

Singapore Management University

## Institutional Knowledge at Singapore Management University

---

Dissertations and Theses Collection (Open Access)

Dissertations and Theses

---

9-2022

### Robustness and cross-lingual transfer: An exploration of out-of-distribution scenario in natural language processing

YU, SICHENG

*Singapore Management University*, [scyu.2018@phdcs.smu.edu.sg](mailto:scyu.2018@phdcs.smu.edu.sg)

Follow this and additional works at: [https://ink.library.smu.edu.sg/etd\\_coll](https://ink.library.smu.edu.sg/etd_coll)



Part of the [Databases and Information Systems Commons](#), and the [Programming Languages and Compilers Commons](#)

---

#### Citation

YU, SICHENG. Robustness and cross-lingual transfer: An exploration of out-of-distribution scenario in natural language processing. (2022).

Available at: [https://ink.library.smu.edu.sg/etd\\_coll/446](https://ink.library.smu.edu.sg/etd_coll/446)

This PhD Dissertation is brought to you for free and open access by the Dissertations and Theses at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Dissertations and Theses Collection (Open Access) by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email [cherylds@smu.edu.sg](mailto:cherylds@smu.edu.sg).

**ROBUSTNESS AND CROSS-LINGUAL  
TRANSFER: AN EXPLORATION OF  
OUT-OF-DISTRIBUTION SCENARIO IN  
NATURAL LANGUAGE PROCESSING**

**SICHENG YU**

**SINGAPORE MANAGEMENT UNIVERSITY**

2022

# **Robustness and Cross-lingual Transfer: An Exploration of Out-Of-Distribution Scenario in Natural Language Processing**

by  
**Sicheng Yu**

Submitted to School of Computing and Information Systems in partial fulfillment  
of the requirements for the Degree of Doctor of Philosophy in Computer Science

## **Dissertation Committee:**

Jing JIANG (Supervisor / Chair)  
Professor of Computer Science  
Singapore Management University

Qianru SUN  
Assistant Professor of Computer Science  
Singapore Management University

Wei GAO  
Assistant Professor of Computer Science  
Singapore Management University

Aixin SUN  
Associate Professor of Computer Science  
Nanyang Technological University

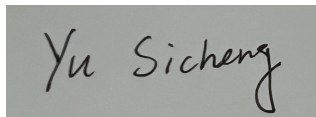
Singapore Management University  
2022

Copyright (2022) Sicheng Yu

I hereby declare that this PhD dissertation is my original work  
and it has been written by me in its entirety.

I have duly acknowledged all the sources of information  
which have been used in this dissertation.

This PhD dissertation has also not been submitted for any degree  
in any university previously.

A rectangular box containing a handwritten signature in black ink. The signature reads "Yu Sicheng" in a cursive style.

---

Sicheng Yu  
27 September 2022

# Robustness and Cross-lingual Transfer: An Exploration of Out-Of-Distribution Scenario in Natural Language Processing

Sicheng Yu

## Abstract

Most traditional machine learning or deep learning methods are based on the premise that training data and test data are independent and identical distributed, *i.e.*, *IID*. However, it is just an ideal situation. In real-world applications, test set and training data often follow different distributions, which we refer to as the out of distribution, *i.e.*, *OOD*, setting. As a result, models trained with traditional methods always suffer from an undesirable performance drop on the OOD test set. It's necessary to develop techniques to solve this problem for real applications.

In this dissertation, we present four pieces of work in the direction of OOD in Natural Language Processing (NLP) which can be further grouped into two sub-categories: adversarial robustness and cross-lingual transfer.

For the sub-category of adversarial robustness, the two work are summarized as follows:

1. We target at the question answering task. Question answering aims to find the answer given a passage, a question and possibly a set of options. Often-times question answering models over rely on some shortcut patterns, *e.g.*, word alignment between question and passage, instead of robust reasoning. Therefore, standard question answering models may fail on adversarial OOD sets where the shortcut fails to work. To this end, we analyze the shortcut in question answering task with the help of causal graphs and propose a counterfactual variable control method to mitigate the problem. The experiment results on different adversarial OOD sets show that our method improves the robustness and interpretability of question answering models.

2. We explore the model debiasing in the scenario of unknown bias where there is no prior knowledge about the bias for natural language understanding tasks. From the causal perspective, vulnerability in deep models is caused by the confounder, *e.g.*, the natural bias. A general method in causal inference for deconfounding is intervention. We propose an automatic and multi-granular intervention method for debiasing the natural language understanding models. With the help of the it, we achieve new state-of-the-art performance on three tasks under their OOD settings.

For the sub-category of cross-lingual transfer, the two work are summarized as follows:

1. We investigate the zero-shot and few-shot cross-lingual understanding tasks where the model is only trained with English data (zero-shot) and very few target language data (few-shot), then we directly apply the model on the target language which is OOD compared to the training data. We propose a counterfactual syntax method which injects the universal syntax into the model and further enforces the model to focus on the syntax information to assist the cross-lingual transfer. Such enriched and utilized syntax information helps the model to attain state-of-the-art performance on three cross-lingual understanding benchmarks.
2. We focus on the issue of translationese artifacts in translate-train method for cross-lingual transfer where we use the translated text of the target language for data augmentation. Although it introduces data of target language in training, it also brings the gap between originals and translationese. We propose an approach to mitigate the gap on the source language and apply it on target languages. The results demonstrate that our approach outperforms several strong baselines.

# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Two OOD scenarios: Robustness and Cross-lingual Transfer . . . . .	2
1.2	Research Contributions . . . . .	3
1.3	Dissertation Structure . . . . .	6
<b>2</b>	<b>Literature Review</b>	<b>8</b>
2.1	Other Out-Of-Distribution Scenarios . . . . .	8
2.1.1	Cross-domain Transfer . . . . .	8
2.1.2	Few-shot Learning . . . . .	9
2.1.3	Long-tailed Learning . . . . .	10
2.2	Adversarial Robustness . . . . .	11
2.2.1	Adversarial Attacks . . . . .	11
2.2.2	Methods . . . . .	13
2.3	Cross-lingual Transfer . . . . .	15
2.3.1	Multilingual Representation . . . . .	15
2.3.2	Cross-lingual Transfer . . . . .	17
<b>I</b>	<b>Adversarial Robustness</b>	<b>18</b>
<b>3</b>	<b>Counterfactual Variable Control for Robust and Interpretable Question Answering</b>	<b>19</b>
3.1	Introduction . . . . .	19

3.2	Counterfactual Variable Control (CVC)	22
3.2.1	Normal Prediction and Counterfactual Prediction	23
3.2.2	CVC Inference	24
3.3	The Implementation of CVC	26
3.3.1	Multi-task Training	26
3.3.2	Counterfactual Inference	28
3.3.3	Summary	30
3.4	Experiments	31
3.4.1	Experimental Settings	31
3.4.2	Implementation Details	34
3.4.3	Results and Analyses	37
3.5	Conclusion	44
<b>4</b>	<b>Interventional Training for Out-Of-Distribution Natural Language Understanding</b>	<b>46</b>
4.1	Introduction	46
4.2	Method	50
4.2.1	Preliminaries	50
4.2.2	Bottom-up Automatic Intervention	51
4.3	Experiment	54
4.3.1	NLU Tasks and Benchmarks	54
4.3.2	Implementation	55
4.3.3	Comparison with SOTAs	56
4.3.4	Ablation Studies	57
4.4	Conclusion	61
<b>II</b>	<b>Cross-lingual Transfer</b>	<b>62</b>
<b>5</b>	<b>COSY: COunterfactual SYntax for Cross-Lingual Understanding</b>	<b>63</b>



5.1	Introduction . . . . .	63
5.2	COSY: COunterfactual SYntax . . . . .	66
5.2.1	Syntax-Aware Networks (SAN) . . . . .	67
5.2.2	Counterfactual Training . . . . .	69
5.3	Experiments . . . . .	71
5.3.1	Datasets . . . . .	71
5.3.2	Implementation . . . . .	72
5.3.3	Results . . . . .	73
5.3.4	Discussion and Analysis . . . . .	74
5.4	Conclusion . . . . .	78
<b>6</b>	<b>Translate-Train Embracing Translationese Artifacts</b>	<b>79</b>
6.1	Introduction . . . . .	79
6.2	Our Approach (TEA) . . . . .	82
6.3	Experiments . . . . .	84
6.4	Conclusion . . . . .	86
<b>7</b>	<b>Conclusion and Future Work</b>	<b>87</b>

# List of Figures

1.1	The organization of the dissertation. . . . .	6
2.1	Examples of cross-domain transfer scenario from multi-domain sentiment analysis dataset [10] with two domains: books and housewares. . . . .	9
2.2	Examples of few-shot learning scenario from FewRel [47] with 3 way 2 shot. . . . .	10
2.3	Example of long-tailed learning scenario borrowed from the figure in Few-NERD [26]. . . . .	11
3.1	We observe multi-choice QA models are “capable” to answer a question without any <i>question</i> data in input (question-muted) during test (b), or during both training and test (c). We conduct these experiments using the BERT-base model [25] on the multi-choice QA benchmark DREAM [150]. (a) shows the normal case for reference. (d) show a training sample on DREAM. . . . .	21
3.2	The SCM of MCQA. <i>P</i> is for <i>passage</i> , <i>Q</i> for <i>question</i> , <i>O</i> for <i>options</i> and <i>A</i> for <i>answer</i> . Particularly, <i>R</i> denotes the comprehensive <i>reasoning</i> . . . . .	21

3.3	Multi-task training framework in our CVC. The complete input ( <i>e.g.</i> , $\mathcal{X} = \{P, Q, O\}$ for MCQA) are fed to the robust branch, while a subset to each shortcut ( <i>e.g.</i> , $\mathcal{X}_n = \{P, O\}$ to the $n$ -th branch). Solid arrows indicate feedforward, and dashed arrows for backpropagation. . . . .	24
3.4	Illustration of the processes for CVC-IV inference and CVC-MV inference. . . . .	25
3.5	The SCM for SEQA task where $Q$ is decomposed to $S, V$ and $E$ . . .	34
3.6	A case study of CVC on MCTest trained on official data. The ground truth is <u>underlined</u> . . . . .	41
3.7	A case study of CVC on SQuAD trained on official data. The distracting sentence from AddVerb is <u>underlined</u> . Only bold tokens in passage are shown in bar chart due to limited page size. . . . .	42
4.1	The proportions of entailment and non-entailment samples with different percentages of lexical overlap. . . . .	47
4.2	(a) Causal graph of NLU tasks, (b) intervention operation, and (c) an example of each node in the causal graph on the NLI task, where the data sample is from MNLI [171]. . . . .	48
4.3	First step of BAI: automatic stratifying where $M_{n_1}$ and $M_{n_2}$ are optimized individually. . . . .	52
4.4	Second step of BAI: bottom-up intervention. The dashed arrows denote the back-propagation. Only the modules (or parameter matrices) with dashed box are updated. . . . .	53
4.5	<b>RQ3.</b> The accuracies of one round of intervention on MNLI with different numbers of environments. . . . .	58

4.6	<b>RQ4.</b> (i) Characteristics and relationship for two partitions. Each sub-graph shows the same analysis setting as in Figure 4.1 in corresponding environment; (ii) Examples for the easy and hard samples for the partition with $n_2=2$ . . . . .	59
5.1	Examples of two sentences in English and Chinese that have the same meaning and share the same syntax in the format of <b>dependency relations</b> and <b>POS tags</b> . . . . .	64
5.2	Illustration of counterfactual syntax generation. Red color highlights the modified syntax with randomized labels. . . . .	65
5.3	The overall pipeline of our COSY. We call the architecture as syntax-aware networks (Section 5.2.1) and the training method as counterfactual training (Section 5.2.2). In this architecture, there are three branches: black, red and blue. Black branch is just the normal attention-based network with additional syntactic information, and only its prediction is used in the testing stage. <b>Red</b> branch and <b>blue</b> branch are novel as they generate the counterfactual syntax samples and drive the counterfactual losses in the training stage—the key functions in COSY. RGAT stands for Relational Graph Attention Network [60, 95]. <b>The modules of RGAT and the modules of Fusion Projection are shared across branches, e.g., two RGAT modules are sharing parameters.</b> Cat denotes concatenation. . . . .	67
5.4	Left: average F1-measure (%) on target languages on MLQA development set (mBERT). Right: average accuracy (%) on target languages on XNLI development set (mBERT). Red dotted line denotes the model performance of using naive fine-tuning. . . . .	75
5.5	F1-measure drop $\Delta$ (%) with a standard normal distribution perturbation on MLQA and XQUAD (mBERT). Two colors denote COSY and SAN-Black. . . . .	76

6.1	QA performance of using original and translated texts (translationese) as training data on TyDiQA dataset. “EM” stands for Exact Match.	80
6.2	The XLM-R [20] classification results of distinguishing the translationese for different languages on TyDiQA [19] when using the classifiers trained with only English pairs (originals and translationese), or randomly initialized (without any training).	81

# List of Tables

2.1	Summary of the adversarial sets utilized in this dissertation. The adversarial sets with * are proposed by us. “M, D, R” denotes MCTest, DREAM, and RACE. “FV” and “PI” denote Fact Verification and Paraphrase Identification. . . . .	13
2.2	Summarization of multilingual pretrained language models. Here the Training corpus denotes the monolingual corpus. . . . .	16
3.1	We conduct MCQA experiments on three datasets, <i>i.e.</i> , MCTest [135], DREAM [150], RACE [79], and SEQA experiments on the SQuAD dataset [129]. “Crowd.”: crowd-sourcing; “-”: not applicable. . . .	33
3.2	Accuracies (%) of conventional training BERT-base MCQA models tested with complete input. “No $X$ ” means the value of input variable $X$ is muted. . . . .	33
3.3	F1 scores (%) of conventional training BERT-base SEQA models tested with complete input. “No $X$ ” means the value of input variable $X$ is muted. . . . .	35

3.4	Accuracies (%) on three MCQA datasets. Models are trained on original training data. BERT-base, BERT-large and RoBERTa-large are backbones. “A.G.” denotes the average improvement over the conventional training (CT) [25] for $Adv^*$ sets. All results on RACE with RoBERTa-large are trained with 1/4 training data due to the resource limitation. “MV” and “IV” are “CVC-MV” and “CVC-IV”. “A” denotes “Adv”. . . . .	35
3.5	SEQA F1-measure (%) on the SQuAD <small>Dev</small> set ( <small>Test</small> set is not public) and adversarial sets. Models are trained on original training data. BERT-base, BERT-large and RoBERTa-large are backbones. “-”: not applicable from original paper. “A.G.”: our average improvement over the conventional training (CT) [25] for $Adv^*$ . “A” denotes “Adv”. Results of CT are from [25] and [97]. Results of QAInformax are from [182] . . . . .	36
3.6	Comparison of ours and related ensembling methods on MCQA with BERT-base. We implement these methods by replacing Eq. 3.10 with their adjustment functions. “A.G.”: our average improvement over the conventional training method (CT) [25] for $Adv^*$ . . . . .	37
3.7	Comparison of ours and related ensembling methods on SEQA with BERT-base. We implement DRiFt by directly changing our adjustment function (Eq. 10) to its. For Bias Product and Learned-Mixin, we first use the corresponding adjustment functions in [17], then we use the TF-IDF released by original paper as the shortcut branch in our implementation. “A.G.”: our average improvement over the conventional training method (CT) [25] for $Adv^*$ . . . . .	38
3.8	The ablation study on SQuAD (BERT-base). (1)-(4) are ablative settings for multi-task training (using CVC-IV); (5)-(9) are ablative settings related to CVC-MV. . . . .	39

3.9	The ablation study on MCTest (BERT-base). (1)-(4) are ablative settings for multi-task training (using CVC-IV inference). “Average” means the average performance on Adv* test sets; (5)-(9) are ablative settings related to CVC-MV inference. . . . .	40
3.10	Accuracies (%) on the MCTest dataset, using different kinds of data augmentation in training with BERT-base. The leftmost column shows which type of adversarial attack for MCQA is used as data enhancement. . . . .	43
3.11	NLI accuracies (%) on Matched Dev and HANS. Our CVC methods are trained only on the original training data (MNLI) with BERT-base.	44
4.1	Comparing our method to SOTAs on three benchmarks. Performance shown is in terms of accuracy. “KB” and “UB” denote known bias version and unknown bias version respectively. Results of Naive Fine-tuning, Reweighting, Product-of-Expert, Learned-Mixin and Regularized-Confidence and with known bias are from [37], [158] and [157]. Results of others are from the original paper.	55
4.2	<b>RQ1.</b> Results of ablative settings on MNLI. “FT” denotes Fine-tuning. . . . .	57
4.3	<b>RQ2.</b> Results of alternative methods for environment stratification on MNLI. . . . .	58
4.4	<b>RQ3.</b> Results of different orders and combinations of environment numbers on MNLI, arrows represent the intervention order. . . . .	60



5.1	Cross-lingual <b>zero-shot</b> performance comparison between COSY and SOTA methods on three benchmark datasets. Note that we report accuracy for XNLI and Exact Match/F1 scores for MLQA and XQUAD. For each dataset, “en.” denotes the results of English while “avg.” is the average performance over all languages. X-R means XLM-R and Naive F.T. is the abbr. of Naive Fine-Tuning. $L$ is the number of target languages. #T denotes the number of training turns, <i>e.g.</i> , STILT augments its training by using each of nine additional datasets. #M is the number of final models, where $1 < O(L) < L$ , and <b>A.D.</b> denotes using additional datasets. . . . .	71
5.2	Results of XNLI under the <b>few-shot</b> setting (mBERT). We report the testing results of English (“en.”), the average results over all non-English languages (“non-en. avg.”) and the average results over all languages (“avg.”). * denotes the results from [113]. More details are available in Appendix. . . . .	72
5.3	The ablation study on MLQA, XQUAD and XNLI (mBERT). We report the average performance of all languages on the test set. . . .	74
5.4	Results of different generation ways for generating counterfactual syntax with mBERT as backbone. “Current” means the current generation way described in Section 5.2. We report the average performance of all languages. . . . .	77
6.1	Main results (Exact Match / F1 scores) on TyDiQA. All methods are with XLM-R as backbone. The “Design” column indicates whether the design of this method considers translationese artifacts. The columns “ar” to “te” represent different target languages. The “avg” column denotes the average performance across the 8 target languages. * indicates our implementation. . . . .	84

6.2	Ablation study on TyDiQA. We report the average EM and F1 performance on the 8 target languages. . . . .	85
6.3	Experiment results of utilizing different language as pivot language for generating $\mathcal{X}_{\text{src, trans}}$ . . . . .	86

# Chapter 1

## Introduction

Machine learning methods have demonstrated their power in diverse areas, *e.g.*, computer vision, natural language processing and recommendation systems. An indispensable assumption for the majority of machine learning methods is that the training data and the test data follow the identical and independent distribution, *i.e.*, *IID*. However, this is just an ideal hypothesis, which is very difficult to achieve in practical applications. For example, a multiple-choice question answering model is trained on the samples that the longest options have higher probability to be the correct answer. The resulted model would perform very well on the test data with similar distribution. However, when it is used to handle the test data with different distribution, *e.g.*, correctness is unrelated to the length of the option, it will likely suffer a significant performance drop.

In this dissertation, we term such test data which is different from the training data as out-of-distribution data, *i.e.*, *OOD*. As mentioned above, it is necessary for the natural language processing (NLP) community to explore how to mitigate the performance drop on OOD, and this is the focus of this dissertation.

## 1.1 Two OOD scenarios: Robustness and Cross-lingual Transfer

There are various types of OOD. In this dissertation, we focus on two types, namely, robustness and cross-lingual transfer, as described below. A more comprehensive survey is conducted in Chapter 2.

- The first type of OOD in this dissertation is *adversarial attacks*, which are designed to expose the vulnerability of models. Models that can defend such attacks are said to have adversarial robustness. A key type of adversarial attack is the attacks towards shortcut correlation in a dataset. A representative example is the HANS adversarial set [105] for natural language inference designed for the MNLI dataset [171]. The natural language inference (NLI) task requires a model to identify the entailment relationship between a “premise” sentence and a “hypothesis” sentence: the relationship between the two sentences is considered “entailment” if the hypothesis can be inferred from the premise, “contradictory” if the two sentences contradict each other, and “neutral” if the two sentences are not related. Although ideally an NLI model needs deep understanding of the semantics of the two sentences to predict their entailment relation, oftentimes an NLI model learns shortcut correlations between some superficial patterns of the two sentences and the entailment label. For example, a common shortcut correlation is that the probability of “entailment” is proportional to the lexical overlap ratio between the two sentences. NLI models heavily relying on such a bias may collapse in specially designed OOD adversarial dataset. For example, “Mike ate an apple” and “An apple ate Mike” have totally the same bag of words; however, their meanings are completely the opposite.
- The second type of OOD is *cross-lingual transfer*, which is unique in natural language processing. Cross-lingual transfer is similar to cross-domain

transfer but the different domains here are different languages. The model is expected to learn the transferability across languages, *e.g.*, a model trained on English is expected to work directly on German text under zero-shot settings.

It is worth noting that the two types of OOD settings do not cover all OOD settings in NLP. However, we do not attempt to comprehensively review all OOD settings in NLP in this dissertation. Rather, we explore the two typical OOD settings above.

## 1.2 Research Contributions

In this thesis, we aim to develop general, task-agnostic methods to tackle adversarial attacks and cross-lingual transfer problems. For adversarial robustness, we focus on developing general methods that can prevent NLP models from relying on shortcut correlations in the training dataset; for cross-lingual transfer, we focus on methods that can overcome the language differences between the source and the target languages. Without loss of generalizability, we adopt several natural language understanding tasks, *e.g.*, question answering and natural language inference, as test beds in this thesis due to the rich set of existing datasets for these tasks.

We summarize our key contributions in adversarial robustness as follows:

- First, we explore the scenario where the shortcut correlation is already known. We focus on the question answering (QA) task for this work although our method presumably can also be applied to other tasks with some modifications. Under the setting of multiple-choice question answering where the model needs to select the best option given a passage and a question, we find that a trained QA model performs surprisingly well when the question is not given. The reason is likely that the QA model overly relies on the shortcut of word matching between the words in a candidate answer and words in the given passage, instead of real comprehension. Thus the model predicts

the option with the highest degree of lexical overlap with the passage while ignoring the question.

Motivated by this, we inspect the QA model through the lens of causality [117]. We formulate the shortcut and robust path as direct path and indirect path in a causal graph. Then we train the model to disentangle the robust path and the direct path through a multi-branch architecture. Each branch encodes either the shortcut (direct path) or the robust path (indirect path). During inference, the indirect effect is computed to enhance the robustness of the QA model. We conduct experiments on multiple-choice question answering and span-extraction question answering. The results on diverse adversarial sets demonstrate the effectiveness of our method.

- Our first work above assumes the accessibility of the bias characteristic and processes the bias samples through reweighting. However, oftentimes the bias is unknown or implicit when we face a new task or dataset. Furthermore, it is demonstrated that sample reweighting with a pre-defined bias model may waste data and mislead the resulted model [1]. In the second piece of work, we also address the aforementioned issues from the perspective of causality but from another angle. We regard the confounding bias as the reason for models to learn spurious correlations. While a common solution is to perform intervention, existing methods handle only known and single confounder [122], but in many NLU tasks the confounders can be both unknown and multifactorial. Thus we propose a novel interventional training method performing multi-granular intervention with identified multifactorial confounders. Our experiments on three NLU tasks, namely, natural language inference, fact verification and paraphrase identification, show that our method achieves state-of-the-art performance.

We summarize our key contributions in cross-lingual transfer as follows:

- Our focus point is zero-shot settings, *i.e.*, models trained on a source language

are directly tested on a target language. Normally, models trained directly on a source language overfit the source language and suffer a significant performance drop on a target language. We tackle this issue by incorporating language-agnostic information, specifically, universal syntax such as dependency relations and POS tags, into language models, based on the observation that universal syntax is transferable across different languages. Specifically, universal dependency and universal POS tags where “universal” denotes the annotation or label for dependency and POS tags are shared across all the languages, *e.g.*, the verb in Chinese is termed as “VERB” and the verb in English is also termed as “VERB”. Our approach includes the design of syntax-aware networks as well as a counterfactual training method to implicitly force the networks to learn not only the semantics but also the syntax. To evaluate our method, we conduct cross-lingual experiments on natural language inference and question answering using mBERT and XLM-R as network backbones. Our approach achieves the state-of-the-art performance without using auxiliary dataset.

- The second work related to cross-lingual transfer is based on a more practical standard approach, *i.e.*, the translate-train approach. The key idea of this approach is to use the translator of the target language to generate training data to mitigate the gap between source and target languages. However, its performance is often hampered by the artifacts in the translated texts (translationese). We discover that such artifacts have common patterns in different languages and can be modeled by deep learning. We thus propose an approach to mitigate such effect on the training data of a source language (whose original and translationese are both available), and apply the learned module to facilitate the inference on the target language. We conduct extensive experiments on the multilingual QA dataset TyDiQA. Our results show that our approach can outperform strong baselines.

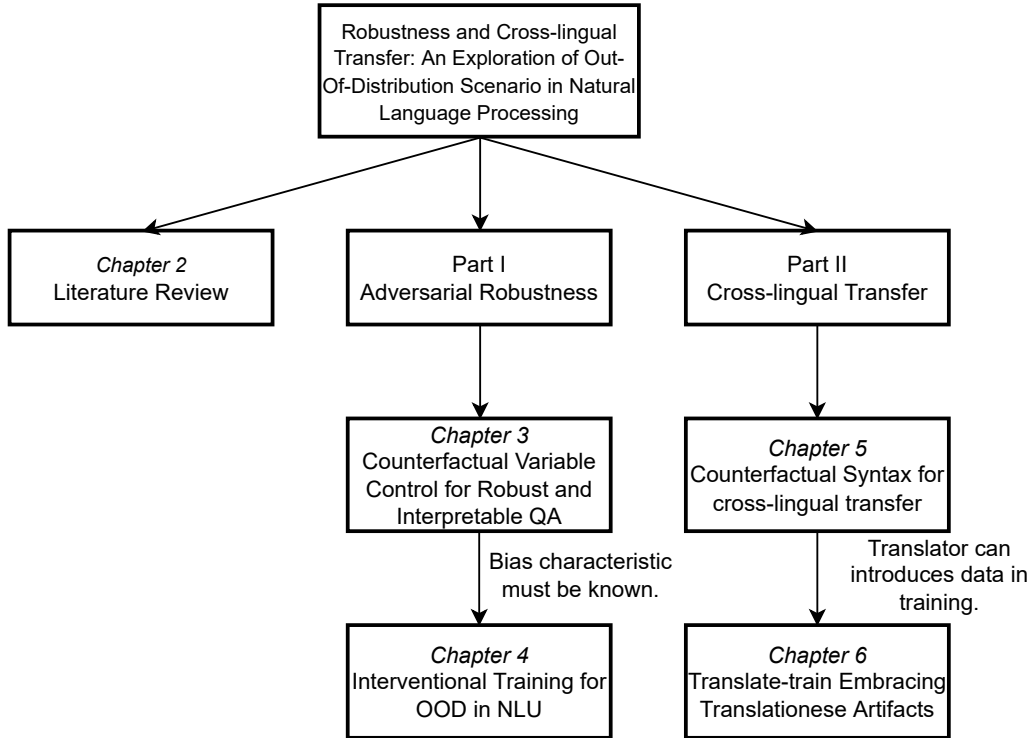


Figure 1.1: The organization of the dissertation.

### 1.3 Dissertation Structure

As shown in Figure 1.1, the remainder of this dissertation is organized as follows: We will first review some related literature corresponding to this dissertation in Chapter 2. Next, we will dive into Part I, adversarial robustness, and Part II, cross-lingual transfer, respectively.

In Part I, we have two chapters (Chapter 3 and Chapter 4). First we will elaborate on our first work of robustness in question answering in Chapter 3 using a novel counterfactual variable control method. This work is under review by the IEEE Transactions on Neural Networks and Learning Systems (TNNLS). Second work targets at a more practical scenario: bias is not known. To address the unknown bias problem, we propose an automatic and multi-factorial intervention method for general natural language understanding tasks. This work was published at the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP).

In Part II, we have two chapters (Chapter 5 and Chapter 6). Chapter 5 covers our second work on cross-lingual OOD by introducing syntax feature into cross-lingual



transfer. This work was published at the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021). In Chapter 6, we go through our next work focusing on the translate-train setting for cross-lingual transfer. Specifically, we explore the effect and solution for translationese artifacts. This work was published at the 60th Annual Meeting of the Association for Computational Linguistics (ACL 2022).

Finally, we present some directions of future work for out-of-distribution NLP research in Chapter 7.

# Chapter 2

## Literature Review

Due to the diverse and complex characteristic of different type of OOD, researchers have proposed various solutions for them. In this chapter, we first give a brief introduction to common scenarios of OOD not mentioned in Chapter 1. Noted that there may also be overlap between the different kinds of OOD scenarios. For example, the differences between the training and the test data may include both domain difference and language difference at the same time. Then we dive into the two scenarios that we focus in this dissertation, *i.e.*, adversarial robustness and cross-lingual transfer, and introduce the existing methods, respectively.

### 2.1 Other Out-Of-Distribution Scenarios

#### 2.1.1 Cross-domain Transfer

In the scenario of cross-domain transfer, data is from a different domain (or a different dataset). We give some examples in Figure 2.1 using sentiment analysis task. The sentiment analysis model is expected to generalize its ability from one domain, *i.e.*, the source domain, to another domain, *i.e.*, the target domain <sup>1</sup>. Most of the existing work can be divided into two groups. The first group of work is based

---

<sup>1</sup>The application may have more than one source domain or more than one target domain in cross-domain transfer.

Domain: Book	Domain: Housewares
<p><u>Label: Positive</u> I received this as a Christmas gift, and I read it again and again. A must-have for comic book fans.</p>	<p><u>Label: Positive</u> I was so thrilled when I unpacked my processor. It is so high quality and professional in both looks and performance.</p>
<p><u>Label: Negative</u> I wish I had the time spent reading this book back so I could use it for better purposes. This book wasted my life</p>	<p><u>Label: Negative</u> It also doesn't work 100% of the time, and we're not sure why. When we fill it, it seems to work fairly well right after but it either does not have as many sprays as it is supposed to, or it isn't working very long.</p>

Figure 2.1: Examples of cross-domain transfer scenario from multi-domain sentiment analysis dataset [10] with two domains: books and housewares.

on the domain generalization setting [88, 89, 90, 109] where the model only meets the data from the source domain without any data from the target domain. Since the target domain is the unseen domain in this setting, this setting is also the most challenging one in cross-domain transfer. The second group of work focus on a more practical setting, *i.e.*, we can collect some unlabeled data from target domain, which is also termed as unsupervised domain adaption [76, 172, 131]. In unsupervised domain adaptation, there are mainly four lines of work. One line of work applies loss modification, *e.g.*, adversarial loss [156, 54] guides the model to generate domain-agnostic feature or the weighting term for each sample [66]. Another line of work resorts to the pivot feature. Specifically, these methods construct the shared feature space for both the source domain and the target domain using the common feature [115, 197, 198, 199]. For example, the word “good” is useful in both of the book domain and houseware domain. The third line of work tries to annotate the unlabeled data with the pseudo-label [183, 200] using the model trained on the source domain. The last line of work comes with the pretrained models, *e.g.*, domain-oriented BERT variants [46, 42, ?].

## 2.1.2 Few-shot Learning

Few-shot learning displays a scenario where we only get access to very few training data [169]. For illustration, Figure 2.2 shows an example of few-shot learning for relation classification task. Here “3 way 2 shot” means there are three classes and

Training data	
<u>Label: Capital of</u>	London is the capital of the U.K. Washington is the capital of the U.S.A.
<u>Label: Member of</u>	Newton served as the president of the Royal Society. Leibniz was a member of the Prussian Academy of Sciences.
<u>Label: Birth name</u>	Samuel Langhorne Clemens, better known by his pen name Mark Twain, was an American writer. Alexei Maximovich Peshkov, primarily known as Maxim Gorky, was a Russian and Soviet writer.

Figure 2.2: Examples of few-shot learning scenario from FewRel [47] with 3 way 2 shot.

each class has two samples in training. In this scenario, it is non-trivial for the model to construct a high-quality data distribution for each class thus the gap between training data and test data is inevitable. The work proposed for few-shot learning include applying the data augmentation in the training [8, 35], meta-learning [32, 149], etc.

### 2.1.3 Long-tailed Learning

In the data collection process, it is hard to keep the label class balance due to the nature of the real world. For example, we show an example in the Figure 2.3 which is borrowed from the Few-NERD [26], Within the parent category “organization”, the child category “company” occupies a very large proportion, while the proportion of the child category “sportsteam” is very small. The frequent class and rare class are called head class and tail class respectively, and such a long-tailed phenomenon is studied as Zipf’s Law [133]. As a result, the model may perform poorly in the tail class due to the lack of training if the training data follows a long-tailed distribution. The solutions for long-tailed learning are mainly derived from several ideas. First, the target is to maintain a balanced dataset or training loss, *e.g.*, resampling, and reweighting [14]. Second, the two-stage strategy [69] is proposed where the backbone is learned in the first stage and the classifier is learned in the second stage with the backbone fixed. Third, the idea of ensemble learning is borrowed

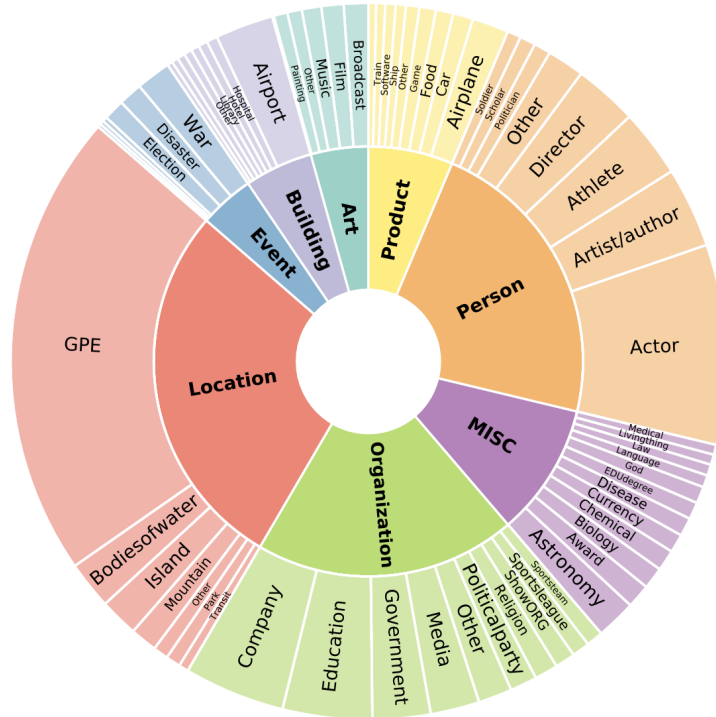


Figure 2.3: Example of long-tailed learning scenario borrowed from the figure in Few-NERD [26].

into long-tailed learning by introducing a multi-expert model [167].

## 2.2 Adversarial Robustness

Although large-scale pre-trained language models have shown their strength in language understanding, they could be easily fooled by simple adversarial attacks, *e.g.*, adding distractor sentences [189] or manipulating the semantic meaning by arranging the words [105]. In this section, we first outline the common types of adversarial attacks and focus on the benchmarks used or proposed in this dissertation. Then we zoom in on the methods for adversarial robustness for certain type of attacks against in this dissertation.

### 2.2.1 Adversarial Attacks

An adversarial sample is usually constructed by perturbation or modification on the original sample. The target is to fool the model, *i.e.*, the model would predict

incorrectly on the adversarial sample. From the angle of the accessibility of the model, the adversarial attack can be distinguished as white-box attack and black-box attack. Our dissertation mainly tackles the black-box attack.

The white-box assumes the validness of the model including its architecture, parameter, intermediate output, gradient, etc. The work belonging to the white-box attack can be roughly summarized into three groups. The first group is based on the Fast Gradient Sign Method (FGSM) based on the gradient of loss. FGSM is originally applied on image [38], and extended by the NLP community. Typical work include TextFool [92] and its upgraded version [140]. The FGSM-based attack usually applies the perturbation on the character level or the word level. The second group resorts to the directional derivatives. A well-known work is HotFlip [28] carried on the character level using swap, insert and delete. The third group utilizes the attention scores within the model computation by replacing the word which receives the largest attention score [11].

Black-box attack lacks the information of the model except for the final output, thus usually build on the heuristics. From the aspect of the relationship between the ground-truth label of the adversarial sample and the original sample, we define two types of black-box attacks, *i.e.*, label-reserved black-box attack and label-altered black-box attack. Label-reserved black-box attack method utilizes paraphrasing to generate semantically equivalent sentences. [61] produces adversarial samples using a syntactically controlled paraphrase network to manipulate the syntactic structure of the sentences. [134] proposes universal replacement rules for semantically equivalent sentences, *e.g.*, adding another question mark to the question sentence. Currently, the label-altered black-box attack considers the characteristics of the task and the original dataset and summarizes the bias existed in the dataset. In the adversarial sample generating process, the attacker modifies the sample so that the bias is misleading. For example, the lexical overlap bias in the natural language inference task may induce the model to predict entailment when the input two sentences share a large ratio of words. The attacker designs the adversarial sample by mak-

ing the two sentences with opposite semantic meanings while maintaining the same bag-of-words [105].

	Bias Type	Adversarial Strategy	Task	Dataset
AddSent [63]	Word alignment	Concatenation adversaries	QA	SQuAD
AddVerb*	Word alignment	Concatenation adversaries	QA	SQuAD
Add1Truth2Opt*	Word alignment	Option Modification	QA	M, D, R
Add2Truth2Opt*	Word alignment	Option Modification	QA	M, D, R
Add1Pas2Opt*	Word alignment	Concatenation adversaries	QA	M, D, R
Add1Ent2Pas*	Word alignment	Word scrambling	QA	M, D, R
HANS [105]	Lexical overlap ratio	Template-based	NLI	MultiNLI
FEVER Symm	Claim-only [142]	Manual generation	FV	FEVER
PAWS [190]	Lexical overlap ratio	Back-translation, Word scrambling	PI	QQP

Table 2.1: Summary of the adversarial sets utilized in this dissertation. The adversarial sets with \* are proposed by us. “M, D, R” denotes MCTest, DREAM, and RACE. “FV” and “PI” denote Fact Verification and Paraphrase Identification.

Finally, we summarize the adversarial sets used in this dissertation in Table 2.1.

## 2.2.2 Methods

There are several streams of work which aim at improving the robustness of model against the black-box attack.

- Many recent work generate adversarial examples to augment the training data such as to make the model more robust against adversarial attacks [134, 97, 63, 168]. They achieve fairly good performance but they have their limitations. First, they need to be aware of the prior knowledge of the adversarial attack, *i.e.*, “in what way to generate adversarial examples”, which is often not available in real applications. Second, their model performance strongly relies on the quality of adversarial examples as well as the training hyperparameters, *e.g.*, augmentation ratios.
- Alternative methods for robustness in NLP include using advanced regularizer [182, 98, 181], training loss [64, 58, 65], sample filtering [177, 12] and model ensembles [17, 13, 49, 157]. Among them, model ensembles are the

most popular method recently. Specifically, model ensemble method first designs a bias model and then trains a target debiased model fused with the bias model. Training instances predicted correctly by the bias model will be down-weighted in the training of the debiased model. Early work mainly revolves around different fusion methods [49, 17, 157, 103] with known bias. Then researchers started looking into unknown bias by designing the bias model with heuristics, *e.g.*, a model trained with very small amount of data [158] or a model with only the bottom layers of the language model [37]. However, instance reweighting based methods rely on either prior knowledge of bias or heuristic design of the bias model. Furthermore, it is pointed out that such bias models may not be able to predict the main model’s reaction of biased samples and reweighting may waste data [1].

- The third stream of work resorts to causal inference. Causal inference [120, 119] measures the causal effect between variables and has been widely applied to various scenarios, *e.g.*, social science [7] and medical science [45]. Causal inference can be applied for debiasing in NLP by measuring the robust causal effect and removing the undesirable spurious causal effect. Our work in Chapter 3 adopts counterfactual method from causal inference. Counterfactual analysis allows us to evaluate the effect of an event or variable by modifying it in a counterfactual scenario, which is contradicted to the factual scenario. Counterfactual methods are also emerging recently in natural language inference [70], semantic parsing [84], story generation [128], dialog systems [196], gender bias [161, 146], and sentiment bias [59]. In Chapter 3, we take the first step towards improving the robustness of QA models based on counterfactual analysis. Another work in Chapter 4 utilizes intervention derived in causal inference. Intervention [116] helps to eliminate the effect of confounders [179, 185, 110]. Invariant Risk Minimization (IRM) [2] is one of the method to implement intervention in deep neural network by learning



a model invariant to different environments [2, 166]. IRM has been widely adopted in computer vision community [77, 139, 23, 96, 166, 154]. In Chapter 4, we propose an interventional training method to eliminate the effect from confounders for natural language understanding tasks.

## 2.3 Cross-lingual Transfer

Cross-lingual learning aims to transfer knowledge from one source natural language to other target languages. In this section, we first review the backbone for the cross-lingual transfer, *i.e.*, multilingual representation. Then we turn to three different cross-lingual transfer settings and their related work.

### 2.3.1 Multilingual Representation

**In the multilingual word embedding era**, the cross-lingual transfer is always based on multilingual embedding. In terms of granularity, multilingual embedding could be divided into two categories.

The first category is based on word level. Typical work on word level depend on word mapping which first trains word embeddings in several languages individually and then maps them to shared space. Some methods directly maximize the similarity using square error [107], orthogonal transformation [3], Canonical Correlation Analysis (CCA) [44], and max-margin based ranking loss [85]. Other methods may incorporate the seed lexicon for a joint multilingual embedding space [4, 148, 81, 55].

The second category is based on the sentence level. Within this category, one group of methods utilizes matrix factorization. For example, FastAlign [27] uses paralleled sentence-pair and unparallelled word mapping for sentence alignment. [145] also extends the method by making use of monolingual data. Other groups of methods may directly bridge the gap between sentence pair [52] or reconstruct the sentence in the target language based on the idea of autoencoder [82].

	Training Objectives	Training Corpus	Parallel Data	Downstream
MBERT	MLM, NSP	Wikipedia	No	NLU
XLM	MLM, TLM, CLM	Wikipedia	Yes	NLU, NLG
XLM-RoBERTa	MLM	CommonCrawl	No	NLU
ERNIE-M	CAMLM, BTMLM	CommonCrawl	Yes	NLU
HICTL	Contrastive Learning	CommonCrawl	Yes	NLU
InfoXLM	Contrastive Learning	CommonCrawl	Yes	NLU
VECO	CAMLM	CommonCrawl	Yes	NLU, NLG

Table 2.2: Summarization of multilingual pretrained language models. Here the Training corpus denotes the monolingual corpus.

**In the pretrained language model era**, multilingual community extend the monolingual pretrained language model [25, 100] for multilinguality and demonstrate their prominent capability on cross-lingual knowledge transfer [175, 125, 56]. The cross-lingual transfer part of our thesis is also built on top of the multilingual pretrained language model. Multilingual BERT [25] is the first multilingual pre-trained language model. The differences between the multilingual version of BERT and its monolingual version are as follows: (1) the training corpus is multilingual while not paralleled; (2) the tokenizer and embedding table are built based on multilinguality as well. The training objectives of Multilingual BERT remain the same, *i.e.*, masked language modeling (MLM) and next sentence prediction (NSP) XLM [80] takes the parallel data into consideration by introducing an additional training objective, *i.e.*, translation language modeling (TLM). In addition to TLM, XLM also augments causal language modeling (CLM) following auto-regressive language modeling. After XLM, XLM-RoBERTa [21] scales the amount of unlabeled data in training. ERNIE-M [114] argues that the parallel data is not fully utilized in XLM and thus proposes cross-attention masked language modeling (CAMLM) and back-translation masked language modeling (BTMLM). Apart from the typical MLM training objective, some work resort to contrastive learning, *e.g.*, HICTL [170] and InfoXLM [15]. Recently, VECO [102] also unifies multilingual natural language understanding (NLU) and natural language generation (NLG) tasks in one multilingual model. All in all, we summarize the features of aforementioned in Table 2.2.

### 2.3.2 Cross-lingual Transfer

There are four settings of cross-lingual transfer attracting attentions from researchers: zero-shot, few-shot and translate-train. Zero-shot setting requires the model trained on source languages has the ability to directly test on target languages while few-shot setting provides a few additional data from target languages. Translate-train setting assumes the availability of a multilingual translator and translates the source language data into target languages as data augmentation.

Our first work on cross-lingual transfer in Chapter 5 focuses on zero-shot and few-shot settings. The bottleneck of these two settings is attributed to two issues: (i) catastrophic forgetting [73, 101], where knowledge learned in the pre-training stage is forgotten in downstream fine-tuning; (ii) lack of language-agnostic features [16, 191] or linguistic discrepancy between the source and the target languages [175, 83]. Existing work can be also roughly divided into two groups. The first proposes to modify the language model by aligning languages with parallel data [191] or strengthening sentence-level representation [170]. The second group focuses on the learning paradigm for fine-tuning on downstream tasks. For instance, some methods adopt meta-learning [113, 178] or intermediate tasks training [124] to learn cross-lingual knowledge. Our work belongs to the later group and fills the blank of using the syntactic information in zero-shot (few-shot) cross-lingual understanding.

Our second work on cross-lingual transfer in Chapter 6 focuses on translate-train setting. Existing translate-train methods explicitly utilize the parallel data [30] or augment more types of data in training [192]. However, the effect of translationese is ignored. The only attempts [5] are conducted for translate-test and zero-shot learning. Translationese artifacts have been widely studied in translation tasks [29, 39, 187, 86, 33]. Some recent work focus on how to mitigate or control the effect of translationese, *e.g.*, tagged training [104, 136, 165]. In contrast, our work concentrates on translate-train and aim at mitigating the artifacts in translationese.

# **Part I**

## **Adversarial Robustness**

## Chapter 3

# Counterfactual Variable Control for Robust and Interpretable Question Answering

### 3.1 Introduction

We explore the adversarial robustness against known bias situation in this chapter by focusing on the QA task. Recently, the error rates on the multiple-choice question answering (MCQA) and span-extraction question answering (SEQA) benchmarks were smashed overnight by large-scale pre-trained models, such as BERT [25], XLNet [180], RoBERTa [100] and Megatron-LM [147]. Impressively, using Megatron-LM achieved an error rate of less than 10% on the large-scale MCQA dataset RACE [79]. However, top-performing models often lack interpretability [31, 71], nor are they robust to adversarial attacks [134, 152, 163]. For example, adding one more question mark at the end of the input question, which is a simple adversarial attack, may decrease the performance of QA models [134]. This vulnerability will raise security concerns when the model is deployed in real-world applications, *e.g.*, intelligent shopping assistants and web search engines. It is thus desirable to figure out why this happens and how to improve the robustness of QA models.

Existing methods for robust QA models mainly resort to robust training. One straightforward way is to generate adversarial examples for training [63, 134]. However, sometimes it is expensive and time-consuming to manually generate adversarial examples, and QA models are still not robust to unseen attacks. On the other hand, recent work focus on regularizing QA models via additional losses. For example, QAInformax [182] maximizes the mutual information between the passage and the question to achieve regularization. However, so far, robust inference has not been fully exploited.

In this chapter, we carefully inspect both the training and the test processes for QA models. We find the aforementioned vulnerability is caused by the fact that the model tends to exploit the *correlations* in the training data. To illustrate this, we show some example results of the BERT-Base MCQA model [25] in Figure 3.1. Supposedly, the model should predict the answer based on the passage, the question, and the options. Surprisingly, the absence of the *question* during only the test stage (Figure 3.1 (b)) or during both the training and the test stages (Figure 3.1 (c)) leads to a limited performance drop. Our hypothesis is that the BERT-Base MCQA model uses a huge amount of network parameters to learn the *shortcut correlation* between the *no-question* inputs (*i.e.*, *passage* and *options*) and the ground-truth *answer* in a brute-force manner. Figure 3.1(d) shows an example where this *shortcut* could be realized by simply aligning the words appearing in both the *passage* and *options*. Can we just conclude from this example that *questions* have little effect on *answers*? We must say no, as this violates our common sense about the causality in QA — *the question causes the answer*.

With the observation above in mind, we take a step further towards robust and interpretable QA systems by figuring out the causality in QA based on causal inference [120, 122]. We begin by analyzing the causal relationships in QA, *i.e.*, associating any two variables based on the causal effect. Inspired by the recent success of causal inference in applications [126, 153, 111], we represent the causal relationships in QA using the Structural Causal Model (SCM) [120]. Figure 3.2(a)

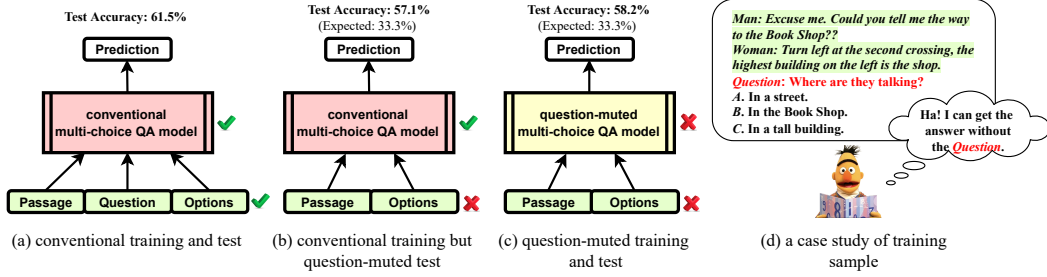


Figure 3.1: We observe multi-choice QA models are “capable” to answer a question without any *question* data in input (question-muted) during test (b), or during both training and test (c). We conduct these experiments using the BERT-base model [25] on the multi-choice QA benchmark DREAM [150]. (a) shows the normal case for reference. (d) show a training sample on DREAM.

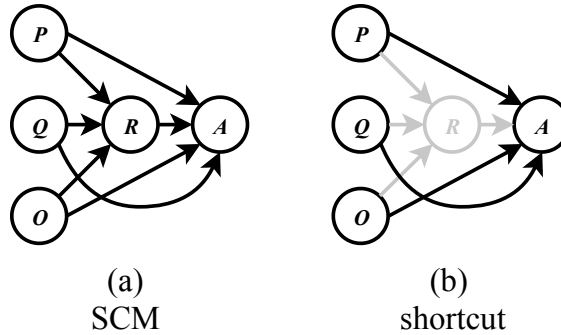


Figure 3.2: The SCM of MCQA.  $P$  is for *passage*,  $Q$  for *question*,  $O$  for *options* and  $A$  for *answer*. Particularly,  $R$  denotes the comprehensive *reasoning*.

shows the SCM for MCQA as an example, where each node denotes a variable (e.g.,  $Q$  for *question* and  $A$  for *answer*) and the directed edge from one node to another represents their causal relation (e.g.,  $Q \rightarrow A$  denotes *question causes answer*). Besides the input and output variables, we introduce an intermediate variable  $R$  to reflect the expected comprehensive *reasoning* among all the inputs. SCM illustrates that not only comprehensive reasoning but also shortcut correlations have effects on the output answer. As highlighted in Figure 3.2(b),  $P$  and  $O$  can directly reach  $A$ , leading to a success rate 24% higher than the random guess shown in Figure 3.1(b). These shortcut correlations are “distractors” against our goal of robust QA, *i.e.*, the prediction should be caused by the comprehensive reasoning.

According to the above causality-based analysis, we expect the robust QA systems to conduct comprehensive reasoning and exclude the shortcut effects for unbiased inference. To alleviate the effects of shortcuts, we propose a novel approach

called Counterfactual Variable Control (CVC) based on the causality theory. CVC in essence includes *counterfactual analysis* [120, 122, 118] and *variable control*. The former allows us to evaluate the effect of an event by modifying it in a counterfactual scenario. The latter, motivated by controlling for variables, aims to explicitly separate the effects of different variables. In this way, we can avoid any interference from controlled variables. To implement CVC in deep models, we realize the SCM as a multi-branch architecture [13, 17] that is composed of a robust branch reflecting the comprehensive reasoning and several shortcut branches. We highlight that CVC training exactly follows the multi-branch training [13], while CVC testing is based on counterfactual analysis to capture the indirect effects of only the comprehensive reasoning. To further evaluate the robustness of CVC, we conduct adversarial attacks that are challenging for shortcut correlations, *e.g.*, words alignment. Experiments are conducted on four QA benchmarks with different backbone networks, *e.g.*, BERT [25] and RoBERTa [100]. The results validate the effectiveness and generalizability of our proposed CVC approach. As shown in the case studies, our CVC can not only achieve robust performance, but also conduct interpretable and reasonable inference processes due to the theoretical foundation of causal inference.

Our main contributions include (i) an overall causality-based analysis using structural causal model for robust QA; (ii) a novel counterfactual variable control (CVC) approach to mitigating the shortcut correlations while preserving the robust comprehensive reasoning in QA; (iii) plugging CVC in different deep backbones and evaluating it on several QA benchmarks; (iv) four types of adversarial attacks for MCQA and one human-annotated adversarial set for SEQQA to evaluate the robustness of QA models.

## 3.2 Counterfactual Variable Control (CVC)

CVC aims to conduct unbiased inference by excluding the shortcut effects. In this section, we use multi-choice question answering (MCQA) as a case study of QA



tasks, and introduce the notations for our proposed Counterfactual Variable Control (CVC). Given a natural language paragraph as passage  $p$ , the models for MCQA are expected to answer the related question  $q$  by selecting the correct answer  $a$  from the candidate options  $o$ . In the following, we use uppercase letters to denote the variables (*e.g.*,  $Q$  for *question*) and lowercase letters for the specific value of a variable (*e.g.*,  $q$  for a specific question).

### 3.2.1 Normal Prediction and Counterfactual Prediction

We further introduce counterfactual notations, *i.e.*, the imagined values of variables as if their ancestors had existed (*i.e.*, uncontrolled) in a counterfactual world [120, 153, 118, 137]. We highlight that our overall notations are general and impose no constraints on the detailed implementation of QA models. Based on the input variables with their normal or counterfactual values, we define the notations for the two cases of model prediction: Normal Prediction and Counterfactual Prediction.

**Normal Prediction (NP)** means that the model makes predictions when the variables are all controlled or uncontrolled. We use the function format  $Y(X = x)$ , abbreviated as  $Y_x$ , to represent the effect of  $X = x$  on  $Y$ . We use this notation to formulate any path on the SCM, and further derive the prediction as:

$$A_{p,q,o,r} = A(P=p, Q=q, O=o, R=r), \quad (3.1)$$

where  $r = R(P = p, Q = q, O = o)$  denotes the normal value of comprehensive reasoning, and  $A_{p,q,o,r}$  denotes the inference logits of the model with realistic inputs values. If all the inputs are controlled (*e.g.*, muting their values as null), the value that  $A$  would obtain can be represented as:

$$A_{p^*,q^*,o^*,r^*} = A(P=p^*, Q=q^*, O=o^*, R=r^*), \quad (3.2)$$

where  $r^* = R(P = p^*, Q = q^*, O = o^*)$ , and  $A_{p^*,q^*,o^*,r^*}$  is the inference logits of the model with null values of input variables, which are denoted as  $p^*$ ,  $q^*$ , and  $o^*$ .

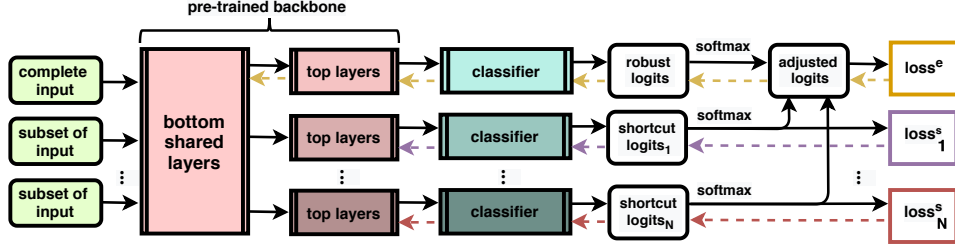


Figure 3.3: Multi-task training framework in our CVC. The complete input (e.g.,  $\mathcal{X} = \{P, Q, O\}$  for MCQA) are fed to the robust branch, while a subset to each shortcut (e.g.,  $\mathcal{X}_n = \{P, O\}$  to the  $n$ -th branch). Solid arrows indicate feedforward, and dashed arrows for backpropagation.

**Counterfactual Prediction (CP)** means that the model predicts the answer when some variables are controlled, but the others are assigned counterfactual values obtained when these variables are uncontrolled. This is a key operation in the *counterfactual analysis* [120, 122, 118]. For example, we control the input variables  $P$ ,  $Q$ , and  $O$  with their values to null (denoted as  $p^*$ ,  $q^*$ , and  $o^*$ ), and assign their child node  $R$  with a counterfactual value  $r = R(P = p, Q = q, O = o)$  obtained when the inputs  $P$ ,  $Q$ , and  $O$  were valid. Similarly, we can control  $R$  as  $r^*$  while assigning its parent nodes  $P$ ,  $Q$ , and  $O$  with counterfactual values  $p$ ,  $q$ , and  $o$ .

To conduct CVC inference, we propose two variants of counterfactual control: (i) controlling only input variables; and (ii) controlling only the mediator variable. For (i), we formulate the value of  $A$  as:

$$A_{p^*, q^*, o^*, r} = A(P = p^*, Q = q^*, O = o^*, R = r), \quad (3.3)$$

For (ii), we have:

$$A_{p, q, o, r^*} = A(P = p, Q = q, O = o, R = r^*), \quad (3.4)$$

### 3.2.2 CVC Inference

Recall that CVC is to preserve only the robust prediction derived by comprehensive reasoning and exclude shortcut correlations. Motivated by the theory of causality [108], CVC can be realized by comparing the fact and its counterpart, *i.e.*, estimating the difference between the normal prediction (NP) and the counterfactual

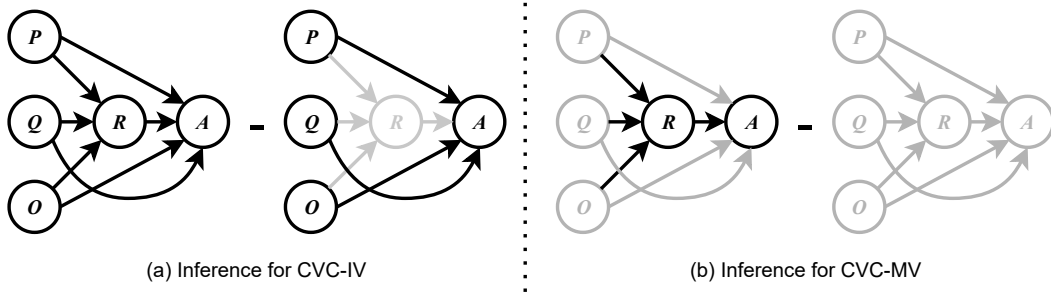


Figure 3.4: Illustration of the processes for CVC-IV inference and CVC-MV inference.

prediction (CP). Intuitively, the importance of a variable can be revealed by controlled experiments. If the difference between the experimental group and control group is large, this variable may have significant effect on the output. We utilize this conclusion from another view. If we know that a variable is essential, we expect the difference to be large. In our case, we expect the difference corresponded to the comprehensive reasoning  $R$  is large, *i.e.*, the model should rely on  $R$  for inference. Following the definition in Section 3.2.1, the idea can be realized by controlling on either inputs (*e.g.*,  $Q$ ) or mediator variables (*e.g.*,  $R$ ). Therefore, CVC can be realized in two ways corresponding to the controlled variables: CVC on Input Variables (CVC-IV) and CVC on Mediator Variables (CVC-MV). We illustrate the inference in Figure 3.4. **CVC on Input Variables (CVC-IV)** is derived as:

$$\text{CVC-IV} = A_{p^*,q^*,o^*,r} - A_{p^*,q^*,o^*,r^*} \quad (3.5)$$

where in  $A_{p^*,q^*,o^*,r}$  the input variables are controlled to be null (*e.g.*,  $p^*$ ) while the mediator variable is set as its counterfactual value, which is obtained by imaging a counterfactual world where the inputs had not been controlled (*i.e.*,  $r$ ).

**CVC on Mediator Variable (CVC-MV)** is derived as:

$$\text{CVC-MV} = A_{p,q,o,r} - A_{p,q,o,r^*}, \quad (3.6)$$

where in  $A_{p,q,o,r^*}$  the input variables are set as their observed values (*e.g.*,  $p$ ) while the mediator variable is controlled, *i.e.*, by imagining a counterfactual world where all inputs had been set to null (*i.e.*,  $r^*$ ).

Note that both CVC-IV and CVC-MV aim to capture the causal effect of comprehensive *reasoning* in QA. The main difference lies in *on which variables to apply the control*. The surgery is on the input variables in CVC-IV and the mediator variable in CVC-MV. The former aims to remove all the shortcut correlations, while the latter preserves only the effect of comprehensive *reasoning* on answer after the subtraction. Experiments further show that CVC-IV and CVC-MV perform differently in various QA settings.

### 3.3 The Implementation of CVC

In this section, we introduce how to implement CVC using deep neural networks, including multi-task training and counterfactual inference strategies. Multi-task training aims to separate robust path and shortcut paths by multi-branch architecture, counterfactual inference conducts unbiased inference based on CVC-IV or CVC-MV in Section 3.2.

#### 3.3.1 Multi-task Training

As illustrated in Figure 3.3, our overall framework implements the SCM in Figure 3.2(a) as multiple neural network branches. The main branch takes all the input variables (*i.e.*, complete input) to learn the causal effect corresponding to the robust path of SCM (*i.e.*,  $P, Q, O \rightarrow R \rightarrow A$ ), which we call comprehensive reasoning branch (or robust branch). The other branches, we call shortcut branches, take a subset of inputs (*i.e.*, part of the variables are muted) to explicitly learn the shortcut correlations corresponding to the shortcut paths of SCM (*e.g.*,  $P, O \rightarrow A$  as  $Q$  is muted). We deploy each branch as the standard QA model with pre-trained backbone [25] where the pre-trained backbone consists of bottom shared layers and top layers. The model is trained via multi-task training, *i.e.*, each branch is optimized using an individual objective. Only the robust branch gradients are propagated to update the bottom shared layers in the backbone.

**Robust branch**  $F^r$  aims to learn comprehensive reasoning for robust QA. It takes the complete input  $\mathcal{X}$ , *e.g.*, the realistic values of *question*, *passage* and *options* in MCQA. The network body, with parameters denoted as  $\theta^r$ , consists of a pre-trained backbone (*e.g.*, BERT) and a classifier (*e.g.*, one FC layer). Its prediction can be formulated as:

$$A^r = F^r(\mathcal{X}; \theta^r). \quad (3.7)$$

Following recent work [13], we fuse the prediction  $A^r$  with shortcut predictions to avoid the robust branch to overfit shortcut correlations. We will elaborate the details and explanations in the paragraph of loss computation.

**Shortcut branches**  $F_n^s$  ( $n = 1, 2, \dots, N$ ) aim to explicitly learn the unrobust correlations between incomplete (controlled) input and the ground truth answer. Each branch takes a subset of variables  $\mathcal{X}_n \subset \mathcal{X}$  as input, setting the other variables as null. Its network, with parameters denoted as  $\theta_n^s$ , has the same architecture with the robust branch. Its prediction can be formulated as:

$$A_n^s = F_n^s(\mathcal{X}_n; \theta_n^s), \quad (3.8)$$

**Loss Computation.** We use cross-entropy loss to optimize all the branches. For the  $n$ -th shortcut branch, we directly minimize the cross-entropy loss over its prediction  $A_n^s$ :

$$\mathcal{L}_n^s = - \sum_i p_i \log \text{softmax}(A_{n,i}^s), \quad (3.9)$$

where  $i$  denotes the  $i$ -th dimension of the prediction, and the one-hot vector  $p$  denotes the encoding of ground truth answer.

For the robust branch, directly optimizing over the robust prediction  $A^r$  cannot avoid the model to learn the correlations as in the conventional QA models, and cannot guarantee the model to learn the pure comprehensive reasoning. We tackle this problem by adjusting  $A^r$  using shortcut predictions  $A_n^s$ . In this way, we can force  $A^r$  to *only preserve the prediction that can never be achieved by shortcuts, i.e.*, the comprehensive reasoning prediction with the complete input variables as input. We

implement this adjustment by fusing the predictions from the robust branch and shortcut branches:

$$A_i^e = \sum_n \hat{\mathbf{p}}_i^r \cdot \hat{\mathbf{p}}_{n,i}^s, \quad (3.10)$$

where  $\hat{\mathbf{p}}_i^r = \text{softmax}(A_i^r)$ ,  $\hat{\mathbf{p}}_{n,i}^s = \text{softmax}(A_{n,i}^s)$  and  $i$  is the  $i$ -th dimension of the prediction. Here we use probabilities instead of logits because we empirically found negative values in logits may reverse the adjustment [17], while probabilities work as normalization and ensure each item in Eq. 3.10 positive value. We then optimize the cross-entropy loss over the adjusted result  $A^e$ :

$$\mathcal{L}^e = - \sum_i \mathbf{p}_i \log \text{softmax}(A_i^e). \quad (3.11)$$

However, we found that empirically the robust branch may focus on only hard samples and ignore easy samples by fusing the branches at the level of predictions. When prediction from shortcut branches are correct with high confidence, logits-level fusion may lead to a very small value in Eq. 3.11. We further propose two variants of losses to tackle this issue at the level of losses:

$$\begin{aligned} \mathcal{L}^{e1} &= - \sum_n \frac{1}{n} \sum_i \mathbf{p}_i \log \text{softmax}(\hat{\mathbf{p}}_i^r \cdot \hat{\mathbf{p}}_{n,i}^s), \\ \mathcal{L}^{e2} &= - \sum_n w_n \sum_i \mathbf{p}_i \log \text{softmax}(\hat{\mathbf{p}}_i^r \cdot \hat{\mathbf{p}}_{n,i}^s), \end{aligned} \quad (3.12)$$

where  $w_n = \text{softmax}(\mathcal{L}_n^s) = \frac{\exp(\mathcal{L}_n^s)}{\sum_{m=1}^n \exp(\mathcal{L}_m^s)}$ .  $w_n$  is a weight used to explicitly enhance the effect of the  $n$ -th shortcut branch on the robust branch. We formulate the overall loss used in multi-task training as follows,

$$\mathcal{L}^{all} = \mathcal{L}^e + \sum_n \mathcal{L}_n^s. \quad (3.13)$$

where  $\mathcal{L}^e$  can be replaced with  $\mathcal{L}^{e1}$  or  $\mathcal{L}^{e2}$ . We empirically found that  $\mathcal{L}^{e2}$  achieves better performance, and conduct the ablation study in experiments.

### 3.3.2 Counterfactual Inference

Different from conventional inference that is based on the posterior probability [25], we propose to use counterfactual inference based on causal effects [122, 118]. In

this section, we introduce how to conduct CVC-IV and CVC-MV inferences given the robust branch  $F^r$  and shortcut branches  $\{F_n^s\}_{n=1}^N$ .

Following the notation formats of NP and CP in Eq. 3.3 and Eq. 3.4 along with the notation of output for each branch in Eq. 3.7 and Eq. 3.8, we can (i) denote the prediction of the  $n$ -th shortcut branch as  $a_n^s = F_n^s(p, o; \theta_n^s)$  and its muted value as  $a_n^{s*} = F_n^s(p^*, o^*; \theta_n^s)$ ; and (ii) denote the prediction of the robust branch as  $a^r = F^r(p, q, o; \theta^r)$  and its muted value as  $a^{r*} = F^r(p^*, q^*, o^*; \theta^r)$ .

In the CVC-IV inference, we mute all the input variables. In this case, we obtain NP as  $A_{a_1^{s*}, \dots, a_N^{s*}, a^{r*}}$  and CP as  $A_{a_1^s, \dots, a_N^s, a^r}$ . Combining Eq. 3.5 and 3.10, we can derive the **CVC-IV inference result** as:

$$\begin{aligned} \text{CVC-IV} &= A_{a_1^{s*}, \dots, a_N^{s*}, a^{r*}} - A_{a_1^s, \dots, a_N^s, a^r} \\ &= \sum_n \hat{p}^r \cdot c_n^s - \sum_n c_n^r \cdot c_n^s, \end{aligned} \quad (3.14)$$

where each element in  $c_n^r$  or  $c_n^s$  is the same constant in  $[0, 1]$ . We highlight that CVC-IV inference corresponds to computing Natural Indirect Effect (NIE) in causal inference [122, 118]. It is equivalent to the normal inference on the robust model, similar to existing work such as Learned-Mixin [17]. Differently, CVC-IV is totally derived from the systematical causal analysis in QA and is thus more explainable than Learned-Mixin which is heuristic.

In the CVC-MV inference, we mute  $A^r$  as  $a^{r*}$ . We can denote the NP as  $A_{a_1^s, \dots, a_N^s, a^{r*}}$ , and the CP as  $A_{a_1^s, \dots, a_N^s, a^{r*}}$ . Combining Eq. 3.6 and 3.10, we can derive the **CVC-MV inference result** as:

$$\begin{aligned} \text{CVC-MV} &= A_{a_1^s, \dots, a_N^s, a^{r*}} - A_{a_1^s, \dots, a_N^s, a^{r*}} \\ &= \sum_n \hat{p}^r \cdot \hat{p}_n^s - \sum_n c_n^r \cdot \hat{p}_n^s, \end{aligned} \quad (3.15)$$

which is an indirect way of making inference using only the robust branch. It corresponds to computing Controlled Indirect Effect (CIE) in causal inference [122, 118].

We empirically find that the hyperparameter  $c_n^r$  makes a clear effect. Therefore, we train a  $c$ -adaptor  $F_n^c$  to adaptively estimate  $c_n^r$ . This can be formulated as:

$$c_n^r = F_n^c(\hat{p}^r, \hat{p}_n^s, \text{Distance}; \theta_n^c), \quad (3.16)$$

---

**Algorithm 1** Counterfactual Variable Control (CVC) algorithm

---

**Stage one: multi-task training****Input:** complete train set data  $\mathcal{X}$  and  $N$  different subsets of train set data  $\{\mathcal{X}_n\}_{n=1}^N$ **Output:**  $F^r$  with parameters  $\theta^r$  and  $\{F_n^s\}_{n=1}^N$  with parameters  $\{\theta_n^s\}_{n=1}^N$ 

- 1: **for** batch in  $\mathcal{X}$  and  $\{\mathcal{X}_n\}_{n=1}^N$  **do**
- 2:   **for**  $n$  in  $\{1, \dots, N\}$  **do**
- 3:     optimize  $\theta_n^s$  with batch of  $\mathcal{X}_n$  by Eq. 3.9;
- 4:   **end for**
- 5:   optimize  $\theta^r$  with batch of  $\mathcal{X}$  by Eq. 3.11 for MCQA (by  $\mathcal{L}^{e2}$  in Eq. 3.12 for SEQQA);
- 6: **end for**

**Stage two: counterfactual inference****Input:**  $F^r$  with parameters  $\theta^r$ ,  $\{F_n^s\}_{n=1}^N$  with parameters  $\{\theta_n^s\}_{n=1}^N$ , complete target test data  $\mathcal{X}'$  along with its subsets  $\{\mathcal{X}'_n\}_{n=1}^N$  and a boolean *USE\_IV*.**Output:** CVC inference result ( $\{F_n^c\}_{n=1}^N$  with parameters  $\{\theta_n^c\}_{n=1}^N$ )

- 1: **if** *USE\_IV* **then**
  - 2:   compute CVC-IV inference result with target data by Eq. 3.14;
  - 3: **else**
  - 4:   optimize  $\{\theta_n^c\}_{n=1}^N$  with  $\mathcal{X}$  and  $\{\mathcal{X}_n\}_{n=1}^N$  by Eq. 3.15, Eq. 3.16 and cross-entropy loss for QA task;
  - 5:   compute CVC-MV inference result with target data  $\mathcal{X}'$  and  $\{\mathcal{X}'_n\}_{n=1}^N$  by Eq. 3.15 and Eq. 3.16;
  - 6: **end if**
- 

where  $F_n^c(x_1, x_2, x_3; \theta_n^c) = \mathbf{W}_n^2 \tanh(\mathbf{W}_n^1[x_1; x_2; x_3])$ ,  $[\cdot]$  is the concatenation operation, and  $\theta_n^c = \{\mathbf{W}_n^1, \mathbf{W}_n^2\}$  are learnable parameters. We implement *Distance* as the Jensen-Shannon divergence [94]  $\text{JS}[\hat{\mathbf{p}}^r || \hat{\mathbf{p}}_n^s]$  between  $\hat{\mathbf{p}}^r$  and  $\hat{\mathbf{p}}_n^s$ . We implement *c*-adaptor using a two-layer MLP and conduct an ablative study to show its efficiency.

### 3.3.3 Summary

We highlight that CVC training follows the supervised training on multi-task networks [13, 17]. These work use similar architecture on other applications. However, our CVC-IV and CVC-MV inference methods are derived from our causal analysis of QA models — our main contribution to the QA methodology.

Algorithm 1 summarizes the pipeline of our proposed Counterfactual Variable



Control (CVC) approach. The approach consists of two stages: multi-task training (Section 3.3.1) and counterfactual inference (Section 3.3.2). Multi-task training aims to train a robust branch  $F^r$  and  $N$  shortcut branches  $\{F_n^s\}_{n=1}^N$ . Counterfactual inference performs the robust and interpretable reasoning for QA.

## 3.4 Experiments

### 3.4.1 Experimental Settings

We evaluate the robustness of CVC for both MCQA and SEQA, using a variety of adversarial attacks [189]. Below we introduce the base datasets followed by the adversarial sets for each base datasets. We conduct multi-task training on the training split of base datasets and conduct inference on original development/test splits of base datasets and adversarial sets.

#### Base Datasets

We show the information of MCQA base datasets in Table 3.1. Specifically, MCQA aims to select the correct answer from several input options given a passage and a question. We conduct experiments on the following benchmark datasets.

- **MCTest** [135] is generated from fictional stories and aims at open-domain machine comprehension. The questions are limited to the level that young children can understand. MCTest consists of two subsets, MC500 and MC160. We use the combination of them in our experiments.
- **DREAM** [150] is a dialogue-based dataset designed by experts to evaluate the comprehensive ability of foreign learners. In addition to simply matching questions, DREAM also contains more challenging questions that requires common-sense reasoning.
- **RACE** [79] is a dataset of English exam from middle and high school reading

comprehension. RACE covers a variety of topics and the proportion of questions that requires reasoning is much larger than other reading comprehension datasets.

Compared to MCQA, options are not provided on the SEQA task. SEQA locates the answer span in a passage given a question. We use the SQuAD dataset for SEQA.

- **SQuAD** [129] is adopted as the benchmark for SEQA where passages are from a set of Wikipedia articles. SQuAD requires several types of reasoning like lexical variation, syntactic variation, etc.

### **Adversarial Sets**

**Adversarial Attacks on MCQA.** To further evaluate the robustness of QA models, we propose four kinds of grammatical adversarial attacks to generate adversarial examples.

- **Add1Truth2Opt** and **Add2Truth2Opt** (Adv1 and Adv2): We replace one (or two) of the wrong options with another one (or two) answers that are correct in other samples with the same passage.
- **Add1Pas2Opt** (Adv3): We replace one of the wrong options with a random distracting sentence extracted from the passage. This distractor does not contain any word that appears in the ground truth option.
- **Add1Ent2Pas** (Adv4): We first choose one of the wrong options with at least one entity, *e.g.*, person name and time, and then replace each entity with another entity of the same type. Then, we add this modified sentence to the end of the passage.

**Adversarial Attacks on SEQA.** For the SEQA task, we utilize three kinds of grammatical adversarial attacks. **AddSent** (Adv1), **AddOneSent** (Adv2) and **AddVerb** (Adv3).

	<b>MCTest</b>	<b>DREAM</b>	<b>RACE</b>	<b>SQuAD</b>
Construction	Crowd.	Exams	Exams	Crowd.
Passage type	Child’s stories	Dialogues	Written text	Wikipedia
# of passages	660	6,444	27,933	23,215
# of questions	2,640	10,197	97,687	107,785
# of options	4	3	4	-

Table 3.1: We conduct MCQA experiments on three datasets, *i.e.*, MCTest [135], DREAM [150], RACE [79], and SEQA experiments on the SQuAD dataset [129]. “Crowd.”: crowd-sourcing; “-”: not applicable.

	MCTest	DREAM	RACE
Random guess	25.0	33.3	25.0
Complete input	68.9	61.5	64.7
No $P$	24.2	32.8	41.6
No $Q$	52.5	57.1	51.0
No $P, Q$	22.4	33.4	34.7

Table 3.2: Accuracies (%) of conventional training BERT-base MCQA models tested with complete input. “No  $X$ ” means the value of input variable  $X$  is muted.

- AddSent and AddOneSent released by [63] add distracting sentences to the passage. The generating process is: firstly perturb the question (*e.g.*, asking another entity) and create a fake answer, then convert the perturbed question into a distracting sentence. The final distracting sentences were filtered by crowdworkers. AddSent is similar to AddOneSent but much harder than AddOneSent. These two settings can be used to measure the model robustness against *entity* or *noun* attacks.
- AddVerb was inspired by above two sets which aims to evaluate the model robustness against *verb* attacks instead of *noun*. we hire expert linguists to annotate the AddVerb following [63]. Examples are as follows. For the question “*What city did Tesla move to in 1880?*”, AddSent sample could be “*Tadakatsu moved to the city of Chicago in 1881.*”, and AddVerb sample could be “*Tesla left the city of Chicago in 1880.*”

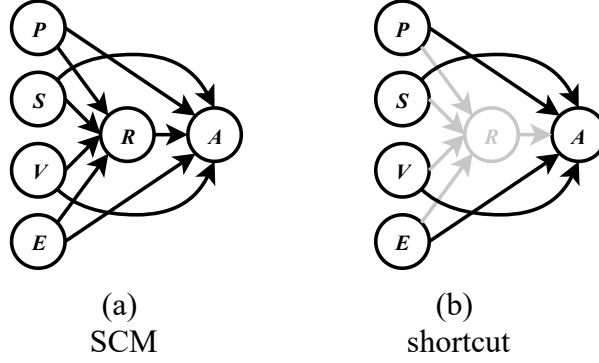


Figure 3.5: The SCM for SEQa task where  $Q$  is decomposed to  $S$ ,  $V$  and  $E$ .

### 3.4.2 Implementation Details

#### General Implementation

We deploy the pre-trained BERT and RoBERTa backbones provided by HuggingFace [174]. The learning rates are fixed to  $3e-5$ ,  $2e-5$  and  $1e-5$  for BERT-base, BERT-large, and RoBERTa-large respectively. A linear warm-up strategy for learning rates is used with the first 10% steps in the whole multi-branch training stage. The batch size is selected amongst  $\{16, 24, 32\}$  for the three backbones. The number of bottom shared layers is fixed to  $5/6$  of the total number of layers in the backbone language model for parameter-efficiency, *e.g.*, sharing 10 layers in bottom shared layers when the BERT-base (12 layers) is adopted as the backbone. The overall experiments are conducted on two pieces of Tesla V100 or two pieces of RTX 2080Ti (depending on the usage of memory). Gradient accumulation and half precision are used to relieve the issue of memory usage. Following [17, 40, 130], we perform model selection for CVC-IV (*i.e.*, choosing the hyperparameters of training epochs) based on the model performance in the development/test sets on the used dataset.

#### MCQA-Specific Implementation

MCQA has two shortcut correlations (see Figure 3.2), *i.e.*,  $Q \rightarrow A$  and  $P \rightarrow A$ <sup>1</sup>. We present the muting experiment results of MCQA in Table 3.2 that can reflect the strength of corresponding direct cause-effects. For example, the results on the row

<sup>1</sup> $O \rightarrow A$  is not discussed here as  $O$  is mandatory and can not be muted

SQuAD	
Complete input	88.1
No $E$	59.4
No $V$	55.1
No $E, V$	15.3
No $Q$	12.4

Table 3.3: F1 scores (%) of conventional training BERT-base SEQA models tested with complete input. “No  $X$ ” means the value of input variable  $X$  is muted.

of “No  $Q$ ” represent the performance of only using  $P \rightarrow A$  and  $O \rightarrow A$  shown in Figure 3.2 (b). We inspect them and notice that the effect from the former one is trivial and negligible compared to the latter. One may argue that  $Q$  is an important cue to predict the answer. Actually, annotators intentionally avoid any easy question-answer pairs when building MCQA datasets. For example, they include a person name in all options of questions about *who*. We thus assume  $Q \rightarrow A$  has been eliminated during well-designed data collection and utilize one shortcut branch (*i.e.*, muting  $Q$ ). Therefore, Eq. 3.11 and 3.12 are equivalent for MCQA ( $N = 1$  and  $w_n = 1$ ). Other MCQA-specific implementation details are the same with the official code of [25].

Data	Method	BERT-base					BERT-large					RoBERTa-large							
		Test	A1	A2	A3	A4	A.G.	Test	A1	A2	A3	A4	A.G.	Test	A1	A2	A3	A4	A.G.
MCTest	CT	68.9	63.9	59.4	20.2	54.8	-	72.3	70.0	66.8	35.5	57.6	-	88.9	88.2	86.6	72.6	84.2	-
	MV	68.1	69.1	65.6	26.8	61.0	+6.1%	73.2	74.3	73.5	38.4	68.4	+6.2%	88.5	89.3	89.6	82.4	83.4	+3.3%
	IV	69.4	70.0	65.4	28.7	59.9	+6.4%	74.4	75.5	75.1	40.4	69.5	+7.6%	87.4	88.1	88.2	82.6	84.2	+2.9%
DREAM	CT	61.5	47.5	39.2	20.9	41.8	-	65.9	50.6	43.0	25.6	48.2	-	84.1	78.2	76.3	57.1	71.8	-
	MV	60.1	49.6	39.9	23.7	45.6	+2.3%	64.0	51.9	46.5	26.3	51.3	+2.2%	82.8	77.9	80.2	66.6	71.4	+3.2%
	IV	60.0	49.2	40.7	25.0	47.1	+3.1%	64.5	52.0	46.2	26.6	51.1	+2.1%	81.7	78.3	79.7	66.7	72.3	+3.4%
RACE	CT	64.7	56.0	50.1	36.6	58.3	-	67.9	61.9	57.9	51.0	61.7	-	78.4	72.4	67.9	65.9	72.1	-
	MV	64.4	56.7	51.7	39.1	59.2	+1.4%	68.5	62.6	58.2	52.0	65.7	+1.5%	78.1	74.4	72.1	68.3	72.4	+2.2%
	IV	64.1	57.0	52.2	38.8	58.6	+1.4%	68.4	63.1	59.1	51.3	65.1	+1.6%	77.6	75.3	73.3	68.6	71.4	+2.6%

Table 3.4: Accuracies (%) on three MCQA datasets. Models are trained on original training data. BERT-base, BERT-large and RoBERTa-large are backbones. “A.G.” denotes the average improvement over the conventional training (CT) [25] for Adv\* sets. All results on RACE with RoBERTa-large are trained with 1/4 training data due to the resource limitation. “MV” and “IV” are “CVC-MV” and “CVC-IV”. “A” denotes “Adv”.

Method	BERT-base					BERT-large					RoBERTa-large				
	Dev	A1	A2	A3	A.G.	Dev	A1	A2	A3	A.G.	Dev	A1	A2	A3	A.G.
CT	88.4	49.9	59.7	44.6	-	90.6	60.2	70.0	50.0	-	93.5	77.0	82.8	61.3	-
QAInformax	88.6	54.5	64.9	-	+4.9%	-	-	-	-	-	-	-	-	-	-
CVC-MV	87.2	55.7	65.3	51.3	+6.0%	90.2	62.6	72.4	52.5	+2.4%	92.6	79.4	84.1	63.2	+1.9%
CVC-IV	86.6	56.3	66.2	51.5	<b>+6.6%</b>	89.4	62.6	71.8	54.1	<b>+2.8%</b>	92.2	79.6	85.0	64.1	<b>+2.5%</b>

Table 3.5: SEQA F1-measure (%) on the SQuAD Dev set (Test set is not public) and adversarial sets. Models are trained on original training data. BERT-base, BERT-large and RoBERTa-large are backbones. “-”: not applicable from original paper. “A.G.”: our average improvement over the conventional training (CT) [25] for Adv\*. “A” denotes “Adv”. Results of CT are from [25] and [97]. Results of QAInformax are from [182]

### SEQA-Specific Implementation

Different from MCQA, we propose to manually separate the *question* ( $Q$ ) of SEQA into corresponding parts: entities & nouns ( $E$ ); verbs & adverbs ( $V$ ); and the remaining stop words & punctuation marks ( $S$ ). As shown in Figure 3.5, the SCM of SEQA contains four input variables as  $P$  (*passage*),  $E$ ,  $V$  and  $S$ . The comprehensive *reasoning* variable  $R$  mediates between these four variables and *answer*  $A$ . The reason why we conduct this partition is twofold: (1)  $P$  is mandatory for SEQA. The lack of  $P$  will result in an invalid prediction. To study the effects of  $Q \rightarrow A$ , what we can do is to split the variable  $Q$  into partitions. (2) Our resulting  $Q$  partitions are intuitive.  $E$  and  $V$  contain the most important semantic meanings. We inspect the empirical effects of all shortcut paths as shown in Table 3.3, and build shortcut branches with  $N = 2$  to represent all shortcut paths in Figure 3.5(b). The first shortcut branch takes  $\mathcal{X}_1 = \{P, S, V\}$  as input and aims to learn  $P, S, V \rightarrow A$ . The second shortcut branch takes  $\mathcal{X}_2 = \{P, S, E\}$  as input and learns  $P, S, E \rightarrow A$ . We empirically use  $\mathcal{L}^{e2}$  to train SEQA models. Other SEQA-specific implementation details are the same with the official code of [25].

Set Method	Test	Adv1	Adv2	Adv3	Adv4	A.G.
<b>MCTest</b>	CT [25]	68.9	63.9	59.4	20.2	54.8 -
	DRiFt [49]	69.6	66.0	61.9	23.0	54.8 +1.9%
	Bias Product [17]	71.0	66.7	63.6	22.8	65.5 +5.1%
	Learned-Mixin [17]	70.5	66.2	60.4	20.2	58.8 +1.8%
	CVC-MV	68.1	69.1	65.6	26.8	61.0 +6.1%
	CVC-IV	69.4	70.0	65.4	28.7	59.9 <b>+6.4%</b>
<b>DREAM</b>	CT [25]	61.5	47.5	39.2	20.9	41.8 -
	DRiFt [49]	60.1	48.5	42.2	23.9	44.7 +2.5%
	Bias Product [17]	58.6	47.5	38.8	22.6	40.2 -0.1%
	Learned-Mixin [17]	60.9	49.2	41.7	20.0	42.3 +1.0%
	CVC-MV	60.1	49.6	39.9	23.7	45.6 +2.3%
	CVC-IV	60.0	49.2	40.7	25.0	47.1 <b>+3.1%</b>
<b>RACE</b>	CT [25]	64.7	56.0	50.1	36.6	58.3 -
	DRiFt [49]	62.0	56.1	53.3	39.3	58.3 <b>+1.7%</b>
	Bias Product [17]	62.3	56.7	53.3	37.0	56.8 +1.0%
	Learned-Mixin [17]	64.3	56.5	51.9	38.0	60.1 +1.4%
	CVC-MV	64.4	56.7	51.7	39.1	59.2 +1.4%
	CVC-IV	64.1	57.0	52.2	38.8	58.6 +1.4%

Table 3.6: Comparison of ours and related ensembling methods on MCQA with BERT-base. We implement these methods by replacing Eq. 3.10 with their adjustment functions. “A.G.”: our average improvement over the conventional training method (CT) [25] for  $\text{Adv}^*$ .

### 3.4.3 Results and Analyses

#### Comparison with Baselines and State-of-the-Arts

Table 3.4 and Table 3.5 show the overall results for MCQA and SEQA, respectively. Note that the adversarial sets  $\text{Adv}$  are used to evaluate the robustness of QA models. We report the average gain on  $\text{Adv}$ , denoted as A.G., to compare CVC with the conventional training methods (CT). From Table 3.4, we can see that both CVC-MV and CVC-IV can surpass the baseline method [25] for defending against adversarial attacks, *e.g.*, by average increase of 7.6% with BERT-large and 3.3% with RoBERTa-large on MCTest. It is worth highlighting the example that CVC-IV on BERT-base gains 8.5% on the most challenging  $\text{Adv}3$  set of MCTest. Besides, our methods are applicable to different backbones like BERT and RoBERTa-large. The results on SEQA in Table 3.5 show similar observation. These results empiri-

Method	Dev	Adv1	Adv2	Adv3	A.G.
CT [25]	88.4	49.9	59.7	44.6	-
DRiFt [49]	85.7	53.7	65.7	48.5	+4.5%
Bias Product [17]	87.8	53.6	65.7	47.3	+4.1%
Learned-Mixin [17]	87.2	53.1	63.9	45.5	+2.1%
CVC-MV	87.2	55.7	65.3	51.3	+6.0%
CVC-IV	86.6	56.3	66.2	51.5	<b>+6.6%</b>

Table 3.7: Comparison of ours and related ensembling methods on SEQa with BERT-base. We implement DRiFt by directly changing our adjustment function (Eq. 10) to its. For Bias Product and Learned-Mixin, we first use the corresponding adjustment functions in [17], then we use the TF-IDF released by original paper as the shortcut branch in our implementation. “A.G.”: our average improvement over the conventional training method (CT) [25] for Adv\*.

cally demonstrate that our CVC strategy is general and model-agnostic.

Compared to state-of-the-art method, our CVC is more robust to adversarial attacks. As shown in Table 3.5, CVC outperforms the state-of-the-art QAInfor-max [182] by an average of 1.7% F1-measure with the same BERT-base backbone. As shown in Table 3.6 and Table 3.7, CVC also outperforms ensemble based methods [17] on MCTest and DREAM datasets. Besides, all the approach achieve less improvement on RACE compared to other two datasets. The possible reason is that RACE is designed for reading comprehension that highlights comprehensive reasoning. Thus, the training data is more debiased. Note that our counterfactual analysis can regard these ensemble based methods as implementation of our CVC-IV.

Also, we notice that CVC-MV often performs worse than CVC-IV on Adv sets but better on in-domain Test (or Dev) sets. The possible reason is that the important hyperparameter of CVC-MV  $c_n^r$  is learned from in-domain data. We will show that augmenting in-domain data with Adv examples greatly improves the performance of CVC-MV in Table 3.10.



Ablative Setting	Dev	Adv1	Adv2	Adv3
(1) w/o first Shct.br.	85.5	52.6	62.5	50.8
(2) w/o second Shct.br.	86.1	57.7	66.1	42.1
(3) use $\mathcal{L}^e$	72.4	45.9	54.9	42.6
(4) use $\mathcal{L}^{e1}$	86.5	53.5	63.2	46.7
CVC-IV (ours)	86.6	56.3	66.2	51.5
(5) same $c_n^r$	85.7	54.3	64.1	51.0
(6) $c_n^r = JS$	85.9	54.3	64.2	51.1
(7) $c_n^r = Euc$	86.0	54.4	64.1	51.2
(8) w/o <i>distance</i>	86.9	55.3	65.0	51.3
(9) w/o $\hat{p}_r$ and $\hat{p}_n$	84.0	53.2	62.6	49.4
CVC-MV (ours)	87.2	55.7	65.3	51.3

Table 3.8: The ablation study on SQuAD (BERT-base). (1)-(4) are ablative settings for multi-task training (using CVC-IV); (5)-(9) are ablative settings related to CVC-MV.

### Ablation Study

**Ablations on SEQA.** Table 3.8 shows the SEQA results in 10 ablative settings to evaluate the importance of shortcut branches, loss functions, and inference strategies: (1) removing the first shortcut branch ( $E$  muted) from the multi-task training; (2) removing the second shortcut branch ( $V$  muted) from the multi-task training; (3) using  $\mathcal{L}^e$  to replace  $\mathcal{L}^{e2}$ ; (4) using  $\mathcal{L}^{e1}$  to replace  $\mathcal{L}^{e2}$ ; (5) setting  $c_n^r$  to the same constant (tuned in  $\{0.2, 0.4, 0.6, 0.8, 1\}$ ) for all input samples; (6) setting  $c_n^r = \mathbf{JS}[\hat{p}^r || \hat{p}_n^s]$  where  $\mathbf{JS}$  denotes Jensen–Shannon divergence; (7) setting  $c_n^r$  as the euclidean distance between  $\hat{p}^r$  and  $\hat{p}_n^s$ ; (8) removing the *distance* item in Eq. 3.16 and (9) removing  $\hat{p}^r$  and  $\hat{p}_n^s$  in Eq. 3.16.

Compared to the ablative results, we can see that our full approach achieves the overall top performance on SEQA. There is one exception. A higher score on Adv1 is achieved (57.7 vs. 55.7) if we do not use the second shortcut branch ( $V$  muted), *i.e.*, the second ablative. However, this setting achieves much lower performance on Adv3 (42.1 vs. 51.3). This observation indicates that this setting without all the shortcut branches cannot make a good trade-off on different adversarial attacks.

**Ablations on MCQA.** Table 3.9 shows the MCQA results in 10 ablative settings.

Ablative Setting	Test	Adv1	Adv2	Adv3	Adv4
(1) one modified Shct.br.	68.3	63.1	58.0	24.8	56.5
(2) two Shct.br. with $\mathcal{L}^e$	70.1	66.8	61.0	24.6	57.1
(3) two Shct.br. with $\mathcal{L}^{e1}$	70.2	66.7	62.1	25.6	56.5
(4) two Shct.br. with $\mathcal{L}^{e2}$	70.8	66.6	61.8	27.1	62.2
CVC-IV (ours)	69.4	70.0	65.4	28.7	59.9
(5) same $c_n^r$	68.1	69.3	64.4	25.6	59.3
(6) $c_n^r = JS$	70.1	67.0	61.9	20.8	62.2
(7) $c_n^r = Euc$	69.8	67.7	61.9	22.3	60.5
(8) w/o <i>distance</i>	66.1	67.9	65.2	27.8	61.0
(9) w/o $\hat{p}_r$ and $\hat{p}_n$	65.6	66.3	64.8	27.4	59.9
CVC-MV (ours)	68.1	69.1	65.6	26.8	61.0

Table 3.9: The ablation study on MCTest (BERT-base). (1)-(4) are ablative settings for multi-task training (using CVC-IV inference). ‘‘Average’’ means the average performance on Adv\* test sets; (5)-(9) are ablative settings related to CVC-MV inference.

Specifically, we (1) use  $\mathcal{X}_1 = \{Q, O\}$  as the input of the only shortcut branch; (2) use two shortcut branches, where the first one takes  $\mathcal{X}_1 = \{P, O\}$  as input and the second one takes  $\mathcal{X}_2 = \{Q, O\}$  as input, and deploy the  $\mathcal{L}^e$  in Eq. 3.11; (3) use the same two shortcut branches as (2), but deploy the  $\mathcal{L}^{e1}$  in Eq. 3.12; (4) use the same two shortcut branches as (2), but  $\mathcal{L}^{e2}$  in Eq. 3.12 is used; The ablative setting of (5)-(9) on MCQA are the same as those used for SEQQA.

Results on (1)-(4) show that considering the shortcut branch with input  $\{Q, O\}$  is not effective for the robustness of model. The reason is that this shortcut branch is hard to train, *i.e.*, not easy to converge (please refer to ‘‘MCQA-specific’’ and Table 3.2). Our empirical conclusions are as follows. Firstly, the shortcut branch with negligible effect magnitude can be ignored when designing the multi-branch architecture. Secondly, if no prior knowledge of the effect magnitude on each shortcut path (of SCM), using  $\mathcal{L}^{e2}$  is the best choice. Results on (5)-(9) show the efficiency of our proposed *c*-adaptor.

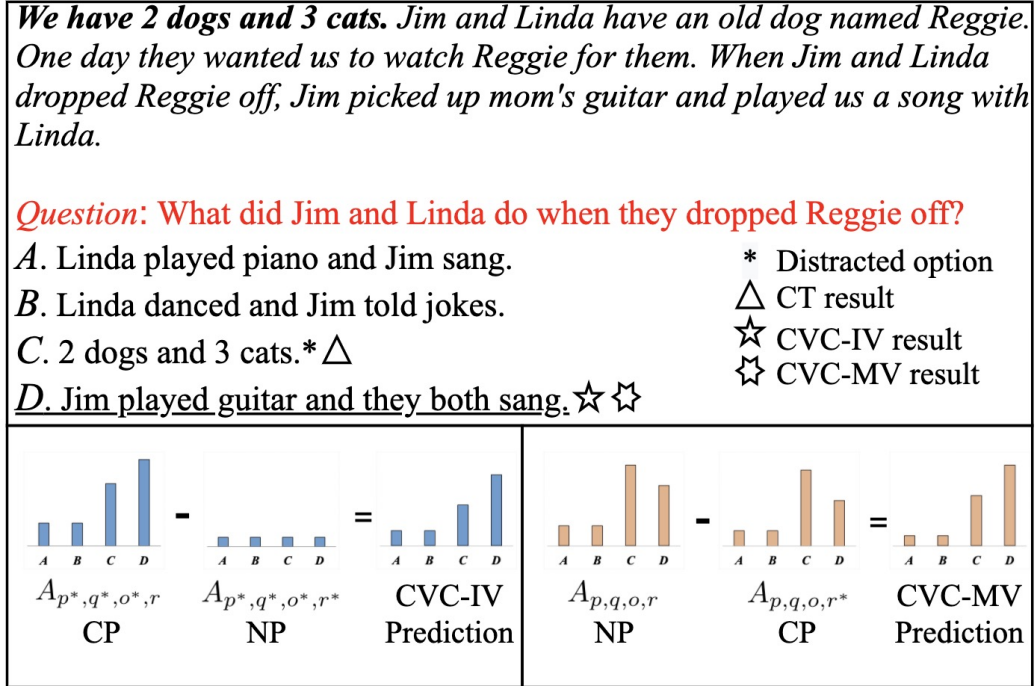


Figure 3.6: A case study of CVC on MCTest trained on official data. The ground truth is underlined.

### Case Studies

We show two examples as case studies to show the interpretability of our approach from two aspects: (1) the disentanglement of robust paths and shortcut in multi-branch architecture, (2) human-like counterfactual inference. Figure 3.6 and Figure 3.7 illustrate two samples from MCQA and SEQA respectively to demonstrate the underlying mechanism of CVC-IV and CVC-MV inference. In Figure 3.6, the conventional training method CT [25] merely aligns the words between *passage* and *options*. This action leads to the wrong choice *C*, which is a confusing choice generated by Adv1. In contrast, both CVC-IV and CVC-MV pick the right answer *D*. On the bottom blocks, we demonstrate the calculation on prediction logits during CVC-IV (Eq. 3.5) and CVC-MV (Eq. 3.6), respectively. We take the CVC-MV as an example to interpret this calculation. Both Normal Prediction (NP)  $A_{p,q,o,r}$  and Counterfactual Prediction (CP)  $A_{p,q,o,r^*}$  contain the logits of *A*, *B*, *C* and *D*. The logit value of *C* is from the word alignment shortcut and it is high in both NP and CP. It thus can be counteracted after the subtraction in CVC-MV. In contrast, the logit

On the other hand, Luther also points out that the Ten Commandments when considered not as God's condemning judgment but as an expression of his eternal will, that is, of the natural law also positively teach how the Christian ought to live. This has traditionally been called the "third use of the law." For Luther, also Christ's life, when understood as an example, **is nothing more than an illustration of the Ten Commandments**, which a Christian should follow in his or her vocations on a daily basis. Luther denied Christ's life a dark story.

*Question: What did Luther consider Christ's life?*

**Ground-truth answer:** illustration of the Ten Commandments

**CT result:** a dark story

**CVC-IV result:** an illustration of the Ten Commandments,

**CVC-MV result:** an illustration of the Ten Commandments,

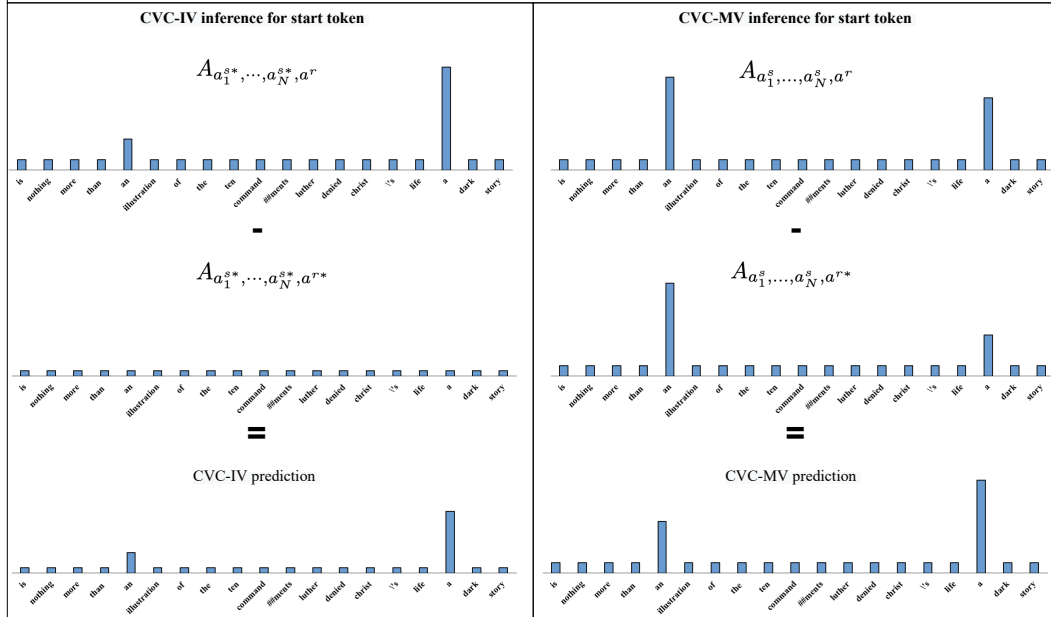


Figure 3.7: A case study of CVC on SQuAD trained on official data. The distracting sentence from AddVerb is underlined. Only bold tokens in passage are shown in bar chart due to limited page size.

value of  $D$  is from the comprehensive *reasoning*. When muting the corresponding variable  $R$  (denoted by  $r^*$  in CP  $A_{p,q,o,r^*}$ ), this value must be reduced. Then it becomes evident after the subtraction in CVC-MV. The sample in Figure 3.7 on SEQA can be interpreted in the same way. The only differences is that the “options” for SEQA are tokens, *e.g.*, which token is the start position for answer span). Note that we normalize the bar chart (the result of the subtraction) for a clear visualization.

## Data Augmentation

Data augmentation with adversarial examples is an intuitive method to improve the model robustness [134, 63]. We conduct experiments on the MCTest dataset to show

		Test	Adv1	Adv2	Adv3	Adv4	AG
<b>Adv1</b>	CT	71.0	70.6	72.1	42.5	60.5	-
	CVC-IV	71.7	73.3	74.9	49.2	63.8	+3.9%
	CVC-MV	71.6	72.9	74.8	48.0	62.7	+3.2%
<b>Adv2</b>	CT	72.3	73.0	75.1	50.1	63.3	-
	CVC-IV	71.8	73.8	76.2	59.8	65.5	+3.5%
	CVC-MV	71.8	74.2	76.6	61.1	65.5	+3.9%
<b>Adv3</b>	CT	67.5	62.7	59.9	70.9	57.1	-
	CVC-IV	67.6	64.5	62.4	70.2	61.6	+2.0%
	CVC-MV	66.8	63.7	62.3	70.3	60.5	+1.5%
<b>Adv4</b>	CT	69.8	65.4	60.2	27.7	63.3	-
	CVC-IV	69.9	66.2	62.4	32.7	61.0	+1.4%
	CVC-MV	67.5	65.6	62.4	25.4	66.7	+0.9%
<b>All</b>	CT	70.5	72.1	74.1	72.5	63.4	-
	CVC-IV	72.7	73.5	76.4	71.9	68.4	+2.0%
	CVC-MV	<b>73.1</b>	<b>74.6</b>	<b>76.6</b>	<b>73.3</b>	<b>73.5</b>	<b>+4.0%</b>

Table 3.10: Accuracies (%) on the MCTest dataset, using different kinds of data augmentation in training with BERT-base. The leftmost column shows which type of adversarial attack for MCQA is used as data enhancement.

the effect of augmentation adversarial data on CT, CVC-IV, and CVC-MV. Specifically, we augment the training data by generating adversarial samples following our adversarial attacks Adv. The results are shown in Table 3.10. Comparing Table 3.10 to the results without data augmentation (Table 3.4), we can observe that models get consistently improved via data augmentation. Comparing the results between CT and CVC, we find that CVC achieves further performance boosts for augmented models. For example, CVC-MV gains an average accuracy increase of 4.0% to “Add All” models when the training data are augmented with all the four kinds of adversarial examples. Note that it is high-cost and time consuming to conduct the data augmentation experiments for SEQA, because the adversarial attacks for SEQA require a lot of human annotations and proofreading.

	Matched Dev	HANS
CT	84.2	62.4
Reweight [17]	83.5	69.2
Bias Product [17]	83.0	67.9
Learned-Mixin [17]	84.3	64.0
Learned-Mixin+H [17]	84.0	66.2
DRiFt-HYPO [49]	84.3	67.1
DRiFt-HAND [49]	81.7	68.7
DRiFt-CBOW [49]	82.1	65.4
Self-debias+Conf-reg [158]	84.5	69.1
Self-debias+Reweight [158]	82.3	69.7
Mind the Trade-off [157]	84.3	70.3
Forgettable <sub>HANS</sub> [177]	84.3	70.4
Forgettable <sub>BoW</sub> [177]	83.4	71.2
Forgettable <sub>BiLSTM</sub> [177]	83.3	71.3
CVC-IV	82.9	70.0
CVC-MV	83.0	<b>71.5</b>

Table 3.11: NLI accuracies (%) on Matched Dev and HANS. Our CVC methods are trained only on the original training data (MNLI) with BERT-base.

### Extension to Natural Language Inference

Our CVC method can also work on other NLP tasks like Natural Language Inference (NLI) task. Following the setting in previous work [17], we train the model on MNLI [171] and evaluate it on an adversarial set, HANS [105]. We use the overlapped tokens in hypothesis and premise as the only bias branch in implementation of CVC. From the results shown in Table 3.11, we observe that CVC-MV outperforms CT by over 9% on the adversarial set, and achieves comparable performance compared to state-of-the-art methods.

## 3.5 Conclusion

We inspect the problem of fragility in QA models, and build the structural causal model to show that the crux is from shortcut correlations. To train robust QA models, we propose a novel CVC approach and implement it on the multi-task training pipeline. We conduct extensive experiments on a variety of QA benchmarks, and

show that our approach can achieve high robustness and good interpretation. Our future work is to enhance the structural causal model by considering the subjective factors, *e.g.*, the preference of dataset annotators and the source of passages.

# Chapter 4

## Interventional Training for Out-Of-Distribution Natural Language Understanding

### 4.1 Introduction

In the previous chapter we concentrate on debiasing for known bias. In this chapter, we turn to a more practical situation of unknown bias. From the era of word embeddings [123] to pre-trained language models [25], researchers of natural language understanding (NLU) have tried to push the performance on benchmark datasets. Traditional settings assume *independent and identical distribution* (IID) in training and testing splits. However, the IID setting cloaks the vulnerability of neural models, *i.e.*, neural models tend to learn non-robust “shortcut” patterns in the training data but fail to make robust predictions on unseen samples. To evaluate the robustness of models, the *out-of-distribution* (OOD) setting draws the attention of the NLU community. For example, the task of natural language inference (NLI) determines whether a hypothesis can be entailed from a premise. We can observe that the lexical overlap between the hypothesis and the premise correlates with the *entailment* label on the benchmark MNLI dataset [171] (as shown in the top part





Figure 4.1: The proportions of entailment and non-entailment samples with different percentages of lexical overlap.

of Figure 4.1). [105] proposed an OOD set named HANS for NLI. As shown in the bottom part of Figure 4.1, HANS does not have the correlation between lexical overlap and the entailment label. NLI models that rely on the lexical overlap heuristic suffer from a significant degradation on HANS [158].

Recently, causal inference has been adopted in NLP to identify robust correlations by analyzing reliable causal effects between variables [188, 110]. From the perspective of causality [120, 119], the crux under a model’s vulnerability is *confounding bias*. We summarize the causal relations behind NLU tasks as a causal graph in Figure 4.2(a).  $X$  represents the input, *e.g.*, a pair of sentences for NLI, and  $Y$  represents a label to be predicted.  $X \rightarrow Y$  represents the desired relation for a robust NLU model, *i.e.*, how to predict the label with reliable understanding of the input.  $X \leftarrow C \rightarrow Y$  denotes a backdoor path of some unreliable relation between  $X$  and  $Y$  confounded by the confounder  $C$ . Examples of  $C$  include nature

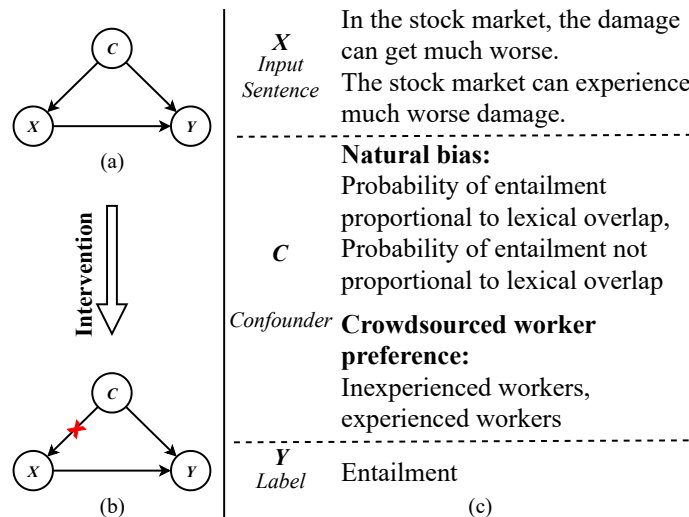


Figure 4.2: (a) Causal graph of NLU tasks, (b) intervention operation, and (c) an example of each node in the causal graph on the NLI task, where the data sample is from MNLI [171].

bias in the dataset [153] or crowdsourced workers preference [36]. For instance, in NLI,  $C$  may represent the degree of lexical overlap between the premise and the hypothesis, which is correlated with the entailment relation in the MNLI dataset (see Figure 4.1).<sup>1</sup> When crowdsourced workers are engaged to create hypotheses for NLI,  $C$  could be the experience level of a worker, with inexperienced workers more likely to write simple sentences with straightforward meanings. As a result, these examples of  $C$  will make  $X$  and  $Y$  spuriously correlated.

A common solution of deconfounding is *intervention* [121, 122], which aims to block the backdoor path (or spurious correlation) by cutting off  $C \rightarrow X$  (see Figure 4.2 (b)). The key idea is to stratify  $X$  into different environments [2, 154], *i.e.*, several subsets of training data, according to the identified confounder. Then the model is expected to make environment-agnostic prediction. By doing so, we are controlling  $X$  and thus break the backdoor path by D-Separation [75]. Figure 4.2(c) depicts an example where the NLI training data is stratified into several environments, *e.g.*, one with obvious trend of lexical overlap bias and another does not. Then the NLI model is trained to fit both environments.

<sup>1</sup>We highlight that the lexical overlap bias is an example for the purpose of illustration and verification only. Our method is designed for situations with unknown confounders.

However, the confounder  $C$  is not always observed. Furthermore, confounders can be multifactorial in NLU, *e.g.*, it may contain both inherent dataset bias and artifacts from crowdsourced workers. Both scenarios make intervention non-trivial. In this chapter, we propose BAI, a bottom-up automatic intervention method, which can (1) identify the unobserved confounder(s) automatically, and (2) perform multi-granular intervention to handle multifactorial confounders. Inspired by [23], the *automatic* stratifying mechanism is realized by maximizing the difference between data in different environments.<sup>2</sup> We further propose a novel *bottom-up* intervention mechanism that aims to address the multifactorial characteristic of  $C$ . While most existing debiasing work only considers a single bias, our bottom-up mechanism enables the model to pick up different confounders in two rounds of interventions. Specifically, based on our preliminary experiments, we find that fine-grained partition (*i.e.*, partition with more environments) results in smaller differences between environments, making environment-agnostic learning easier. Thus we start from a fine-grained partition. We then move on to a coarse-grained partition to further block the backdoor effect via  $C$  and make the learning environment-agnostic.

We apply BAI on three OOD benchmarks for NLU tasks. The results show that our method outperforms state-of-the-art methods, *e.g.*, achieving 7 percentage points of absolute gains from the previous best method under OOD setting of Quora Question Pairs (QQP) [190], a benchmark dataset for paraphrase identification.

**Contributions:** (1) we analyze the issue of NLU vulnerability from the perspective of causality analysis; (2) we propose a bottom-up automatic intervention method to perform intervention for unobserved and multifactorial confounders; and (3) extensive experiments on three OOD benchmarks demonstrate that our method outperforms state-of-the-art methods.

---

<sup>2</sup>Here an environment refers to a subset of training data. A partition is an assignment of the whole training set into multiple environments, *e.g.*, a partition with five environments.

## 4.2 Method

### 4.2.1 Preliminaries

**Causal Intervention** is the core idea of this chapter. We formulate NLU tasks with a causal graph [122], which illustrates the causal relationships between variables with a directed acyclic graph. As shown in Figure 4.2, each node represents a variable, *e.g.*, a pair of sentences or a label for NLU tasks, and each directed edge denotes that the head node has direct effect on the tail node.

Naïve model training, *i.e.*, empirical risk minimization (ERM) [159], indiscriminately learns both spurious correlation  $X \leftarrow C \rightarrow Y$  and causal correlation  $X \rightarrow Y$ . Specifically, by applying Bayes’ rule on Figure 4.2(a), we can obtain:

$$P(Y|X) = \sum_c P(Y|X, c) \underline{P(c|X)}, \quad (4.1)$$

where the bias is introduced via  $P(C|X)$ . For example, consider the NLI task. Let  $X$  be a pair of two sentences (premise and hypothesis) and  $Y$  the entailment label. Let  $C$  represents the degree of lexical overlap between the two sentences in  $X$ , and let  $c_1$  and  $c_2$  denote two situations: having obvious lexical overlap and having little or no lexical overlap. Typically on IID training data of NLI,  $P(c_1|X)$  is larger than  $P(c_2|X)$ , and thus  $P(c_1|X)$  tends to dominate the overall term,  $P(Y|X)$ . In other words, model tends to learn  $P(Y|X)$  from  $c_1$  instead of  $X$ .

In contrast, causal intervention in Figure 4.2(b) yields:

$$P(Y|do(X)) = \sum_c P(Y|X, c) \underline{P(c)}, \quad (4.2)$$

where the  $do(X)$  denotes that intervention is conducted on  $X$ . With  $do$  operation,  $c$  is no longer associated with  $X$  and thus the model treats  $c_1$  and  $c_2$  fairly subject to the prior distribution of  $C$ .

**Invariant Risk Minimization** [2] (IRM) is one of the popular tool for intervention in deep neural networks. Given the stratified environments, IRM targets at a robust model which is invariant to environments. In this work, we utilize two versions of IRM.

Given the input  $X$ , model  $f$  and the partition of environments  $\mathcal{E}$ , the original version of IRM [2] minimizes the objective:

$$\text{IRM}_{v1} = \sum_{e \in \mathcal{E}} \text{XE}(f(X^e), Y) + \lambda \cdot \|\nabla_{\mathbf{w}|\mathbf{w}=\mathbf{1.0}} \text{XE}(\mathbf{w} \cdot f(X^e), Y)\|^2, \quad (4.3)$$

where  $X^e$  denotes the data in the environment of  $e$  and XE denotes cross-entropy loss.  $\mathbf{w}$  is a fixed dummy classifier. The second term measures the optimality of  $\mathbf{w}$  for each environment to encourage the model to make environment-invariant predictions. This version of IRM is unstable due to the second-order derivatives.

Another version of IRM [154] initializes individual classifier  $\mathbf{W}_e$  for each environment  $e$  while all environments share one feature extractor. Here we denote the model for the environment  $e$  as  $f^e = \mathbf{W}_e \circ \Phi$  where  $\Phi$  is a feature extractor, e.g., BERT. The corresponding loss is written as:

$$\text{IRM}_{v2} = \sum_{e \in \mathcal{E}} \text{XE}(f^e(X^e), Y) + \lambda \cdot \text{Var}_{e' \in \mathcal{E}}(\mathbf{W}_{e'}). \quad (4.4)$$

The second term is the variance of classifier weights, which encourages optimal classifiers for different environments to be close to each other.

## 4.2.2 Bottom-up Automatic Intervention

To implement intervention on NLU tasks with the unobserved and multi-factorial confounder, we propose a Bottom-up Automatic Intervention (BAI) method using IRM. Figure 4.3 and Figure 4.4 show the overall pipeline of BAI. It consists of two components: *automatic stratification* and *bottom-up intervention*. The automatic stratification component generates partition of environments by maximizing the difference between data in different environments based on a reference model. The bottom-up intervention component performs intervention at two levels of granularity.

**Automatic Stratification** generates the partition of environments with unobserved confounder. A good partition is achieved when a reference model behaves differently under different environments. Inspired by [23], we first train a reference model

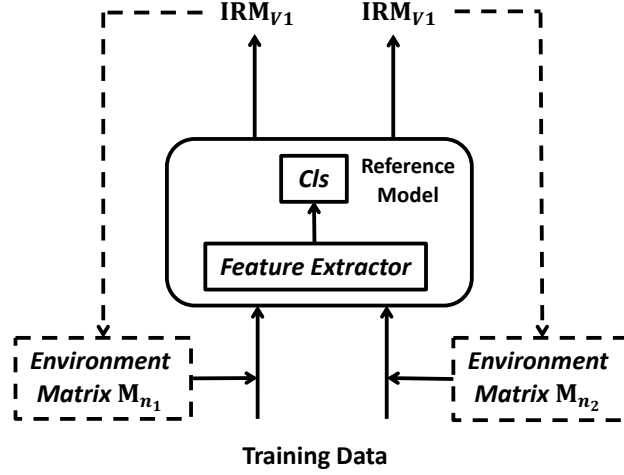


Figure 4.3: First step of BAI: automatic stratifying where  $M_{n_1}$  and  $M_{n_2}$  are optimized individually.

$f_{\text{ref}}$  through the naïve trained BERT [25]<sup>3</sup>. Note the second term of Eq. 4.3 is to make environment-invariant prediction, that is, to minimize the difference of data behavior across environments. Inversely, our goal is to maximize the difference of data behavior by magnifying the second term of IRM.

As shown in Figure 4.3, we initialize an environment matrix  $M \in \mathbb{R}^{D \times N}$  indicating the belonging of each training sample to each environment, where  $D$  and  $N$  denote the number of training data and pre-defined environments, respectively.  $M^{i,j}$  is the probability of  $i$ -th sample belonging to  $j$ -th environment.  $\text{IRM}_{v_2}$  is not applicable since the naïve trained reference model only has one classifier. Thus we derive  $M$  by fixing the reference model  $f_{\text{ref}}$  and maximizing the second term of  $\text{IRM}_{v_1}$  as follows:

$$\max_M \sum_{e \in \mathcal{E}} \|\nabla_{\mathbf{w}|\mathbf{w}=1.0} \text{XE}(\mathbf{w} \cdot f_{\text{ref}}(X^e), Y)\|^2, \quad (4.5)$$

where  $\mathcal{E}$  is the partition of environments determined by  $M$ . Note that max operation makes the back-propagation of gradients from  $M$  infeasible. To address this issue, we deploy the Gumbel Softmax trick [62] to re-formulate the discrete sampling as:

$$\mathcal{E} = g(M) = \text{Gumbel-Softmax}(M). \quad (4.6)$$

<sup>3</sup>Here we expect the reference model to have bias since a biased model is able to recognize bias sample since the bias sample would be predicted easily with high confidence.

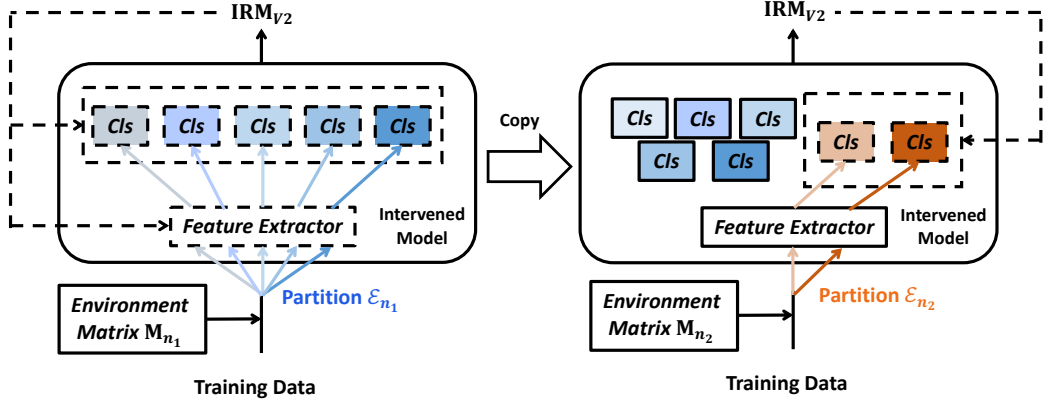


Figure 4.4: Second step of BAI: bottom-up intervention. The dashed arrows denote the back-propagation. Only the modules (or parameter matrices) with dashed box are updated.

We term the environment matrix with  $n$  environments as  $M_n$ . Specifically, we deploy automatic stratifying to extract two environments matrices, *i.e.*, fine-grained  $M_{n_1}$  and coarse-grained  $M_{n_2}$  ( $n_1 > n_2$ ), for bottom-up intervention.

**Bottom-Up Intervention** adopts multi-granular partitions for intervention in a bottom-up fashion, to derive a robust model  $f_{\text{int}}$ . As shown in Figure 4.4, bottom-up intervention consists of two rounds of intervention deployed by  $\text{IRM}_{v2}$  due to its stability and scalability.

We first generate fine-grained partition  $\mathcal{E}_{n_1}$  and coarse-grained partition  $\mathcal{E}_{n_2}$  from  $M_{n_1}$  and  $M_{n_2}$  (see Figure 4.4), where the number of environments in  $\mathcal{E}_{n_1}$  is larger than that in  $\mathcal{E}_{n_2}$ . Second, we start from the fine-grained partition  $\mathcal{E}_{n_1}$  and train the intervened robust model  $f_{\text{int}}$ . Similarly, we decompose  $f_{\text{int}} = \mathbf{W} \circ \Phi$  where  $\Phi$  is feature extractor, *e.g.*, BERT, and  $\mathbf{W}$  is a set of learned classifiers. We use  $\mathbf{W}_e$  to represent the classifier exclusive to environment  $e$  and  $\mathbf{W}\{\mathcal{E}\}$  to denote the set of classifiers for  $\mathcal{E}$  partition, that is,  $\mathbf{W}\{\mathcal{E}_{n_1}\} = \{\mathbf{W}_e \mid e \in \mathcal{E}_{n_1}\}$  represents all classifiers for partition  $\mathcal{E}_{n_1}$ . The feature extractor and the classifiers of  $\mathcal{E}_{n_1}$  in bottom fine-grained intervention are optimized by:

$$\min_{\Phi, \mathbf{W}\{\mathcal{E}_{n_1}\}} \sum_{e \in \mathcal{E}_{n_1}} \text{XE}(f_{\text{int}}^e(X^e), Y) + \lambda \cdot \text{Var}_{e' \in \mathcal{E}_{n_1}}(\mathbf{W}_{e'}), \quad (4.7)$$

Then we conduct the intervention of coarse-grained partition  $\mathcal{E}_{n_2}$ . To prevent the catastrophic forgetting, *i.e.*, the intervention with new partition may make the

model forget the invariant property on previous partition, we incorporate the idea from continual learning [91, 132]. Specifically, we fix the parameter of model  $f_{\text{int}}$  including the feature extractor and  $n_1$  classifiers for  $\mathcal{E}_{n_1}$ . Then we augment  $n_2$  classifiers for the new partition  $\mathcal{E}_{n_2}$ , resulting in  $n_1 + n_2$  classifiers. Here we only optimize the  $n_2$  augmented classifiers during training as:

$$\min_{\mathbf{W}_{\{\mathcal{E}_{n_2}\}}} \sum_{e \in \mathcal{E}_{n_2}} \text{XE}(f_{\text{int}}^e(X^e), Y) + \lambda \cdot \text{Var}_{e' \in \mathcal{E}_{n_1} \cup \mathcal{E}_{n_2}}(\mathbf{W}_{e'}), \quad (4.8)$$

where the first term is based on the new partition  $\mathcal{E}_{n_2}$  while the second term computes the variance of classifier weights across all  $n_1 + n_2$  classifiers.

**Inference** is based on the design of  $\text{IRM}_{v_2}$  [154]. Since we are not able to distinguish which environment the input data belongs to, we simply average the weight of  $n_1 + n_2$  classifiers for inference:

$$\hat{Y} = f_{\text{int}}^{\bar{e}}(X) = \bar{\mathbf{W}} \cdot \Phi(X), \quad (4.9)$$

where  $\bar{\mathbf{W}}$  denotes the mean weight of all classifiers.

## 4.3 Experiment

### 4.3.1 NLU Tasks and Benchmarks

We apply our method on three NLU tasks to evaluate the effectiveness of our method. Specifically, we train on the original training set and evaluate on both the IID and the OOD evaluation sets. The accuracy is reported for all the benchmark datasets.

**Natural Language Inference** aims to classify the relationship between two sentences, *i.e.*, a premise and a hypothesis, into three classes: “entailment”, “contradiction” and “neutral”. It has been observed that NLI models may rely on the lexical overlap bias [105]. We adopt MNLI [171] and HANS [105] as the IID and OOD sets, respectively.



Method	MNLI		FEVER		QQP	
	IID	OOD	IID	OOD	IID	OOD
	Dev	HANS	Dev	Symmetric	Dev	PAWS
Naïve Fine-tuning	84.5	62.4	85.6	63.1	91.0	33.5
Reweighting (KB)	83.5	69.2	84.6	66.5	89.5	50.8
Product-of-Expert (KB)	82.9	67.9	86.5	66.2	88.8	58.1
Learned-Mixin	84.0	64.9	83.1	64.9	86.6	56.8
Regularized-Confidence (KB)	84.5	69.1	86.4	66.2	89.0	36.0
Reweighting (UB)	82.3	69.7	87.1	65.5	85.2	57.4
Product-of-Expert (UB)	81.9	66.8	85.9	65.8	86.1	56.3
Regularized-Confidence (UB)	84.3	67.1	87.6	66.0	89.0	43.0
Forgettable Examples	83.1	70.5	87.1	67.0	89.0	48.8
Self-Debiasing	83.2	71.2	-	-	90.2	46.5
EIIL	83.9	69.9	89.2	68.1	87.9	57.3
BAI (Ours)	82.3	<b>72.7</b>	90.1	<b>69.1</b>	84.2	<b>65.0</b>

Table 4.1: Comparing our method to SOTAs on three benchmarks. Performance shown is in terms of accuracy. “KB” and “UB” denote known bias version and unknown bias version respectively. Results of Naive Fine-tuning, Reweighting, Product-of-Expert, Learned-Mixin and Regularized-Confidence and with known bias are from [37], [158] and [157]. Results of others are from the original paper.

**Fact Verification** also takes in a pair of sentences, *i.e.*, a claim and an evidence, and requires the model to give the position of the evidence towards the claim. The labels are “support”, “refutes”, and “not enough information”. Fact verification models often suffer from the claim-only bias [158]. In this paper, we use FEVER [155] as the IID data and FEVER Symmetric [142] as the OOD data.

**Paraphrase Identification** identifies whether a sentence is paraphrase of another sentence. A sentence pair is labeled as “duplicate” if the two sentences share the same semantic meaning, otherwise “non-duplicate”. Similar to NLI, lexical overlap bias exists in paraphrase identification. We use QQP [164] in training as the IID set and PAWS [190] as the OOD set.

### 4.3.2 Implementation

BERT-base [25] from HuggingFace’s Transformers [173] is deployed as the feature extractor for fair and direct comparison with previous methods. For standard hyperparameters for the training of NLU model, we use the same configu-

ration as [157, 158], *i.e.*, 3 epochs of training, learning rate of  $5e-5$  for NLI and  $2e-5$  for fact verification and paraphrase identification. Unlike previous methods [17, 40, 18, 141, 37] which are directly evaluated on the OOD set, we only perform checkpoint selection on the OOD set. We choose hyperparameters exclusive to our method according to the analysis on the NLI task (see RQ3) and deploy the same configuration for the other two tasks to avoid hyperparameter tuning. Specifically, we set the learning rate to  $1e-2$  for automatic stratification to optimize the environment matrix, and  $n_1 = 5$  and  $n_2 = 2$  for bottom-up intervention. We also fix  $\lambda$  to  $1e2$ . Note the coarse-grained partition may require multiple turns of training to achieve better performance. The average results over 5 runs with different random seeds are reported.

### 4.3.3 Comparison with SOTAs

In this section, we compare our method with the following baselines: **Naïve Fine-tuning** [25] directly fine-tunes the pre-trained language model on the downstream NLU tasks; **Reweighting** [17] reweights each training sample according to the confidence on bias model; **Product-of-Expert** [53] trains the robust model fused with the bias model by sum of logits; **Learned-Mixin** [17] utilizes a different fusion method. **Regularized-Confidence** [157] enhances the model in a knowledge distillation fashion; Unknown bias version methods in [158] adopt the bias model trained only with a small number of data; **Forgettable Examples** [176] trains the model with an additional round with the forgotten data; **Self-Debiasing** [37] utilizes bottom layers of model as the bias model; **EIII** [23] is the IRM method that inspired this paper, which is originally applied to CV.

Table 4.1 summarizes the performance comparison between BAI and the above SOTA methods. Overall, BAI achieves the top performance on all the OOD sets. Specifically, BAI significantly outperforms naïve Fine-tuning by doubling the accuracy on PAWS (65.0% vs. 33.5%), which demonstrates that BAI with causality-

<b>Ablative Setting</b>	<b>Dev</b>	<b>HANS</b>
Naïve FT	84.5	62.4
(a) Randomized Environment	84.0	62.4
(b) w/o Regularizer	83.0	66.8
(c) One Intervention	83.9	69.9
(d) Naive FT+Multiple Classifiers	84.4	62.6
Full Method	82.3	<b>72.7</b>

Table 4.2: **RQ1**. Results of ablative settings on MNLI. “FT” denotes Fine-tuning.

theoretic basis is effective for OOD generalization on NLU tasks. Also, BAI surpasses SOTA methods with 6.9% gains over previous best result on PAWS, which shows the superiority of BAI over reweighting based methods.

We also observe a trade-off between IID and OOD on MNLI and QQP across most of the methods, i.e., performance gains on OOD are achieved with the sacrifice of IID performance. It is because naïve fine-tuning fits IID training data well. Interestingly, the IID test data of FEVER benefits from debiasing methods, which suggests that the data distribution of the IID test data may be different from that of the training data.

### 4.3.4 Ablation Studies

In this section, we conduct extensive ablation studies to evaluate the components in our BAI and answer the following research questions.

**RQ1:** *How does each component of BAI contribute to the performance gains?*

**Answer:** We design four ablative settings: (a) Replacing the learned environment matrix with a randomly initialized one; (b) Removing the regularizer term in Eq. 4.7 and 4.8; (c) Replacing bottom-up intervention with single intervention, *i.e.*, removing Eq. 4.8. (d) Using the same number of classifiers on naïve fine-tuning model as our BAI.

As reported in Table 4.2, the settings (a) and (d) prove that the environment partition is vital in our method and the improvement of our method is not from the added

Stratifying Method	Dev	HANS
No Stratifying	84.5	62.4
(1) Domain Information	84.2	63.2
(2) Confidence	84.0	67.7
(3) Lexical Overlap	83.8	65.6
Automatic Stratifying (Ours)	83.9	<b>69.9</b>

Table 4.3: **RQ2**. Results of alternative methods for environment stratification on MNLI.

parameters<sup>4</sup>. Result of (b) reveals that both the regularizer term and the design of one classifier for one environment contribute to the gains in our method. Finally, the full method with bottom-up intervention outperforms (c), which demonstrates the effectiveness of multi-granular intervention.

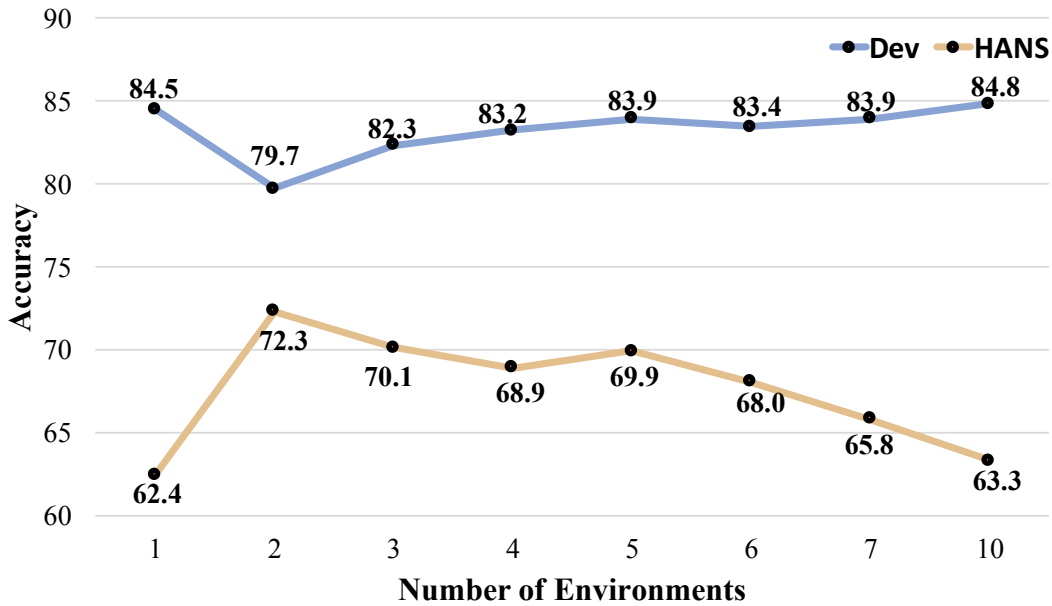


Figure 4.5: **RQ3**. The accuracies of one round of intervention on MNLI with different numbers of environments.

**RQ2:** *Is there any other solution for stratification?*

**Answer:** Yes. We evaluate several alternative methods for partition on MNLI according to the attached information of training samples: (1) Domain information, *i.e.*, “fiction”, “government”, “slate”, “telephone” and “travel”; (2) Confidence of prediction [17]. We calculate the highest confidence or the options and the confi-

<sup>4</sup>BAI introduces 0.008% more parameters compared to that of Naïve Fine-tuning.

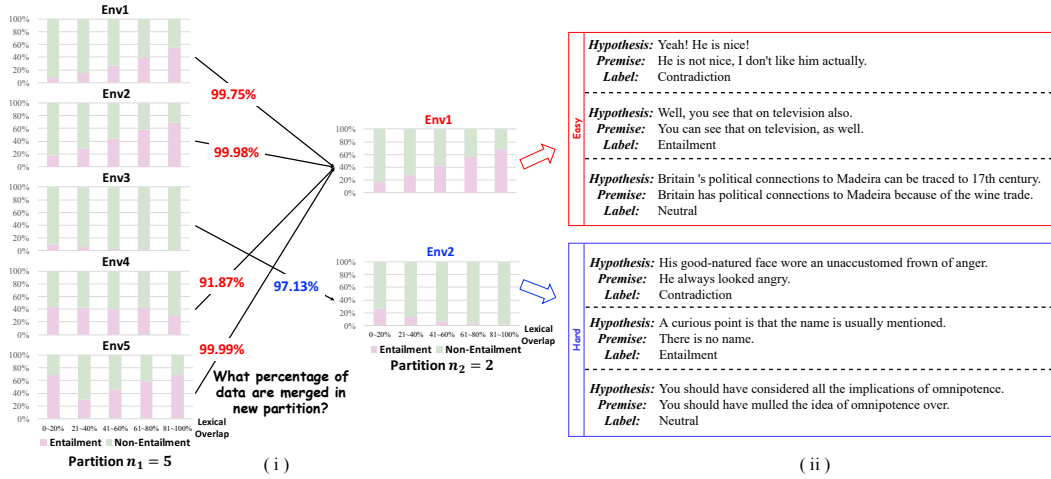


Figure 4.6: **RQ4.** (i) Characteristics and relationship for two partitions. Each sub-graph shows the same analysis setting as in Figure 4.1 in corresponding environment; (ii) Examples for the easy and hard samples for the partition with  $n_2 = 2$ .

dence for the ground-truth label. All the samples are grouped into environments by K-Means [48] according to the two confidence scores; (3) Prior knowledge of bias, *i.e.*, lexical overlap bias in Figure 4.1. We also group them into different environments by K-Means. For fairness, we fix the number of environments as 5, which is the number of domains in the setting (1). We compare the above settings with our model trained using only one intervention in Eq. 4.7.

As summarized in Table 4.3, the results show that directly using domain information as basis for environments stratifying has very few gains, *i.e.*, 0.8%. Although intervention based on domain information is beneficial for every domain, such intervention does not provide a good partition for debiasing as the lexical bias still exists. Stratifying based on confidence and lexical overlap shows considerable improvements compared to that of no stratifying, which demonstrates the two factors are indeed related to the confounder of MNLI. Note that the automatic stratifying method is designed for unobserved confounder, which outperforms the simple heuristics in settings (2) and (3) without using any prior knowledge of bias.

**RQ3:** *How to set the number of environments?*

**Answer:** We first analyze the situation of only one round of intervention and visualize the performance trend in Figure 4.5. Note that setting the number of en-

Order & Combination	Dev	HANS
$\mathcal{E}_2 \rightarrow \mathcal{E}_5$	81.7	70.1
$\mathcal{E}_5 \rightarrow \mathcal{E}_3$	83.7	71.4
$\mathcal{E}_5 \rightarrow \mathcal{E}_3 \rightarrow \mathcal{E}_2$	81.3	73.5
$\mathcal{E}_5 \rightarrow \mathcal{E}_2$ (Config in Table 4.1)	81.1	73.3

Table 4.4: **RQ3.** Results of different orders and combinations of environment numbers on MNLI, arrows represent the intervention order.

vironments as one equals to naïve fine-tuning, *i.e.*, no stratification. Overall, there is a trade-off in the results between Dev and HANS, *i.e.*, IID and OOD performances. This phenomenon is particularly prominent in  $\mathcal{E}_2$ . The reason is that only one intervention forces the model to focus on only one confounder. In this case, it forces the model to pay much attention on the harder samples, *i.e.*, the confounder of crowdsourced worker preference, leading to significant performance drop on dev set (see RQ4 for more details). With the number of environments increasing, the gaps between the environments are also smaller, *i.e.*, the OOD performance of ten environments is close to that of the naïve fine-tuning.

We further analyze the multiple interventions. We conduct experiments with the number of interventions in different orders or combinations. The experiment results are summarized in Table 4.4. We observe that applying the partition with two environments in the final intervention is better and increasing the turns of intervention only brings marginal improvements. Thus, we simply fix  $\mathcal{E}_5 \rightarrow \mathcal{E}_2$  for all tasks in our paper.

**RQ4:** *What is each environment like?*

**Answer:** Figure 4.6 inspects each environment in two partitions, *i.e.*,  $\mathcal{E}_5$  and  $\mathcal{E}_2$ , on MNLI and summarizes the characteristic for each environment.  $\mathcal{E}_2$  can be regarded as a coarse variant of  $\mathcal{E}_5$ , *i.e.*, the first environment of  $\mathcal{E}_2$  partition combines four environments of  $\mathcal{E}_5$ . We can see that both partitions contain environments with distinct characteristics.  $\mathcal{E}_2$  focuses more on crowdsourced worker preference while  $\mathcal{E}_5$  shows each environment with more diverse situation for the nature bias, *i.e.*, lexical overlap bias.

We further investigate the crowdsourced worker preference in  $\mathcal{E}_2$ , *i.e.*, the difficulty of the samples in these two environments is distinguishable. Samples in the second environment are more challenging compared to the first one. As depicted in Figure 4.6 (ii), reasoning of easy samples is straightforward, *i.e.*, `nice` versus `not nice` and `do not like`. In contrast, hard examples require a deep understanding of the semantic meaning. For instance, the hard samples with contradiction and entailment as labels expect the model to have the ability to identify the current situation, *e.g.*, `no name for now`, and the usual situation, *e.g.*, `name is usually mentioned in the past`. The above inspection reveals that BAI helps to generate meaningful and multifactorial partition.

## 4.4 Conclusion

In this chapter, we explore how to improve the robustness of NLU models under OOD setting, and propose a bottom-up automatic intervention method for debiasing. The experiment results demonstrate the superiority of our model over state-of-the-art methods. In future work, we will consider two improvements on BAI. First, we target at an end-to-end framework for intervention and dynamic learn the partition of environment for NLU tasks. Second, we want to ease the trade-off effect between IID and OOD sets.

## **Part II**

# **Cross-lingual Transfer**



# Chapter 5

## COSY: COunterfactual SYntax for Cross-Lingual Understanding

### 5.1 Introduction

With the emergence of BERT [25], large-scale pre-trained language models have become an indispensable component in the solutions to many natural language processing (NLP) tasks. Recently, large-scale multilingual transformer-based models, such as mBERT [25], XLM [80] and XLM-R [21], have been widely deployed as backbones in cross-lingual NLP tasks [175, 125, 72]. However, these models trained on a single resource-rich language, *e.g.*, English, all suffer from a large drop of performance when tested on different target languages, *e.g.*, Chinese and German—where the setting is called *zero-shot cross-lingual transfer*. For example, on the XQUAD dataset, mBERT achieves a 24 percentage points lower exact match score on the target language Chinese than on the training language English [57]. This indicates that this model has seriously overfitted English.

An intuitive way to tackle this is to introduce language-agnostic information—the most transferable feature across languages, which is lacking in existing multilingual language models [16]. In our work, we propose to exploit reliable language-agnostic information—syntax in the form of universal dependency relations and

English: I bought two new laptops yesterday .  
 Chinese: 我 昨天 买了 两台 新的 电脑 。

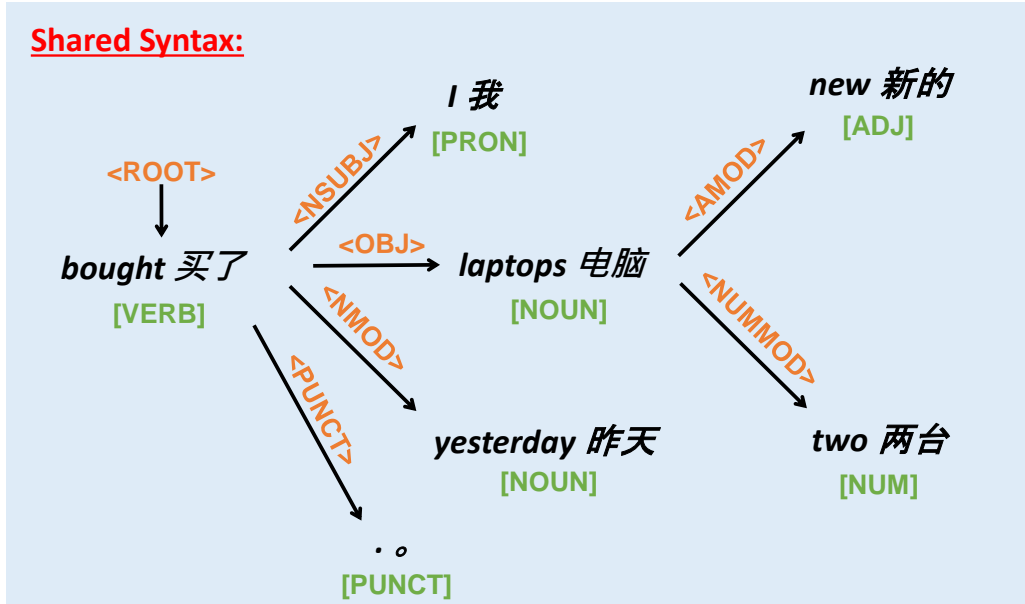


Figure 5.1: Examples of two sentences in English and Chinese that have the same meaning and share the same syntax in the format of **dependency relations** and **POS tags**.

universal POS tags [24, 112, 194, 193]. As illustrated in Figure 5.1, the sentences in Chinese and English share the same meaning but have different word orders. The order difference hampers the transferability between English and Chinese in conventional language models (with sequential words as input). In contrast, it is clear from Figure 5.1 that the two sentences share identical dependency relations and POS tags. Thus, we can incorporate such universal syntax<sup>1</sup> information to enhance the transferability across different languages. To achieve this learning objective in deep models, we design syntax-aware networks that incorporate the encodings of dependency relations and POS tags into the encoding of semantics.

However, we find that empirically the conventional attention-based incorporation of syntax, *e.g.*, relational graph attention networks [60], has little effect on improving the model. One possible reason is that the learning process may be dom-

<sup>1</sup>In the rest of this chapter, syntax denotes universal syntax for simplicity.

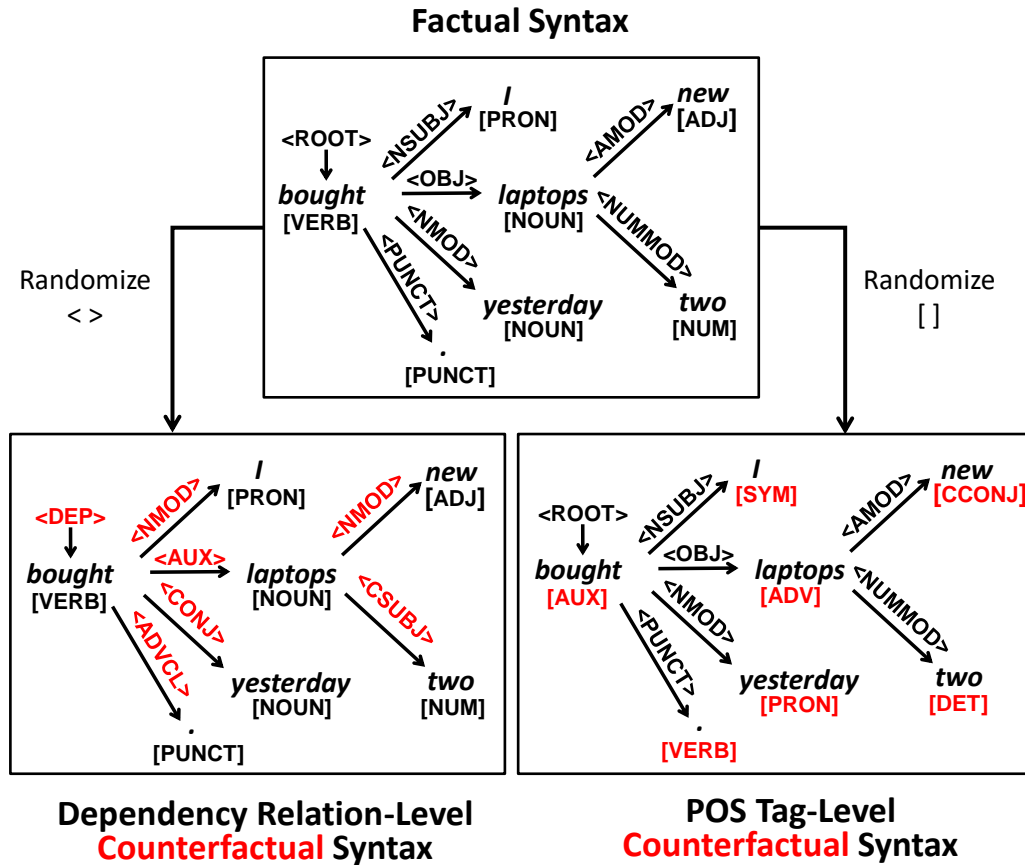


Figure 5.2: Illustration of counterfactual syntax generation. Red color highlights the modified syntax with randomized labels.

inated by the pre-trained language models due to their strength in semantic representation learning, which leads to an overfitted model. This raises the question of *how to induce the model to focus more on syntax while maintaining its original capability of representing semantics?* To this end, we propose a novel COunterfactual SYntax (COSY) method, inspired by causal inference [138, 120] and contrastive learning [50].

The intuition behind COSY is to create copies of training instances with their syntactic features altered (see the “counterfactual” syntax in Figure 5.2), and to force the encodings of the counterfactual instances to be different from the encodings of their corresponding factual instances. In this way, the model would learn to put more emphasis on the syntactic information when learning how to encode an instance, and such encodings are likely to perform well across languages. We evaluate

our COSY method on both question answering (QA) and natural language inference (NLI) under cross-lingual settings. Experimental results show that, without using any additional data, COSY is superior to the state-of-the-art methods.

**Contributions:** 1) we develop a syntax-aware network that incorporates transferable syntax in language models; 2) we propose a novel counterfactual training method that addresses the technical challenge of emphasizing syntax; and 3) extensive experiments on three benchmarks demonstrate the effectiveness of our method for cross-lingual tasks.

## 5.2 COSY: COunterfactual SYntax

COSY aims to leverage the syntactic information, *e.g.*, dependency relations and POS tags, to increase the transferability of cross-lingual language models. Specifically, COSY implicitly forces the networks to learn to encode the input not only based on semantic features but also based on syntactic features through syntax-aware networks and a counterfactual training method.

As illustrated in Figure 5.3, COSY consists of three branches with each branch based on syntax-aware networks (SAN) indicated by a distinct color. The main branch (in black) is the factual branch that uses factual syntax as input. The **red** and **blue** branches are counterfactual branches using counterfactual dependency relations and counterfactual POS tags as input, respectively. The counterfactual training method guides the black branch to put more emphasis on syntactic information with the help of other two branches. Note that the **red** and **blue** branches work for counterfactual training, and only the prediction from the black branch is used in testing.

Below, we first elaborate the modules of SAN in Section 5.2.1, and then introduce the counterfactual training method in Section 5.2.2.

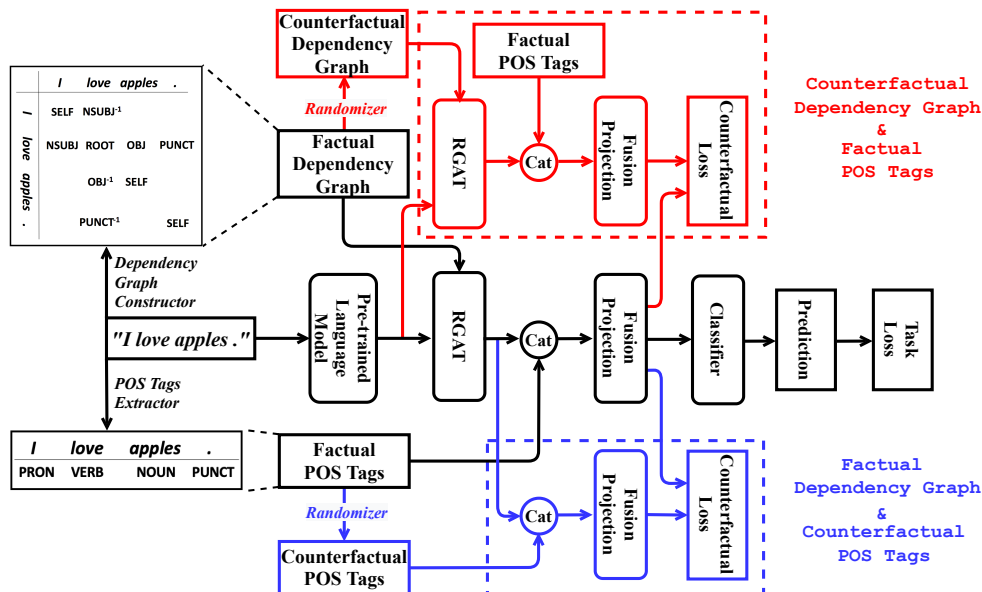


Figure 5.3: The overall pipeline of our COSY. We call the architecture as syntax-aware networks (Section 5.2.1) and the training method as counterfactual training (Section 5.2.2). In this architecture, there are three branches: black, red and blue. Black branch is just the normal attention-based network with additional syntactic information, and only its prediction is used in the testing stage. Red branch and blue branch are novel as they generate the counterfactual syntax samples and drive the counterfactual losses in the training stage—the key functions in COSY. RGAT stands for Relational Graph Attention Network [60, 95]. **The modules of RGAT and the modules of Fusion Projection are shared across branches, e.g., two RGAT modules are sharing parameters.** Cat denotes concatenation.

### 5.2.1 Syntax-Aware Networks (SAN)

As shown in Figure 5.3, SAN contains four major modules: a set of feature extractors, a relational graph attention network (RGAT), fusion projection, and a classifier. In this section, we use the route in the black branch as an example to elaborate each module. The set of feature extractors include three components: a pre-trained language model, a dependency graph constructor and a POS tags extractor.

**Pre-trained Language Model.** Following previous work [57], we deploy a pre-trained multi-lingual language model, e.g., mBERT [25], to encode each input sentence into contextual features. Given a sequence of tokens with a length of  $S$ , we denote the derived contextual features as  $\mathbf{H} = [\mathbf{h}_1, \dots, \mathbf{h}_S] \in \mathbb{R}^{S \times d}$ , where  $d$  is the dimensionality of each hidden vector.

**Dependency Graph Constructor.** We use it to construct the (factual) dependency

graph for each input sentence. In this work, the Stanza toolkit [127] is used to extract the universal dependency relations as the first step. Then, the dependency graph can be represented as  $G = \{V, R, E\}$ , where the nodes  $V$  are tokens, the edges  $E$  denote the existence of dependency relations, and the set  $R$  contains the relation types for  $E$ . Each edge  $e_{ij} \in E$  consists of a triplet  $(v_i, v_j, r)$  where  $v_i, v_j \in V$  and  $r \in R$ .

As shown in Figure 5.3, we define three kinds of relation types in  $R$ : 1) a forward syntactic relation, e.g., love  $\xrightarrow{\text{OBJ}}$  apples; 2) an inverse syntactic relation, e.g., apples  $\xrightarrow{\text{OBJ}^{-1}}$  love; and 3) a self loop SELF that allows the information to flow from a node to itself. Note that we regard the ROOT relation as a self-loop. In this way, we obtain 75 different types of relations in total, and thus denote the embedding matrix as  $\mathbf{R} \in \mathbb{R}^{75 \times d'}$ .

**POS Tags Extractor.** We deploy the same Stanza toolkit [127] to assign (factual) POS tags  $P$  for all tokens. We obtain 17 different types of POS tags and denote the embedding matrix as  $\mathbf{T} \in \mathbb{R}^{17 \times d'}$ .

**Relational Graph Attention Networks (RGAT).** RGAT is one of the standard backbones to incorporate the dependency graph [60, 95]. Given the (factual) dependency graph  $G$  with the contextual features of each node, RGAT can generate the relation-aware features (for each node). Details are given below. Suppose  $e_{ij}$  is the directed edge from node  $v_i$  to node  $v_j$  and the dependency relation  $r$ . The importance score of  $v_j$  from  $v_i$  is computed as:

$$s(v_i, v_j) = \text{Concat}(\mathbf{e}_{ij}^s, \mathbf{e}_{ij}^r) \cdot \mathbf{W}_{Attm}, \quad (5.1)$$

where  $\mathbf{W}_{Attm} \in \mathbb{R}^{(d/2+d') \times 1}$  maps a vector to a scalar,  $\mathbf{e}_{ij}^r$  is the embedding of the dependency relation between  $v_i$  and  $v_j$  from  $\mathbf{R}$ , and  $\mathbf{e}_{ij}^s$  is computed by element-wise multiplication between  $v_i$  and  $v_j$ :

$$\mathbf{e}_{ij}^s = (\mathbf{h}_i \cdot \mathbf{W}_Q) \circ (\mathbf{h}_j \cdot \mathbf{W}_K), \quad (5.2)$$

where  $\mathbf{W}_K \in \mathbb{R}^{d \times d/2}$  and  $\mathbf{W}_Q \in \mathbb{R}^{d \times d/2}$  are the learnable parameters for key and query projections [160], and  $\mathbf{h}_i$  and  $\mathbf{h}_j$  denote their contextual features extracted

from pre-trained language models. Then, the importance scores are normalized across  $\mathcal{N}_j$  to obtain the attention score of  $v_j$  from  $v_i$ :

$$\alpha(v_i, v_j) = \frac{\exp(s(v_i, v_j))}{\sum_{k \in \mathcal{N}_j} \exp(s(v_k, v_j))}, \quad (5.3)$$

where  $\mathcal{N}_j$  denotes the set of nodes pointing to  $v_j$ . The relation-aware features of  $v_j$  is computed as the weighted sum of all nodes in  $\mathcal{N}_j$  with corresponding attention scores. After computing all nodes, we get the relation-aware features  $\hat{\mathbf{H}} = [\hat{\mathbf{h}}_1, \dots, \hat{\mathbf{h}}_S] \in \mathbb{R}^{S \times d}$ .

**Fusion Projection.** We fuse the relation-aware features  $\hat{\mathbf{H}}$  with the (factual) POS tags information before feeding them into the classifier. Given POS tags  $P$ , the fused features for each token are represented by

$$\mathbf{f}_j = \text{Concat}(\hat{\mathbf{h}}_j, \mathbf{p}_j) \cdot \mathbf{W}_F, \quad (5.4)$$

where  $\mathbf{W}_F \in \mathbb{R}^{(d+d') \times d}$  are learnable parameters of fusion projection and  $\mathbf{p}_j$  is the corresponding embedding of the POS tag of the  $j$ -th token from  $\mathbf{T}$ . The fused features of the entire sequence are denoted as  $\mathbf{F} = [\mathbf{f}_1, \dots, \mathbf{f}_S] \in \mathbb{R}^{S \times d}$ .

**Classifier.** It is designed based on the specific task, such as NLI or QA, following [25].

## 5.2.2 Counterfactual Training

Recall that the challenge in the effective utilization of syntax is how to induce the model to focus more on syntax while maintaining its original representation capability of semantics. Inspired by counterfactual analysis [120, 119, 122] and contrastive learning [43], we propose a counterfactual training method by incorporating counterfactual syntax (counterfactual dependency graph and counterfactual POS tags) on the red and blue branches in Figure 5.3. Each branch is designed to guide the model to focus on one type of syntax, *i.e.*, dependency graph or POS tags.

**Counterfactual Dependency Graph** is utilized on the **red** branch with factual POS tags in Figure 5.3. We build a counterfactual dependency graph by maintaining

graph structure and nodes, and replacing each type of relation (except for a self-loop SELF) with a randomized (counterfactual) type. We name it  $G^-$ . We feed  $G^-$  and  $\mathbf{H}$  into RGAT to obtain the counterfactual relation-aware features denoted as  $\hat{\mathbf{H}}^-$ . Then, we fuse  $\hat{\mathbf{H}}^-$  with the factual POS tags to derive the counterfactual features  $\mathbf{F}^{cf1} = [\mathbf{f}_1^{cf1}, \dots, \mathbf{f}_S^{cf1}]$  on the red branch. Finally, we can calculate the similarity between the factual and the counterfactual features, by leveraging the dot-product operation, as follows,

$$\mathcal{L}_{cf1} = \frac{1}{S} \sum_i^S \mathbf{f}_i \cdot \mathbf{f}_i^{cf1}. \quad (5.5)$$

This counterfactual loss forces the model to emphasize the syntactic information related to dependency relations.

**Counterfactual POS Tags** are utilized with the factual dependency graph on the [blue](#) branch in Figure 5.3. We create counterfactual POS tags  $P^-$  from factual POS tags  $P$  by randomly selecting a POS tag for each token. Accordingly, we replace each embedding  $\mathbf{p}_i$  by  $\mathbf{p}_i^-$ . Given the relation-aware features  $\hat{\mathbf{H}}$  from the black branch, we then feed the embeddings of counterfactual POS tags in Eq. 5.4 and get the counterfactual features as  $\mathbf{F}^{cf2} = [\mathbf{f}_1^{cf2}, \dots, \mathbf{f}_S^{cf2}]$ . Finally, we can calculate the similarity between the factual and the counterfactual features (on the blue branch) by leveraging the dot-product operation, as follows,

$$\mathcal{L}_{cf2} = \frac{1}{S} \sum_i^S \mathbf{f}_i \cdot \mathbf{f}_i^{cf2}. \quad (5.6)$$

This counterfactual loss forces the model to emphasize the syntactic information related to POS tags. The overall loss function used in training is as follows,

$$\mathcal{L} = \mathcal{L}_{task} + \lambda(\mathcal{L}_{cf1} + \mathcal{L}_{cf2}), \quad (5.7)$$

where  $\mathcal{L}_{task}$  is the task-specific loss, *i.e.*, a cross-entropy loss, and  $\lambda$  is a scale to balance between the task-specific loss and our proposed counterfactual losses.



	Method	#T	#M	A.D.	XNLI		MLQA		XQUAD	
					en.	avg.	en.	avg.	en.	avg.
mBERT	Naive F.T.	1	1	No	82.1	68.4	67.0 / 80.2	44.2 / 61.4	72.2 / 83.5	51.0 / 66.7
	XMAML-One	$L$	$O(L)$	Yes	82.1	69.6	-	-	-	-
	LAKM	1	1	Yes	-	-	66.8 / 80.0	-	-	-
	COSY (Ours)	1	1	No	82.2	<b>70.1</b>	67.2 / 80.4	<b>45.2 / 62.1</b>	72.6 / 83.6	<b>53.2 / 68.1</b>
X-R <sub>base</sub>	Naive F.T.	1	1	No	84.6	75.1	- / 80.1	- / 65.1	71.6 / 83.1	55.9 / 71.8
	XMAML-One	$L$	$O(L)$	Yes	-	-	- / 80.2	- / 66.1	-	-
	COSY (Ours)	1	1	No	84.3	<b>75.6</b>	67.7 / 80.7	<b>48.5 / 66.5</b>	74.0 / 85.1	<b>57.3 / 73.4</b>
X-R <sub>large</sub>	Naive F.T.	1	1	No	88.7	80.0	70.6 / 83.5	53.2 / 71.6	75.7 / 86.5	60.6 / 76.8
	STILT	9	1	Yes	89.6	81.6	70.8 / 84.1	54.4 / 72.8	77.4 / 88.3	63.3 / 78.7
	XMAML-One	$L$	$O(L)$	Yes	-	-	- / 84.3	- / <b>73.2</b>	-	-
	COSY (Ours)	1	1	No	89.2	<b>81.9</b>	70.9 / 84.2	<b>54.7 / 73.2</b>	77.7 / 88.0	<b>64.0 / 79.7</b>

Table 5.1: Cross-lingual **zero-shot** performance comparison between COSY and SOTA methods on three benchmark datasets. Note that we report accuracy for XNLI and Exact Match/F1 scores for MLQA and XQUAD. For each dataset, “en.” denotes the results of English while “avg.” is the average performance over all languages. X-R means XLM-R and Naive F.T. is the abbr. of Naive Fine-Tuning.  $L$  is the number of target languages. **#T** denotes the number of training turns, *e.g.*, STILT augments its training by using each of nine additional datasets. **#M** is the number of final models, where  $1 < O(L) < L$ , and **A.D.** denotes using additional datasets.

## 5.3 Experiments

In this section, we evaluate our COSY method for cross-lingual understanding under both zero-shot and few-shot settings. For the zero-shot setting, we use English for training and evaluate the model on different target languages. For the few-shot setting, we follow the implementation in [113] and use the development set of the target languages for model fine-tuning<sup>2</sup>.

### 5.3.1 Datasets

We evaluate our method on the natural language inference (NLI) and the question answering (QA) tasks. We briefly introduce the datasets used in our experiments as follows.

**Natural Language Inference (NLI).** Given two sentences, NLI asks for the relationship between the two sentences, which can be entailment, contradiction or

<sup>2</sup>All the results and analyses are under the zero-shot settings by default, except for Table 5.2.

Method	en.	non-en.	avg.	avg.
Naive F.T.*	81.9	70.3	71.2	
XMAML-One*	82.4	70.7	71.6	
COSY (Ours)	82.6	<b>71.9</b>	<b>72.7</b>	

Table 5.2: Results of XNLI under the **few-shot** setting (mBERT). We report the testing results of English (“en.”), the average results over all non-English languages (“non-en. avg.”) and the average results over all languages (“avg.”). \* denotes the results from [113]. More details are available in Appendix.

neutral. We conduct experiments on XNLI [22] and evaluate our method on 13 target languages<sup>3</sup>.

**Question Answering (QA).** In this chapter, we consider the QA task that asks the model to locate the answer from a passage given a question. We conduct experiments on MLQA [87] and XQUAD [6]. COSY is evaluated on 7 languages on MLQA and 10 languages on XQUAD (with Thai excluded).

### 5.3.2 Implementation

In data preprocessing, we feed the same syntactic information to each of the sub-words in the same word after tokenization. Our implementation of pre-trained language models (mBERT and XLM-R) is based on HuggingFaces’s Transformers [173]. We select the checkpoint and set hyper-parameters, *e.g.*, learning rate and  $\lambda$  in the loss function, based on the performance on the corresponding development sets. We select learning rate amongst  $\{7.5e-6, 1e-5, 3e-5\}$  and fix the batch size to 32. We select dimension  $d'$  amongst  $\{100, 300\}$ .  $\lambda$  in counterfactual loss is set to 0.1 (see Figure 5.4). A linear warm up strategy for learning rate is adopted with first 10% optimization steps. Adam [74] is adopted as the optimizer. All experiments are conducted on a workstation with dual NVIDIA V100 32GB GPUs.

<sup>3</sup>We remove Thai (th) and Swahili (sw) from our experiments since these two languages are not supported by Stanza.

### 5.3.3 Results

We compare our method with naive fine-tuning and the state-of-the-art methods. The overall results on three benchmarks are presented in Table 5.1 (zero-shot) and Table 5.2 (few-shot).

**Comparison with Naive Fine-tuning.** Naive Fine-tuning [175, 93, 57] is to directly fine-tune the pre-trained language model on downstream tasks as in [25]. From Table 5.1 and Table 5.2, we can observe that COSY consistently outperforms the naive fine-tuning method on all datasets, *e.g.*, by average 1.9 percentage points (accuracy) and 2.9 percentage points (F1) on XNLI and XQUAD with XLM-R<sub>large</sub> in the zero-shot setting. These observations demonstrate the effectiveness of COSY and suggest that universal syntax as language-agnostic features can enhance the transferability for cross-lingual understanding. Furthermore, the results show that COSY is able to work with different backbones and thus is model-agnostic.

**Comparison with the State of the Art.** We first outline the SOTA zero-shot (few-shot) cross-lingual methods we compared with as follows: (1) XMAML-one [113] borrows the idea from meta-learning. Specifically, XMAML-one utilizes an auxiliary language development data in training, *e.g.*, using the development set of Spanish in training to assist German on MLQA. XMAML-One reports the results based on the most beneficial auxiliary language. (2) STILT [124] augments intermediate task training before fine-tuning on the target task, *e.g.*, adding training of HellaSwag [186] before training on the NLI task. STILT also reports results with the most beneficial intermediate task. (3) LAKM [184] first mines knowledge phrases along with passages from the Web. Then these Web data are used to enhance the phrase boundaries through a masked language model objective. Note that LAKM is only evaluated on three languages of MLQA.

On the one hand, we observe that COSY surpasses the compared SOTA methods over all evaluation metrics. Although meta-learning methods [32, 41, 151] advance the state-of-the-art performance for few-shot learning, our COSY still outperforms

Ablative Setting	MLQA		XQUAD		XNLI
	EM	F1	EM	F1	Acc
Naive F.T.	44.2	61.4	51.0	66.7	68.4
(1) SAN-Black	44.3	61.4	51.6	66.9	68.7
(2) SAN-Black+Gate	44.5	61.5	51.9	67.1	68.7
(3) SAN-Black, Red	44.9	61.7	52.8	67.8	69.9
(4) SAN-Black, Blue	44.7	61.8	52.2	67.4	69.7
(5) COSY	<b>45.2</b>	<b>62.1</b>	<b>53.2</b>	<b>68.1</b>	<b>70.1</b>

Table 5.3: The ablation study on MLQA, XQUAD and XNLI (mBERT). We report the average performance of all languages on the test set.

the meta-learning-based method, *i.e.*, XMAML-One, with 1.1 percentage points in the few-shot setting. On the other hand, the superiority of COSY is also reflected in other aspects, which are shown in Table 5.1. Specifically, COSY does not require additional datasets and cumbersome data selection process, which is more convenient and resources saving.

### 5.3.4 Discussion and Analysis

**Ablation Study.** In Table 5.3, we show the MLQA, XQUAD and XNLI results in 4 ablative settings, to evaluate the approach when we (1) only utilize the SAN-Black branch; (2) utilize the SAN-Black branch with an intuitive gate mechanism to control the information of pre-trained language model and syntax; (3) utilize the SAN-Black branch and SAN-Red branch; (4) utilize the SAN-Black branch and SAN-Blue branch.

Compared to the ablative results, we can see that our full method achieves the overall top performance in all settings. Syntax features are incorporated into the models in (1)-(5) and all of them outperform the naive fine-tuning method, which demonstrates the effectiveness of universal syntax. By analyzing the settings one by one, we can observe that SAN-Black only attains limited improvement compared to naive fine-tuning since syntax is incorporated in the model by overlooked. Gate mechanism (2) fails to solve the overlooking issue. Both of (3) and (4) with

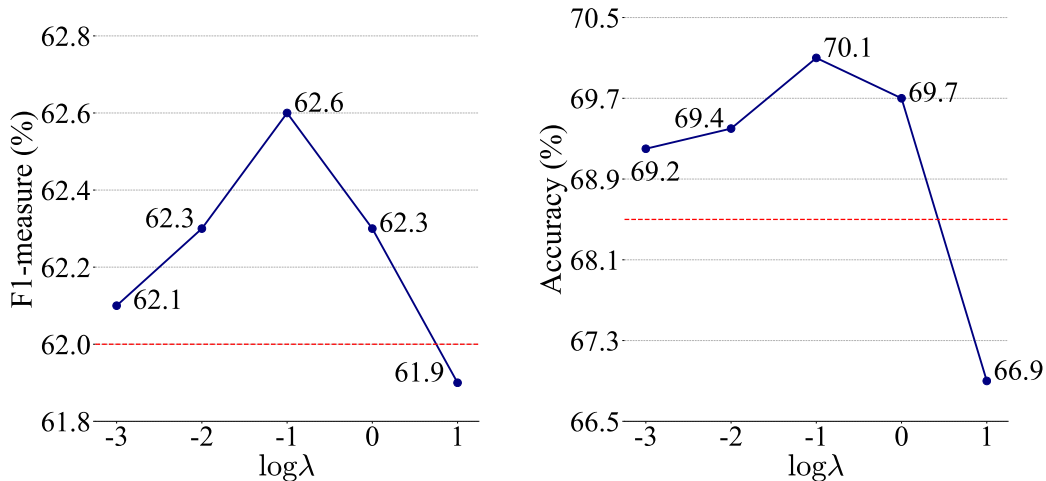


Figure 5.4: Left: average F1-measure (%) on target languages on MLQA development set (mBERT). Right: average accuracy (%) on target languages on XNLI development set (mBERT). Red dotted line denotes the model performance of using naive fine-tuning.

counterfactual training are able to bring gains compared to (1), and the results indicate that dependency relations are more effective compared to POS labels. We also observe that our full method (5) does not accumulate the gains from (3) and (4). One explanation could be that part of the information provided by the dependency relations and POS labels overlaps. For instance, if we see an edge of relation,  $\text{word}_a \xrightarrow{\text{AMOD}} \text{word}_b$ , we may infer that  $\text{word}_a$  is NOUN and  $\text{word}_b$  is ADJ.

**Effect of  $\lambda$ .** We now study the impact of the scale value  $\lambda$  with counterfactual losses. For clarity, we show the results with different values of  $\log \lambda$  in Figure 5.4. We can observe that COSY attains the highest results when  $\lambda = 0.1$  on both MLQA and XNLI. As the value drops, the effect of counterfactual loss is also smaller and the performance is getting closer to that from naive fine-tuning (red dotted line). If a large value of  $\lambda$  is applied, *e.g.*,  $\lambda = 1$ , the model begins to over-emphasize the syntax and semantics are overlooked, which leads to significant decrease on performance.

**Effect of COSY.** In this part, we first study whether counterfactual training method indeed guides the model to focus more on syntactic information. We conduct analysis on the COSY and SAN-Black. Since it is non-trivial to measure the utilization

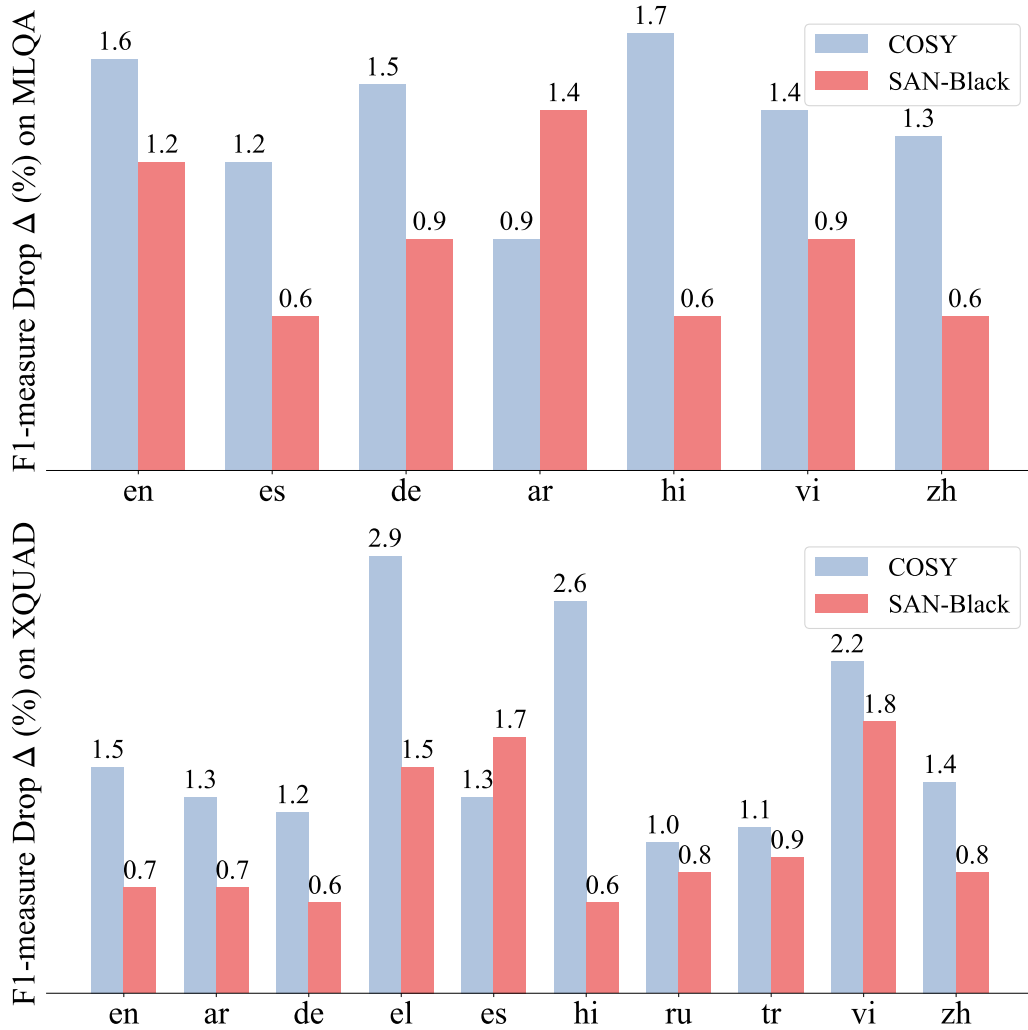


Figure 5.5: F1-measure drop  $\Delta$  (%) with a standard normal distribution perturbation on MLQA and XQUAD (mBERT). Two colors denote COSY and SAN-Black.

of syntax in a straightforward way, we adopt a standard way to measure the importance of the neurons in deep models [67]. Specifically, we perturb the syntactic features with a Gaussian noise to test data and check whether our model would be more easily affected by the syntax perturbation. If so, then it verifies that our model indeed relies more on syntax. The results are shown in Figure 5.5. We can discover that the performance drop of COSY is larger compared to that with SAN-Black.

Meanwhile, we also explore whether COSY is beneficial for yielding more meaningful syntax embedding than SAN-Black. Specifically, we compute the correlation score (absolute cosine similarity) between the embedding of syntactic relation and the corresponding inverse relation from the same type. For COSY, we observe

	MLQA		XQUAD	
	EM	F1	EM	F1
(1)	44.8	61.7	52.2	67.3
(2)	45.1	62.0	53.1	68.1
(3)	44.9	61.9	52.7	67.8
(4)	45.0	62.0	53.2	68.0
Current	45.2	62.1	53.2	68.1

Table 5.4: Results of different generation ways for generating counterfactual syntax with mBERT as backbone. “Current” means the current generation way described in Section 5.2. We report the average performance of all languages.

that the score of the related types are  $42.4\times$  larger than that of two randomly selected embeddings (average over 10000 times). However, for SAN-Black, its score is only  $1.4\times$  larger than that of two randomly selected embeddings. It demonstrates that COSY attains more meaningful syntax representations than SAN-Black.

**Counterfactual Syntax Generation.** Here we analyze other alternative ways of counterfactual syntax generation. Specifically, we design the following variants and report the results in Table 5.4: (1) we not only replace edge types, but also replace connections for counterfactual dependency graph construction; (2) for each input sequence, we create 5 counterfactual dependency graphs, 5 sets of counterfactual POS tags, and the counterfactual loss is the average over the 5 sets; (3) we replace the factual syntax with a fixed type, *e.g.*, a type of padding instead of a random type from all types; (4) in each generating process, we only replace 50% of the factual syntax.

Comparing (1) with the result of “SAN-Black,Blue” in Table 5.3, we can see that (1) does not work. We believe that randomly changing connections in  $G^-$ , *e.g.*, an edge is created from the first token to the last token in a long passage, may have a significant effect to  $\hat{H}^-$ , it is undesirable for further optimization of counterfactual loss. Results from (2) and (4) suggest that the number of the generated counterfactual syntax and ratio of randomizing do not play an important role in COSY. It is also discovered that randomizing with all types is better than simple replacement with a fixed type.

## 5.4 Conclusion

We study how to effectively plug in syntactic information for cross-lingual understanding. Specifically, we propose a novel counterfactual-syntax-based approach to emphasize the importance of syntax in cross-lingual models. We conduct extensive experiments on three cross-lingual benchmarks, and show that our approach can outperform the SOTA methods without additional dataset. For future work, we will combine our approach with other orthogonal methods, *e.g.*, meta-learning, to further improve its effectiveness.



# Chapter 6

## Translate-Train Embracing

## Translationese Artifacts

### 6.1 Introduction

Cross-lingual transfer has drawn wide attention in recent years [57, 93]. It aims to reuse NLP models trained on a *source* language for the task of a *target* language. The most intuitive method is transfer learning, *i.e.*, leveraging pre-trained multilingual language models (LMs) such as mBERT [25] and XLM-R [20]. These pre-trained LMs encode different languages into a joint space of multilingual representations [175, 83], and they perform well especially for zero-shot cross-lingual tasks [175, 83]. Another method orthogonal to this is called translate-train [57, 30]. It translates training data from the source language into the target language and uses the translated texts for training. This work focuses on this method.

Translate-train mitigates the language gap between the source and the target languages in multilingual inference tasks in a straightforward manner, as it directly generates the needed target training samples. This generation process uses a pre-learned translator, which introduces artifacts in the translated texts (*i.e.*, translationese<sup>1</sup>). In other words, translationese often exhibits features such as stylistic ones

---

<sup>1</sup>We refer original texts written by humans as *originals* and the translated texts as *translationese*

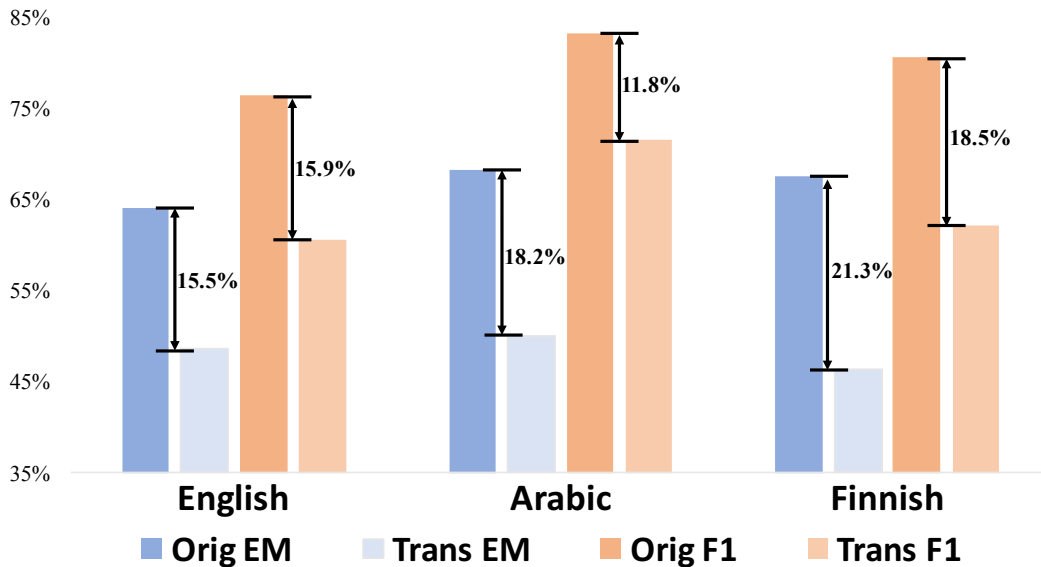


Figure 6.1: QA performance of using original and translated texts (translationese) as training data on TyDiQA dataset. “EM” stands for Exact Match.

that are different from originals and thus can mislead model training [143, 162, 9]. Figure 6.1 shows the quantitative comparison using different training data (originals vs. translationese) but originals for testing. Taking English as example, we train one model with originals English data and another model with translationese English data generated by back-translation. The test data is originals English data. It is clear that using models trained with translationese is significantly inferior.

In this chapter, we aim to tackle this issue by studying the learnability and transferrability of the artifact patterns in translationese. We conduct experiments to first investigate if such patterns are recognizable or transferrable by deep learning models. Specifically, we train a binary classifier to distinguish originals from translationese using the training data of only one language. We then test it on other languages. Our intuition are the following: 1) If the model converges, then it means we can learn the patterns of the artifacts. 2) If the trained model recognizes the translationese of other languages, then it means the model can transfer the learned patterns between different languages. Our results in Figure 6.2 validate both: 1) the model converges well and achieves 97% accuracy on the training language, and 2) for simplicity.

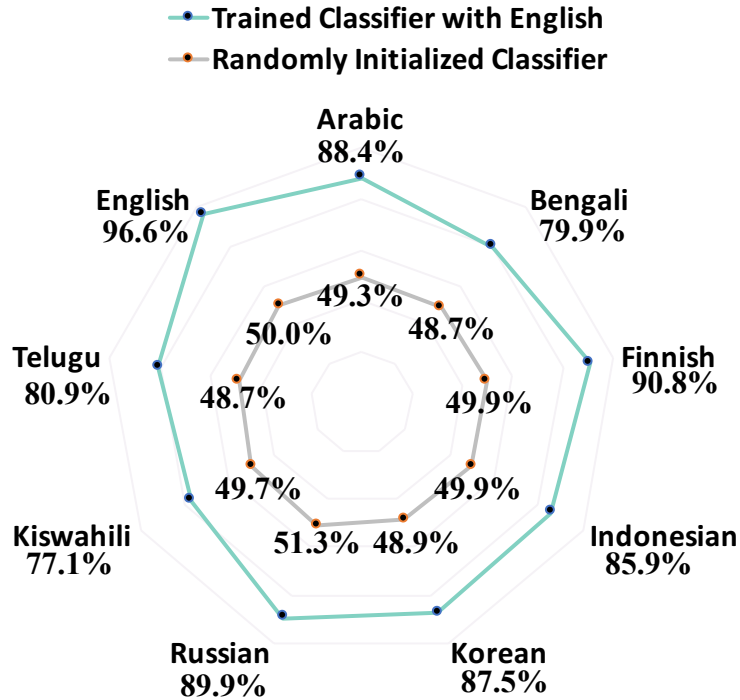


Figure 6.2: The XLM-R [20] classification results of distinguishing the translationese for different languages on TyDiQA [19] when using the classifiers trained with only English pairs (originals and translationese), or randomly initialized (without any training).

it also performs reasonably well on other languages (77% ~ 91%).

Based on the above intuitions and validations, we propose an approach named Translationese Embracing Artifacts (TEA) and implement it using the domain mapping technique [195]. TEA explicitly learns the mapping function from originals to translationese using the source language containing pairwise training data (originals and the corresponding translationese). It minimizes the distance between the mapped representation of originals and the representation of the corresponding translationese (*e.g.*, generated by back-translation) [144]. It then applies this function on the inference of target languages which do not have any originals for training. For evaluation, we conduct experiments on the multilingual QA tasks using the TyDiQA dataset [19]<sup>2</sup>. Our results show that TEA outperforms translate-train baselines as well as the related methods of mitigating translationese in machine

<sup>2</sup>Note that our approach is generic and can be implemented into tackling other multilingual NLP tasks.

translation [104, 165].

## 6.2 Our Approach (TEA)

Let  $\mathbf{x}$ , a question-passage pair, represents the input, and  $\mathbf{y}$  is the output label sequence that indicates the start and end positions of the answer span in the passage.  $\mathcal{X}$  denotes the domain of  $\mathbf{x}$  and  $\mathcal{Y}$  is the set of answers. The input  $\mathbf{x}$  comes from different languages, and it can be either originals or translationese during training. Specifically, we use  $\mathcal{X}_{\text{src, orig}}$  to denote the domain of source language originals, and define  $\mathcal{X}_{\text{trgt, orig}}$  and  $\mathcal{X}_{\text{trgt, trans}}$  in a similar way. We further use back-translation [144] to generate source language translationese, denoted by  $\mathcal{X}_{\text{src, trans}}$ , for the purpose of modeling the difference between originals and translationese.

The goal is to learn a mapping  $f : \mathcal{X}_{\text{trgt, orig}} \rightarrow \mathcal{Y}$ , *i.e.*, taking target language originals as input. However, during training, we only have  $\mathcal{D}_{\text{src, orig}} \in \mathcal{X}_{\text{src, orig}} \times \mathcal{Y}$  and  $\mathcal{D}_{\text{trgt, trans}} \in \mathcal{X}_{\text{trgt, trans}} \times \mathcal{Y}$ . The challenge is that a mapping function  $f$  learned from either  $\mathcal{D}_{\text{src, orig}}$  or  $\mathcal{D}_{\text{trgt, trans}}$  may not work well for  $\mathcal{X}_{\text{trgt, orig}}$ . Based on the observations from Figure 6.2, we could learn to mitigate the translationese artifacts for target languages by a original-to-translationese mapping trained with the source language. We therefore break down  $\mathcal{X}$  to  $\mathcal{Y}$  into following steps:

**Multilingual Projection (MP):** First, input  $\mathbf{x}$  is projected into a language-agnostic multilingual space by using a pre-trained multilingual LM. We use  $\mathcal{X}_{\text{ml}}$  to denote the projected multilingual space, and  $f_{\text{MP}}$  is a multilingual projection (*i.e.*, LM) that maps an input  $\mathbf{x}$  in any language into  $\mathcal{X}_{\text{ml}}$ .

**Original-to-Translationese Projection (OTP):** Suppose  $\mathcal{X}_{\text{ml}}$  consists of two subspaces:  $\mathcal{X}_{\text{ml}} = \mathcal{X}_{\text{ml, orig}} \cup \mathcal{X}_{\text{ml, trans}}$ , where  $\mathcal{X}_{\text{ml, orig}}$  and  $\mathcal{X}_{\text{ml, trans}}$  denote the multilingual representations of any originals and translationese, respectively. To closing the gap between originals and translationese, we define an original-to-translationese projection function  $f_{\text{OTP}} : \mathcal{X}_{\text{ml, orig}} \rightarrow \mathcal{X}_{\text{ml, trans}}$  to convert the representation of a piece of originals to its corresponding representation of translationese.

**Language-Agnostic QA (QA):** The last step is a language-agnostic classifier for QA task itself. We use  $f_{QA} : \mathcal{X}_{ml, trans} \rightarrow \mathcal{Y}$  to denote this function.

Given an input  $\mathbf{x}$ , depending on whether it is from originals or translationese, we use different compositions of the functions above to map  $\mathbf{x}$  to  $\mathbf{y}$ :

$$\mathbf{y} = \begin{cases} f_{QA} \circ f_{OTP} \circ f_{MP}(\mathbf{x}) & \mathbf{x} \in \mathcal{X}_{*, orig}, \\ f_{QA} \circ f_{MP}(\mathbf{x}) & \mathbf{x} \in \mathcal{X}_{*, trans}. \end{cases} \quad (6.1)$$

Here  $\circ$  represents the composition of two functions, *i.e.*,  $f \circ g(x) = f(g(x))$ , and  $*$  denotes source language or target languages. More concretely, for  $(\mathbf{x}, \mathbf{y}) \in \mathcal{D}_{src, orig}$ , we use  $\mathcal{X}_{src, orig} \xrightarrow{f_{MP}} \mathcal{X}_{ml, orig} \xrightarrow{f_{OTP}} \mathcal{X}_{ml, trans} \xrightarrow{f_{QA}} \mathcal{Y}$ ; for  $(\mathbf{x}, \mathbf{y}) \in \mathcal{D}_{trgt, trans}$ , we use  $\mathcal{X}_{trgt, trans} \xrightarrow{f_{MP}} \mathcal{X}_{ml, trans} \xrightarrow{f_{QA}} \mathcal{Y}$ .

As suggested in Section 6.1, we make use of the source language translationese to learn the  $f_{OTP}$ . Specifically, for  $(\mathbf{x}, \mathbf{y}) \in \mathcal{D}_{src, orig}$ , we represents  $\mathbf{x}' \in \mathcal{X}_{src, trans}$  as its corresponding translationese, *i.e.*, generated by back-translation [144] through a pivot language. Let  $\{(\mathbf{x}, \mathbf{x}')\} \in \mathcal{D}_{src, pairs}$  denotes all the pairs of originals and translationese in the source language. Then, we minimize the similarity between  $f_{OTP}(f_{MP}(\mathbf{x}))$  and  $f_{MP}(\mathbf{x}')$  to optimize  $f_{OTP}$ .

In summary, the loss function consists of the following three components:

$$\begin{aligned} L = & \sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{D}_{src, orig}} l(f_{QA} \circ f_{OTP} \circ f_{MP}(\mathbf{x}), \mathbf{y}) \\ & + \sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{D}_{trgt, trans}} l(f_{QA} \circ f_{MP}(\mathbf{x}), \mathbf{y}) \\ & + \sum_{(\mathbf{x}, \mathbf{x}') \in \mathcal{D}_{src, pairs}} \text{cos}(f_{OTP}(f_{MP}(\mathbf{x})), f_{MP}(\mathbf{x}')), \end{aligned} \quad (6.2)$$

where  $l(\cdot, \cdot)$  is standard cross entropy loss and  $\text{cos}(\cdot, \cdot)$  is the cosine similarity function.

**Model Details.** For  $f_{MP}$ , we use multilingual pre-trained LM, XLM-R [20]. For  $f_{OTP}$ , we utilize a self-attention layer as in transformer [160] followed by a linear layer.  $f_{QA}$  is also implemented by a linear layer following a standard way [25].

## 6.3 Experiments

**Dataset.** We conduct experiments on the TyDiQA dataset [19]. TyDiQA is the only existing large-scale multilingual benchmark dataset where test data is original text written by humans. Specifically, we evaluate our approach on the gold-passage sub-task of TyDiQA, which includes 9 languages. We set English as the source language and others as target languages and report the performance on the target languages. During training, we utilize translated training data in *all* target languages for joint training. We use Exact Match (EM) and F1 scores as our evaluation metrics.

**Implementation.** We use the pre-trained multilingual language model, XLM-R [20], as our backbone. Translations of the English training data for target languages are from XTREME [57] and translationese English is translated by Google Cloud Translation<sup>3</sup>.

Method	Design	ar	bn	fi	id	ko	ru	sw	te	avg
STT	F	40.4/67.6	47.8/64.0	53.2/70.5	61.9/77.4	10.9/31.9	42.1/67.0	48.1/66.1	43.6/70.1	43.5/64.3
FILTER	F	50.8/72.8	56.6/70.5	57.2/73.3	59.8/76.8	12.3/33.1	46.6/68.9	65.7/77.4	50.4/69.9	49.9/67.8
STT*	F	58.0/76.6	54.6/70.2	59.0/74.8	64.7/80.2	48.0/61.6	49.5/71.2	58.7/74.6	57.0/76.2	56.2/73.2
TAG*	T	56.9/76.4	55.5/70.0	59.4/75.2	64.4/79.6	48.6/61.7	49.1/70.4	60.7/76.0	57.8/76.4	56.5/73.2
TST*	T	58.4/75.5	60.2/72.2	58.3/74.4	65.5/78.9	49.3/62.6	49.0/69.7	63.5/76.7	56.2/76.1	57.6/73.3
GRL*	T	57.6/75.6	58.4/72.6	59.7/74.8	65.3/79.9	49.6/62.2	49.1/70.4	62.9/76.9	58.2/77.0	57.6/73.7
TEA	T	56.5/76.1	60.2/74.9	60.9/76.5	63.6/79.3	48.6/61.4	51.5/72.0	66.7/78.9	60.7/78.7	<b>58.6/74.7</b>

Table 6.1: Main results (Exact Match / F1 scores) on TyDiQA. All methods are with XLM-R as backbone. The “Design” column indicates whether the design of this method considers translationese artifacts. The columns “ar” to “te” represent different target languages. The “avg” column denotes the average performance across the 8 target languages. \* indicates our implementation.

**Baselines.** We compare our model with the following baselines: (1) Standard Translate-Train (STT) [25], which is a standard fine-tuning approach for translate-train. (2) FILTER [30], which is an advanced translate-train method that fully utilizes the parallel data. (3) Tagging (TAG) [104], which distinguishes originals and translationese by adding a tag to each. (4) Two-Stage Training (TST) [165], which is another approach to address the gap between translationese and originals. It first uses the combination of them for training followed by another round of training

<sup>3</sup><https://cloud.google.com/translate>

only on originals. (5) Gradient Reversal Layer (GRL) [34], which is a general DA method.

**Main results.** The comparison between our approach and the baselines is summarized in Table 6.1. We can observe the following: (1) Our TEA outperforms all baselines. For instance, TEA surpasses STT by 2.4% (EM) and 1.5% (F1) on average. This demonstrates the effectiveness of our method. (2) Methods considering translationese artifacts generally perform better than methods without such design, which reinforces the importance of mitigating translationese artifacts. (3) Compared to other baselines for translationese artifacts, TEA still shows its superiority. We highlight that our OTP module for explicit projection is better than implicit DA approaches, *e.g.*, TAG only uses different tag to distinguish the translationese from originals.

Settings	EM	F1
STT	56.2	73.2
(1) STT+ $\mathcal{X}_{\text{src, trans}}$	56.6	73.2
(2) STT+params	56.3	73.5
(3) TOP	57.9	74.1
(4) MLP in OTP	56.7	73.3
(5) MSE loss	58.0	73.9
Full method	<b>58.6</b>	<b>74.7</b>

Table 6.2: Ablation study on TyDiQA. We report the average EM and F1 performance on the 8 target languages.

**Ablation studies.** We conduct in-depth ablation studies to analyze TEA. Specifically, we explore the following settings: (1) Since we use 11% more data in TEA (unlabeled  $\mathcal{X}_{\text{src, trans}}$ ) compared to STT, here we add labeled  $\mathcal{X}_{\text{src, trans}}$  in STT. (2) Since we use additional 0.38% parameters (OTP) in our method compared to STT, here we add the same OTP module in STT. (3) We replace the Original-to-Translationese Projection (OTP) by Translationese-to-Original Projection (TOP). (4) We replace the self-attention layer in OTP with a multi-layer perceptron (MLP). (5) We replace the cosine similarity function in loss with mean square function.

The results are summarized in Table 6.2. Compared to the variants, our full

method performs best over all settings. (1)/(2) incorporate additional data/parameters, which demonstrates the improvement of our method is not caused by the two factors. (3) proves that TOP still mitigates the artifacts, but OTP obtaining better performance. We argue that it is because most of the training data is translationese. (4) and (5) demonstrate the effectiveness of our loss function and architecture.

Settings	Language Family	EM	F1
Scottish (gd)	Indo-European	58.8	74.0
Korean (ko)	Koreanic	57.8	74.0
Chinese (zh)	Sino-Tibetan	57.6	73.8
German (de)	Indo-European	58.6	74.7

Table 6.3: Experiment results of utilizing different language as pivot language for generating  $\mathcal{X}_{\text{src, trans}}$ .

**Pivot Languages Analysis.** Here we study the effect of pivot language used in generating  $\mathcal{X}_{\text{src, trans}}$ . Specifically, we select four pivot languages, *i.e.*, German (de)<sup>4</sup>, Scottish (gd), Korean (ko) and Chinese (zh), for evaluation. We fix our approach and only replace the  $\mathcal{X}_{\text{src, trans}}$  used in OTP. The results are reported in Table 6.3. We observe that pivot languages from Indo-European family is superior to that from other language families. We think this is because other target language training data in translate-train are translated from English and English is from Indo-European family.

## 6.4 Conclusion

We aim to mitigate the translationese artifacts when training translate-train models. After verifying the transferability of the translationese patterns across languages, we propose the TEA approach that learns to mitigate artifacts using a source language and to facilitate the inference on unseen target languages. Our approach is simple and generic and our results on multilingual QA shows its efficiency.

<sup>4</sup>German (de) is the default pivot language in this work.



# Chapter 7

## Conclusion and Future Work

In the previous chapters, we elaborate on the necessity of mitigating the gap in the out-of-distribution scenario for natural language processing tasks, and propose methods for two scenarios, *i.e.*, adversarial robustness and cross-lingual transfer, which show obvious improvement and outperform state-of-the-art results on different natural language understanding tasks.

For adversarial robustness, we propose two methods. First, we concentrate on the question answering task, especially, multiple-choice question answering and span-extraction question answering. We discover that the question answering model may take the shortcut, *i.e.*, spurious correlation, for the prediction instead of comprehensive reasoning. Towards a robust question answering model, we formulate the inference process using the structural causal model and argue that the robust reasoning is equivalent to the indirect effect of the input variables, *e.g.*, passage and question. Thus we propose counterfactual variable control (CVC) to measure the indirect effect implemented on the deep models. We evaluate CVC on 7 different adversarial sets on four question answering datasets with different backbones. The results demonstrate the effectiveness. Second, we explore a more general and challenging setting for adversarial robustness, *i.e.*, unknown bias, without the help of reweighting with the bias model. We adopt intervention from causal inference to mitigate the bias, *i.e.*, confounder. Two difficulties are that the confounder may be

unobserved and multi-factorial. To this end, we propose bottom-up automatic intervention (BAI), which consists of two components: automatic stratification and bottom-up intervention, to counter each difficulty. We apply BAI on three benchmarks under the OOD setting and outperform state-of-the-art methods.

For cross-lingual transfer, we have one study for zero-shot and few-shot cross-lingual transfer and another study for translate-train cross-lingual transfer. Under zero-shot and few-shot settings, we resort to the universal syntax for language-agnostic features, specifically, universal POS tags and universal dependency relations. To further facilitate the usage of augmented syntactic feature in multilingual pretrained language models, we propose a counterfactual training method to guide the model to focus on the augmented features named COunterfactual SYntax (COSY). We evaluate COSY on three multilingual benchmarks from two natural language understanding tasks, *i.e.*, question answering and natural language inference. The experimental results show that COSY outperforms other state-of-the-art methods with fewer parameters and training steps. Our second piece of work dives into the translate-train setting for cross-lingual transfer. Translate-train serves as the data augmentation for target languages by translating the data in the source language to the target language. We discover that such translated data, *i.e.*, translationese, may hamper the model’s performance on data directly written by humans, *i.e.*, original text. To address this problem, we first empirically show that the gap between translationese and originals is transferable across languages. Then we propose a domain mapping method named Translationese Embracing Artifacts (TEA) to close the gap. We test TEA on a human-written question answering dataset and observe strong performance against other translate-train and translationese-aware methods.

In summary, we design novel approaches to address different OOD scenarios in adversarial robustness and cross-lingual transfer. We hope that this dissertation will attract more attention to OOD scenarios rather than only focusing on the IID test set and inspire more work in the future.

We also list several possible directions worth exploring in the future below.

One direction is to apply more advanced causal inference methods in NLP to solve complex OOD problems. Currently, the development of causal inference in NLP is still in its infancy. Most of the methods rely on simple counterfactual or intervention formulation. How to effectively apply other causal inference techniques in NLP is underexplored. For example, (i) judging whether the prediction is fair in certain OOD scenario based on the definition of counterfactual fairness [78, 106], *e.g.*, gender and race bias in the text. (ii) automatically building the causal graph for natural language processing. There is already a large body of work on identifying the causal structure for real-world data or system [68, 51]. In NLP, we want to point out two possibilities of usage. One is to automatically explore the causal structure within the text itself, *e.g.*, extracting the causal relationship between facts and knowledge [99]. Second, automatically figuring out the causal structure of the data generation process or the inference process, *e.g.*, Figure 3.2 and Figure 4.2.

Second, some special designs for the mixture of different types of OOD are also interesting for further exploration. For example, when we need to use multilingual NLP tasks in cross-border business for community question answering, we may have the Chinese data from the Zhihu platform and the target deployment place of the system is the Reddit platform. In this case, two OOD scenarios are mixed, *i.e.*, cross-domain transfer and cross-lingual transfer. One may say that we could directly apply the existing cross-domain or cross-lingual transfer method to address this scenario. However, the performance may deteriorate even more since the gap is relatively larger compared to the case in this dissertation. How to disentangle the mixed OOD types and reconstruct them for diverse requirements could be indispensable for these applications.

There are still many open questions in the area of mitigating the gap of OOD for NLP tasks. Hope the aforementioned directions could inspire more interesting works in this direction.

# Bibliography

- [1] H. Amirkhani and M. T. Pilehvar. Don't discard all the biased instances: Investigating a core assumption in dataset bias mitigation techniques. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4720–4728, 2021.
- [2] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.
- [3] M. Artetxe, G. Labaka, and E. Agirre. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 2289–2294, 2016.
- [4] M. Artetxe, G. Labaka, and E. Agirre. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, 2017.
- [5] M. Artetxe, G. Labaka, and E. Agirre. Translation artifacts in cross-lingual transfer learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7674–7684, 2020.
- [6] M. Artetxe, S. Ruder, and D. Yogatama. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.
- [7] R. M. Baron and D. A. Kenny. The moderator–mediator variable distinction in social psychological research: Conceptual, strategic, and statistical considerations. *Journal of personality and social psychology*, 51(6):1173, 1986.
- [8] S. Benaim and L. Wolf. One-shot unsupervised cross domain translation. *advances in neural information processing systems*, 31, 2018.
- [9] Y. Bizzoni, T. S. Juzek, C. Espana-Bonet, K. D. Chowdhury, J. van Genabith, and E. Teich. How human is machine translationese? comparing human and machine translations of text and speech. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 280–290, 2020.
- [10] J. Blitzer, M. Dredze, and F. Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 440–447, 2007.
- [11] M. Blohm, G. Jagfeld, E. Sood, X. Yu, and N. T. Vu. Comparing attention-based convolutional and recurrent neural networks: Success and limitations in machine reading comprehension. *arXiv preprint arXiv:1808.08744*, 2018.

- [12] R. L. Bras, S. Swayamdipta, C. Bhagavatula, R. Zellers, M. E. Peters, A. Sabharwal, and Y. Choi. Adversarial filters of dataset biases. *arXiv preprint arXiv:2002.04108*, 2020.
- [13] R. Cadene, C. Dancette, M. Cord, D. Parikh, et al. Rubi: Reducing unimodal biases for visual question answering. In *NeurIPS*, pages 839–850, 2019.
- [14] K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019.
- [15] Z. Chi, L. Dong, F. Wei, N. Yang, S. Singhal, W. Wang, X. Song, X.-L. Mao, H.-Y. Huang, and M. Zhou. Infoclm: An information-theoretic framework for cross-lingual language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3576–3588, 2021.
- [16] R. Choenni and E. Shutova. What does it mean to be language-agnostic? probing multilingual sentence encoders for typological properties. *arXiv preprint arXiv:2009.12862*, 2020.
- [17] C. Clark, M. Yatskar, and L. Zettlemoyer. Don’t take the easy way out: Ensemble based methods for avoiding known dataset biases. In *EMNLP*, pages 4060–4073, 2019.
- [18] C. Clark, M. Yatskar, and L. Zettlemoyer. Learning to model and ignore dataset bias with mixed capacity ensembles. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 3031–3045, 2020.
- [19] J. H. Clark, E. Choi, M. Collins, D. Garrette, T. Kwiatkowski, V. Nikolaev, and J. Palomaki. Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470, 2020.
- [20] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, É. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, 2020.
- [21] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, É. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.
- [22] A. Conneau, R. Rinott, G. Lample, A. Williams, S. Bowman, H. Schwenk, and V. Stoyanov. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.
- [23] E. Creager, J.-H. Jacobsen, and R. Zemel. Environment inference for invariant learning. In *International Conference on Machine Learning*, pages 2189–2200. PMLR, 2021.

- [24] M.-C. de Marneffe, T. Dozat, N. Silveira, K. Haverinen, F. Ginter, J. Nivre, and C. D. Manning. Universal stanford dependencies: A cross-linguistic typology. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, 2014.
- [25] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186, 2019.
- [26] N. Ding, G. Xu, Y. Chen, X. Wang, X. Han, P. Xie, H. Zheng, and Z. Liu. Few-nerd: A few-shot named entity recognition dataset. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3198–3213, 2021.
- [27] C. Dyer, V. Chahuneau, and N. A. Smith. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, 2013.
- [28] J. Ebrahimi, A. Rao, D. Lowd, and D. Dou. Hotflip: White-box adversarial examples for text classification. In *ACL*, pages 31–36, 2018.
- [29] S. Edunov, M. Ott, M. Ranzato, and M. Auli. On the evaluation of machine translation systems trained with back-translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2836–2846, 2020.
- [30] Y. Fang, S. Wang, Z. Gan, S. Sun, and J. Liu. Filter: An enhanced fusion method for cross-lingual language understanding. *arXiv preprint arXiv:2009.05166*, 2020.
- [31] S. Feng, E. Wallace, A. Grissom II, M. Iyyer, P. Rodriguez, and J. Boyd-Graber. Pathologies of neural models make interpretations difficult. In *EMNLP*, pages 3719–3728, 2018.
- [32] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 2017.
- [33] M. Freitag, D. Grangier, and I. Caswell. Bleu might be guilty but references are not innocent. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 61–71, 2020.
- [34] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.
- [35] H. Gao, Z. Shou, A. Zareian, H. Zhang, and S.-F. Chang. Low-shot learning via covariance-preserving adversarial augmentation networks. *Advances in Neural Information Processing Systems*, 31, 2018.
- [36] M. Geva, Y. Goldberg, and J. Berant. Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. 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 1161–1166, 2019.

- [37] A. Ghaddar, P. Langlais, M. Rezagholizadeh, and A. Rashid. End-to-end self-debiasing framework for robust nlu training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1923–1929, 2021.
- [38] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.
- [39] Y. Graham, B. Haddow, and P. Koehn. Translationese in machine translation evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 72–81, 2020.
- [40] G. Grand and Y. Belinkov. Adversarial regularization for visual question answering: Strengths, shortcomings, and side effects. In *Proceedings of the Second Workshop on Shortcomings in Vision and Language*, pages 1–13, 2019.
- [41] J. Gu, Y. Wang, Y. Chen, V. O. Li, and K. Cho. Meta-learning for low-resource neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.
- [42] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith. Don’t stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, 2020.
- [43] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2006.
- [44] A. Haghighi, P. Liang, T. Berg-Kirkpatrick, and D. Klein. Learning bilingual lexicons from monolingual corpora. In *Proceedings of ACL-08: Hlt*, pages 771–779, 2008.
- [45] J. A. Hall, M. A. Milburn, and A. M. Epstein. A causal model of health status and satisfaction with medical care. *Medical care*, pages 84–94, 1993.
- [46] X. Han and J. Eisenstein. Unsupervised domain adaptation of contextualized embeddings for sequence labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4238–4248, 2019.
- [47] X. Han, H. Zhu, P. Yu, Z. Wang, Y. Yao, Z. Liu, and M. Sun. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809, 2018.
- [48] J. A. Hartigan and M. A. Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1):100–108, 1979.
- [49] H. He, S. Zha, and H. Wang. Unlearn dataset bias in natural language inference by fitting the residual. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 132–142, 2019.
- [50] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

- [51] C. Heinze-Deml, M. H. Maathuis, and N. Meinshausen. Causal structure learning. *Annual Review of Statistics and Its Application*, 5:371–391, 2018.
- [52] K. M. Hermann and P. Blunsom. Multilingual distributed representations without word alignment. *arXiv preprint arXiv:1312.6173*, 2013.
- [53] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.
- [54] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pages 1989–1998. Pmlr, 2018.
- [55] Y. Hoshen and L. Wolf. Non-adversarial unsupervised word translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 469–478, 2018.
- [56] T.-Y. Hsu, C.-L. Liu, and H.-y. Lee. Zero-shot reading comprehension by cross-lingual transfer learning with multi-lingual language representation model. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 2019.
- [57] J. Hu, S. Ruder, A. Siddhant, G. Neubig, O. Firat, and M. Johnson. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International Conference on Machine Learning*, 2020.
- [58] P.-S. Huang, R. Stanforth, J. Welbl, C. Dyer, D. Yogatama, S. Gowal, K. Dvijotham, and P. Kohli. Achieving verified robustness to symbol substitutions via interval bound propagation. In *EMNLP*, pages 4074–4084, 2019.
- [59] P.-S. Huang, H. Zhang, R. Jiang, R. Stanforth, J. Welbl, J. Rae, V. Maini, D. Yogatama, and P. Kohli. Reducing sentiment bias in language models via counterfactual evaluation. In *EMNLP: Findings*, pages 65–83, 2020.
- [60] T. Ishiwatari, Y. Yasuda, T. Miyazaki, and J. Goto. Relation-aware graph attention networks with relational position encodings for emotion recognition in conversations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 2020.
- [61] M. Iyyer, J. Wieting, K. Gimpel, and L. Zettlemoyer. Adversarial example generation with syntactically controlled paraphrase networks. In *NAACL*, 2018.
- [62] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.
- [63] R. Jia and P. Liang. Adversarial examples for evaluating reading comprehension systems. In *EMNLP*, pages 2021–2031, 2017.
- [64] R. Jia, A. Raghunathan, K. Göksel, and P. Liang. Certified robustness to adversarial word substitutions. In *EMNLP*, pages 4120–4133, 2019.
- [65] H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and T. Zhao. Smart: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. *arXiv preprint arXiv:1911.03437*, 2019.



- [66] J. Jiang and C. Zhai. Instance weighting for domain adaptation in nlp. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 264–271, 2007.
- [67] Á. Kádár, G. Chrupała, and A. Alishahi. Representation of linguistic form and function in recurrent neural networks. *Computational Linguistics*, 2017.
- [68] M. Kalisch and P. Bühlmann. Causal structure learning and inference: a selective review. *Quality Technology & Quantitative Management*, 11(1):3–21, 2014.
- [69] B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng, and Y. Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations*, 2019.
- [70] D. Kaushik, E. Hovy, and Z. C. Lipton. Learning the difference that makes a difference with counterfactually-augmented data. In *ICLR*, 2020.
- [71] D. Kaushik and Z. C. Lipton. How much reading does reading comprehension require? a critical investigation of popular benchmarks. In *EMNLP*, pages 5010–5015, 2018.
- [72] P. Keung, V. Bhardwaj, et al. Adversarial learning with contextual embeddings for zero-resource cross-lingual classification and ner. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 2019.
- [73] P. Keung, Y. Lu, J. Salazar, and V. Bhardwaj. On the evaluation of contextual embeddings for zero-shot cross-lingual transfer learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 2020.
- [74] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [75] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.
- [76] W. M. Kouw and M. Loog. A review of domain adaptation without target labels. *IEEE transactions on pattern analysis and machine intelligence*, 43(3):766–785, 2019.
- [77] D. Krueger, E. Caballero, J.-H. Jacobsen, A. Zhang, J. Binas, D. Zhang, R. Le Priol, and A. Courville. Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, pages 5815–5826. PMLR, 2021.
- [78] M. J. Kusner, J. Loftus, C. Russell, and R. Silva. Counterfactual fairness. *Advances in neural information processing systems*, 30, 2017.
- [79] G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy. Race: Large-scale reading comprehension dataset from examinations. In *EMNLP*, pages 785–794, 2017.
- [80] G. Lample and A. Conneau. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*, 2019.
- [81] G. Lample, A. Conneau, M. Ranzato, L. Denoyer, and H. Jégou. Word translation without parallel data. In *International Conference on Learning Representations*, 2018.

- [82] S. Lauly, A. Boulanger, and H. Larochelle. Learning multilingual word representations using a bag-of-words autoencoder. *arXiv preprint arXiv:1401.1803*, 2014.
- [83] A. Lauscher, V. Ravishankar, I. Vulić, and G. Glavaš. From zero to hero: On the limitations of zero-shot cross-lingual transfer with multilingual transformers. *arXiv preprint arXiv:2005.00633*, 2020.
- [84] C. Lawrence and S. Riezler. Improving a neural semantic parser by counterfactual learning from human bandit feedback. In *ACL*, pages 1820–1830, 2018.
- [85] A. Lazaridou, G. Dinu, and M. Baroni. Hubness and pollution: Delving into cross-space mapping for zero-shot learning. 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 270–280, 2015.
- [86] G. Lembersky, N. Ordan, and S. Wintner. Adapting translation models to translationese improves smt. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 255–265, 2012.
- [87] P. Lewis, B. Oğuz, R. Rinott, S. Riedel, and H. Schwenk. Mlqa: Evaluating cross-lingual extractive question answering. *arXiv preprint arXiv:1910.07475*, 2019.
- [88] D. Li, Y. Yang, Y.-Z. Song, and T. Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.
- [89] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017.
- [90] H. Li, S. J. Pan, S. Wang, and A. C. Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5400–5409, 2018.
- [91] Z. Li and D. Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.
- [92] B. Liang, H. Li, M. Su, P. Bian, X. Li, and W. Shi. Deep text classification can be fooled. *arXiv preprint arXiv:1704.08006*, 2017.
- [93] Y. Liang, N. Duan, Y. Gong, N. Wu, F. Guo, W. Qi, M. Gong, L. Shou, D. Jiang, G. Cao, et al. Xglue: A new benchmark dataset for cross-lingual pre-training, understanding and generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 2020.
- [94] J. Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151, 1991.
- [95] H. Linmei, T. Yang, C. Shi, H. Ji, and X. Li. Heterogeneous graph attention networks for semi-supervised short text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 2019.
- [96] J. Liu, Z. Hu, P. Cui, B. Li, and Z. Shen. Heterogeneous risk minimization. *arXiv preprint arXiv:2105.03818*, 2021.

- [97] K. Liu, X. Liu, A. Yang, J. Liu, J. Su, S. Li, and Q. She. A robust adversarial training approach to machine reading comprehension. In *AAAI*, 2020.
- [98] X. Liu, H. Cheng, P. He, W. Chen, Y. Wang, H. Poon, and J. Gao. Adversarial training for large neural language models. *arXiv preprint arXiv:2004.08994*, 2020.
- [99] X. Liu, D. Yin, Y. Feng, Y. Wu, and D. Zhao. Everything has a cause: Leveraging causal inference in legal text analysis. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1928–1941, 2021.
- [100] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.
- [101] Z. Liu, G. I. Winata, A. Madotto, and P. Fung. Exploring fine-tuning techniques for pre-trained cross-lingual models via continual learning. *arXiv preprint arXiv:2004.14218*, 2020.
- [102] F. Luo, W. Wang, J. Liu, Y. Liu, B. Bi, S. Huang, F. Huang, and L. Si. Veco: Variable and flexible cross-lingual pre-training for language understanding and generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3980–3994, 2021.
- [103] R. K. Mahabadi, Y. Belinkov, and J. Henderson. End-to-end bias mitigation by modelling biases in corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8706–8716, 2020.
- [104] B. Marie, R. Rubino, and A. Fujita. Tagged back-translation revisited: Why does it really work? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5990–5997, 2020.
- [105] T. McCoy, E. Pavlick, and T. Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *ACL*, pages 3428–3448, 2019.
- [106] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35, 2021.
- [107] T. Mikolov, Q. V. Le, and I. Sutskever. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*, 2013.
- [108] S. L. Morgan and C. Winship. *Counterfactuals and causal inference*. Cambridge University Press, 2015.
- [109] K. Muandet, D. Balduzzi, and B. Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pages 10–18. PMLR, 2013.
- [110] G. Nan, J. Zeng, R. Qiao, Z. Guo, and W. Lu. Uncovering main causalities for long-tailed information extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9683–9695, 2021.
- [111] Y. Niu, K. Tang, H. Zhang, Z. Lu, X.-S. Hua, and J.-R. Wen. Counterfactual vqa: A cause-effect look at language bias. *arXiv preprint arXiv:2006.04315*, 2020.

- [112] J. Nivre, M.-C. De Marneffe, F. Ginter, Y. Goldberg, J. Hajic, C. D. Manning, R. McDonald, S. Petrov, S. Pyysalo, N. Silveira, et al. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, 2016.
- [113] F. Nooralahzadeh, G. Bekoulis, J. Bjerva, and I. Augenstein. Zero-shot cross-lingual transfer with meta learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 2020.
- [114] X. Ouyang, S. Wang, C. Pang, Y. Sun, H. Tian, H. Wu, and H. Wang. Ernie-m: Enhanced multilingual representation by aligning cross-lingual semantics with monolingual corpora. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 27–38, 2021.
- [115] S. J. Pan, X. Ni, J.-T. Sun, Q. Yang, and Z. Chen. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th international conference on World wide web*, pages 751–760, 2010.
- [116] J. Pearl. [bayesian analysis in expert systems]: comment: graphical models, causality and intervention. *Statistical Science*, 8(3):266–269, 1993.
- [117] J. Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688, 1995.
- [118] J. Pearl. Direct and indirect effects. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 411–420, 2001.
- [119] J. Pearl. Causal inference. *Causality: Objectives and Assessment*, 2010.
- [120] J. Pearl et al. Causal inference in statistics: An overview. *Statistics surveys*, 3:96–146, 2009.
- [121] J. Pearl, M. Glymour, and N. P. Jewell. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.
- [122] J. Pearl and D. Mackenzie. *The book of why: the new science of cause and effect*. Basic Books, 2018.
- [123] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.
- [124] J. Phang, P. M. Htut, Y. Pruksachatkun, H. Liu, C. Vania, K. Kann, I. Calixto, and S. R. Bowman. English intermediate-task training improves zero-shot cross-lingual transfer too. *arXiv preprint arXiv:2005.13013*, 2020.
- [125] T. Pires, E. Schlinger, and D. Garrette. How multilingual is multilingual bert? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.
- [126] J. Qi, Y. Niu, J. Huang, and H. Zhang. Two causal principles for improving visual dialog. *arXiv preprint arXiv:1911.10496*, 2019.
- [127] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, and C. D. Manning. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2020.

- [128] L. Qin, A. Bosselut, A. Holtzman, C. Bhagavatula, E. Clark, and Y. Choi. Counterfactual story reasoning and generation. In *EMNLP*, pages 5046–5056, 2019.
- [129] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. Squad: 100,000+ questions for machine comprehension of text. In *EMNLP*, pages 2383–2392, 2016.
- [130] S. Ramakrishnan, A. Agrawal, and S. Lee. Overcoming language priors in visual question answering with adversarial regularization. In *NeurIPS*, pages 1541–1551, 2018.
- [131] A. Ramponi and B. Plank. Neural unsupervised domain adaptation in nlp—a survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6838–6855, 2020.
- [132] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.
- [133] W. J. Reed. The pareto, zipf and other power laws. *Economics letters*, 74(1):15–19, 2001.
- [134] M. T. Ribeiro, S. Singh, and C. Guestrin. Semantically equivalent adversarial rules for debugging nlp models. In *ACL*, pages 856–865, 2018.
- [135] M. Richardson, C. J. Burges, and E. Renshaw. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *EMNLP*, pages 193–203, 2013.
- [136] P. Riley, I. Caswell, M. Freitag, and D. Grangier. Translationese as a language in “multilingual” nmt. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7737–7746, 2020.
- [137] N. Roese. Counterfactual thinking. *Psychological Bulletin*, 121(1):133–148, 1997.
- [138] N. J. Roese. Counterfactual thinking. *Psychological bulletin*, 1997.
- [139] E. Rosenfeld, P. K. Ravikumar, and A. Risteski. The risks of invariant risk minimization. In *International Conference on Learning Representations*, 2020.
- [140] S. Samanta and S. Mehta. Generating adversarial text samples. In *European Conference on Information Retrieval*, pages 744–749. Springer, 2018.
- [141] V. Sanh, T. Wolf, Y. Belinkov, and A. M. Rush. Learning from others’ mistakes: Avoiding dataset biases without modeling them. In *International Conference on Learning Representations*, 2020.
- [142] T. Schuster, D. Shah, Y. J. S. Yeo, D. R. F. Ortiz, E. Santus, and R. Barzilay. Towards debiasing fact verification models. 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 3410–3416, 2019.
- [143] L. Selinker. Interlanguage. *International Review of Applied Linguistics in Language Teaching*, pages 209–231, 1972.
- [144] R. Sennrich, B. Haddow, and A. Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, 2016.

- [145] T. Shi, Z. Liu, Y. Liu, and M. Sun. Learning cross-lingual word embeddings via matrix co-factorization. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 567–572, 2015.
- [146] S. Shin, K. Song, J. Jang, H. Kim, W. Joo, and I.-C. Moon. Neutralizing gender bias in word embedding with latent disentanglement and counterfactual generation. In *EMNLP: Findings*, pages 3126–3140, 2020.
- [147] M. Shoeybi, M. Patwary, R. Puri, P. LeGresley, J. Casper, and B. Catanzaro. Megatron-lm: Training multi-billion parameter language models using gpu model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.
- [148] S. L. Smith, D. H. Turban, S. Hamblin, and N. Y. Hammerla. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. *arXiv preprint arXiv:1702.03859*, 2017.
- [149] J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.
- [150] K. Sun, D. Yu, J. Chen, D. Yu, Y. Choi, and C. Cardie. Dream: A challenge data set and models for dialogue-based reading comprehension. *TACL*, 7:217–231, 2019.
- [151] Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.
- [152] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.
- [153] K. Tang, Y. Niu, J. Huang, J. Shi, and H. Zhang. Unbiased scene graph generation from biased training. *arXiv preprint arXiv:2002.11949*, 2020.
- [154] D. Teney, E. Abbasnejad, and A. van den Hengel. Unshuffling data for improved generalization in visual question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1417–1427, 2021.
- [155] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal. Fever: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, 2018.
- [156] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.
- [157] P. A. Utama, N. S. Moosavi, and I. Gurevych. Mind the trade-off: Debiasing nlu models without degrading the in-distribution performance. *arXiv preprint arXiv:2005.00315*, 2020.
- [158] P. A. Utama, N. S. Moosavi, and I. Gurevych. Towards debiasing nlu models from unknown biases. In *EMNLP*, pages 7597–7610, 2020.
- [159] V. Vapnik. Principles of risk minimization for learning theory. In *Proceedings of the 4th International Conference on Neural Information Processing Systems*, pages 831–838, 1991.

- [160] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, 2017.
- [161] J. Vig, S. Gehrmann, Y. Belinkov, S. Qian, D. Nevo, Y. Singer, and S. Shieber. Investigating gender bias in language models using causal mediation analysis. *Advances in Neural Information Processing Systems*, 33, 2020.
- [162] V. Volansky, N. Ordan, and S. Wintner. On the features of translationese. *Digital Scholarship in the Humanities*, 30(1):98–118, 2015.
- [163] E. Wallace, S. Feng, N. Kandpal, M. Gardner, and S. Singh. Universal adversarial triggers for attacking and analyzing nlp. In *EMNLP*, pages 2153–2162, 2019.
- [164] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, 2018.
- [165] S. Wang, Z. Tu, Z. Tan, S. Shi, M. Sun, and Y. Liu. On the language coverage bias for neural machine translation. In *Findings of the 59th Annual Meeting of Association for Computational Linguistics*, 2021.
- [166] T. Wang, C. Zhou, Q. Sun, and H. Zhang. Causal attention for unbiased visual recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3091–3100, 2021.
- [167] X. Wang, L. Lian, Z. Miao, Z. Liu, and S. Yu. Long-tailed recognition by routing diverse distribution-aware experts. In *International Conference on Learning Representations*, 2020.
- [168] Y. Wang and M. Bansal. Robust machine comprehension models via adversarial training. In *NAACL*, pages 575–581, 2018.
- [169] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34, 2020.
- [170] X. Wei, Y. Hu, R. Weng, L. Xing, H. Yu, and W. Luo. On learning universal representations across languages. *arXiv preprint arXiv:2007.15960*, 2020.
- [171] A. Williams, N. Nangia, and S. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL*, pages 1112–1122, 2018.
- [172] G. Wilson and D. J. Cook. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(5):1–46, 2020.
- [173] T. Wolf, J. Chaumond, L. Debut, V. Sanh, C. Delangue, A. Moi, P. Cistac, M. Funtowicz, J. Davison, S. Shleifer, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020.
- [174] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew. Huggingface’s transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

- [175] S. Wu and M. Dredze. Beto, bentz, becas: The surprising cross-lingual effectiveness of bert. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 2019.
- [176] Y. Yaghoobzadeh, S. Mehri, R. T. des Combes, T. J. Hazen, and A. Sordoni. Increasing robustness to spurious correlations using forgettable examples. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3319–3332, 2021.
- [177] Y. Yaghoobzadeh, R. Tachet, T. J. Hazen, and A. Sordoni. Robust natural language inference models with example forgetting. *arXiv preprint arXiv:1911.03861*, 2019.
- [178] M. Yan, H. Zhang, D. Jin, and J. T. Zhou. Multi-source meta transfer for low resource multiple-choice question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.
- [179] X. Yang, H. Zhang, and J. Cai. Deconfounded image captioning: A causal retrospect. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
- [180] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*, pages 5754–5764, 2019.
- [181] M. Ye, C. Gong, and Q. Liu. Safer: A structure-free approach for certified robustness to adversarial word substitutions. *arXiv preprint arXiv:2005.14424*, 2020.
- [182] Y.-T. Yeh and Y.-N. Chen. Qainfomax: Learning robust question answering system by mutual information maximization. In *EMNLP*, pages 3361–3366, 2019.
- [183] J. Yu, M. El-karef, and B. Bohnet. Domain adaptation for dependency parsing via self-training. In *Proceedings of the 14th International Conference on Parsing Technologies*, pages 1–10, 2015.
- [184] F. Yuan, L. Shou, X. Bai, M. Gong, Y. Liang, N. Duan, Y. Fu, and D. Jiang. Enhancing answer boundary detection for multilingual machine reading comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.
- [185] Z. Yue, H. Zhang, Q. Sun, and X. Hua. Interventional few-shot learning. *arXiv preprint arXiv:2009.13000*, 2020.
- [186] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.
- [187] M. Zhang and A. Toral. The effect of translationese in machine translation test sets. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 73–81, 2019.
- [188] W. Zhang, H. Lin, X. Han, and L. Sun. De-biasing distantly supervised named entity recognition via causal intervention. *arXiv preprint arXiv:2106.09233*, 2021.
- [189] W. E. Zhang, Q. Z. Sheng, A. Alhazmi, and C. Li. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transactions on Intelligent Systems and Technology*, 11(3):1–41, 2020.



- [190] Y. Zhang, J. Baldridge, and L. He. Paws: Paraphrase adversaries from word scrambling. In *NAACL*, pages 1298–1308, 2019.
- [191] W. Zhao, S. Eger, J. Bjerva, and I. Augenstein. Inducing language-agnostic multilingual representations. *arXiv preprint arXiv:2008.09112*, 2020.
- [192] B. Zheng, L. Dong, S. Huang, W. Wang, Z. Chi, S. Singhal, W. Che, T. Liu, X. Song, and F. Wei. Consistency regularization for cross-lingual fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3403–3417, 2021.
- [193] J. T. Zhou, H. Zhang, D. Jin, and X. Peng. Dual adversarial transfer for sequence labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
- [194] J. T. Zhou, H. Zhang, D. Jin, H. Zhu, M. Fang, R. S. M. Goh, and K. Kwok. Dual adversarial neural transfer for low-resource named entity recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.
- [195] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.
- [196] Q. Zhu, W. Zhang, T. Liu, and W. Y. Wang. Counterfactual off-policy training for neural dialogue generation. In *EMNLP*, pages 3438–3448, 2020.
- [197] Y. Ziser and R. Reichart. Neural structural correspondence learning for domain adaptation. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 400–410, 2017.
- [198] Y. Ziser and R. Reichart. Deep pivot-based modeling for cross-language cross-domain transfer with minimal guidance. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 238–249, 2018.
- [199] Y. Ziser and R. Reichart. Pivot based language modeling for improved neural domain adaptation. In *Proceedings of the 2018 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long papers)*, pages 1241–1251, 2018.
- [200] Y. Zou, Z. Yu, B. Kumar, and J. Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European conference on computer vision (ECCV)*, pages 289–305, 2018.