] Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella

Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The lambada dataset: Word prediction requiring a broad discourse context. *arXiv preprint arXiv:1606.06031*, 2016. 98, 99

[189] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002. 58, 92, 132

[190] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*, 2019.

[191] Daniel S Park, Yu Zhang, Ye Jia, Wei Han, Chung-Cheng Chiu, Bo Li, Yonghui Wu, and Quoc V Le. Improved noisy student training for automatic speech recognition. *arXiv preprint arXiv:2005.09629*, 2020.

[192] Sungrae Park, JunKeon Park, Su-Jin Shin, and Il-Chul Moon. Adversarial dropout for supervised and semi-supervised learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[193] Sree Hari Krishnan Parthasarathi and Nikko Strom. Lessons from building acoustic models with a million hours of speech. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6670–6674. IEEE, 2019.

[194] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017. 104

[195] Anselmo Peñas, Yusuke Miyao, Álvaro Rodrigo, Eduard H Hovy, and Noriko Kando. Overview of clef qa entrance exams task 2014. In *CLEF (Working Notes)*, pages 1194–1200, 2014. 56, 59, 99

[196] Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. Cross-sentence n-ary relation extraction with graph lstms. *TACL*, 5:101–115, 2017. URL `https://transacl.org/ojs/index.php/tacl/article/view/1028`.

[197] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 4, 29, 65, 104

[198] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018. 3, 4, 29, 105

[199] Matthew E Peters, Mark Neumann, IV Logan, L Robert, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. Knowledge enhanced contextual word representations. *arXiv preprint arXiv:1909.04164*, 2019.

[200] Siyuan Qiao, Wei Shen, Zhishuai Zhang, Bo Wang, and Alan Yuille. Deep co-training for semi-supervised image recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 135–152, 2018.

[201] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf*, 2018.

[202] Ilija Radosavovic, Piotr Dollár, Ross Girshick, Georgia Gkioxari, and Kaiming He. Data distillation: Towards omni-supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4119–4128, 2018.

[203] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016. 55, 58, 99, 105

[204] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015. 116, 117, 127, 132, 134, 136

[205] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. In *Advances in neural information processing systems*, pages 3546–3554, 2015.

[206] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized evolution for image classifier architecture search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4780–4789, 2019.

[207] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? *International Conference on Machine Learning*, 2019.

[208] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. *arXiv preprint arXiv:1612.00563*, 2016. 127

[209] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *ICML*, 2014.

[210] Matthew Richardson, Christopher JC Burges, and Erin Renshaw. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *EMNLP*, volume 3, page 4, 2013. 56, 58, 63

[211] Ellen Riloff and Janyce Wiebe. Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 105–112, 2003.

[212] Álvaro Rodrigo, Anselmo Peñas, Yusuke Miyao, Eduard H Hovy, and Noriko Kando. Overview of clef qa entrance exams task 2015. In *CLEF (Working Notes)*, 2015. 56, 59, 99

[213] Dan Roth and Wen-tau Yih. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, pages 1–8, 2004.

[214] Aruni Roy Chowdhury, Prithvijit Chakrabarty, Ashish Singh, SouYoung Jin, Huaizu Jiang, Liangliang Cao, and Erik G. Learned-Miller. Automatic adaptation of object detectors to

new domains using self-training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[215] Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*, 2015. 115, 127

[216] Devendra Singh Sachan, Manzil Zaheer, and Ruslan Salakhutdinov. Revisiting lstm networks for semi-supervised text classification via mixed objective function. 2018.

[217] J Sachs, P Tung, and RYH Lam. How to construct a cloze test: Lessons from testing measurement theory models. *Perspectives*, 1997. 107

[218] Sunil Kumar Sahu, Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. Inter-sentence relation extraction with document-level graph convolutional neural network. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4309–4316, 2019. URL `https://www.aclweb.org/anthology/P19-1423/`.

[219] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 1163–1171, 2016.

[220] Julian Salazar, Davis Liang, Zhiheng Huang, and Zachary C Lipton. Invariant representation learning for robust deep networks. In *Workshop on Integration of Deep Learning Theories, NeurIPS*, 2018.

[221] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

[222] Carissa Schoenick, Peter Clark, Oyvind Tafjord, Peter Turney, and Oren Etzioni. Moving beyond the turing test with the allen ai science challenge. *Communications of the ACM*, 60(9):60–64, 2017. 99

[223] John Schulman, Pieter Abbeel, and Xi Chen. Equivalence between policy gradients and soft q-learning. *arXiv preprint arXiv:1704.06440*, 2017. 120, 121, 124, 126, 127

[224] H Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371, 1965.

[225] Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*, 2015.

[226] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *ACL*, 2016. 92

[227] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*, 2016. 98, 105

[228] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *Proceedings of the International Conference on Learning Representations*, 2017.

[229] Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. Minimum risk training for neural machine translation. *arXiv preprint arXiv:1512.02433*, 2015. 116, 127

[230] Tianxiao Shen, Myle Ott, Michael Auli, and Marc'Aurelio Ranzato. Mixture models for diverse machine translation: Tricks of the trade. *arXiv preprint arXiv:1902.07816*, 2019.

[231] Yelong Shen, Po-Sen Huang, Ming-Wei Chang, and Jianfeng Gao. Implicit reasonet: Modeling large-scale structured relationships with shared memory. *arXiv preprint arXiv:1611.04642*, 2016. xxi, 72, 77, 78

[232] Weiwei Shi, Yihong Gong, Chris Ding, Zhiheng MaXiaoyu Tao, and Nanning Zheng. Transductive semi-supervised deep learning using min-max features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 299–315, 2018.

[233] Hideyuki Shibuki, Kotaro Sakamoto, Yoshinobu Kano, Teruko Mitamura, Madoka Ishioroshi, Kelly Y Itakura, Di Wang, Tatsunori Mori, and Noriko Kando. Overview of the ntcir-11 qa-lab task. In *NTCIR*, 2014. 56, 59, 99

[234] Patrice Y Simard, Yann A LeCun, John S Denker, and Bernard Victorri. Transformation invariance in pattern recognitionâĂŤtangent distance and tangent propagation. In *Neural networks: tricks of the trade*, pages 239–274. Springer, 1998.

[235] Carl-Johann Simon-Gabriel, Yann Ollivier, Leon Bottou, Bernhard Schölkopf, and David Lopez-Paz. First-order adversarial vulnerability of neural networks and input dimension. In *International Conference on Machine Learning*, pages 5809–5817, 2019.

[236] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015.

[237] Adam Skory and Maxine Eskenazi. Predicting cloze task quality for vocabulary training. In *Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 49–56. Association for Computational Linguistics, 2010. 110

[238] Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1279. URL `https://www.aclweb.org/anthology/P19-1279`.

[239] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning With Neural Tensor Networks for Knowledge Base Completion. In *Advances in Neural Information Processing Systems 26*, pages 926–934. 2013. 71, 72, 78

[240] Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. N-ary relation extraction using graph-state LSTM. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2226–2235, 2018. URL `https://www.aclweb.org/anthology/D18-1246/`.

[241] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of*

machine learning research, 15(1):1929–1958, 2014.

[242] Robert Stanforth, Alhussein Fawzi, Pushmeet Kohli, et al. Are labels required for improving adversarial robustness? *arXiv preprint arXiv:1905.13725*, 2019.

[243] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. YAGO: A Core of Semantic Knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, pages 697–706, 2007. 71

[244] Qianru Sun, Xinzhe Li, Yaoyao Liu, Shibao Zheng, Tat-Seng Chua, and Bernt Schiele. Learning to self-train for semi-supervised few-shot classification. *arXiv preprint arXiv:1906.00562*, 2019.

[245] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014. 115, 127

[246] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[247] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[248] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

[249] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[250] Mingxing Tan and Quoc V Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, 2019.

[251] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204, 2017.

[252] Wilson L Taylor. âĂIJcloze procedureâĂİ: a new tool for measuring readability. *Journalism Bulletin*, 30(4):415–433, 1953. 97

[253] Joshua B Tenenbaum and William T Freeman. Separating style and content. *NIPS*, 1997.

[254] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996. 74

[255] Kristina Toutanova and Danqi Chen. Observed Versus Latent Features for Knowledge Base and Text Inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 57–66, 2015. 72, 78

[256] Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. Representing Text for Joint Embedding of Text and Knowledge Bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Pro-*

cessing, pages 1499–1509, 2015.

[257] Hugo Touvron, Andrea Vedaldi, Matthijs Douze, and Hervé Jégou. Fixing the train-test resolution discrepancy. *arXiv preprint arXiv:1906.06423*, 2019.

[258] Annie Tremblay. Proficiency assessment standards in second language acquisition research. *Studies in Second Language Acquisition*, 33(3):339–372, 2011. 97

[259] Trieu H Trinh, Minh-Thang Luong, and Quoc V Le. Selfie: Self-supervised pretraining for image embedding. *arXiv preprint arXiv:1906.02940*, 2019.

[260] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. Newsqa: A machine comprehension dataset. *arXiv preprint arXiv:1611.09830*, 2016. xxi, 55, 58, 60, 99, 100

[261] Peter D Turney. Domain and function: A dual-space model of semantic relations and compositions. *Journal of Artificial Intelligence Research*, 44:533–585, 2012.

[262] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. *arXiv preprint arXiv:1702.05464*, 2017.

[263] Betty van Aken, Benjamin Winter, Alexander Löser, and Felix A. Gers. How does BERT answer questions?: A layer-wise analysis of transformer representations. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pages 1823–1832, 2019. doi: 10.1145/3357384.3358028. URL `https://doi.org/10.1145/3357384.3358028`.

[264] Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge Belongie. Learning from noisy large-scale datasets with minimal supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 839–847, 2017.

[265] Patrick Verga, Emma Strubell, and Andrew McCallum. Simultaneously self-attending to all mentions for full-abstract biological relation extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 872–884, 2018. URL `https://www.aclweb.org/anthology/N18-1080/`.

[266] Vikas Verma, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. *arXiv preprint arXiv:1903.03825*, 2019.

[267] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 3156–3164. IEEE, 2015. 115, 127, 132

[268] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledge base. 2014.

[269] Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Pa-*

pers), volume 1, pages 189–198, 2017. 105

[270] Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. Switchout: an efficient data augmentation algorithm for neural machine translation. *arXiv preprint arXiv:1808.07512*, 2018.

[271] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge Graph Embedding by Translating on Hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1112–1119. 2014. 71, 72, 77, 78

[272] Zhuoyu Wei, Jun Zhao, and Kang Liu. Mining inference formulas by goal-directed random walks. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1379–1388, Austin, Texas, November 2016. Association for Computational Linguistics. 78

[273] Robert West, Evgeniy Gabrilovich, Kevin Murphy, Shaohua Sun, Rahul Gupta, and Dekang Lin. Knowledge Base Completion via Search-based Question Answering. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 515–526, 2014. 71

[274] Jason Weston, Frédéric Ratle, Hossein Mobahi, and Ronan Collobert. Deep learning via semi-supervised embedding. In *Neural Networks: Tricks of the Trade*, pages 639–655. Springer, 2012.

[275] Ronald J Williams and Jing Peng. Function optimization using connectionist reinforcement learning algorithms. *Connection Science*, 3(3):241–268, 1991. 127

[276] Sam Wiseman and Alexander M Rush. Sequence-to-sequence learning as beam-search optimization. *arXiv preprint arXiv:1606.02960*, 2016. 116, 134, 136

[277] Lijun Wu, Yiren Wang, Yingce Xia, QIN Tao, Jianhuang Lai, and Tie-Yan Liu. Exploiting monolingual data at scale for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4198–4207, 2019.

[278] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*, 2019.

[279] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[280] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057, 2015. 115, 127

[281] Ruochen Xu and Yiming Yang. Cross-lingual distillation for text classification. *ACL*, 2017.

[282] Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. Classifying relations via long short term memory networks along shortest dependency paths. In *proceedings of the*

2015 conference on empirical methods in natural language processing, pages 1785–1794, 2015.

[283] Yichong Xu, Jingjing Liu, Jianfeng Gao, Yelong Shen, and Xiaodong Liu. Towards human-level machine reading comprehension: Reasoning and inference with multiple strategies. *arXiv preprint arXiv:1711.04964*, 2017. 105

[284] I. Zeki Yalniz, Herv'e J'egou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. Billion-scale semi-supervised learning for image classification. *Arxiv 1905.00546*, 2019.

[285] Bishan Yang and Tom Mitchell. Leveraging knowledge bases in LSTMs for improving machine reading. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1436–1446, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1132. URL `https://www.aclweb.org/anthology/P17-1132`.

[286] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *Proceedings of the International Conference on Learning Representations*, 2015. 72, 76, 78

[287] Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. *arXiv preprint arXiv:1603.08861*, 2016.

[288] Zhilin Yang, Junjie Hu, Ruslan Salakhutdinov, and William W Cohen. Semi-supervised qa with generative domain-adaptive nets. *arXiv preprint arXiv:1702.02206*, 2017. 28, 53, 55

[289] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764, 2019.

[290] Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. DocRED: A large-scale document-level relation extraction dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 764–777, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1074. URL `https://www.aclweb.org/anthology/P19-1074`.

[291] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pages 189–196, 1995.

[292] Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6210–6219, 2019.

[293] Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. Semantic parsing via staged query graph generation: Question answering with knowledge base. 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)*, pages 1321–1331, Beijing, China, July 2015. Association for Computational Lin-

guistics. 71

[294] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. Qanet: Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv:1804.09541*, 2018.

[295] Bowen Yu, Zhenyu Zhang, Tingwen Liu, Bin Wang, Sujian Li, and Quangang Li. Beyond word attention: Using segment attention in neural relation extraction. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 5401–5407, 2019. doi: 10.24963/ijcai.2019/750. URL `https://doi.org/10.24963/ijcai.2019/750`.

[296] Mo Yu, Wenpeng Yin, Kazi Saidul Hasan, Cícero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. Improved neural relation detection for knowledge base question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 571–581, 2017. doi: 10.18653/v1/P17-1053. URL `https://doi.org/10.18653/v1/P17-1053`.

[297] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[298] Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Central moment discrepancy (cmd) for domain-invariant representation learning. *ICLR*, 2017. 4, 86, 96

[299] Richard S Zemel, Yu Wu, Kevin Swersky, Toniann Pitassi, and Cynthia Dwork. Learning fair representations. *ICML*, 2013. 91, 92, 96

[300] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/C14-1220`.

[301] Runtian Zhai, Tianle Cai, Di He, Chen Dan, Kun He, John Hopcroft, and Liwei Wang. Adversarially robust generalization just requires more unlabeled data. *arXiv preprint arXiv:1906.00555*, 2019.

[302] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S^4l: Self-supervised semi-supervised learning. In *Proceedings of the IEEE international conference on computer vision*, 2019.

[303] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *ICLR*, 2017. 85

[304] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[305] Richard Zhang. Making convolutional networks shift-invariant again. In *International Conference on Machine Learning*, 2019.

[306] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for

text classification. In *Advances in neural information processing systems*, pages 649–657, 2015.

[307] Xingcheng Zhang, Zhizhong Li, Chen Change Loy, and Dahua Lin. Polynet: A pursuit of structural diversity in very deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 718–726, 2017.

[308] Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, 2017. xxiii, 104, 141, 148

[309] Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 35–45, 2017. URL `https://nlp.stanford.edu/pubs/zhang2017tacred.pdf`.

[310] Yuhao Zhang, Peng Qi, and Christopher D. Manning. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2205–2215, 2018. URL `https://www.aclweb.org/anthology/D18-1244/`.

[311] Yuhao Zhang, Peng Qi, and Christopher D Manning. Graph convolution over pruned dependency trees improves relation extraction. *arXiv preprint arXiv:1809.10185*, 2018.

[312] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. ERNIE: enhanced language representation with informative entities. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1441–1451, 2019. URL `https://www.aclweb.org/anthology/P19-1139/`.

[313] Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. Joint extraction of entities and relations based on a novel tagging scheme. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1227–1236, 2017. doi: 10.18653/v1/P17-1113. URL `https://doi.org/10.18653/v1/P17-1113`.

[314] Giulio Zhou, Subramanya Dulloor, David G Andersen, and Michael Kaminsky. Edf: Ensemble, distill, and fuse for easy video labeling. *arXiv preprint arXiv:1812.03626*, 2018.

[315] Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 207–212, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-2034. URL `https://www.aclweb.org/anthology/P16-2034`.

[316] Xiaojin Zhu, Zoubin Ghahramani, and John D Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International confer-*

ence on Machine learning (ICML-03), pages 912–919, 2003.

[317] Xiaojin Jerry Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2005.

[318] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015. 99

[319] Brian D Ziebart. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Carnegie Mellon University, 2010. 120, 121

[320] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008. 127

[321] Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas, November 2016. Association for Computational Linguistics.

[322] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018.

[323] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin D Cubuk, and Quoc V Le. Rethinking pre-training and self-training. *arXiv preprint arXiv:2006.06882*, 2020.

[324] Geoffrey Zweig and Christopher JC Burges. The microsoft research sentence completion challenge. Technical report, Technical Report MSR-TR-2011-129, Microsoft, 2011. 98