Transfer Learning and Robustness for Natural Language Processing

by

Di Jin

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2020

Author . . . .
Department of Mechanical Engineering
August 20th, 2020

Certified by . . . .
Peter Szolovits
Professor
Thesis Supervisor

Accepted by . . . .
Nicolas Hadjiconstantinou
Chairman, Department Committee on Graduate Theses

Transfer Learning and Robustness for Natural Language Processing

by

Di Jin

Submitted to the Department of Mechanical Engineering
on August 20th, 2020, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Mechanical Engineering

Abstract

Teaching machines to understand human language is one of the most elusive and long-standing challenges in Natural Language Processing (NLP). Driven by the fast development of deep learning, state-of-the-art NLP models have already achieved human-level performance in various large benchmark datasets, such as SQuAD, SNLI, and RACE. However, when these strong models are deployed to real-world applications, they often show poor generalization capability in two situations: 1. There is only a limited amount of data available for model training; 2. Deployed models may degrade significantly in performance on noisy test data or natural/artificial adversaries. In short, performance degradation on low-resource tasks/datasets and unseen data with distribution shifts imposes great challenges to the reliability of NLP models and prevent them from being massively applied in the wild.

This dissertation aims to address these two issues. Towards the first one, we resort to *transfer learning* to leverage knowledge acquired from related data in order to improve performance on a target low-resource task/dataset. Specifically, we propose different transfer learning methods for three natural language understanding tasks: multi-choice question answering, dialogue state tracking, and sequence labeling, and one natural language generation task: machine translation. These methods are based on four basic transfer learning modalities: multi-task learning, sequential transfer learning, domain adaptation, and cross-lingual transfer. We show experimental results to validate that transferring knowledge from related domains, tasks, and languages can improve the target task/dataset significantly.

For the second issue, we propose methods to evaluate the robustness of NLP models on text classification and entailment tasks. On one hand, we reveal that although these models can achieve a high accuracy of over 90%, they still easily crash over paraphrases of original samples by changing only around 10% words to their synonyms. On the other hand, by creating a new challenge set using four adversarial strategies, we find even the best models for the aspect-based sentiment analysis task cannot reliably identify the target aspect and recognize its sentiment accordingly. On the contrary, they are easily confused by distractor aspects. Overall, these findings raise great concerns of robustness of NLP models, which should be enhanced to ensure their long-run stable service.

Thesis Supervisor: Peter Szolovits
Title: Professor

Acknowledgments

Five years of Ph.D. life is a long marathon, accompanied with laughs and tears, sky-high peaks and dark troughs, and highlights and desperation. I cannot imagine that I could keep moving forward without the companionship and encouragement of my mentors, friends, and family, all along the way. So in the very end of this journey, I would like to express my many thanks to all of you.

First and foremost, I am immensely grateful to my advisor, Peter Szolovits, for your support, patience, insights, as well as all the guidance and help you provided throughout my Ph.D. study. About three years ago, I was at the darkest time ever in my life when I did not want to continue my Ph.D. study any more in my previous major of optical imaging and I was desperate to find a new direction that I really love and can prevent me from quitting. It is you that offered me the opportunity to enter the field of computer science and artificial intelligence, which I would like to devote my entire life to and burn my energies for. Moreover, the freedom you gave me to pursue my interests and follow my curiosity has made all the difference and led to my current achievements. You are definitely one of the most important persons in my life who have changed my fate.

I am very fortunate to have met mentors along the way who took a chance on me and helped me grow. First of all, I am grateful to John Leonard, Jim Glass, Sanjay Sarma, and Brian Subirana for being my thesis committee members and helping me improve my thesis. Furthermore, I am grateful to Peter So, Renjie Zhou, Dilek Hakkani-tur, Tag-young Chung, Bill Gao, Hadrien Glaude, Tristan Naumann, Marzyeh Ghassemi, Frank Dernoncourt, Tianyi Zhou, and Peng Fei, working with you has been one of the high points of my Ph.D.. Especially, Renjie, you are not only my ever best mentor but also my good friend and I always feel very warm with your generous help.

I would also like to thank Wei-Hung Weng, Matthew McDermott, Willie Boag, Emily Alsentzer, Geeticka Chauhan, Elena Sergeeva, Harry Tzu-Ming Hsu, Tiffany So Yeon Min, Joao Palotti, Heather Berlin, and all other members of the MIT Clinical

Decision Making Group for your inspiring discussions, helpful feedback, and efficient collaborations.

I am grateful for all the amazing people I had the chance to get to know at numerous conferences. I am looking forward to seeing you all at future events.

Finally, I thank my family for your love and support, and all my friends for your help and company. It is you who make my Ph.D. life colorful and full of joys.

Contents

List of Figures

List of Tables

List of Abbreviations

ABSA	Aspect-based Sentiment Analysis
AD	Adversarial Discriminator
AT	Adversarial Training
BPE	Byte Pair Encoding
CRF	Conditional Random Field
CV	Computer Vision
DAMT	Domain Adaptation for Machine Translation
DNN	Deep Neural Networks
DST	Dialogue State Tracking
EDST	Extractive Dialogue State Tracking
FCNN	Full-connected Neural Network
GAN	Generative Adversarial Nets
GRAD	Generalized Resource-Adversarial Discriminator
LM	Language Modeling
MAN	Multi-step Attention Network
MCQA	Multiple-choice Question Answering
ML	Machine Learning
MRC	Machine Reading Comprehension
MRQA	Machine Reading for Question Answering
MT	Machine Translation
MTL	Multi-task Learning
NER	Named Entity Recognition
NLG	Natural Language Generation

NLI	Natural Language Inference
NLP	Natural Language Processing
NLU	Natural Language Understanding
NMT	Neural Machine Translation
POS	Part-of-speech
QA	Question Answering
SLU	Spoken Language Understanding
USE	Universal Sentence Encoder

Chapter 1

Introduction

1.1 Motivation

Language is the hallmark of human intelligence, distinguishing us from animals [Pinker, 2003]. Developing systems that can understand and generate human language is considered one of the most challenging tasks along the path to Artificial General Intelligence [Russell and Norvig, 2002]. Although still far from being perfect, we have witnessed significant progress in natural language processing (NLP) and computational linguistics over the last half a century.

As shown by Figure 1-1, NLP research starts from the 1950s and an early achievement is the famous IBM-Georgetown experiment, where researchers demonstrated successful machine translation of 60 sentences from Russian to English for the first time [Hutchins, 2005]. Soon later in 1957, Noam Chomsky published his book "Syntactic Structures" [Chomsky and Lightfoot, 1957], which sparked the development of phrase-structure grammar and played an important role in that era of the 1960s–1970s. Starting from the 1970s, rule-based [Winograd, 1972] and frame-based systems [Minsky, 1974] were composed by humans to capture the semantics of text; however, these rules/frames were only able to deal with limited problems that they were designed for and their performance was also far from satisfactory. Ever since the 1990s, statistical approaches to NLP were introduced and started to revolutionize this field, which utilizes statistical models to automatically learn the rules from the data

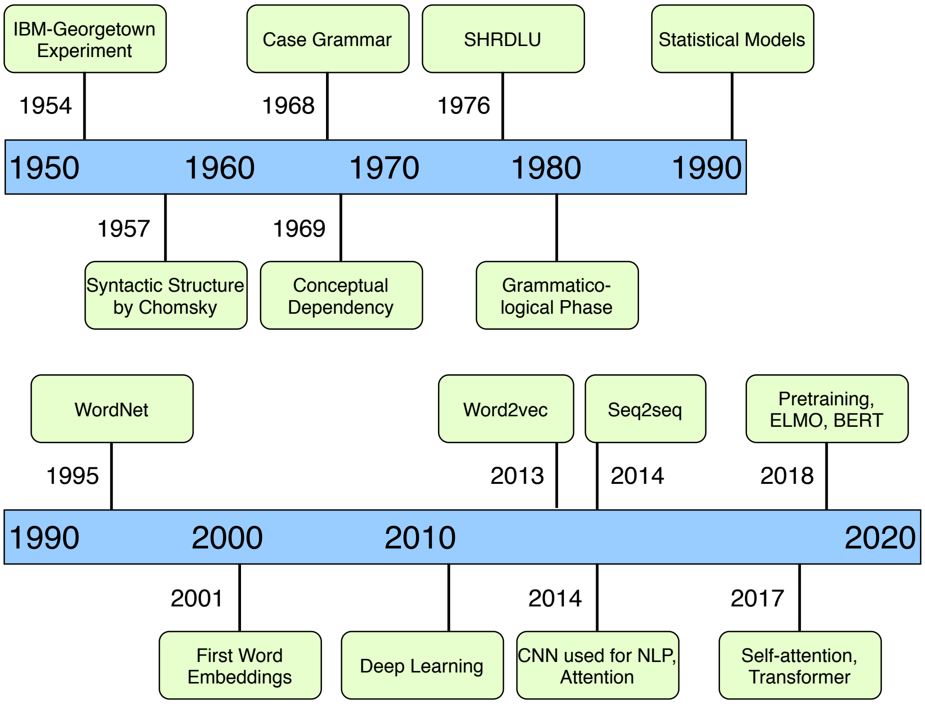


Figure 1-1: Brief history of NLP development.

[Manning et al., 1999]. In this way, human labor was freed from crafting multifarious rules and turned into collecting features that reflect the linguistic properties of text.

In the last decade, deep learning, as a particular category of machine learning (ML) models, has been continuously making breakthroughs in almost every area of NLP and has become without any doubt the model of choice when learning from data [LeCun et al., 2015]. On the one hand, deep learning removes the need for feature engineering by automatically learning the lexical, syntactic, and semantic features via its hierarchical architecture. On the other, the performance of every single task has been boosted unprecedentedly by deep learning compared with all non-deep learning models [Devlin et al., 2019].

Powered by a large quantity of high-quality data samples, supervised learning in NLP has rivaled human performance in many important tasks such as machine translation (MT) [Edunov et al., 2018], question answering (QA) [Lan et al., 2019], and natural language inference (NLI) [Lan et al., 2019]. Table 1.1 summarizes the comparison between the state-of-the-art models' and human performance for the several most important NLP tasks and datasets, which validates that the current best NLP models can be competent compared with human performance on a broad range of benchmark datasets. Despite all these unprecedented successes, many deep learning models are found to be poor at generalizing to unseen data by recent studies [Belinkov and Bisk, 2018]. Over-parameterized learning algorithms can (nearly) perfectly fit the training data but are not able to adapt well to test data with distribution shifts [Brown et al., 2020]. This phenomenon is referred to as data interpolation or, informally, as memorization of the training data [Feldman, 2019].

Tasks	Datasets	SOTA Models	SOTA Perf.	Human Perf.
Text Classification	SST-2	ERNIE	97.5	97.81
Natural Language Inference	MNLI	T5	92.2	92.01
	RTE	T5	92.8	93.61
Paraphrasing	MRPC	ALBERT + DAAF	94.0	86.31
Common Sense	ReCoRD	T5	94.1	91.72
Question Answering	SQuAD 1.1	LUKE	90.2	82.03
	SQuAD 2.0	ELECTRA+ATRLP+PV	89.6	86.83
	BoolQ	T5	91.2	89.02
	MultiRC	T5	88.1	81.82

Table 1.1: Comparison between state-of-the-art models' and human performance for important NLP tasks and datasets. "Perf." denotes performance.

Generalization has long been an important topic in machine learning [Moody, 1992] and plays a vital role in applying advanced ML systems into real-life production. Models would easily have poor generalization capability with a large gap between performance on training and test sets under two scenarios: training data is scarce so that not enough coverage over the data distribution of test data can be provided; training and test data have distribution shifts [Goodfellow et al., 2016]. This thesis will focus on investigating the generalization of models from the NLP perspective by addressing these two issues.

The above-mentioned first issue is very common given the plethora of languages, tasks, and domains in the real world. As our world develops, new human needs constantly arise; new tasks—from mining drug-to-drug interaction among large biomedical literature databases, searching for recorded court cases relevant to a current customer's case, to building a conversation agent to talk with people fluently—more and more need to be solved by NLP automatically, accurately, and efficiently. Unfortunately, due to the time constraints and lack of available resources, it is not feasible to gather enough data to train well a powerful deep leaning model to fulfill its potential for every new setting.

What's more, our language is diverse; the number of human languages in the world is estimated to be between 5,000 and 7,000. However, the majority of current NLP research and industry services are focused on English and several Asian and European languages spoken by the largest populations. It is still important to apply the cutting-edge techniques in NLP to those minor languages and benefit those people speaking these minor languages. However, one of the main obstacles is the scarcity of labeled data.

Standard supervised learning that relies on plenty of labeled data would easily break down when facing real-world challenges since manually annotating abundant examples for every setting is infeasible. Fortunately, transfer learning can be leveraged to ameliorate this failing by transferring knowledge from other related sources to the target task. It is actually not a brand-new concept but has long been driving many fundamental advances in NLP such as the latent semantic analysis [Deerwester et al., 1990], Brown clusters [Brown et al., 1993], continuous word embeddings [Mikolov et al., 2013], and most recent prevalent pre-trained contextualized word embeddings [Devlin et al., 2019]. These milestone-level works can all be deemed as particular forms of transfer learning, since their proposed methods are aiming at the same target: transferring knowledge from a general-purpose source task to a more specialized target task.

With regards to the second issue, although current deep learning algorithms have achieved very high benchmarking performances, recent studies [Belinkov and Bisk, 2018; Jia and Liang, 2017; Szegedy et al., 2014] show that they are still brittle in a way similar to early rule-based systems: they can easily conform to the characteristics of the data they have been trained on but are not able to adapt when conditions change. In general, the generalization capability of trained models on unseen data with natural/artificial noise or distribution shifts can be referred as *robustness*. A recent body of work has found that neural networks are susceptible to natural noise, spurious correlations or artifacts that exist in data, as well as adversarial examples that are distortions of inputs and can easily fool the networks [Hendrycks et al., 2019; Tsipras et al., 2018]. This phenomenon has serious implications that our current models fail to robustly learn the underlying concepts.

1.2 Contributions

This thesis first focuses on exploring effective approaches to improving deep learning models on datasets or domains with inadequate or zero labeled data for both Natural Language Understanding (NLU) and Natural Language Generation (NLG) tasks. More specifically, techniques based on transfer learning have been investigated as the driving forces, and three main dimensions of transfer learning for NLP have all been covered: transfer across domains, transfer across tasks, and transfer across languages. Secondly, we will evaluate the robustness of the current most powerful NLP models for text classification and entailment tasks by proposing a novel adversarial attack method and introducing a new challenge test set via adversarially revising the original data. Overall, the contributions of this thesis are as follows:

- For NLU, we propose a multi-stage and multi-task transfer learning strategy specifically for the multi-choice question answering (MCQA) task. This strategy can in general improve performance on the four most representative MCQA datasets by over 9% in accuracy. Combined with powerful pre-trained large-scale language models, we can achieve close-to-human performance even with only several hundreds of annotated samples. This work was published at AAAI 2020 [Jin et al., 2020a].

- Still for NLU tasks, we have two other works. One is to propose an adversarial domain adaptation method, termed as Dual Adversarial Transfer Network (DATNet), for the low-resource named entity recognition tasks, where we found that this method can be effective for both cross-domain and cross-language transfer learning [Zhou et al., 2019]. The other is to transfer machine reading comprehension (MRC) models to the dialogue state tracking (DST) task for task-oriented dialogue systems, which aims to estimate the current belief state of a dialog given all the preceding conversation. By leveraging the MRC datasets, this method outperforms the existing approaches by a large margin in few-shot scenarios when the availability of in-domain data is limited. More importantly, even without any state tracking data, i.e., a zero-shot scenario, this approach can achieve greater than 90% average accuracy owing to the transferred knowledge from related sources [Gao et al., 2020].

- For NLG, we propose a simple but effective approach to the semi-supervised domain adaptation scenario for machine translation, where the aim is to improve the performance of a translation model on the target domain consisting of only non-parallel data with the help of supervised source domain data. This approach iteratively trains a Transformer-based MT model via three training objectives: language modeling, back-translation, and supervised translation. This method achieved substantial performance improvements, up to +19.31 BLEU over the strongest baseline and +47.69 BLEU over the unadapted model [Jin et al., 2020c].

- To examine the generalization and robustness of current most popularly used NLP models, we present TextFooler, a simple but strong baseline for natural language adversarial attacking [Jin et al., 2020b]. By applying it to two fundamental natural language tasks, text classification and textual entailment, we demonstrate that even the strongest BERT model is still fragile to subtle changes in text samples. Besides, we introduce a new challenge set for the aspect-based sentiment analysis (ABSA) task by adversarially crafting new samples from original data. This new test set can comprehensively and fairly evaluate how robust an ABSA model is, especially for the accurate identification of aspects [Xiang et al., 2020]. We argue that future works on improving the robustness of deep learning models could also contribute to better transfer learning performance.

1.3 Thesis Outline

The rest of this thesis is organized as follows:

In Chapter 2, we provide an overview of background information that is relevant in order to understand the contents of this thesis. We review fundamentals of deep learning and discuss neural network-based methods and tasks in NLP. We further introduce the existing transfer learning methods and current progress for probing the robustness of deep learning models for NLP.

Chapter 3 presents our work on transfer learning for NLU tasks. We first introduce in detail the work on utilizing the multi-stage and multi-task transfer learning framework to improve performance on the low-resource MCQA datasets. We then go through the exploration of transfer learning methods applied to the sequence labeling and dialogue state tracking tasks.

In Chapter 4, we propose a domain adaptation method for the NLG task, specifically the machine translation task. Based on the iterative back-translation strategy, we can significantly improve the translation performance on a target domain that has zero labeled data with the help of supervised source domain data.

In Chapter 5, we focus on examining the generalization and robustness of deep learning models for NLP, which is among the most influential factors for transfer learning performance. We propose a textual adversarial attack method and introduce an adversarially crafted challenge set so as to reveal that even high-performing models can easily crash on subtle changes in the test samples.

Chapter 6 contains our conclusion, where we summarize our findings and provide an outlook into the future.

1.4 Where NLP meets Mechanical Engineering?

We are now in the era when more and more interdisciplinary studies/researches are emerging. Projects/problems with high and broad impacts always integrate information, data, techniques, tools, perspectives, concepts, and/or theories from two or more disciplines or bodies of specialized knowledge to advance fundamental understanding [Klein and Newell, 1997]. Healthcare engineering is one of the most representative examples, covering all engineering disciplines such as biomedical, chemical, civil, computer, electrical, environmental, industrial, information, materials, mechanical, software, and systems engineering [Chyu et al., 2015]. As our technology advances, it is a growing trend that computer science and mechanical engineering should collaborate and contribute together to revolutionary impacts.

Machine learning has already been changing the Mechanical Engineering landscape, providing data-driven insights to understand complex phenomena and more accurate results and analysis, in just a fraction of the time it takes compared to traditional methods. Among various machine learning algorithms, NLP has been playing a pivotal role in establishing the interaction between a human and a machine. Think of human-centered robotics, which studies the intersections of human behavior and machine automation. Any communications between human and robots rely on effective understanding of human natural language by the robots, which is impossible without the help of mature NLP techniques. Autonomous driving is a more specific example in this scenario, where people still need to give commands to vehicles by voice even if no other human inputs are ever needed. Such fluent and efficient conversations or dialogues between human and machines are actually built upon various aspects of text understanding and generating capabilities, such as part-of-speech tagging, named entity recognition, syntactic parsing, coreference resolution, text classification, natural language inference, paraphrasing, discourse relation classification, question answering, machine translation, controlled text generation, etc. Overall, the profound progress in these sub-fields made by decades of effort from the NLP community should ultimately benefit Mechanical Engineering whenever machines need to understand and talk to people.

1.5 Publications

This thesis primarily relates to the following publications:

- **Jin, Di**, Shuyang Gao, Jiun-Yu Kao, Tagyoung Chung and Dilek Z. Hakkani-Tür. "MMM: Multi-stage Multi-task Learning for Multi-choice Reading Comprehension." AAAI (2020).

- **Jin, Di**, Zhijing Jin, Joey Tianyi Zhou and Peter Szolovits. "Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment." AAAI (2020).

- **Jin, Di**, Zhijing Jin, Joey Tianyi Zhou and Peter Szolovits. "A Simple Baseline to Semi-Supervised Domain Adaptation for Machine Translation." ArXiv abs/2001.08140 (2020)

- Zhou, Joey Tianyi, Hao Zhang, **Di Jin**, Hongyuan Zhu, Meng Fang, Rick Siow Mong Goh and Kenneth K Kwok. "Dual Adversarial Neural Transfer for Low-Resource Named Entity Recognition." ACL (2019).

- Gao, Shuyang, Sanchit Agarwal, **Di Jin**, Tagyoung Chung, and Dilek Hakkani-Tur. "From Machine Reading Comprehension to Dialogue State Tracking: Bridging the Gap." Proceedings of the Second Workshop on NLP for Conversational AI at ACL (2020).

- Xing, Xiaoyu, Zhijing Jin, **Di Jin**, Bingning Wang, Qi Zhang and Xuanjing Huang. "Tasty Burgers, Soggy Fries: Probing Aspect Robustness in Aspect-Based Sentiment Analysis." Submitted to EMNLP (2020).

While not directly related, the following articles have also been completed over the course of the PhD:

- Yan, Ming, Hao Zhang, **Di Jin**, and Joey Tianyi 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, pp. 7331-7341. (2020).

- **Jin, Di**, Zhijing Jin, Joey Tianyi Zhou, Lisa Orii, and Peter Szolovits. "Hooks in the Headline: Learning to Generate Headlines with Controlled Styles." ACL (2020).

- Jin, Zhijing*, **Di Jin***, Jonas Mueller, Nicholas Matthews, and Enrico Santus. "IMaT: Unsupervised Text Attribute Transfer via Iterative Matching and Translation." In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 3088-3100. 2019.

- **Jin, Di**, and Peter Szolovits. "Advancing PICO Element Detection in Biomedical Text via Deep Neural Networks." Bioinformatics (2020).

- **Jin, Di** and Peter Szolovits. "Hierarchical Neural Networks for Sequential Sentence Classification in Medical Scientific Abstracts." EMNLP (2018).

- **Jin, Di**, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang and Peter Szolovits. "What Disease does this Patient Have? A New Open Domain Question Answering Dataset on the Medical Domain." Submitted to EMNLP (2020).

- **Jin, Di**, and Peter Szolovits. "Pico element detection in medical text via long short-term memory neural networks." In Proceedings of the BioNLP 2018 workshop, pp. 67-75. 2018.

- **Jin, Di**, Franck Dernoncourt, Elena Sergeeva, Matthew McDermott, and Geeticka Chauhan. "MIT-MEDG at SemEval-2018 task 7: Semantic relation classifi-

*Equal Contributions

cation via convolution neural network." In Proceedings of The 12th International Workshop on Semantic Evaluation, pp. 798-804. 2018.

- Zhou, Joey Tianyi, Hao Zhang, **Di Jin**, Xi Peng, Yang Xiao, and Zhiguo Cao. "Roseq: Robust sequence labeling." IEEE transactions on neural networks and learning systems (2019).
- Yang, Xuewen, Heming Zhang, **Di Jin**, Yingru Liu, Chi-Hao Wu, Jianchao Tan, Dongliang Xie, Jue Wang, Xin Wang. "Fashion Captioning: Towards Generating Accurate Descriptions with Semantic Rewards." Submitted to ECCV (2020).
- Alsentzer, Emily, John R. Murphy, Willie Boag, Wei-Hung Weng, **Di Jin**, Tristan Naumann, and Matthew McDermott. "Publicly available clinical BERT embeddings." arXiv preprint arXiv:1904.03323 (2019).
- Zhang, Hao, Chunyu Fang, Xinlin Xie, Yicong Yang, Wei Mei, **Di Jin**[†], and Peng Fei[†]. "High-throughput, high-resolution deep learning microscopy based on registration-free generative adversarial network." Biomedical optics express 10, no. 3 (2019): 1044-1063.

[†] Corresponding Author

Chapter 2

Background

This chapter provides background knowledge to set the stage for the subsequent chapters. It first gives an overview of NLP tasks we will cover in the subsequent chapters (2.1). Then we introduce the concept of transfer learning, define its taxonomy, and go through its current research progress (2.2). Lastly we provide an overview of robustness of NLP models (2.3).

2.1 Natural Language Processing Tasks

NLP aims to teach computers to understand natural language, however, language itself is conceptual and abstract. We thus need to take a more concrete view by defining a series of specific tasks so that computers can understand a piece of text from different aspects or angles. We will now review the main NLP tasks that will be tackled in this thesis. Examples for these tasks are shown in Table 2.1, 2.2, and 2.3.

2.1.1 Text Classification

Text classification is the process of assigning tags or categories to text that is composed of a contiguous sequence of words according to its content and semantics. It is one of the fundamental tasks in NLP with broad applications such as sentiment analysis, topic classification, spam detection, and intent detection.

Text	**Bill**	**and**	**Peter**	**are**	**hugging**	**one**	**another**
POS Tagging	NNP	CC	NNP	VBP	VBG	CD	DT
Chunking	O	O	O	O	VP	NP	NP
NER	PERSON	O	PERSON	O	O	NUMBER	O
Paraphrasing	Bill and Peter are embracing each other						
NLI-Entailment	There are men showing affection						
NLI-Contradiction	Bill and Peter are playing soccer ball						
NLI-Neutral	Two men are celebrating after a game						
MT to French	Bill et Peter s'embrassent						

Table 2.1: Annotations for all tasks discussed in this thesis (except machine reading comprehension) for an example sentence. NNP: proper noun, singular; CC: coordinating conjunction; VBP: verb, non-3rd person singular present; VBG: verb, gerund or present participle; CD: cardinal number; DT: Determiner; VP: verb phrase; NP: noun phrase.

Sentiment Analysis Sentiment analysis is the task of classifying the polarity of a given text, which is the most popular task among all text classification applications. Usually this polarity is binary (positive or negative) or ternary (positive, negative, or neutral). Most datasets belong to domains that contain a large number of emotive texts such as movie and product reviews or tweets. In review domains, star ratings are generally used as a proxy for sentiment. The advanced variant requires models to analyze the sentiment not only towards the whole text but also with respect to a particular target aspect within the text [Pontiki et al., 2016].

2.1.2 Sequence Labeling

Sequence labeling involves the algorithmic assignment of a categorical label to each word in a text. It can be further divided into the following three tasks:

Part-of-speech (POS) tagging POS tagging is the task of tagging a word in a text with its corresponding part-of-speech, which is a category of words with similar grammatical properties. Common English parts of speech are noun, verb, adjective, adverb, pronoun, preposition, conjunction, etc. Parts of speech can be arbitrarily fine-grained and are typically based on the chosen tag set. And they vary greatly between languages due to cross-lingual differences. The current research trend for

POS tagging is to build one model universally for multiple languages. For example, the currently most popularly used dataset for this task, Universal Dependencies [Nivre et al., 2016], contains more than 150 treebanks in 90 languages. When applying a POS tagger to a new domain, current models particularly struggle with word-tag combinations that have not been seen before.

Chunking Chunking aims to identify continuous spans of tokens that form syntactic units. The main difference between chunking and POS tagging is that chunking focuses more on representing higher order structures such as noun phrases or verb phrases.

Named Entity Recognition (NER) NER is the task of detecting and tagging entities in text with their corresponding type, such as PERSON or LOCATION. Entity categories are pre-defined and differ based on the application. Common categories are person names, organizations, locations, time expressions, monetary values, etc. NER has become a common component of information extraction systems in many domains such as drug-to-drug interaction discovery based on biomedical literature. Although current models have achieved very high performance (>92% F_1) on the canonical CoNLL-2003 newswire dataset [Sang and De Meulder, 2003], current NER systems do not generalize well to new domains such as medical and legislation domains.

Overall, both POS tagging and chunking act mostly on the grammatical and syntactic level, while NER captures more of the semantic and meaning-related aspects of the text.

2.1.3 Paraphrasing

Paraphrasing forms a restatement of the meaning of a text or passage using other words. Applications of paraphrasing are varied including information retrieval, question answering, text summarization, and plagiarism detection. It is also widely used as an approach of data augmentation as it can generate new texts with the same semantics to expand existing corpora.

2.1.4 Natural Language Inference

NLI, also known as textual entailment, aims to examine a directional relation between two text fragments. More specifically, the two texts are termed premise (p) and hypothesis (h), respectively, and NLI classifies the relation between them into three categories: entailment, contradiction, and neutral. The entailment relation holds true if, and only if, a human would be justified in inferring the proposition expressed by h after reading the proposition expressed by p. In contrast, the contradiction relation is chosen when a human reading p would reject the proposition expressed by h. If neither above relations exist, the neutral relation is selected. NLI is essential in tasks ranging from information retrieval to semantic parsing to commonsense reasoning. And it is explicitly or implicitly an important part of many down-stream larger NLP systems such as question answering, multi-document summarization, and dialogue management, as they need to recognize that a particular target meaning can be inferred from different text variants.

2.1.5 Machine Translation

MT investigates the use of software to translate text or speech from one language to another. Since its birth in the 1950s, MT has long been a popular topic in the field of NLP and enjoys many successful applications into real-world production. While no system provides the holy grail of fully automatic high-quality machine translation of unrestricted text, many fully automated systems produce reasonable output. The quality of it can be substantially improved if the domain is restricted and controlled.

2.1.6 Machine Reading Comprehension

The task of machine reading comprehension (MRC) aims to answer comprehension questions by reading over a passage of text. Just as we use reading comprehension tests to measure how well a human has understood a piece of text, it can play the same role for evaluating how well computer systems understand human language. It is an instance of question answering, as it is essentially a question answering problem

CNN/Daily Mail (cloze style)
Passage: (@entity4) if you feel a ripple in the force today, it may be the news that the official @entity6 is getting its first gay character. according to the sci-fi website @entity9, the upcoming novel "@entity11" will feature a capable but flawed @entity13 official named @entity14 who "also happens to be a lesbian." the character is the first gay figure in the official @entity6–the movies, television shows, comics and books approved by @entity6 franchise owner @entity22–according to @entity24, editor of "@entity6" books at @entity28 imprint @entity26.
Question: characters in " " movies have gradually become more diverse
Answer: @entity6

MCTest (multiple choice)
Passage: Timmy liked to play games and play sports but more than anything he liked to collect things. He collected bottle caps. He collected sea shells. He collected baseball cards. He has collected baseball cards the longest. He likes to collect the thing that he has collected the longest the most. He once thought about collecting stamps but never did. His most expensive collection was not his favorite collection. Timmy spent the most money on his bottle cap collection.
Question: Which is Timmy's most expensive collection?
Options: A. Sea Shells; B. Baseball Cards; C. Stamps; D. Bottle Cap
Answer: D. Bottle Cap

SQuAD (span prediction)
Passage: Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24–10 to earn their third Super Bowl title. The game was played on February 7, 2016, at Levi's Stadium in the San Francisco Bay Area at Santa Clara, California....
Question: What is the AFC short for?
Answer: American Football Conference

NarrativeQA (free-form text)
Passage: ...In the eyes of the city, they are now considered frauds. Five years later, Ray owns an occult bookstore and works as an unpopular children s entertainer withWinston; Egon has returned to Columbia University to conduct experiments into human emotion; and Peter hosts a pseudo-psychic television show. Peter's former girlfriend Dana Barrett has had a son, Oscar, with a violinist whom she married then divorced when he received an offer to join the London Symphony Orchestra....
Question: How is Oscar related to Dana?
Answer: He is her son

Table 2.2: A few examples from representative MRC datasets: CNN/Daily Mail [Hermann et al., 2015], MCTest [Richardson et al., 2013], SQuAD [Rajpurkar et al., 2016], and NarrativeQA [Kočiskỳ et al., 2018].

over a short passage of text. Depending on the question and answer format, we can divide existing MRC tasks into four categories:

Cloze style The question contains a placeholder. One example of it can be like "Tottenham manager Juande Ramos has hinted he will allow _____ to leave if the Bulgaria striker makes it clear he is unhappy.". The systems must guess which word or entity completes the question based on the passage, and the answer is either chosen from a pre-defined set of choices or the full vocabulary/ontology.

Multiple choice For this type, several hypothesized options are provided and the correct answer is chosen from them. Each option can be a word, a phrase or a sentence.

Span prediction This category is also referred to as *extractive question answering* and the answer must be a single span (i.e., several consecutive words) in the passage.

Free-form answer This category allows the answer to be any free-text form.

Table 2.2 gives an example for each of the above-mentioned categories.

2.1.7 Dialogue State Tracking

Dialogue state tracking (DST) is a key component in task-oriented dialogue systems, which can track what has happened in a dialog, incorporating system outputs, user speech, context from previous turns, and other external information. Its output is used by the *dialog policy/management* to decide what action the dialogue system should take next. As a kind of context-aware language understanding task, DST aims to extract user goals or intents hidden in human-machine conversation and represent them as a compact dialogue state, i.e., a set of slots and their corresponding values. Table 2.3 gives one example for illustration from a multi-turn and multi-domain dialogue, where the DST system aims to fill in the slot values for a set of predefined slot names of two domains: hotel and attraction. Specifically in this example, the "Hotel" domain contains four slot names: name, book people, book stay, and book day. In the first turn of dialogue, the DST system should be able to identify that “Huntingdon Marriott Hotel” from the user utterance, “I am planning a trip to Cambridge soon and want to stay at Huntingdon Marriott Hotel.”, is the slot value for the slot name of “Name”. Subsequently in the second turn, “6”, “4”, and “Tuesday” should be extracted out from the user utterance by the DST system as the slot values for the three slot names of “Book People”, “Book Stay”, and “Book Day”, respectively. The same pattern also applies to the following turns and other domains.

Turn	Utterance	Hotel				Attraction	
		Name	Book people	Book stay	Book day	Type	Area
1	**S:** Hi! How can I help you? **U:** I am planning a trip to Cambridge soon and want to stay at **Huntingdon Marriott Hotel**.	Huntingdon Marriott Hotel	None	None	None	None	None
2	**S:** Sure, how many days and how many people? **U:** We have **6** people staying for **4** night starting from **Tuesday**.	Huntingdon Marriott Hotel	6	4	Tuesday	None	None
3	**S:** Done! Is there anything else I can help you with today? **U:** Any recommendations if I want to visit a **museum** in the **west** part of town?	Huntingdon Marriott Hotel	6	4	Tuesday	Museum	West
4	**S:** The Museum of Fine Arts is the most popular one, which is close to your hotel. **U:** Cool! I will go there.	Huntingdon Marriott Hotel	6	4	Tuesday	Museum	West

Table 2.3: An example of DST for a multi-turn and multi-domain dialogue. “Hotel” and “Attraction” represent two domains. And each domain contains a set of predefined slot names and the DST system aims to find the corresponding slot values that can be extracted from the user utterances or a ontology that is before-hand built. Utterances starting with “S” are from the system agents while those starting with “U” are from users.

2.2 Transfer Learning

In this section, we will first give a definition of transfer learning, and then provide a taxonomy by reviewing its four prevalent settings in NLP. Lastly, we conduct a literature review over each of the four settings.

2.2.1 Definition

The traditional supervised learning paradigm breaks down when we do not have sufficient labeled data for the desired task or domain to train a reliable model. Transfer learning allows us to deal with this scenario by leveraging the data of some related task or domain, known as the source task and source domain. We transfer the knowledge gained by solving the source task in the source domain to the target task and target domain as illustrated by Figure 2-1. Specifically for neural network based models that are used throughout this thesis, this knowledge relates to the learned representation.

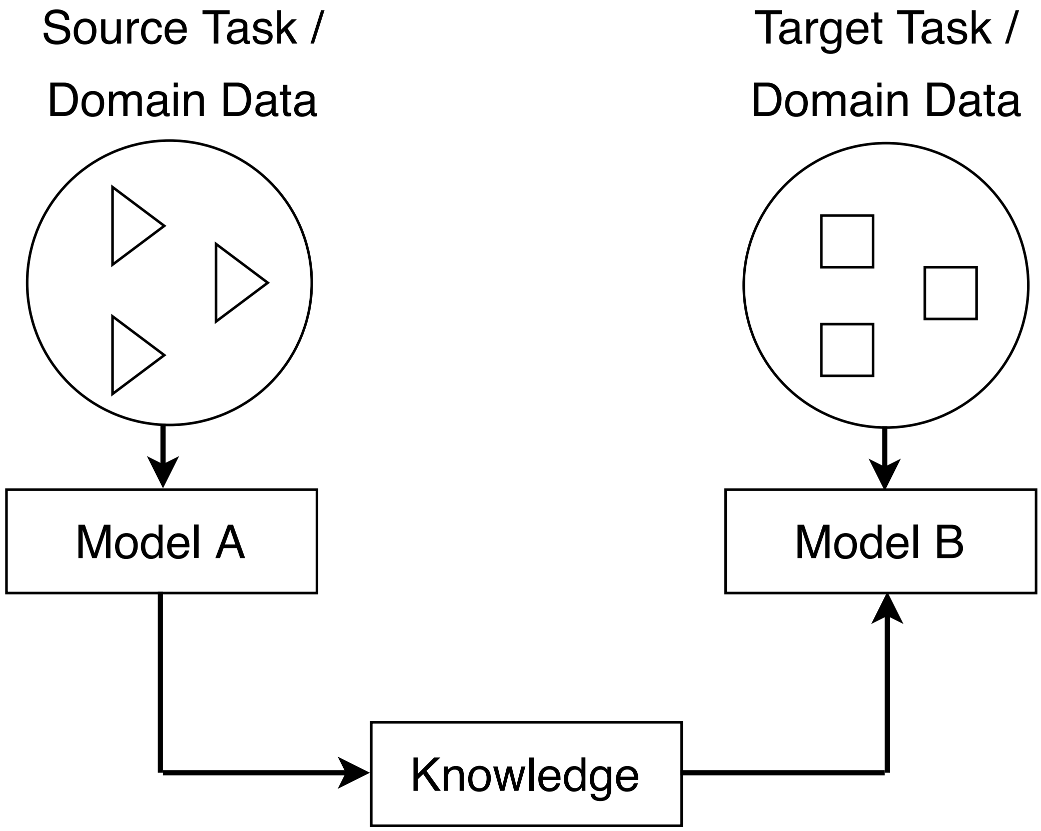


Figure 2-1: The transfer learning setup.

We will now proceed to a more formal definition. At first, we introduce some notations and definitions. We denote a domain of data as $\mathcal{D}$, and it consists of two components: a feature space $\mathcal{X}$ and a marginal probability distribution $P(X)$, where $X = \{x^{(1)}, ..., x^{(n)}\} \in \mathcal{X}$ (n is the number of samples).

Given a specific domain, $\mathcal{D} = \{\mathcal{X}, P(X)\}$, a task $\mathcal{T}$ also consists of two components: a label space $\mathcal{Y}$ and a conditional probability distribution $P(Y|X)$ that cannot be fully observed but can be approximated by learning from the available paired data samples $\{x^{(i)}, y^{(i)}\}$, where $x^{(i)} \in X$ and $y^{(i)} \in \mathcal{Y}$. For example, in the binary sentiment analysis task, $x^{(i)}$ can be a piece of a movie review and $y^{(i)}$ can be either the "positive" or the "negative" label.

We now consider the typical transfer learning scenario, where there is one source domain $\mathcal{D}_S = \{(x_S^{(1)}, y_S^{(1)}), ..., (x_S^{(n_S)}, y_S^{(n_S)})\}$, and one target domain $\mathcal{D}_T = \{(x_T^{(1)}, y_T^{(1)}), ..., (x_T^{(n_T)}, y_T^{(n_T)})\}$. Here $x_S^{(i)} \in \mathcal{X}_S$, $y_S^{(i)} \in \mathcal{Y}_S$, $x_T^{(i)} \in \mathcal{X}_T$, $y_T^{(i)} \in \mathcal{Y}_T$, and $0 \leq n_T \ll n_S$. We now give a unified definition of transfer learning [Pan and Yang, 2009].

Definition 1 (Transfer Learning)**.** *Given a source domain $\mathcal{D}_S$ and its associated learning task $\mathcal{T}_S$, a target domain $\mathcal{D}_T$ and its learning task $\mathcal{T}_T$,* ***transfer learning*** *aims to help improve the learning of the target predictive function $f_T(\cdot) = P(Y_T|X_T)$ in $\mathcal{D}_T$ using the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$, or $\mathcal{T}_S \neq \mathcal{T}_T$.*

2.2.2 Taxonomy

In the above definition, a domain is defined as a pair $\mathcal{D} = \{\mathcal{X}, P(X)\}$, and a task also as a pair $\mathcal{T} = \{\mathcal{Y}, P(Y|X)\}$. When the source and target domains are the same, i.e., $\mathcal{D}_S = \mathcal{D}_T$, and their learning tasks are the same, i.e., $\mathcal{T}_S = \mathcal{T}_T$, the learning problem becomes a normal supervised learning problem, where we expect our data to be i.i.d.. The transfer learning paradigm requires that either the domains or the tasks should be different, and we can thus categorize transfer learning under two sub-settings, *inductive transfer learning* and *transductive transfer learning*, based on different situations between the source and target domains and tasks. Here we give the formal definitions for these two settings:

Definition 2 (Inductive Transfer Learning)**.** *Given a source domain $\mathcal{D}_S$ and its associated learning task $\mathcal{T}_S$, a target domain $\mathcal{D}_T$ and its learning task $\mathcal{T}_T$,* ***inductive transfer learning*** *aims to help improve the learning of the target predictive function $f_T(\cdot) = P(Y_T|X_T)$ in $\mathcal{D}_T$ using the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{T}_S \neq \mathcal{T}_T$.*

Definition 3 (Transductive Transfer Learning)**.** *Given a source domain $\mathcal{D}_S$ and its associated learning task $\mathcal{T}_S$, a target domain $\mathcal{D}_T$ and its learning task $\mathcal{T}_T$,* ***transductive transfer learning*** *aims to help improve the learning of the target predictive* *function* $f_T(\cdot) = P(Y_T|X_T)$ *in* $\mathcal{D}_T$ *using the knowledge in* $\mathcal{D}_S$ *and* $\mathcal{T}_S$*, where* $\mathcal{D}_S \neq \mathcal{D}_T$ *and* $\mathcal{T}_S = \mathcal{T}_T$*.*

Next we discuss each setting in mode detail:

Inductive transfer learning: In this setting, the target task is different from the source task, no matter whether the source and target domains are the same or not. In this case, some labeled data in the target domain are required to induce the objective predictive model for use in the target domain. In this case, the most important distinction is whether the tasks are learned sequentially or simultaneously. Learning multiple tasks at the same time is known as *multi-task learning*, while we will use *sequential transfer learning* to denote the sequential case. We will discuss these two methods later in detail.

Transductive transfer learning: In this setting, the source and target tasks are the same, while the source and target domains are different. In this situation, zero or very few labeled data in the target domain are available while a lot of labeled data in the source domain are available. In addition, according to different situations between the source and target domains, we can further categorize the transductive transfer learning setting into two cases.

1. $\mathcal{X}_S \neq \mathcal{X}_T$. The feature spaces of the source and target domain are different. In the context of NLP, this scenario usually corresponds to the *cross-lingual learning* or *cross-lingual adaptation*.

2. $P(X_S) \neq P(X_T)$ while $\mathcal{X}_S = \mathcal{X}_T$. The marginal probability distributions of source and target domain are different. This scenario is generally known as *domain adaptation*.

In short, the complete taxonomy for transfer learning for NLP can be seen in Figure 2-2.

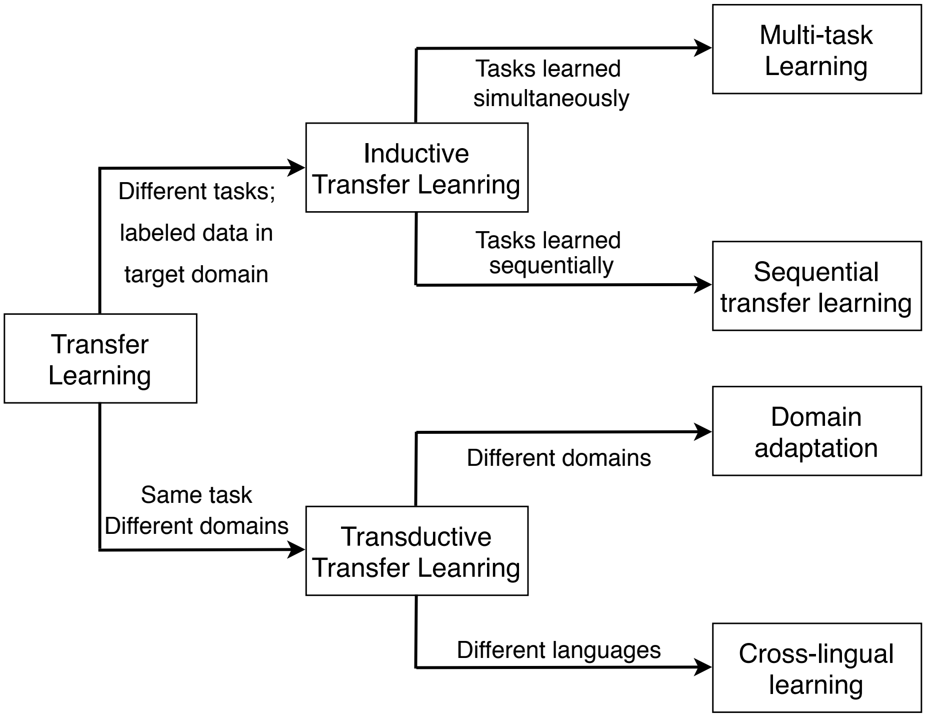

Multi-task Learning
Tasks learned simultaneously
Inductive Transfer Leanring
Different tasks; labeled data in target domain
Tasks learned sequentially
Sequential transfer learning
Transfer Learning
Domain adaptation
Different domains
Same task Different domains
Transductive Transfer Leanring
Different languages
Cross-lingual learning

Figure 2-2: A taxonomy for transfer learning for NLP.

Considering the importance of the three above-mentioned transfer learning settings: multi-task learning, sequential transfer learning, and domain adaptation, we will elaborate them in the following three sub-sections.

2.2.3 Multi-task Learning

2.2.3.1 Introduction

Multi-task learning (MTL) is a natural fit in situations where we are interested in obtaining predictions for multiple tasks at once. Such scenarios are common for instance in autonomous driving perception where we might want to recognize the cars, pedestrians, traffic lights, traffic signs, etc., simultaneously from the same images [Chowdhuri et al., 2019]; or in Bioinformatics where we might want to predict symptoms for multiple diseases altogether [Harutyunyan et al., 2019]. In scenarios such as drug discovery, where tens or hundreds of active compounds should be predicted, MTL accuracy increases continuously with the number of tasks [Ramsundar et al., 2015].

In many situations, however, we only care about performance on one task. In this case, we still would like to find out tasks relevant to the main task as the auxiliary tasks and conduct MTL so that model generalization can be improved by leveraging the domain-specific information contained in the training signals of related tasks [Caruana, 1997].

Suppose we have N tasks in total; for each task, we can obtain its corresponding loss $\mathcal{L}$ using the data from that task. Then the MTL is implemented by optimizing the weighted sum of task losses $\frac{1}{N}\sum_{i=1}^{N}\lambda_i\mathcal{L}_i$, where λ is the weight for each task.

2.2.3.2 Two Parameters Sharing Paradigms for MTL

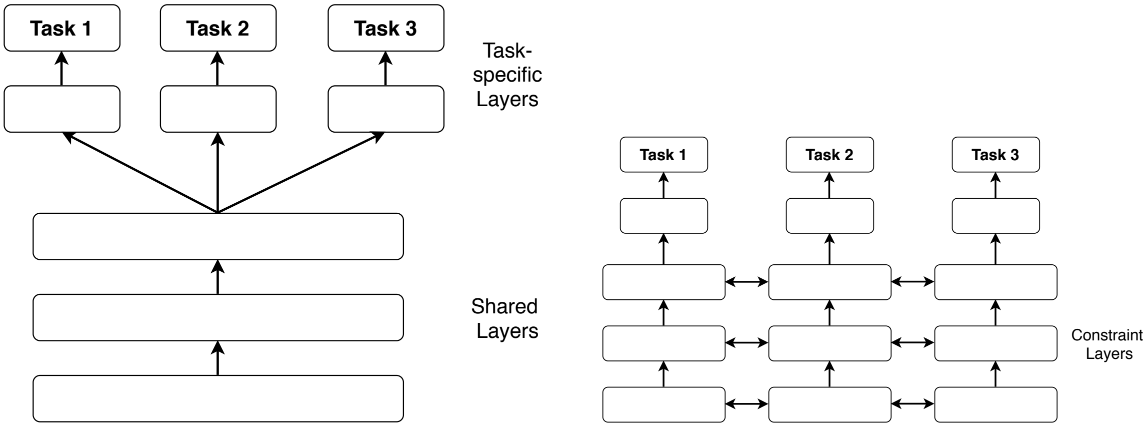
In the context of deep learning, multi-task learning is typically done with either hard or soft parameter sharing of hidden layers.

Hard parameter sharing Hard parameter sharing is the most commonly used approach to MTL in neural networks. It is implemented by sharing the bottom hidden layers across all tasks, while keeping the upper output layers specific to each task, which is illustrated by Figure 2-3a. Hard parameter sharing greatly reduces the risk of overfitting, since the number of parameters can be reduced by the order of N times when there are N tasks.

Soft parameter sharing In this sharing paradigm, each task has its own model with its own parameters. The distance between the parameters of each model is then regularized/minimized in order to encourage the parameters to be similar, which is illustrated by Figure 2-3b. The metrics for measuring the distance can be l_2 distance [Duong et al., 2015] and the trace norm [Yang and Hospedales, 2017].

2.2.3.3 Auxiliary Tasks

When we are looking for auxiliary tasks to help improve the main target task via the MTL framework, the selection criterion is that the auxiliary tasks should be related to the main task in some way and that they should be helpful for predicting



(a) Hard parameter sharing

(b) Soft parameter sharing

Figure 2-3: Two parameter sharing paradigms for MTL

the main task. However, we still do not have a good notion of when two tasks should be considered similar or related. Prior to the deep learning era, there were scattered studies that proposed approaches to judging the relatedness of two tasks for machine learning models. For instance, Ben-David and Schuller [2003] proposes that two tasks are $\mathcal{F}$-related if the data for both tasks can be generated from a fixed probability distribution using a set of transformations $\mathcal{F}$. One example that conforms to this requirement could be object recognition with data from cameras with different positions, angles, and lighting conditions. Xue et al. [2007] define two classification tasks as similar when the two classification boundaries are close, that is, when the weight vectors of two classifiers are similar. For deep learning models, there are no theoretical frameworks that can calculate the task relatedness but many empirical results to demonstrate which two tasks are related and which ones are not. The most comprehensive study on task relatedness to date has been done in computer vision (CV), via sequential transfer learning: Zamir et al. [2018] propose a task taxonomy that organizes 26 CV tasks based on how well a model pretrained on one task transfers to another task. In the following, several most common types of auxiliary tasks used in NLP will be introduced.

2.2.3.4 Common Types of Auxiliary Tasks

Related supervised task Using a related supervised task as an auxiliary task for MTL is the classical choice. Prominent examples include: Changpinyo et al. [2018] empirically demonstrate that jointly training several sequence tagging tasks such as chunking, POS tagging, and NER improves upon either independent or pairwise learning of the tasks; McCann et al. [2018] introduce the Natural Language Decathlon (decaNLP), a challenge that casts ten tasks: question answering, machine translation, summarization, natural language inference, sentiment analysis, semantic role labeling, relation extraction, goal-oriented dialogue, semantic parsing, and commonsense pronoun resolution, as question answering over a context, and they present a new multitask question answering network (MQAN) that jointly learns all tasks without any task-specific modules or parameters.

Adversarial An adversarial loss can be added to any classification based task by crafting small perturbations to the original data that can adversarially maximize the training error [Miy]. By training the models over those artificially created adversarial perturbations, they can be made more robust to unseen noises and thus can generalize better [Zhu et al., 2019]. An adversarial auxiliary task might also help to combat bias and ensure more privacy by encouraging the model to learn representations that do not contain information that would allow the reconstruction of sensitive user attributes [Li et al., 2018b].

Learning the inverse It is interesting to see that learning the inverse of the task together with the main task might also be useful in some circumstances. For example, in unsupervised machine translation, the back-translation technique is the back-bone, where the model is trained to translate in both translation directions: from language A to B and back from B to A [Lample et al., 2017]. Xia et al. [2017] show that this has applications not only to MT, but also to image classification (with image generation as its inverse) and sentiment classification (paired with sentence generation).

Representation learning The goal of an auxiliary task in MTL is to enable the model to learn representations that are shared or helpful for the main task. We can use some unsupervised tasks that are known to induce general-purpose representations such as language modeling and denoised reconstruction to help improve the representation learning. Rei [2017a] propose a sequence labeling framework with a secondary training objective, learning to predict surrounding words for every word in the dataset. This language modeling objective incentivises the system to learn general-purpose patterns of semantic and syntactic composition, which are also useful for improving accuracy on different sequence labeling tasks.

2.2.4 Sequential Transfer Learning

This section gives an overview of sequential transfer learning, arguably the most frequently used transfer learning scenario in NLP and machine learning.

2.2.4.1 Introduction

We define *sequential transfer learning* as the setting where we first train the model on the source task and then subsequently on the target task separately. The goal of sequential transfer learning is to transfer information from the model trained on the source task to improve performance of the target model. When we start to train the model on the target task, we initialize all or most parameters by those optimized in the training phase on the source task. Compared to multi-task learning, sequential transfer learning is useful mainly in three scenarios:

- Data for the source and target tasks are not available at the same time.

- The amount of data in the source task is much larger than that in the target task, in which multi-task learning would be inefficient and the data-imbalance problem would hurt the performance of the target task.

- We need to adapt from one source task to multiple target tasks and these adaptations may not happen at the same time.

Generally, sequential transfer learning is expensive when training the source model, but enables fast adaptation to a target task, while multi-task learning may be expensive when training the target model.

One of the greatest disadvantages of sequential transfer learning over multi-task learning is the catastrophic forgetting problem, which happens when the model is being trained on the target task but gradually forgets the previously seen tasks (i.e., when the model is being trained on the subsequent task, it will gradually lose the information gained by solving previous tasks and thus its performance on them will drop). Continual learning has been actively investigated recently to prevent or alleviate this phenomenon [Parisi et al., 2019].

When the model is trained on the source task, we can call this process the *pre-training* phase, while the transferring process from the source task to the target task is named the *adaptation* phase. In the following, we will introduce these two phase in more detail.

2.2.4.2 Pretraining

In the pretraining phase, we are always facing the choice of the appropriate and useful source task. Over the last decade, various kinds of source tasks have been proposed, where some of them can benefit specific target tasks while others are useful for a wide range of target tasks. And we consider a pretraining task to be *universal* if it helps on most NLP tasks, which is a long-standing challenge in representation learning and has recently received increasing attention.

We can categorize these miscellaneous pretraining tasks into three kinds based on the *source of supervision*:

- **Traditional Supervision**: Traditional supervision requires manually labeling each training example.

- **Distant Supervision**: Large amounts of noisily supervised data can be collected via web-crawling or obtained with heuristics or domain expertise without human annotation, and these distantly labeled data can be used to train the source model.

- **No Supervision**: In this case, it is easy to obtain a large amount of unlabeled text on which unsupervised training can be implemented, such as autoencoding and language modeling.

In the following, we will elaborate these three scenarios:

Supervised Pretraining Supervised pretraining leverages any existing tasks and datasets, which are chosen based on the particular down-stream task. For instance, Zoph et al. [2016] train a machine translation model on a high-resource language pair and then transfer this model to a low-resource language pair. The large open-domain Stanford Question Answering Dataset (SQuAD) has been widely used as the pretraining task to benefit more specialized QA domains [Min et al., 2017; Wiese et al., 2017]. Some pretraining tasks can benefit more than one downstream task and discovering such *universal* supervised pretraining tasks has been a hot topic recently. Here we summarize the most representative examples among them: paraphrasing [Wieting et al., 2016], natural language inference [Cer et al., 2018; Conneau et al., 2017], translation [McCann et al., 2017], constituency parsing [Subramanian et al., 2018], and image captioning [Kiela et al., 2018].

Distantly Supervised Pretraining The most profound example for distantly supervised pretraining is sentiment analysis. Originally, binarized emoticons were used as noisy labels, but later also hashtags and emojis have been used to pretrain the deep neural networks, which enables training models on millions of tweets [Mohammad, 2012; Severyn and Moschitti, 2015; Suttles and Ide, 2013]. Felbo et al. [2017] scale up this strategy to predict a large number of emojis from 1246 million tweets containing one of 64 common emojis, and obtain state-of-the-art performance on 8 benchmark datasets within sentiment, emotion and sarcasm detection using a single pretrained model. Another example could be that Yang et al. [2017a] use a range of external information, such as punctuation, automatic segmentation, and POS, to generate silver labels so that they can pretrain a neural word segmentation model to benefit 6 down-stream datasets.

Unsupervised Pretraining Unsupervised pretraining, is a much more scalable approach and closer to the way humans learn, without requiring millions of labeled examples. It only needs access to unlabeled human-written text and can create labels using sentences and words that compose these texts. It is also referred to as *self-supervised pretraining* since the supervision labels/signals are coming from itself. Unsupervised pretraining actually has a long history in NLP and many breakthroughs along the direction of learning representations of words from unlabeled data can be considered as forms of unsupervised pretraining. In the first half of the last decade, word embeddings were invented and greatly developed via learning the word-word co-occurrence statistics based on matrix factorization [Deerwester et al., 1990; Levy et al., 2015] or statistical language modeling [Mikolov et al., 2013; Pennington et al., 2014]. These word embeddings self-learned from large-scale unlabeled text enable the representations of discrete words in low-dimensional dense and continuous vectors, which is one of the foundations of all current deep learning models for NLP. Later on in the last half decade, contextualized word embeddings were brought out and achieved remarkable improvements over all NLP tasks. They pretrain a whole neural network on unlabeled text via various objectives: sequence autoencoding [Dai and Le, 2015], next (previous) sentence prediction [Kiros et al., 2015], and character/word level conditional/masked language modeling [Devlin et al., 2019; Lample and Conneau, 2019; Peters et al., 2018]. Advancement from the previous normal word embedding to current contextualized word embeddings actually introduces one key paradigm shift: going from just initializing the first layer of our models to pretraining the entire model with hierarchical representations. The pretraining step enables the models to learn complex language phenomena such as compositionality, polysemy, anaphora, long-term dependencies, agreement, negation, and many more [Rogers et al., 2020], which provides the down-stream tasks a much better start point than training from scratch.

2.2.4.3 Adaptation

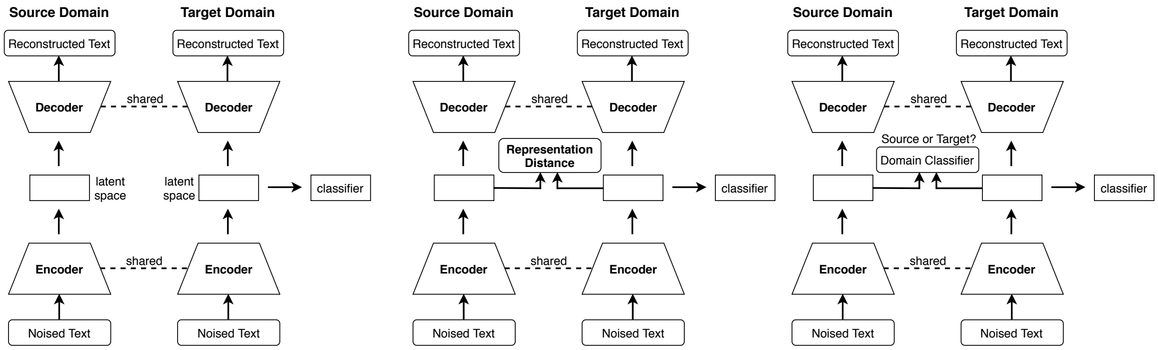
There are many fewer works on the second stage: *adaptation*. In general, there are two main ways to adapt a pretrained model to a target task: *feature extraction* and *fine-tuning*.

- **Feature extraction**: In feature extraction, a model's weights are frozen and the pretrained representations are used in a downstream model similar to classic feature based approaches. For instance, the pretrained word embeddings are used as the features of words in text and they remain static and do not update during the target task training process.

- **Fine-tuning**: In contrast, fine-tuning involves updating the pretrained representations. In this case, the pretrained parameters are used as initialization for the parameters of the model on the downstream task. Some special fine-tuning approaches have been proposed to facilitate this process. For instance, Howard and Ruder [2018] propose several novel techniques, such as discriminative fine-tuning, slanted triangular learning rates, and gradual unfreezing, to retain previous knowledge obtained in pretraining as much as possible and avoid catastrophic forgetting during fine-tuning. Rebuffi et al. [2017] propose residual adapters as a small number of target task specific parameters and only fine-tune them in the adaptation stage while freezing all other parameters, which adds flexibility to the model compared with features extraction but makes the fine-tuning process much cheaper.

2.2.5 Domain Adaptation

In machine learning, training and test data are typically assumed to be i.i.d.. However, there are many real-world cases where models are applied to data with different distributions from the data seen during training. Generally, this would happen when we want to obtain a decent model on a domain with no or very few supervised data and we can find a related domain with a sufficient number of training samples. This



(a) Basic setting (b) Distance minimization (c) Domain-adversary

Figure 2-4: Representation learning approaches to domain adaptation.

gives rise to the problem of domain adaptation, which deals with adapting from a training distribution to a different distribution at test time.

To put it more formally, assume we have two sets of data: a source domain S providing labeled training instances and a target domain T providing instances on which the classifier is meant to be deployed. We do not make the assumption that these are drawn from the same distribution, but rather that S is drawn from a distribution p_S and T from a distribution p_T. The learning problem of domain adaptation consists in finding a function realizing a good transfer from S to T, i.e., it is trained on data drawn from p_S and generalizes well on data drawn from p_T.

There have been many efforts in developing various domain adaptation methods in NLP and we can categorize them into three major types: representation learning, data selection, and self-training. We will introduce them as follows.

2.2.5.1 Representation Learning

Representation learning aims to find a function that can encode the text from different domains into a common latent space so that the commonalities between the source and target domains can be made use of. Glorot et al. [2011] propose the first approach that applied a deep neural network to learn a common representation for domain adaptation. As illustrated by Figure 2-4a, the proposed protocol is to train a shared stacked denoising autoencoder (SDA) to reconstruct the noised input text from both the source and target domains. In this reconstruction process, the encoder is induced to learn the common representation of the source and target domains. In the second step, a linear support vector machine (SVM) model is trained on the latent representation of the source domain training examples and then tested on that of the targer domain test examples. This method has demonstrated decent performance on domain adaptation, and afterwards Chen et al. [2012] propose replacing the SDA with a marginalized denoising autoencoder (mSDA) to resolve two crucial limitations of SDA: high computational cost and lack of scalability to high-dimensional features. Besides reconstructing the noised source and target domain text to the original text in the corresponding domain, Zhou et al. [2016] propose additionally transforming the source domain examples to the target domain and vice versa, which can effectively model the domain-specific features as well as the commonality of domains.

The above-mentioned latent feature learning approaches *implicitly* bring the representations of the source and target domains input samples close to each other by using the same model to reconstruct them. For further improvement, later works propose approaches to *explicitly* enforcing that the latent spaces of the two domains overlap as much as possible, which can be divided into the following two methods:

Representation Distance Minimization Approaches: As shown in Figure 2-4b, this line of works proposes measuring the distance between the latent representation of the source domain examples and that of the target domain, and the minimization of this distance can better lead to domain invariant representations. An early approach in this line by Tzeng et al. [2014] propose using Maximum Mean Discrepancy (MMD) as the measurement metric of this distance, and later on Zhuang et al. [2015] introduce the KL divergence as an alternative. As applications and extensions of this direction, Wang et al. [2018b] minimize a label-aware extension of MMD between the source and target domain hidden representations in an LSTM for medical NER. He et al. [2018] combine this method with self-ensemble bootstrapping for semi-supervised learning, which can make better use of the unlabeled target data for classifier refinement.

Domain-Adversarial Approaches: The most widely used approach in this line employs an adversarial loss [Ganin and Lempitsky, 2015]. As shown in Figure 2-4c, this method adds an auxiliary classifier upon the latent representations, and it is trained to predict the domain of the input example. More importantly, the gradients of this classifier are reversed so that the encoder is encouraged to learn representations that will maximally confuse the classifier and will not allow it to differentiate between the domains. This classification-based adversarial component is very effective at reducing the difference between the source and the target domain distributions; however, it may make the training process unstable. To solve this issue, Arjovsky et al. [2017] propose minimizing the approximated Wasserstein distance (also known as Earth Mover Distance) between the distributions for the source and target domains.

All previously mentioned approaches focus either on mapping representations from one domain to the other, or on learning to extract features that are invariant to the domain from which they were extracted. However, they ignore the individual characteristics of each domain, which also play key roles in prediction. For example, the word “beast” can be a positive indicator of camera quality, but irrelevant to restaurants or movies. Also, “easy” is frequently used in the electronics domain to express positive sentiment (e.g. the camera is easy to use), while expressing negative sentiment in the movie domain (e.g. the ending of this movie is easy to guess). Bousmalis et al. [2016] first bring out this issue and provide a solution to it by partitioning the latent representation of samples into two subspaces: one component which is private to each domain and one which is shared across domains. More specifically, the shared space is obtained by using the above-mentioned techniques such as the denoised auto-encoding, the minimization of representation distance, and the domain-adversarial technique. And the domain-specific space is enforced to be orthogonal to the shared space so that it can store the information that is unique to each domain. This idea was first demonstrated on computer vision tasks, and Kim et al. [2017b] apply it to the Spoken Language Understanding (SLU) system that includes three sub-tasks: domain classification, intent classification, and semantic slots filling, in a single BiLSTM. Liu et al. [2018] expand the domain-specific representations with domain knowledge, collected by attending over a memory network of domain training data.

2.2.5.2 Data Selection

It has been widely observed that not all data in the source domain are equally helpful to create the best-performing system on the target domain [Tan et al., 2017; van der Wees et al., 2017]. Fortunately, this issue can be solved by applying intelligent data selection, which automatically selects the most similar samples in the source domain to the target domain data. The key to the success of data selection relies on accurate evaluation on the similarity of data between the two domains at the instance level. Almost all data selection methods are comprised of two key components: instance representation and similarity metric. In short, we first obtain the representations of instances from both the source and target domains and then we measure the similarity of these representations between two domains with a particular metric. We will elaborate what options we generally have for each component as follows:

Instance Representation:

- **Term distributions:** The relative frequency distributions of terms in the vocabulary have been successfully used to gauge similarity with respect to a target domain [Plank and van Noord, 2011; Wu and Huang, 2016]. The underlying assumption is that similar domains have more terms in common than dissimilar domains. The term distribution of a domain D is a vector $t \in \mathcal{R}^{|V|}$ where t_i is the probability of the i-th word in the vocabulary V appearing in D. Term distributions, however, only capture superficial occurrence statistics, which likely cannot express a more nuanced spectrum of domain similarity.

- **Word embeddings:** A weighted sum of pre-trained word embeddings can be used to represent an instance of text [Perone et al., 2018; Ruder et al., 2017a]. Frequent words can be discounted by weighting the word embedding v_{w_i} of each word w_i occurring in the document d with the word's smoothed inverse probability $\sqrt{\frac{a}{p(w_i)}}$, where $p(w_i)$ is the probability of w_i appearing in domain D and a is a smoothing factor.

- **Sentence embeddings:** Several pre-trained sentence encoders can be utilized to encode the instances into latent representations [Cer et al., 2018; Conneau et al., 2017; Kiros et al., 2015].

- **Autoencoder representations:** Denoising autoencoders have been successfully used in recent work on domain adaptation [Glorot et al., 2011; Zhuang et al., 2015]. Their representations are typically created to be domain-invariant, but might still capture information that is beneficial for modeling domain similarity.

- **Topic distributions:** Topic distributions have proven to be convincing features for POS tagging [Plank and van Noord, 2011] and sentiment analysis [Lu et al., 2011].

Domain Similarity Metrics:

- **Jensen-Shannon divergence:** Jensen-Shannon divergence is one of the most frequently used measures of domain similarity [Remus, 2012] and has been shown to outperform other similarity metrics [Ruder et al., 2017b]. It is a smoothed, symmetric variant of KL divergence, and given two different probability distributions P and Q, it can be written as $D_{JS}(P||Q) = \frac{1}{2}[D_{KL}(P||M) + D_{KL}(Q||M)]$, where $M = \frac{1}{2}(P+Q)$, and D_{KL} is the KL divergence: $D_{KL}(P||Q) = \sum_{i=1}^{n} p_i \frac{p_i}{q_i}$.

- **Cosine similarity:** Cosine similarity is traditionally used to measure the similarity between vectors a and b:

$$cos(a,b) = \frac{a \cdot b}{||a|| \cdot ||b||}.$$

- **Proxy $\mathcal{A}$ distance:** The A distance [Ben-David et al., 2007] aims to identify the subset A in a family of subsets $\mathcal{A}$ on which the source domain distribution P

and the target domain distribution Q differ the most and is defined as follows:

$$d_{\mathcal{A}}(P, Q) = 2 \sup_{A \in \mathcal{A}} |Pr_P(A) - Pr_Q(A)|.$$

In practice, the proxy $\mathcal{A}$ distance is widely used: first of all as many examples from the source domain as the target domain are sampled; then all source domain examples are labeled with 0 while all target domain examples with 1 and a logistic regression model is trained on this balanced binary dataset; finally the probability of belonging to the target domain inferred from the logistic regression model is used as the similarity score for each source domain example.

- **Language model perplexity:** Another similarity metric that has been successfully used in Machine Translation is a sentence's perplexity as determined by language models trained separately on both the source and target domain [Axelrod et al., 2011; Duh et al., 2013; van der Wees et al., 2017]. More specifically, the bilingual cross-entropy difference (CED) is widely used as defined by:

$$CED = (H_{T,s_a} - H_{S,s_a}) + (H_{T,s_b} - H_{S,s_b}),$$

where S and T denote the source and target domains, respectively; $s = (s_a, s_b) \in S$ represents the sentence pairs between the language pairs a and b in the source domain; H_{T,s_a} and H_{S,s_a} denote the perplexity scores of sentence s_a obtained by the language model trained on the target and source domain text of language a, respectively.

- **Reconstruction loss:** A motivation behind this metric is that in an ideal case, if the data from the source domain are similar and useful for the target domain, then one should be able to find a pair of encoding and decoding functions such that the reconstruction errors on the source domain data and the target domain data are both small [Tan et al., 2017]. So we can train a pair of encoder and decoder on the target domain text via the reconstruction task and then rank the similarity of source domain samples by their reconstruction loss.

2.2.5.3 Self-training

Self-training is one of the earliest and simplest approaches to semi-supervised learning [Yarowsky, 1995]. Its application to domain adaptation has covered a variety of tasks: parsing [Reichart and Rappoport, 2007; Sagae, 2010], summarization [Sandu et al., 2010], NER [Ciaramita and Chapelle, 2010], sentiment analysis [He and Zhou, 2011], etc. As the name implies, self-training leverages a model's own predictions on unlabeled data in order to obtain additional information that can be used during training. Typically, unlabeled examples with confident predictions (i.e., that have a probability higher than a threshold) are used as labeled instances during the next iteration.

The main downside of self-training is that the model is not able to correct its own mistakes and errors may be amplified as iterations go on. A solution to this issue is multi-view training [Blum and Mitchell, 1998], which proposes to train different models with different *views* of the data. These views can differ in various ways such as in the features they use, in the architectures of the models, or in the data on which the models are trained. Ideally the views complement each other and the models can collaborate in improving each other's performance.

Tri-training [Zhou and Li, 2005] is one of the best known multi-view training methods. It leverages the agreement of three independently trained models instead of relying on a single model to reduce the bias of predictions on unlabeled data and make the pseudo-labels more accurate. The main requirement for tri-training is that the initial models are diverse, which can be achieved using different model architectures. The most common way to obtain diversity for tri-training is to obtain different variations of the original training data using bootstrap sampling [Ruder and Plank, 2018]. The three models are then trained on these bootstrap samples. An unlabeled data point is added to the training set of a model if the other two models agree on its label. Training stops when the classifiers do not change anymore.

2.2.6 Summary

In short, in this section we have given a comprehensive review of transfer learning. We first introduce its definition, then discuss about its taxonomy by categorizing it into inductive and transductive transfer learning, and finally describe each of the three most important transfer learning settings in detail: multi-task learning, sequential transfer learning, and domain adaptation. This section would serve as the background for Chapter 3 and 4.

2.3 Robustness

? first discovered that deep neural networks can be made to misclassify an image by applying a certain hardly perceptible perturbation and, even worse, the same perturbation can cause a different network, that was trained on a different subset of the dataset, to misclassify the same input. Such a shocking discovery soon sparked an enormous number of follow-up works and discussions around the topic of deep learning robustness [Goodfellow et al., 2015; Papernot et al., 2016]. More surprisingly, such adversaries can not only affect the image classification and recognition of digital images, but also transfer to the physical world and affect real-world systems [Chen et al., 2018]. Athalye et al. [2018] manufactured the first physical adversarial objects using a 3D printing technique and successfully fooled a neural model to make different classification predictions on images that are taken of the same object but from different viewpoints, as illustrated by Figure 2-5.

In addition to such problems in computer vision, more and more evidence pointing to the existence of adversarial attacks in NLP has also been found. Belinkov and Bisk [2018] confront NMT models with synthetic and natural sources of noise and find that state-of-the-art models fail to translate even moderately noisy texts that humans have no trouble comprehending. Jia and Liang [2017] propose an adversarial evaluation scheme for a question answering dataset, SQuAD, and test whether systems can answer questions about paragraphs that contain adversarially inserted sentences, which are automatically generated to distract computer systems without changing

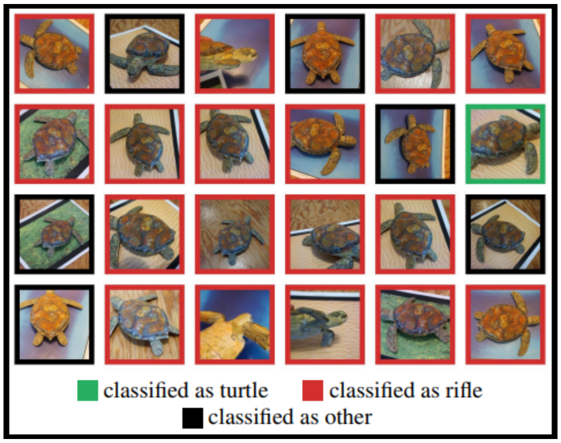
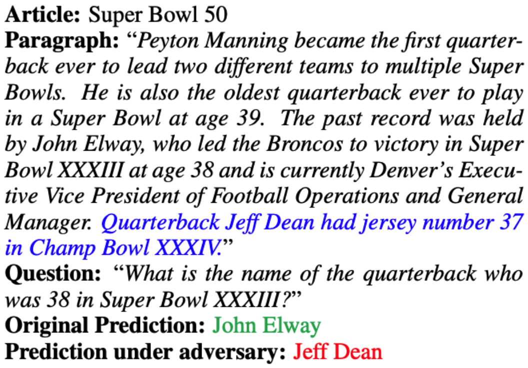


Figure 2-5: Randomly sampled poses of a 3D-printed turtle adversarially perturbed to classify as a rifle at every viewpoint. Randomly sampled poses of a 3D-printed turtle adversarially perturbed to classify as a rifle at every viewpoint. This figure is copied from Athalye et al. [2018].



Article: Super Bowl 50
Paragraph: "*Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager. Quarterback Jeff Dean had jersey number 37 in Champ Bowl XXXIV.*"
Question: "*What is the name of the quarterback who was 38 in Super Bowl XXXIII?*"
Original Prediction: John Elway
Prediction under adversary: Jeff Dean

Figure 2-6: An adversarial example from the SQuAD dataset. An example from the SQuAD dataset. The BiDAF Ensemble model originally gets the answer correct, but is fooled by the addition of an adversarial distracting sentence (in blue). This figure is copied from Jia and Liang [2017].

the correct answer or misleading people. Figure 2-6 gives one example. In this adversarial setting, the accuracy of sixteen published models drops from an average of 75% F1 score to 36%. Over the last two or three years, there has been growing interest in investigating the adversarial robustness of NLP models, including new methods for generating adversarial examples and better approaches to defending against these adversaries [Alzantot et al., 2018; Ebrahimi et al., 2017; Gao et al., 2018; Kuleshov et al., 2018; Li et al., 2018a; Pruthi et al., 2019; Zang et al., 2020].

Chapter 3

Transfer Learning for Natural Language Understanding

This chapter introduces new transfer learning methods for three types of Natural Language Understanding tasks: multi-choice question answering, dialogue state tracking, and named entity recognition. We demonstrate that these methods specialized for each of the three tasks can achieve significant improvements in performance. And they are summarized as follows:

In Section 3.1, we introduce **MMM**, a **M**ulti-stage **M**ulti-task learning framework for multi-choice reading comprehension (this section is based on this work: Jin et al. [2019a]). This method involves two sequential stages: a coarse-tuning stage using out-of-domain datasets and a multi-task learning stage using a larger in-domain dataset to help our model generalize better with limited data. Furthermore, we propose a novel multi-step attention network (MAN) as the top-level classifier for this task. We demonstrate that MMM, as a new transfer learning strategy, can significantly advance the state-of-the-art on the four most representative MCQA datasets.

In Section 3.2, we will demonstrate how to make transfer learning still effective even when the source and target tasks are distant by transferring the machine reading comprehension task to the dialogue state tracking task (this section is based on this work: Gao et al. [2020]).[1] Specifically, we divide the slot types in dialogue state

[1] This work is an extension of my internship project at Amazon Alexa AI: Jin et al. [2020a]. I

into *categorical* or *extractive* to borrow the advantages from both *multiple-choice* and *span-based* reading comprehension models. By leveraging MRC datasets, our method outperforms the existing approaches by a large margin in few-shot scenarios when the availability of in-domain data is limited.

In Section 3.3, we propose a new neural transfer method termed Dual Adversarial Transfer Network for addressing low-resource Named Entity Recognition (this section is based on this work: Zhou et al. [2019]).[2] To address the noisy and imbalanced training data, we propose a novel Generalized Resource-Adversarial Discriminator (GRAD). Additionally, adversarial training is adopted to boost model generalization. In experiments, we examine the effects of different components in DATNet across domains and languages and show that significant improvement can be obtained, especially for low-resource data.

3.1 Multi-stage Multi-task Learning for Multi-choice Reading Comprehension

3.1.1 Introduction

Building a system that comprehends text and answers questions is challenging but fascinating, which can be used to test the machine's ability to understand human language [Chen, 2018; Hermann et al., 2015]. Many machine reading comprehension (MRC) based question answering (QA) scenarios and datasets have been introduced over the past few years, which differ from each other in various ways, including the source and format of the context documents, whether external knowledge is needed, and the format of the answer, to name a few. We can divide these QA tasks into two categories: 1) extractive/abstractive QA such as SQuAD [Rajpurkar et al., 2018] and HotPotQA [Yang et al., 2018]. 2) multiple-choice QA (MCQA) tasks such as

was involved from the idea proposal stage and provided the code and trained model parameters for the multi-choice question answering module.

[2] I was involved from the idea refinement stage, conducted part of experiments, and helped write the manuscript.

MultiRC [Khashabi et al., 2018], and MCTest [Ostermann et al., 2018].

In comparison to extractive/abstractive QA tasks, the answers of the MCQA datasets are in the form of open, natural language sentences and not restricted to spans in text. Various question types exist, such as arithmetic, summarization, common sense, logical reasoning, language inference, and sentiment analysis. Therefore it requires more advanced reading skills for the machine to perform well on this task. Table 3.1 shows one example from one of MCQA datasets, DREAM [Sun et al., 2019b]. To answer the first question in Table 3.1, the system needs to comprehend the whole dialogue and use some common sense knowledge to infer that such a conversation can only happen between classmates rather than brother and sister. For the second question, the implicit *inference* relationship between the utterance *"You'll forget your head if you're not careful."* in the passage and the answer option *"He is too careless."* must be figured out by the model to obtain the correct answer. Many MCQA datasets were collected from language or science exams, which were purposely designed by educational experts and consequently require non-trivial reasoning techniques [Lai et al., 2017]. As a result, the performance of machine readers on these tasks can more accurately gauge comprehension ability of a model.

Recently large and powerful pre-trained language models such as BERT [Devlin et al., 2019] have been achieving the state-of-the-art (SOTA) results on various tasks; however, its potency on MCQA datasets has been severely limited by the availability of enough data. For example, the MCTest dataset has two variants: MC160 and MC500, which are curated in a similar way, and MC160 is considered easier than MC500 [Richardson et al., 2013]. However, BERT-based models perform much worse on MC160 compared with MC500 (8–10% gap) since the data size of the former is about three times smaller. To tackle this issue, we investigate how to improve the generalization of BERT-based MCQA models with the constraint of limited training data using four representative MCQA datasets: DREAM, MCTest, TOEFL, and SemEval-2018 Task 11.

We propose **MMM**, a **M**ulti-stage **M**ulti-task learning framework for **M**ulti-choice question answering. Our framework involves two sequential stages: a coarse-

Dialogue
W: Come on, Peter! It's nearly seven.
M: I'm almost ready.
W: We'll be late if you don't hurry.
M: One minute, please. I'm packing my things.
W: The teachers won't let us in if we are late.
M: Ok. I'm ready. Oh, I'll have to get my money.
W: You don't need money when you are having the exam, do you?
M: Of course not. Ok, let's go... Oh, my god. I've forgot my watch.
W: ***You'll forget your head if you're not careful.***
M: My mother says that, too.
Question 1: What's the relationship between the speakers?
A. Brother and sister. **B.** Mother and son. **C.** Classmates. √
Question 2: What does the woman think of the man?
A. He is very serious. **B.** ***He is too careless.*** √ **C.** He is very lazy.

Table 3.1: Data samples of DREAM dataset. (√: the correct answer)

tuning stage using out-of-domain datasets and a multi-task learning stage using a larger in-domain dataset. For the first stage, we coarse-tuned our model with natural language inference (NLI) tasks. For the second multi-task fine-tuning stage, we leveraged the current largest MCQA dataset, RACE, as the in-domain source dataset and simultaneously fine-tuned the model on both source and target datasets via multi-task learning. Through extensive experiments, we demonstrate that the two-stage sequential fine-tuning strategy is the optimal choice for the BERT-based model on MCQA datasets. Moreover, we also propose a Multi-step Attention Network (MAN) as the top-level classifier instead of the typical fully-connected neural network for this task and obtained better performance. Our proposed method improves BERT-based baseline models by at least 7% in absolute accuracy for all the MCQA datasets (except the SemEval dataset that already achieves 88.1% for the baseline). As a result, by leveraging BERT and its variant, RoBERTa [Liu et al., 2019c], our approach advanced the SOTA results for all the MCQA datasets, surpassing the previous SOTA by at least 16% in absolute accuracy (except the SemEval dataset). Source code is provided at: https://github.com/jind11/MMM-MCQA.

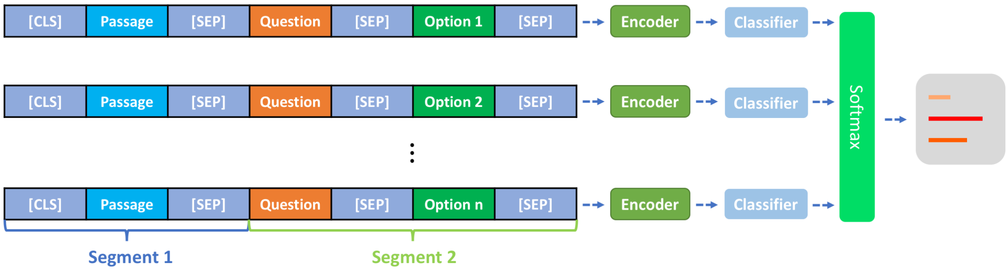


Figure 3-1: Model architecture. "Encoder"is a pre-trained sentence encoder such as BERT. "Classifier" is a top-level classifier.

3.1.2 Methods

In MCQA, the inputs to the model are a passage, a question, and answer options. The passage, denoted as P, consists of a list of sentences. The question and each of the answer options, denoted by Q and O, are both single sentences. A MCQA model aims to choose one correct answer from answer options based on P and Q.

3.1.2.1 Model Architecture

Figure 3-1 illustrates the model architecture. Specifically, we concatenate the passage, question and one of the answer options into a long sequence. For a question with n answer options, we obtain n token sequences of length l. Afterwards, each sequence will be encoded by a sentence encoder to get the representation vector $H \in \mathbb{R}^{d \times l}$, which is then projected into a single value $p = C(H)$ $(p \in \mathbb{R}^1)$ via a top-level classifier C. In this way, we obtain the logit vector $\mathbf{p} = [p_1, p_2, ..., p_n]$ for all options of a question, which is then transformed into the probability vector through a softmax layer. We choose the option with highest logit value p as the answer. Cross entropy loss is used as the loss function. We used the pre-trained bidirectional transformer encoder, i.e., BERT and RoBERTa as the sentence encoder. The top-level classifier will be detailed in the next subsection.

3.1.2.2 Multi-step Attention Network

For the top-level classifier upon the sentence encoder, the simplest choice is a two-layer fully-connected neural network (FCNN), which consist of one hidden layer with *tanh* activation and one output layer without activation. This has been widely adopted when BERT is fine-tuned for downstream classification tasks and performs very well [Devlin et al., 2019]. Inspired from the success of the attention network widely used in the span-based QA task [Seo et al., 2016], we propose the multi-step attention network (MAN) as our top-level classifier. Similar to the dynamic or multi-hop memory network [Kumar et al., 2016; Liu et al., 2017b], MAN maintains a state and iteratively refines its prediction via multi-step reasoning.

The MAN classifier works as follows. A pair of question and answer option together is considered as a whole segment, denoted as QO. Suppose the sequence length of the passage is p and that of the question and option pair is q. We first construct the working memory of the passage $H^P \in \mathbb{R}^{d\times p}$ by extracting the hidden state vectors of the tokens that belong to P from H and concatenating them together in the original sequence order. Similarly, we obtain the working memory of the (question, option) pair, denoted as $H^{QO} \in \mathbb{R}^{d\times q}$. Alternatively, we can also encode the passage and (question, option) pair individually to get their representation vectors H^P and H^{QO}, but we found that processing them in a pair performs better.

We then perform K-step reasoning over the memory to output the final prediction. Initially, the initial state $\mathbf{s}^0$ in step 0 is the summary of H^P via self-attention: $\mathbf{s}^0 = \sum_i \alpha_i H_i^P$, where $\alpha_i = \frac{exp(w_1^T H_i^P)}{\sum_j exp(w_1^T H_j^P)}$. In the following steps $k \in \{1, 2, ..., K-1\}$, the state is calculated by:

$$\mathbf{s}^k = GRU(\mathbf{s}^{k-1}, \mathbf{x}^k), \tag{3.1}$$

where $\mathbf{x}^k = \sum_i \beta_i H_i^{QO}$ and $\beta_i = \frac{exp(w_2^T[\mathbf{s}^{k-1}; H_i^{QO}])}{\sum_j exp(w_2^T[\mathbf{s}^{k-1}; H_j^{QO}])}$. Here $[x; y]$ is concatenation of the vectors x and y. The final logit value is determined using the last step state:

$$P = w_3^T[\mathbf{s}^{K-1}; \mathbf{x}^{K-1}; |\mathbf{s}^{K-1} - \mathbf{x}^{K-1}|; \mathbf{s}^{K-1} \cdot \mathbf{x}^{K-1}]. \tag{3.2}$$

	DREAM	**MCTest**	**SemEval-2018 Task 11**	**TOEFL**	**RACE**
construction method	exams	crowd.	crowd.	exams	exams
passage type	dialogues	child's stories	narrative text	narrative text	written text
# of options	3	4	2	4	4
# of passages	6,444	660	2,119	198	27,933
# of questions	10,197	2,640	13,939	963	97,687
non-extractive answer* (%)	83.7	45.3	89.9	-	87.0

Table 3.2: Statistics of MCQA datasets. (crowd.: crowd-sourcing; $\star$: answer options are not text snippets from reference documents.)

Basically, the MAN classifier calculates the attention scores between the passage and (question, option) pair step by step dynamically such that the attention can refine itself through several steps of deliberation. The attention mechanism can help filter out irrelevant information in the passage against the (question, option) pair.

3.1.2.3 Two Stage Training

We adopt a two-stage procedure to train our model with both in-domain and out-of-domain datasets as shown in Figure 3-2.

Coarse-tuning Stage We first fine-tune the sentence encoder of our model with natural language inference (NLI) tasks. For exploration, we have also tried to fine-tune the sentence encoder on other types of tasks such as sentiment analysis, paraphrasing, and span-based question answering at this stage. However, we found that only the NLI task shows robust and significant improvements for our target multi-choice task. See Section 3.1.5 for details.

Multi-task Learning Stage After the coarse-tuning stage, we simultaneously fine-tune our model on a large in-domain source dataset and the target dataset together via multi-task learning. We share all model parameters including the sentence encoder as well as the top-level classifier for these two datasets.

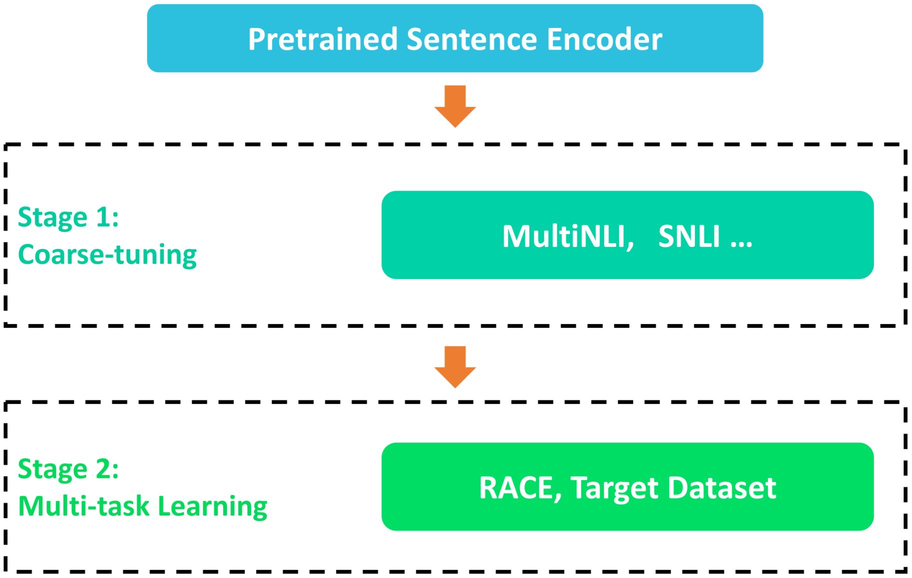


Figure 3-2: Multi-stage and multi-task fine-tuning strategy.

3.1.3 Experimental Setup

3.1.3.1 Datasets

We use four MCQA datasets as the target datasets: DREAM [Sun et al., 2019b], MCTest [Richardson et al., 2013], TOEFL [Ostermann et al., 2018], and SemEval-2018 Task 11 [Tseng et al., 2016], which are summarized in Table 3.2. For the first coarse-tuning stage with NLI tasks, we use MultiNLI [Williams et al., 2017] and SNLI [Young et al., 2014] as the out-of-domain source datasets. For the second stage, we use the current largest MCQA dataset, i.e., RACE [Lai et al., 2017] as the in-domain source dataset. For all datasets, we use the official train/dev/test splits.

3.1.3.2 Speaker Normalization

Passages in the DREAM dataset are dialogues between two persons or more. Every utterance in a dialogue starts with the speaker's name. For example, in the utterance "m: How would he know?", "m" is the abbreviation of "man" indicating that this utterance is from a man. More than 90% of utterances have the speaker names as "w," "f," and "m," which are all abbreviations. However, the speaker names mentioned in the questions are full names such as "woman" and "man." In order to make it clear for the model to learn which speaker the question is asking about, we used a speaker normalization strategy by replacing "w" or "f" with "woman" and "m" with "man" for the speaker names in the utterances. We found this simple strategy is quite effective, providing us with 1% improvement. We will always use this strategy for the DREAM dataset for our method unless explicitly mentioned.

3.1.3.3 Multi-task Learning

For the multi-task learning stage, at each training step, we randomly selected a dataset from the two datasets (RACE and the target dataset) and then randomly fetched a batch of data from that dataset to train the model. This process was repeated until the predefined maximum number of steps or the early stopping criterion has been met. We adopted the proportional sampling strategy, where the probability of sampling a task is proportional to the relative size of each dataset compared to the cumulative size of all datasets [Liu et al., 2019b].

3.1.3.4 Training Details

We used a linear learning rate decay schedule with warm-up proportion of 0.1. We set the dropout rate as 0.1. The maximum sequence length is set to 512. We clipped the gradient norm to 5 for the DREAM dataset and 0 for other datasets. The learning rate and number of training epochs vary for different datasets and encoder types, which are summarized in the Table 3.3. The model architecture and training settings for the NLI task are the same as those in [Devlin et al., 2019].

More than 90% of passages have more than 512 words in the TOEFL dataset, which exceed the maximum sequence length that BERT supports, thus we cannot process the whole passage within one forward pass. To solve this issue, we propose the sliding window strategy, in which we split the long passage into several snippets of length 512 with overlaps between subsequent snippets and each snippet from the same passage will be assigned with the same label. In the training phase, all snippets will be used for training, and in the inference phase, we aggregate the logit vectors of all snippets from the same passage and pick the option with the highest logit value as the prediction. In experiments, we found the overlap of 256 words is the optimal, which can improve the BERT-Base model from accuracy of 50.0% to 53.2%. We adopted this sliding window strategy **only** for the TOEFL dataset.

Datasets	BERT-Base	BERT-Large	RoBERTa-Large
DREAM	2e-5 / 8	2e-5 / 8	1e-5 / 10
MCTest	1e-5 / 8	5e-6 / 8	5e-6 / 10
TOEFL	5e-6 / 8	1e-5 / 8	5e-6 / 10
SemEval	2e-5 / 8	2e-5 / 8	1e-5 / 10
RACE	5e-5 / 5	2e-5 / 5	1e-5 / 10

Table 3.3: Optimal learning rate (left number) and number of training epochs (right number) for different datasets and encoder types.

3.1.4 Results

We first evaluate our method on the DREAM dataset. The results are summarized in Table 3.4. In the table, we first report the accuracy of the SOTA models in the leaderboard. We then report the performance of our re-implementation of fine-tuned models as another set of strong baselines, among which the RoBERTa-Large model has already surpassed the previous SOTA. For these baselines, the top-level classifier is a two-layer FCNN for BERT-based models and a one-layer FCNN for the RoBERTa-Large model. Lastly, we report model performance of our proposed method, ***MMM***, which uses all our improved techniques: MAN classifier + speaker normalization + two stage learning strategies. As direct comparisons, we also list the accuracy increment between MMM and the baseline with the same sentence encoder, shown in parentheses, from which we can see that the performance augmentation is over 9% for BERT-Base and BERT-Large. Although the RoBERTa-Large baseline has already outperformed the BERT-Large baseline by around 18%, MMM gives us another ∼4% improvement, pushing the accuracy closer to the human performance. Overall, MMM has achieved a new SOTA, i.e., test accuracy of 88.9%, which exceeds the previous best by 16.9%.

Model	**Dev**	**Test**
FTLM++ [Sun et al., 2019b]	58.1	58.2
BERT-Large [Devlin et al., 2019]	66.0	66.8
XLNet [Yang et al., 2019]	-	72.0
BERT-Base	63.2	63.2
BERT-Large	66.2	66.9
RoBERTa-Large	85.4	85.0
BERT-Base+MMM	72.6 (9.4)	72.2 (9.0)
BERT-Large+MMM	75.5 (9.3)	76.0 (9.1)
RoBERTa-Large+MMM	**88.0** (2.6)	**88.9** (3.9)
Human Performance	93.9*	95.5*
Ceiling Performance	98.7*	98.6*

Table 3.4: Accuracy on the DREAM dataset. Performance marked by $\star$ is reported by [Sun et al., 2019b]. Numbers in parentheses indicate the accuracy increased by MMM compared to the baselines.

Dataset	**Previous Single-Model SOTA**	**Baselines**			**+MMM**			**Human**
		BERT-B	**BERT-L**	**RoBERTa-L**	**BERT-B**	**BERT-L**	**RoBERTa-L**	**Scores**
MC160	80.0 [Sun et al., 2018]	63.8	65.0	81.7	85.4 (21.6)	89.1 (**24.1**)	**97.1** (15.4)	97.7*
MC500	78.7 [Sun et al., 2018]	71.3	75.2	90.5	82.7 (11.4)	86.0 (10.8)	**95.3** (4.8)	96.9*
TOEFL	56.1 [Chung et al., 2017]	53.2	55.7	64.7	60.7 (7.5)	66.4 (10.7)	**82.8** (18.1)	–
SemEval	88.8 [Sun et al., 2018]	88.1	88.7	94.0	89.9 (1.8)	91.0 (2.3)	**95.8** (1.8)	98.0†

Table 3.5: Performance in accuracy (%) on test sets of other datasets: MCTest (MC160 and MC500), TOEFL, and SemEval. Performance marked by $\star$ is reported by [Richardson et al., 2013] and that marked by † is from [Ostermann et al., 2018]. Numbers in the parentheses indicate the accuracy increased by MMM. "-B" means the base model and "-L" means the large model.

Settings	**DREAM**	**MC160**	**MC500**
Full Model	**72.6**	**86.7**	**83.5**
– Second-Stage Multi-task Learning	68.5	72.5	78.0
– First-Stage Coarse-tuning on NLI	69.5	80.8	81.8
– MAN	71.2	85.4	81.5
– Speaker Normalization	71.4	—	—

Table 3.6: Ablation study on the DREAM and MCTest-MC160 (MC160) datasets. Accuracy (%) is on the development set.

We also test our method on three other MCQA datasets: MCTest including MC160 and MC500, TOEFL, and SemEval-2018 Task 11. The results are summarized in Table 3.5. Similarly, we list the previous SOTA models with their scores for comparison. We compared our method with the baselines that use the same sentence encoder. Except for the SemEval dataset, our method can improve the BERT-Large model by at least 10%. For both MCTest and SemEval datasets, our best scores are very close to the reported human performance. The MC160 and MC500 datasets were curated in almost the same way [Richardson et al., 2013] with only one difference that MC160 is around three times smaller than MC500. We can see from Table 3.5 that both the BERT and RoBERTa baselines perform much worse on MC160 than MC500, although questions in MC160 are somewhat easier to be solved than those in MC500 [Richardson et al., 2013]. We think the reason is that the data size of MC160 is not large enough to fine-tune the large models well because they have a huge number of trainable parameters. However, by leveraging the transfer learning techniques we proposed, we can significantly improve the generalization capability of BERT and RoBERTa models on the small datasets so that the best performance of MC160 can even surpass that of MC500. This demonstrates the effectiveness of our method.

To better understand why MMM can be successful, we conducted an ablation study by removing one feature at a time from the BERT-Base model. The results are shown in Table 3.6. We see that the removal of the second stage multi-task learning part hurts our method most significantly, indicating that the majority of improvement is coming from the knowledge transferred from the in-domain dataset. The first stage of coarse-tuning using NLI datasets is also very important, which provides the model with enhanced language inference ability. As for the top-level classifier, i.e., the MAN module, if we replace it with a typical two-layer FCNN as in [Devlin et al., 2019], we have 1–2% performance drop. Lastly, for the DREAM dataset, the speaker normalization strategy gives us another ∼1% improvement.

3.1.5 Discussion

3.1.5.1 Why does natural language inference help?

As shown in Table 3.6, coarse-tuning on NLI tasks can help improve the performance of MCQA. We conjecture one of the reasons is that, in order to pick the correct answer, we need to rely on the language inference capability in many cases. As an example in Table 3.1, the utterance highlighted in the bold and italic font in the dialogue is the evidence sentence from which we can obtain the correct answer to Question 2. There is no token overlap between the evidence sentence and the correct answer, indicating that the model cannot solve this question by surface matching. Nevertheless, the correct answer is an *entailment* to the evidence sentence while the wrong answers are not. Therefore, the capability of language inference enables the model to correctly predict the answer. We can deem the passage and the pair of (question, answer) as a pair of *premise* and *hypothesis*. Then the process of choosing the right answer to a certain question is similar to the process of choosing the *hypothesis* that can best entail the *premise*. In this sense, part of the MCQA task can be deemed to be an NLI task. This also agrees with the argument that NLI is a fundamental ability of a natural language processing model and it can help support other tasks that require a higher level of language processing abilities [Welleck et al., 2018]. We provided several more examples that require language inference reading skills in Table 3.7; they are wrongly predicted by the BERT-Base baseline model but can be correctly solved by exposing the model to NLI data with the coarse-tuning stage. These examples reveal to us that exposing the model to NLI data can help enhance its language inference ability, which is required by all these examples to get correct answers.

Dialogue 1: man: Wonderful day, isn't it? Want to join me for a swim? woman: ***If you don't mind waiting while I get prepared.***
Question: What does the woman mean? **A.** She is too busy to go. **B.** She doesn't want to wait long. × **C.** She's willing to go swimming. √
Dialogue 2: woman: Shall we go to a play or to a movie? man: ***It's all the same to me.***
Question: What does this man mean? **A.** It makes no difference to him which they go to. √ **B.** He does not want to go to either one. × **C.** The play and the movie are about the same subject.
Dialogue 3: woman: I'm sorry, Mr Wilson. I got up early but the bus was late. man: ***Your bus is always late, Jane.***
Question: What does the man mean? **A.** Jane used the same excuse again. √ **B.** Jane stayed up too late last night. **C.** Jane always gets up early. ×

Table 3.7: Examples from the DREAM dataset. √ marks the correct answer and the answer chosen by the NLI data enhanced BERT-Base model while × marks the answer predicted by the BERT-Base baseline model. These examples are wrongly solved by the BERT-Base baseline model but get correct predictions by inserting the first stage of coarse-tuning using NLI data.

3.1.5.2 Can other tasks help with MCQA?

By analyzing the MCQA datasets, we found that some questions ask about the attitude of one person towards something and in some cases, the correct answer is simply a paraphrase of the evidence sentence in the passage. This finding naturally leads to the question: could other kinds of tasks such as sentiment classification, or paraphrasing also help with MCQA problems?

To answer this question, we select several representative datasets for five categories as the up-stream tasks: sentiment analysis, paraphrase, span-based QA, NLI, and

MCQA.[3] We conduct experiments where we first train the BERT-Base models on each of the five categories and then further fine-tune our models on the target dataset: DREAM and MC500 (MCTest-MC500). For the sentiment analysis category, we used the Stanford Sentiment Treebank (SST-2) dataset from the GLUE benchmark [Wang et al., 2018a] (around 60k train examples) and the Yelp dataset[4] (around 430k train examples). For the paraphrase category, three paraphrasing datasets are used from the GLUE benchmark: Microsoft Research Paraphrase Corpus (MRPC), Semantic Textual Similarity Benchmark (STS-B), and Quora Question Pairs (QQP), which are denoted as "GLUE-Para.". For the span-based QA, we use the SQuAD 1.1, SQuAD 2.0[5], and MRQA[6] which is a joint dataset including six popular span-based QA datasets, including HotpotQA, NaturalQuestionsShort, NewsQA, SearchQA, SQuAD 1.1, and TriviaQA-web.

Table 3.8 summarizes the results. We see that sentiment analysis datasets do not help much with our target MCQA datasets. But the paraphrase datasets do bring some improvements for MCQA. For span-based QA, only SQuAD 2.0 helps to improve the performance of the target dataset. Interestingly, although MRQA is much larger than other QA datasets (at least six times larger), it makes the performance worst. This suggests that span-based QA might not be the appropriate source tasks for transfer learning for MCQA. We hypothesize that this could be due to the fact that most of the questions are non-extractive (e.g., 84% of questions in DREAM are non-extractive) while all answers are extractive in the span-based QA datasets.

For completeness of our experiments, we also used various NLI datasets: MultiNLI, SNLI, Question NLI (QLI), Recognizing Textual Entailment (RTE), and Winograd NLI (WNLI) from the GLUE benchmark. We used them in three kinds of combinations: MultiNLI alone, MultiNLI plus SNLI denoted as "NLI", and combining all five datasets together, denoted as "GLUE-NLI". As the results show in Table 3.8, NLI and GLUE-NLI are comparable and both can improve performance on the target dataset

[3] We list NLI and MCQA here to explore their individual contribution to the improvement of our target dataset.

[4] https://www.yelp.com/dataset/challenge

[5] https://rajpurkar.github.io/SQuAD-explorer/

[6] https://mrqa.github.io/

Task Type	**Dataset Name**	**DREAM**	**MC500**
-	Baseline	63.2	69.5
Sentiment Analy.	SST-2	62.7	69.5
	Yelp	62.5	71.0
Paraphrase	GLUE-Para.	64.2	72.5
Span-based QA	SQuAD 1.1	62.1	69.5
	SQuAD 2.0	64.0	74.0
	MRQA	61.2	68.3
NLI	MultiNLI	67.0	79.5
	NLI	68.4	80.0
	GLUE-NLI	68.6	79.0
Combination	GLUE-Para.+NLI	68.0	79.5
Multi-choice QA	RACE	**70.2**	**81.2**

Table 3.8: Transfer learning results for DREAM and MC500. The BERT-Base model is first fine-tuned on each source dataset and then further fine-tuned on the target dataset. Accuracy is on the the development set. A two-layer FCNN is used as the classifier.

by a large margin.

Lastly, among all these tasks, using other data for the MCQA task itself, i.e., pretraining on the RACE dataset, can help most to boost performance. This result agrees with the intuition that the in-domain dataset can be the ideal data for transfer learning.

In conclusion, we find that for out-of-domain datasets, the NLI datasets can be most helpful to the MCQA task, indicating that natural language inference capability should be an important foundation of the MCQA systems. A larger in-domain dataset, i.e. another MCQA dataset, can also be very useful.

3.1.5.3 NLI dataset helps with convergence

The first stage of coarse-tuning with NLI data can not only improve the accuracy but also help the model converge faster and better. Especially for the BERT-Large and RoBERTa-Large models that have many more trainable parameters, convergence is very sensitive to the optimization settings. However, with the help of NLI datasets, convergence for large models is no longer an issue, as shown in Figure 3-3. Under the same optimization hyper-parameters, compared with the baseline, coarse-tuning

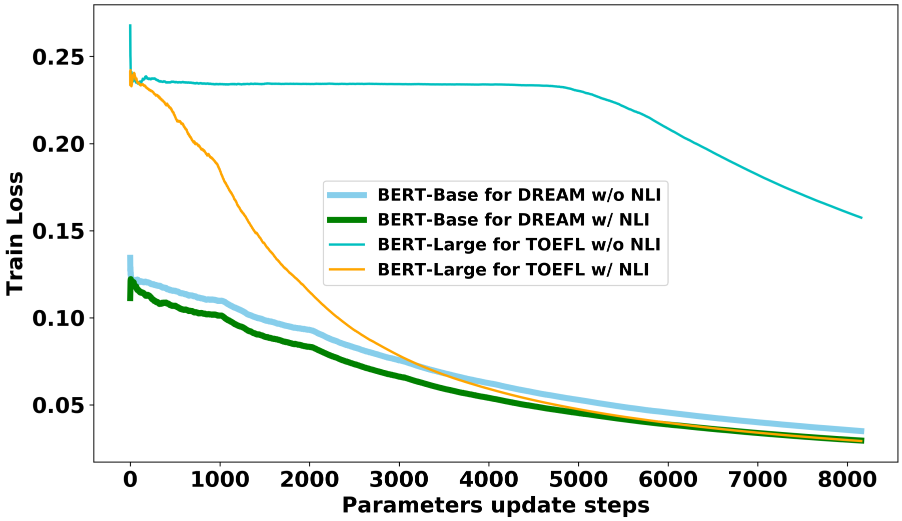


Figure 3-3: Train loss curve with respect to optimization steps. With prior coarse-tuning on NLI data, convergence becomes much faster and easier.

can make the training loss of the BERT-Base model decrease much faster. More importantly, for the BERT-Large model, without coarse-tuning, the model does not converge at all during the first several thousands of iterations, which can be completely resolved with the help of NLI data.

3.1.5.4 Multi-stage or Multi-task

In a typical scenario where we have one source and one target dataset, we naturally have a question about whether we should simultaneously train a model on them via multi-task learning or first train on the source dataset then on the target sequentially. Many previous works adopted the latter way. Chung et al. [2017]; Phang et al. [2018]; Sun et al. [2018] and Chung et al. [2017] demonstrated that the sequential fine-tuning approach outperforms the multi-task learning setting in their experiments. However, we had contradictory observations in our experiments. Specifically, we conducted a pair of control experiments: one is that we first fine-tune the BERT-Base model on the source dataset RACE and then further fine-tune on the target dataset, and the other is that we simultaneously train the model on RACE and the target dataset via multi-task learning. The comparison results are shown in Table 3.9. We see that compared

Setting Configuration	**DREAM**	**MC160**	**MC500**
BERT-Base → RACE → Target	70.2	80.0	81.2
BERT-Base → {RACE, Target}	**70.7**	**80.8**	**81.8**
BERT-Base → {RACE, Target, NLI}	70.5	87.0	82.5
BERT-Base → NLI → {RACE, Target}	**71.2**	**88.3**	**83.5**

Table 3.9: Comparison between multi-task learning and sequential fine-tuning. BERT-Base model is used and the accuracy is on the development set. Target refers to the target dataset in transfer learning. A two-layer FCNN instead of MAN is used as the classifier.

with sequential fine-tuning, the multi-task learning achieved better performance. We conjecture that in the sequential fine-tuning setting, while the model is being fine-tuned on the target dataset, some information or knowledge learned from the source dataset may be lost since the model is no longer exposed to the source dataset in this stage. In comparison, this information can be kept in the multi-task learning setting and thus can better help improve the target dataset.

Now that the multi-task learning approach outperforms the sequential fine-tuning setting, we naturally arrive at another question: what if we merged the coarse-tuning and multi-task learning stages together? That is, what if we simultaneously trained on the NLI, the source datasets, and the target datasets alltogether under the multi-task learning framework? We also conducted a pair of control experiments for investigation. The results in Table 3.9 show that casting the fine-tuning process on these three datasets into separate stages performs better, indicating that multi-stage training is also necessary. Considering that the NLI dataset is an out-of-domain dataset while RACE is in-domain with respect to the target datasets, we can obtain a good practice: we first separate the source datasets into two categories: out-of-domain and in-domain, based on the type of the target dataset; then we can adopt a multi-stage training strategy, that is, first fine-tune the model on the out-of-domain source datasets, then fine-tune on the in-domain source datasets and the target dataset together via multi-task learning.

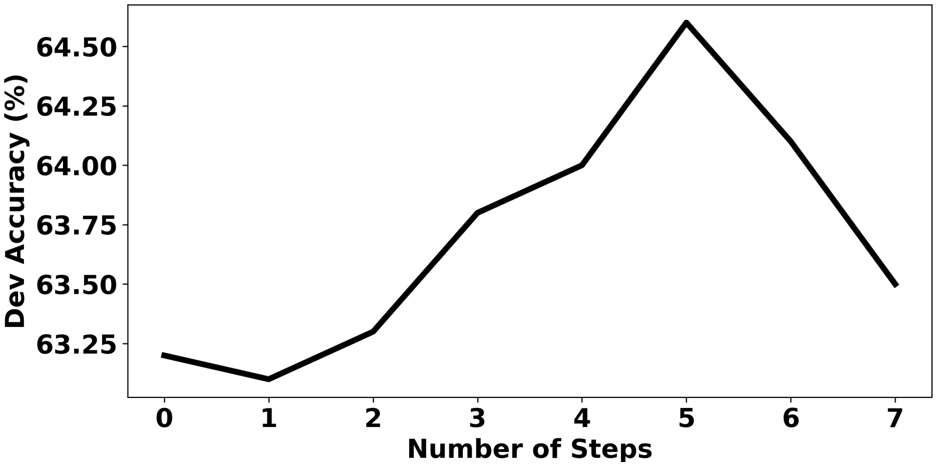


Figure 3-4: Effects of the number of reasoning steps for the MAN classifier. 0 steps means using FCNN instead of MAN. The BERT-Base model and DREAM dataset are used.

3.1.5.5 Multi-steps reasoning is important

Previous results show that the MAN classifier shows improvement compared with the FCNN classifier, but we are also interested in how the performance changes while varying the number of reasoning steps K, as shown in Figure 3-4. $K = 0$ means that we do not use MAN but FCNN as the classifier. We observe that there is a gradual improvement as we increase $K = 1$ to $K = 5$, but after 5 steps the improvements start to degrade. This verifies that an appropriate number of steps of reasoning is important for the memory network to reflect its benefits.

3.1.5.6 Could the source dataset be benefited?

So far we have been discussing the case where we do multi-task learning with the source dataset RACE and various much smaller target datasets to help improve the targets. We also want to see whether our proposed techniques can also benefit the source dataset itself. Table 3.10 summarizes the results of the BERT-Base model on the RACE dataset obtained by adding the coarse-tuning stage, adding the multi-task training together with DREAM, and adding the MAN module. From this table, we see that all three techniques can yield improvements over the baseline model for the source dataset RACE, among which the NLI coarse-tuning stage can help elevate the

Settings	RACE-M	RACE-H	RACE
BERT-Base	73.3	64.3	66.9
+NLI	**74.2**	**66.6**	**68.9**
+DREAM	72.4	66.1	67.9
+MAN	71.2	**66.6**	67.9

Table 3.10: Ablation study for the RACE dataset. The accuracy is on the development set. All parts of MMM improve this source dataset. RACE-M and RACE-H are subsets of the RACE dataset, where RACE-M is collected from middle school examinations while RACE-H is from high school.

Model	RACE-M	RACE-H	RACE
Official Reports:			
BERT-Base	71.7	62.3	65.0
BERT-Large	76.6	70.1	72.0
XLNet-Large	85.5	80.2	81.8
RoBERTa-Large	86.5	81.3	83.2
BERT-Base+MMM	74.8	65.2	68.0
BERT-Large+MMM	78.1	70.2	72.5
XLNet-Large+MMM	86.8	81.0	82.7
RoBERTa-Large+MMM	**89.1**	**83.3**	**85.0**

Table 3.11: Comparison of the test accuracy of the RACE dataset between our approach MMM and the official reports that are from the dataset leaderboard.

scores most.

Since we found all parts of MMM can work well for the source dataset, we tried to use them to improve the accuracy on RACE. The results are shown in Table 3.11. We used four kinds of pre-trained sentence encoders: BERT-Base, BERT-Large, XLNet-Large, and RoBERTa-Large. For each encoder, we listed the official report of scores from the leaderboard. Compared with the baselines, MMM leads to improvements ranging from 0.5% to 3.0% in accuracy. Our best result is obtained by the RoBERTa-Large encoder.

Major Types	Sub-types	Percent	Accuracy
Matching	Keywords	23.3	94.3
	Paraphrase	30.7	84.8
Reasoning	Arithmetic	12.7	73.7
	Common Sense	10.0	60.0
	Others	23.3	77.8

Table 3.12: Error analysis on DREAM. The column of "Percent" reports the percentage of question types among 150 samples that are from the development set of the DREAM dataset that are wrongly predicted by the BERT-Base baseline model. The column of "Accuracy" reports the accuracy of our best model (RoBERTa-Large+MMM) on these samples.

3.1.5.7 Error Analysis

In order to investigate how well our model performs for different types of questions, we did an error analysis by first randomly selecting 150 samples from the common wrong predictions on the development set of the DREAM dataset, obtained by three BERT-Base baseline models, each of which was individually trained with different random seeds. We then manually classified them into several question types based on the following criterion:

- **Matching:**
  - **Keywords:** The correct answer is a phrase and can match a span of text in the passage.
  - **Paraphrase:** The correct answer is a sentence and is a paraphrase to the evidence sentence in the passage.
- **Reasoning:**
  - **Arithmetic:** The correct answer is a number and some calculations must be conducted to get it.
  - **Common Sense:** Some common sense is needed to answer the question.

– **Others:** Other kinds of questions that need some reasoning to obtain the answer.

The annotation results are shown in Table 3.12. And we can see that the BERT-Base baseline model still does not do well on matching problems. We then evaluate our best model on these samples and report the accuracy of each question type in the last column of Table 3.12. We find that our best model can improve upon every question type significantly, especially for the matching problems, and most surprisingly, our best model can even greatly improve its ability on solving the arithmetic problems, achieving an accuracy of 73.7%.

Dialogue:
man: Good morning. May I help yon? woman: I'd like to rent a car, please. man: Okay. Full-size, mid-size, or compact, madam? woman: Compact is OK. What's the rate? man: 78 dollars a day. woman: And I'd like to have insurance just in case. man: If you want full coverage insurance, it will be 8 dollars per day. woman: All right, I'll take that, too. man: OK. Please fill in this form.
Question: How much will the woman pay in total? **A.** 70 dollars. **B.** 78 dollars. **C.** 86 dollars. √
Model Prediction: C. 86 dollars.

Table 3.13: Example of arithmetic question correctly solved by our best model. This example is from the development set of the DREAM dataset. √ marks the correct answer.

By evaluating the accuracy of our best model on each of these question types, we found our model can even do very well on the arithmetic problems. In order to verify whether our model really has the ability of doing math, we sampled some arithmetic questions that are correctly predicted by our model, made small alterations to the passage or (question, answer) pair, and then checked whether our model can still make correct choices. Table 3.13 shows one arithmetic problem and our model can

Dialogue:
man: Good morning. May I help yon?
woman: I'd like to rent a car, please.
man: Okay. Full-size, mid-size, or compact, madam?
woman: Compact is OK. What's the rate?
man: 78 dollars a day.
woman: And I'd like to have insurance just in case.
man: If you want full coverage insurance, it will be ***7*** dollars per day.
woman: All right, I'll take that, too.
man: OK. Please fill in this form.

Question: How much will the woman pay in total?
A. ***71*** dollars.
B. 78 dollars.
C. ***85*** dollars. √

Model Prediction: A. 71 dollars.

Table 3.14: Adversarial example that forces our best model to make wrong predictions and is crafted by slightly revising the example in Table 3.13. The revisions are highlighted in bold and italic font. √ marks the correct answer.

get it right. The correct answer “86 dollars” should be the addition of the car rent “78 dollars” and the car insurance “8 dollars”, and it seems that our model can perform this simple calculation. However, if we simply changed the car insurance price from 8 dollars to 7 dollars in the passage, the model would obtain the wrong prediction, “71 dollars”, as shown in Table 3.14. We also curated another type of adversarial example by revising the passage so that the woman in the dialogue does not want the car insurance, in which the correct answer should be only the car rental price “78 dollars”. As shown in Table 3.15, our model again makes the wrong choice,“70 dollars”. These two adversarial examples strongly disprove that our model really has the ability to solve mathematical questions.

3.1.5.8 Lessons Learned

There are several lessons we have learned during experiments and we will discuss them here for references.

- We have tried adding an unsupervised learning objective of masked language

Dialogue:

man: Good morning. May I help yon?
woman: I'd like to rent a car, please.
man: Okay. Full-size, mid-size, or compact, madam?
woman: Compact is OK. What's the rate?
man: 78 dollars a day.
woman: And I'd like to have insurance just in case.
man: If you want full coverage insurance, it will be 7 dollars per day.
woman: ***Oh, that's too expansive for me. Then I would rather not have the insurance.***
man: ***All right, I will cancel that for you.*** Please fill in this form.

Question: How much will the woman pay in total?
A. 70 dollars.
B. 78 dollars. √
C. 86 dollars.

Model Prediction: A. 70 dollars.

Table 3.15: Adversarial example that forces our best model to make wrong predictions and is crafted by slightly revising the example in Table 3.13. The revisions are highlighted in bold and italic font. √ marks the correct answer.

modeling to the multi-task learning framework while fine-tuning BERT on the DREAM dataset, following the work from Rei [2017b]. However, this strategy cannot bring in any improvement. Furthermore, we have also tried adding more corpora that contain dialogues (e.g., datasets from Eric et al. [2019] and Dinan et al. [2019]) for language modeling training in this multi-task learning framework since the context in DREAM is also based on dialogues, but this trail has also shown no significant improvement.

- If choosing FCNN instead of MAN as the classifier, there is no need to add any hidden layers (i.e., only one output layer without activation is enough), especially for the RoBERTa model.

- The training performance of BERT/RoBERTa models can be influenced by the effective batch size, which is the product of batch size per GPU, number of GPUs used, and number of gradient accumulation steps. In practice, the effective batch size should be larger than 12. In the meantime, the number of gradient accumulation steps should not be too larger and it is preferable to be less than 6.

3.1.6 Related Work

3.1.6.1 Reading Comprehension for Question Answering

There is increasing interest in machine reading comprehension (MRC) for question answering (QA). The extractive QA tasks primarily focus on locating text spans from the given document/corpus to answer questions [Rajpurkar et al., 2018]. Answers in abstractive datasets such as MS MARCO [Nguyen et al., 2016], SearchQA [Dunn et al., 2017], and NarrativeQA [Kočiskỳ et al., 2018] are human-generated and based on source documents or summaries in free text format. However, since annotators tend to copy spans as answers [Reddy et al., 2019], the majority of answers are still extractive in these datasets. The multi-choice QA datasets are collected either via crowd sourcing, or collected from examinations designed by educational experts [Lai et al., 2017]. In this type of QA datasets, besides token matching, a significant portion of questions require multi-sentence reasoning and external knowledge [Ostermann et al., 2018].

Progress of research for MRC first relies on the breakthrough of the sentence encoder, from the basic LSTM to the pre-trained transformer based model [Devlin et al., 2019], which has elevated the performance of all MRC models by a large margin. Besides, the attention mechanisms between the context and the query can empower the neural models with higher performance [Seo et al., 2016]. In addition, some techniques such as answer verification [Hu et al., 2019b], multi-hop reasoning [Xiao et al., 2019], and synthetic data augmentation can be also helpful.

3.1.6.2 Transfer Learning

Transfer learning has been widely proved to be effective across many domains in NLP. In the QA domain, the most well-known example of transfer learning would be fine-tuning the pre-trained language model such as BERT to the downstream QA datasets such as SQuAD [Devlin et al., 2019]. Besides, multi-task learning can also be deemed as a type of transfer learning, since during the training of multiple datasets from different domains for different tasks, knowledge will be shared and transferred from each task to others, which has been used to build a generalized QA model [Talmor and Berant, 2019]. However, no previous works have demonstrated that the knowledge from the NLI datasets can also be transferred to improve the MCQA task.

3.1.6.3 Multi-task Learning

Multi-task learning has long been demonstrated to be effective in various NLP tasks such as QA [Talmor and Berant, 2019], sequential labeling [Sanh et al., 2019], language inference [Liu et al., 2019a], and sentence classification [Liu et al., 2019b]. It is also related to transfer learning since during the training of multiple different datasets from different domains for different tasks, knowledge will be shared and transferred from each task to others [Talmor and Berant, 2019]. More importantly, multi-tasks learning is now widely used to build a model that can learn general-purpose representations for natural language understanding so that one model can be applied to many diverse tasks [Liu et al., 2019b; McCann et al., 2018; Phang et al., 2018; Radford et al., 2018; Wang et al., 2018a].

3.1.7 Summary

We propose MMM, a multi-stage multi-task transfer learning method on the multiple-choice question answering tasks. Our two-stage training strategy and the multi-step attention network achieved significant improvements for MCQA. We also did detailed analyses to explore the importance of both our training strategies as well as different kinds of in-domain and out-of-domain datasets. It is noteworthy that our proposed transfer learning strategy can actually be generalized to various NLP tasks, where for any given target dataset, we can find its corresponding out-of-domain and in-domain source datasets, and then we train the model on the out-of-domain source datasets first, and subsequently fine-tune the model on the combination of the in-domain datasets and the target datasets via multi-task training. This strategy should always be effective at improving the target dataset.

3.2 Bridging the Gap From Machine Reading Comprehension to Dialogue State Tracking

3.2.1 Introduction

Building a task-oriented dialogue system that can comprehend users' requests and complete tasks on their behalf is a challenging but fascinating problem. Dialogue state tracking (DST) is at the heart of task-oriented dialogue systems. It tracks the *state* of a dialogue during the conversation between a user and a system. The *state* is typically defined as the *(slot_name, slot_value)* pair that represents, given a slot, the value that the user provides or system-provided value that the user accepts. More details about DST can refer to Section 2.1.7.

Despite the importance of DST in task-oriented dialogue systems, few large datasets are available. To address this issue, several methods have been proposed for data collection and bootstrapping the DST system. These approaches either utilize a Wizard-of-Oz setup via crowd sourcing [Budzianowski et al., 2018; Wen et al., 2017] or a Machines Talking To Machines (M2M) framework [Shah et al., 2018]. Currently the most comprehensive dataset with state annotations is MultiWOZ [Budzianowski et al., 2018], which contains seven domains with around 10,000 dialogues. However, compared to other NLP datasets, MultiWOZ is still relatively small, especially for training data-intensive neural models. In addition, it is also non-trivial to get a large amount of clean labeled data given the nature of task-oriented dialogues [Eric et al., 2019].

Another thread of approaches have tried to utilize data in a more efficient manner. These approaches [Wu et al., 2019; Zhou and Small, 2019] usually train the models on several domains and perform zero-shot or few-shot learning on unseen domains. However, these methods require slot definitions to be similar between the training data and the unseen test data. If such systems are given a completely new slot type, the performance would degrade significantly. Therefore, these approaches still rely on considerable amount of DST data to cover a broad range of slot categories.

Machine reading comprehension research aims to develop techniques to understand human written language. Although this is in general a very difficult task, recent works have shown impressive advances [Chen, 2018; Rajpurkar et al., 2016]. Considering that both the MRC and DST tasks focus on the *comprehension* of given context (e.g., a paragraph, or a conversation), we are inspired to formulate the DST problem as an instance of MRC, which should allow us to take advantage of those advances and of the abundant MRC data that could help to overcome the scarcity of labeled DST data.

Building upon this motivation, we formulate the DST task as a MRC one by specially designing a question for each slot in the dialogue state, similar to Gao et al. [2019]. For instance, we formulate a question like “What type of food does the user want to eat?” for the slot of “Food” in the domain of “Restaurant”. In order to solve these artificially crafted questions, we divide the slots into two types: *categorical* and *extractive*, based on the number of slot values in the ontology. For instance, in MultiWOZ, slots such as *parking* take values of *{Yes, No, Don't Care}* and can thus be treated as *categorical*. In contrast, slots such as *hotel-name* may accept an unlimited number of possible values and these are treated as *extractive*. Accordingly, we propose two machine reading comprehension models for dialogue state tracking. For categorical slots, we use multiple-choice reading comprehension models where an answer has to be chosen from a limited number of options. And for the extractive dialogue state tracking, span-based reading comprehension is applied, where the answer can be found in the form of a span in the conversation.

To summarize our approach and contributions:

- We divide the dialogue state slots into categorical and extractive types and use MRC techniques for state tracking. Our approach can leverage the recent advances in the field of machine reading comprehension, including both multiple-choice and span-based reading comprehension models.

- We propose a two-stage training strategy. We first coarse-train the state tracking models on reading comprehension datasets, then fine-tune them on the target state tracking dataset.

- We show the effectiveness of our method under three scenarios: First, when 100% of training data is all used for training, we show our method achieves close to the current state-of-the-art on MultiWoz 2.1 in terms of joint goal accuracy.[7] Second, in a few-shot setting, when only 1–10% of the training data is available, we show our methods significantly outperform the previous methods for five test domains in MultiWoz 2.0. In particular, we achieve 45.91% joint goal accuracy with just 1% (around 20–30 dialogues) of hotel domain data compared to the previous best result of 19.73% [Wu et al., 2019]. Third, in a zero-shot setting where no state tracking data is used for training, our models still achieve considerable average slot accuracy. More concretely, we show that 13 out of 30 slots in MultiWOZ 2.1 can achieve an average slot accuracy of greater than 90% without any training.

3.2.2 Related Works

Traditionally, dialogue state tracking methods [Lee et al., 2019; Liu and Lane, 2017; Mrkšić et al., 2016b; Nouri and Hosseini-Asl, 2018; Zhong et al., 2018] assume a fully-known fixed ontology for all slots where the output space of a slot is constrained by the values in the ontology. However, such approaches cannot handle previously unseen values and do not scale well for slots such as *restaurant-name* that can take potentially unbounded sets of values. To alleviate these issues, Rastogi et al. [2017] and Goel et al. [2018] generate and score slot-value candidates from the ontology, dialogue context n-grams, slot tagger outputs, or a combination of them. However, these approaches suffer if a reliable slot tagger is not available or if the slot value is longer than the candidate n-grams. Xu and Hu [2018] proposed an attention-based pointing mechanism to find the start and end of the slot value to better tackle the

[7] Joint goal accuracy checks whether all predicted states exactly matches the ground truth state for all slots.

issue of unseen slot values. Gao et al. [2019] proposed using a MRC framework for state tracking. They track slot values by answering the question "what is the value of the slot?" through attention-based pointing to the dialogue context. Although these approaches are more practical and scalable, they suffer when the exact slot value does not appear in the context as expected or if the value is not *pointable*. More recently, hybrid approaches have attempted to combine the benefits of both using a predefined ontology (closed vocabulary) and dynamically generating candidate set or pointing (open vocabulary) approaches. Goel et al. [2019] select between the two approaches per slot based on the development set. Wu et al. [2019] utilize a pointer generator network to either copy from the context or generate from the vocabulary.

Perhaps the most similar to our work is by Zhang et al. [2019b] and Zhou and Small [2019] where they divide slot types into span-based (extractive) slots and pick-list (categorical) slots and use a QA framework to point or pick values for these slots. A major limitation of these works is that they utilize heuristics to determine which slots should be categorical and which non-categorical. Moreover, in these settings most of the slots are treated as categorical (21/30 and 25/30), even though some of them have a very large number of possible values, e.g., *restaurant-name*. This is not scalable, especially when the ontology is large, not comprehensive, or when new domains/slots can occur at test time as in the DSTC8 dataset [Rastogi et al., 2019].

There are recent efforts toward building or adapting dialog state tracking systems in low data scenarios [Wu et al., 2019; Zhou and Small, 2019]. The general idea in these approaches is to treat all but one domain as in-domain data while the other domains as out-of-domain data; a model is first trained on the in-domain data and then tested on the out-of-domain data either directly (zero shot) or after being fine-tuned on a small percentage (1%-10%) of the out-of-domain data (few shot). A major drawback of these approaches is that they require several labeled in-domain examples in order perform well on the unseen domain. This limits these approaches to in-domain slots and slot definitions and they do not generalize very well to new slots or to a completely unseen target domain. This also requires large amounts of labeled data in the source domain, which may not be available in a real-world scenario. Our proposed approach, on the other hand, utilizes domain-agnostic QA datasets with zero or a small percentage of DST data and significantly outperforms these approaches in low-resource settings.

3.2.3 Methods

Slot Name	# Possible Values	Exact Match Rate	Extractive	Categorical
hotel.semi.type	3	61.1%	×	✓
hotel.semi.internet	3	62.1%	×	✓
hotel.semi.parking	4	63.1%	×	✓
restaurant.semi.pricerange	4	97.8%	✓	✓
hotel.semi.pricerange	6	97.7%	✓	✓
hotel.semi.area	6	98.8%	✓	✓
attraction.semi.area	6	99.0%	✓	✓
restaurant.semi.area	6	99.2%	✓	✓
hotel.semi.stars	7	99.2%	✓	✓
hotel.book.people	8	98.2%	✓	✓
hotel.book.stay	8	98.9%	✓	✓
train.semi.day	8	99.3%	✓	✓
restaurant.book.day	8	98.7%	✓	✓
restaurant.book.people	8	99.1%	✓	✓
hotel.book.day	11	98.1%	✓	✓
train.book.people	12	94.7%	✓	×
train.semi.destination	27	98.2%	✓	×
attraction.semi.type	27	86.6%	✓	×
train.semi.departure	31	97.6%	✓	×
restaurant.book.time	67	97.2%	✓	×
hotel.semi.name	78	88.7%	✓	×
taxi.semi.arriveby	97	91.9%	✓	×
restaurant.semi.food	103	96.4%	✓	×
taxi.semi.leaveat	108	81.1%	✓	×
train.semi.arriveby	156	91.5%	✓	×
attraction.semi.name	158	84.3%	✓	×
restaurant.semi.name	182	93.9%	✓	×
train.semi.leaveat	201	87.4%	✓	×
taxi.semi.destination	251	87.9%	✓	×
taxi.semi.departure	253	84.6%	✓	×

Table 3.16: Slot statistics for MultiWOZ 2.1. We classify the slots into extractive or categorical based on their exact match rate in conversation as well as number of possible values. 3 slots are categorical only, 12 slots are both extractive and categorical, the remaining 15 slots are extractive only.

3.2.3.1 Dialogue State Tracking as Reading Comprehension

Dialogue as Paragraph For a given dialogue at turn t, let us denote the user utterance tokens and the agent utterance tokens as $\mathbf{u}_t$ and $\mathbf{a}_t$ respectively. We concatenate the user utterance tokens and the agent utterance tokens at each turn to construct a sequence of tokens as $\mathbf{D}_t = \{\mathbf{u}_1, \mathbf{a}_1, ..., \mathbf{u}_t\}$. $\mathbf{D}_t$ can be viewed as the paragraph that we are going to ask questions on at turn t.

Slot as Question We can formulate a natural language question $\mathbf{q}_i$, for each slot s_i in the dialogue state. Such a question describes the meaning of that slot in the dialogue state. Examples of (slot, question) pairs can be seen in Table 3.17 and 3.18. We formulate questions by considering characteristics of domain and slot. In this way, DST becomes finding an answer to the question $\mathbf{q}_i$ given the paragraph $\mathbf{D}_t$. Note that Gao et al. [2019] formulate the dialogue state tracking problem in a similar way but their question formulation *"what is the value of a slot ?"* is more abstract, whereas our questions are more concrete and meaningful to the dialogue.

3.2.3.2 Span-based MRC to Extractive DST

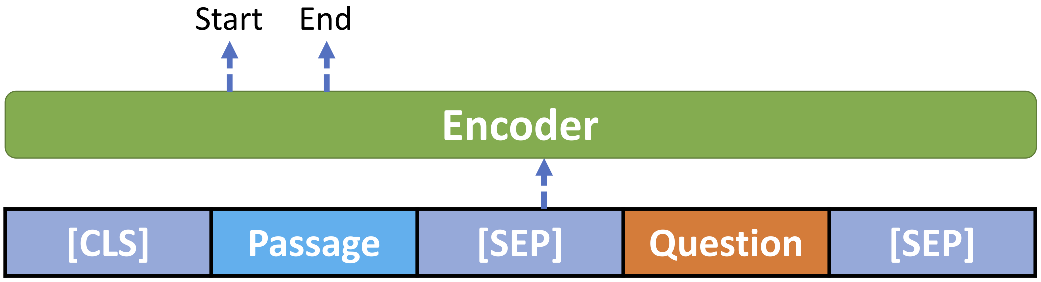


Figure 3-5: Model architecture for extractive state tracking. "Encoder"is a pre-trained sentence encoder such as BERT.

For many slots in the dialogue state such as names of attractions, restaurants, and departure times, one can often find their values in the dialogue context with exact matches. Slots with a wide range of values fit this description. Table 3.16 shows the exact match rate for each slot in MultiWOZ 2.1 dataset [Budzianowski et al., 2018; Eric et al., 2019] where slots with a large number of possible values tend to have higher exact match rate ($\geq 80\%$). We call tracking such slots *extractive dialogue stack tracking (EDST)*.

This problem is similar to span-based MRC where the goal is to find a span in the passage that best answers the question. Therefore, for EDST, we adopt the simple

Dialogue
U: I'm so hungry. Can you find me a place to eat in the city centre?
A: I'm happy to help! There are a great deal of restaurants there. What type of food did you have in mind?
U: I do not care, it just needs to be expensive.
A: Fitzbillies restaurant serves British food would that be okay?
U: Yes, may I have the address?
restaurant.semi.food: What type of food does the user want to eat?
Answer: [52-53] *(I do not care, it just needs to be expensive)*
restaurant.semi.name: What is the name of the restaurant where the user wants to eat?
Answer: [53-55] *(Fitzbillies restaurant)*

Table 3.17: Sample dialogue from MultiWOZ dataset showing framing of extractive DST to span-based MRC. The span text (or *don't care* user utterance) is also shown in italics.

BERT-based question answering model used by Devlin et al. [2019], which has shown strong performance on multiple datasets [Rajpurkar et al., 2016, 2018; Reddy et al., 2019]. In this model as shown in Figure 3-5, the slot question and the dialogue are represented as a single sequence. The probability of a dialogue token t_i being the start of the slot value span is computed as $p_i = \frac{e^{\mathbf{s}\cdot\mathbf{T}_i}}{\sum_j e^{\mathbf{s}\cdot\mathbf{T}_j}}$, where $\mathbf{T}_j$ is the embedding of each token t_j and $\mathbf{s}$ is a learnable vector. A similar formula is applied for finding the end of the span.

Handling *None* Values At any given turn in the conversation, there are typically, many slots that have not been mentioned or accepted yet by the user. All these slots must be assigned a *None* value in the dialogue state. We can view such cases as *no answer exists* in the reading comprehension formulation. Similar to dev for the SQuAD 2.0 task, we assign the answer span with start and end at the beginning token [CLS] for these slots.

Handling *Don't Care* Values To handle *don't care* value in EDST, a span is also assigned to *don't care* in the dialogue. We find the dialogue turn when the slot value first becomes *don't care* and set the start and end of the *don't care* span to be the start and end of the user utterance of this turn. See Table 3.17 for an example.

3.2.3.3 Multiple-Choice Reading Comprehension to Categorical Dialogue State Tracking

Dialogue
U: I am looking for a place to to stay that has cheap price range it should be in a type of hotel A: Okay , Do you have a specific area you want to stay in? U: No, I just need to make sure it's cheap. Oh, and I need parking.
hotel.semi.area: What is the area that the user wants to book a hotel in? **A.** East **B.** West **C.** North **D.** South **E.** Centre **F.** Don't Care ✓ **G.** Not Mentioned
hotel.semi.parking: Does the user want parking at the hotel? **A.** Yes ✓ **B.** No **C.** Don't Care **D.** Not Mentioned

Table 3.18: Sample dialogue from MultiWOZ dataset showing framing of categorical DST to multiple-choice MRC.

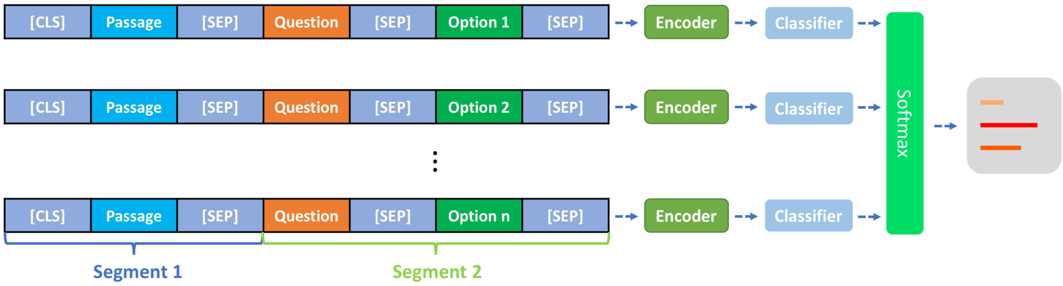


Figure 3-6: Model architecture for categorical dialog state tracking. "Encoder" is a pre-trained sentence encoder such as BERT. "Classifier" is a top-level fully connected layer.

The other type of slots in the dialogue state cannot be filled through exact match in the dialogue context in a large number of cases. For example, a user might express intent for hotel parking as *"oh! and make sure it has parking"* but the slot *hotel-parking* only accepts values from *{Yes, No, Don't Care}*. In this case, the state tracker needs to infer whether or not the user wants parking based on the user utterance and to select the correct value from the list. These kinds of slots may not have exact-match spans in the dialogue context but usually require a limited number of values to choose from.

Tracking these type of slots is surprisingly similar to multiple-choice question answering tasks. In comparison to span-based MRC tasks, the answers of the MCQA datasets [Lai et al., 2017; Sun et al., 2019b] are often in the form of open, natural language sentences and are not restricted to spans in text. Following the traditional models of MCQA [Devlin et al., 2019; Jin et al., 2019a], we concatenate the slot question, the dialogue context and one of the answer choices into a long sequence. We then feed this sequence into a sentence encoder to obtain a logit vector. Given a question, we can get m logit vectors assuming there are m answer choices. We then transform these m logit vectors into a probability vector through a fully connected layer and a softmax layer; see Figure 3-6 for details.

Handling *None* and *Don't Care* Values For each question, we simply add two additional choices "not mentioned" and "do not care" in the answer options, representing *None* and *don't care*, as shown in Table 3.18. It is worth noting that certain slots not only accept a limited number of values but also their values can be found as an exact-match span in the dialogue context. For these slots, both extractive and categorical DST models can be applied, as shown in Table 3.16.

3.2.4 Experiments

3.2.4.1 Datasets

	# of passages	**# of examples**
MRQA (span-based)	386,384	516,819
DREAM (multi-choice)	6,444	10,197
RACE (multi-choice)	27,933	97,687
MultiWOZ	8,420	298,978*

Table 3.19: Statistics of datasets used. (*: we only report the number of positive examples (a non-empty value) in MultiWOZ for fair comparison.)

MultiWOZ We use the largest available multi-domain dialogue dataset with state annotation: MultiWOZ 2.0 [Budzianowski et al., 2018] and MultiWOZ 2.1 [Eric et al., 2019], an enhanced, less noisy version of the MultiWOZ 2.0 dataset, which contains 7 distinct domains across 10K dialogues. We exclude the *hospital* and *police* domains, which have very few dialogues. This results in 5 remaining domains *attraction*, *restaurant*, *taxi*, *train*, *hotel* with a total of 30 (domain, slot) pairs in the dialog state following Wu et al. [2019] and Zhang et al. [2019b].

Reading Comprehension Datasets For a span-based MRC dataset, we use the dataset from the Machine Reading for Question Answering (MRQA) 2019 shared task [Fisch et al., 2019] that was focused on extractive question answering. MRQA contains six distinct datasets across different domains: SQuAD, NewsQA, TriviaQA, SearchQA, HotpotQA, and NaturalQuestions. In this dataset, any answer to a question is a segment of text or span in a given document. For a multiple-choice MRC dataset, we leverage the current largest multiple-choice QA dataset, RACE [Lai et al., 2017] as well as a dialogue-based multiple-choice QA dataset, DREAM [Sun et al., 2019b]. Both of these datasets are collected from English language exams that are carefully designed by educational experts to assess the comprehension level of English learners. Table 3.19 summarizes the statistics of these datasets. It is worth noting that for MultiWOZ, although the number of examples is significantly more than for multiple-choice QA datasets, the number of distinct questions is only 30 due to the limited number of slot types.

3.2.4.2 Canonicalization for Extractive Dialogue State Tracking

For extractive dialogue state tracking, it is common that the model will choose a span that is either a super-set of the correct reference or has a similar meaning as the correct value but with a different wording. Following this observation, we adopt a simple canonicalization procedure after our span-based model prediction. If the predicted value does not exist in the ontology of the slot, then we match the prediction with the value in the ontology that is closest to the predicted value in terms of edit distance.[8] Note that this procedure is only applied at model *inference* time. At

[8] we use the function *get_ closest_ matches* of *difflib* in Python for this implementation.

training time for extractive dialogue state tracking, the ontology is not required.

3.2.4.3 Two-stage Training

A two-stage training procedure is used to train the extractive and categorical dialogue state tracking models with both types of reading comprehension datasets (DREAM, RACE, and MRQA) and the dialogue state tracking dataset (MultiWOZ).

Reading Comprehension Training Stage For the categorical dialogue state tracking model, we coarse-tune the model on DREAM and RACE. For extractive dialogue state tracking model, we coarse-tune the model on the MRQA dataset as a first step.

Dialog State Tracking Training Stage After being trained on the reading comprehension datasets, we expect our models to be capable of answering (passage, question) pairs. In this phase, we further fine-tune these models on the MultiWOZ dataset.

3.2.5 Results and Analyses

3.2.5.1 DST with Full Training Data

Joint Goal Accuracy	
SpanPtr [Xu and Hu, 2018]	29.09%
FJST [Eric et al., 2019]	38.00%
HyST [Goel et al., 2019]	39.10%
DSTreader [Gao et al., 2019]	36.40%
TRADE [Wu et al., 2019]	45.96%
DS-DST [Zhang et al., 2019b]	**51.21**%
DSTQA w/span [Zhou and Small, 2019]	49.67%
DSTQA w/o span [Zhou and Small, 2019]	51.17%
STARC (this work)	49.48%

Table 3.20: Joint Goal Accuracy on MultiWOZ 2.1 test set.

We use the full data in MultiWOZ 2.1 to test our models. For the first 15 slots with the lowest number of possible values (from *hotel.semi.type* to *hotel.book.day* in Table 3.16, we use our proposed categorical dialogue state tracking model, whereas for the remaining 15 slots, we use the extractive dialogue state tracking model. We use the pre-trained word embedding RoBERTa-Large [Liu et al., 2019c] in our experiment.

Table 3.20 summarizes the results. We can see that our model, STARC (**S**tate **T**racking **A**s **R**eading **C**omprehension), achieves close to the state-of-the-art accuracy on MultiWOZ 2.1 in the full data setting. It is worth noting that the best performing approach, DS-DST [Zhang et al., 2019b], cherry-picks 9 slots as span-based slots whereas the remaining 21 slots are treated as categorical. Further, the second best result DSTQA w/o span [Zhou and Small, 2019] does not use a span-based model for any slot. Unlike these state-of-the-art methods, our method simply categorizes the slots based on the number of values in the ontology. As a result, our approach uses fewer (15 compared to 21 in DS-DST) and more reasonable (only those with few values in the ontology) categorical slots. Thus, our approach is more practical to be applied in a real-world scenario.

Ablation	Dev Accuracy
STARC (this work)	**53.95**%
– MRC Coarse Tuning	52.35%
– Canonicalization	51.07%
– MRC Coarse Tuning – Canonicalization	50.84%
– Categorical Model	47.86%
– Categorical Model – Canonicalization	41.86%
DS-DST Threshold-10	49.08%
DS-DST Span Only	40.39%

Table 3.21: Ablation study with different aspects of our model and other comparable approaches. The numbers reported are joint goal accuracy on the MultiWOZ 2.1 development set.

Ablation Study We also run an ablation study to understand which component of our model helps with accuracy. Table 3.21 summarizes the results. For fair comparison, we also report the numbers for DS-DST Threshold-10 [Zhang et al., 2019b], where they also use the first 15 slots with a categorical model and the remaining with an extractive model. We observe that both two-stage the training strategy using reading comprehension data and canonicalization play important roles in achieving higher accuracy. Without the categorical model (using an extractive model for all slots), STARC is still able to achieve joint goal accuracy of 47.86%. More interestingly, if we remove the categorical model as well as the canonicalization, the performance drops drastically, but is still slightly better than using a purely extractive model in DS-DST.

Error Type	Extractive	Categorical
ref not none, predicted none	43.7%	31.4%
ref none, predicted not none	25.6%	58.4%
ref not none, predicted not none	30.6%	10.0%

Table 3.22: Type of errors made by each model.

Handling *None* Value Through error analysis of our models, we have learned that the models' performance on the *None* value has a significant impact on the overall accuracy. Table 3.22 summarizes our findings. We found that the plurality of errors for the extractive model comes from cases where ground-truth is not *None* but the model predicted *None*. For the categorical model, the opposite was true. The majority of errors were from the model predicting not the *None* value but the ground-truth is actually *None*. We leave further investigation of this issue for future work.

3.2.5.2 Few shot from MRC to DST

	Hotel			**Restaurant**			**Attraction**			**Train**			**Taxi**		
	1%	*5%*	*10%*	*1%*	*5%*	*10%*	*1%*	*5%*	*10%*	*1%*	*5%*	*10%*	*1%*	*5%*	*10%*
TRADE	19.73	37.45	41.42	42.42	55.70	60.94	35.88	57.55	63.12	59.83	69.27	71.11	63.81	66.58	70.19
DSTQA	N/A	50.18	53.68	N/A	58.95	64.51	N/A	**70.47**	**71.60**	N/A	70.35	74.50	N/A	70.90	74.19
STARC	**45.91**	**52.59**	**57.37**	**51.65**	**60.49**	**64.66**	**40.39**	65.34	66.27	**65.67**	**74.11**	**75.08**	**72.58**	**75.35**	**79.61**

Table 3.23: Joint goal accuracy for few-shot experiments. Best numbers reported by TRADE and DSTQA are also shown.

In the few-shot setting, our models (both extractive and categorical) are pre-trained on reading comprehension datasets and we randomly select a limited amount of *target* domain data for fine-tuning. We evaluate our model with 1%, 5% and 10%

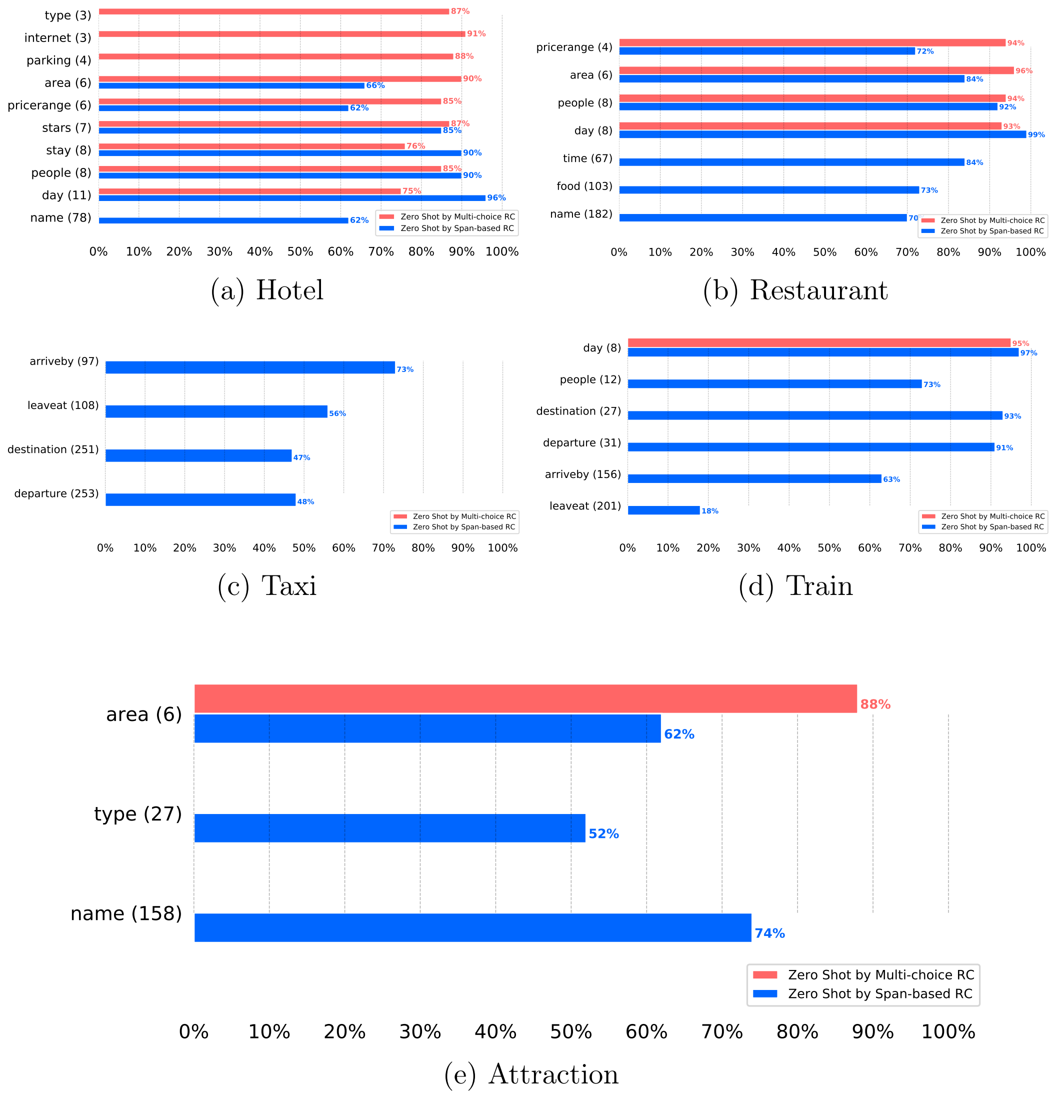
of training data in the target domain. Table 3.23 shows the results of our model under this setting for five domains in MultiWOZ 2.0[9]. We also report the few-shot results for two other models: TRADE [Wu et al., 2019] and DSTQA [Zhou and Small, 2019], where they perform the same few-shot experiments but pre-trained with a hold-out strategy, i.e., training on the other four domains in MultiWOZ and fine-tuning on the held-out domain. We can see that under all three different data settings, our model outperforms the TRADE and DSTQA models (except the attraction domain for DSTQA) by a large margin. Especially in the 1% data setting for the hotel domain, which contains the largest number of slots (10) among all the five domains, the joint goal accuracy dropped to 19.73% for TRADE while our model can still achieve relatively high joint goal accuracy of 45.91%. This significant performance difference can be attributed to pre-training our models on reading comprehension datasets, which gives our model the ability to comprehend passages or dialogues (which we have empirically verified in the next section). The formulation of dialogue state tracking as a reading comprehension task helps the model to transfer comprehension capability.

3.2.5.3 Zero shot from MRC to DST

In zero-shot experiments, we want to investigate how would the reading comprehension models behave on the MultiWOZ dataset *without* any training on state tracking data. To do so, we train our models on reading comprehension datasets and test on MultiWOZ 2.1. Note that, in this setting, we only take labels in MultiWOZ 2.1 that are not missing, ignoring the data that is “None" in the dialogue state. For zero-shot experiments from multiple-choice MRC to DST, we take the first fifteen slots in Table 3.16 that are classified as categorical. For zero shot from span-based MRC to DST, we take twenty-seven slots which are extractive except the first three slots in Table 3.16.

Figure 3-7 summarizes the results for the hotel, restaurant, taxi, train, and attraction domains in MultiWOZ 2.1. We can see that most of the slots have an average

[9] We show results on MultiWOZ 2.0 rather than 2.1 for the purpose of comparison to previous works.



(a) Hotel

(b) Restaurant

(c) Taxi

(d) Train

(e) Attraction

Figure 3-7: Zero-shot average slot accuracy using multi-choice and span-based MRC to DST in hotel, restaurant, taxi, and train domain of MultiWOZ 2.1. The number in parentheses indicates the number of possible values that a slot can take.

accuracy of at least 50% or above in both multiple-choice MRC and span-based MRC approaches, indicating the effectiveness of MRC data. For some slots such as *hotel.stay*, *hotel.people*, *hotel.day*, *restaurant.people*, *restaurant.day*, and *train.day*, we are able to achieve very high zero-shot accuracy (greater than 90%). The zero-shot setting in TRADE [Wu et al., 2019], where the transfer is from the four source domains to the held-out target domain, fails completely on certain slot types like *hotel.name*. In contrast, our zero-shot experiments from MRC to DST are able to transfer almost

Example (Span-based MRC model prediction is bolded)	**Ground Truth**
Dialogue: "*. . . .A: sure , what area are you thinking of staying, U:* ***i do not have an area preference*** *but it needs to have free wifi and parking at a moderate price. . . .*" Question: "*which area is the hotel at?*" (hotel.semi.area)	don't care
Dialogue: "*U: i am looking for something fun to do on the* ***east side of town*** *. funky fun house is my favorite place on the east side...* Question: "*which area is the restaurant at?*" (restaurant.semi.area)	east
Dialogue: "*U: I need 1 that leaves after 13:30 for bishops stortford how about the tr8017 ? A:* ***it leaves at 15:29*** *and arrives at 16:07 in bishops stortford ....*" Question: "*what time will the train leave from the departure location?*" (train.semi.leaveat)	15:29
Dialogue: "*U: hello i want to see some* ***authentic architectures*** *in cambridge!...*" Question: "*what is the type of the attraction?*" (attraction.semi.type)	architecture
Dialogue: "*...A: can i help you with anything else ? U: i would like to book a taxi* ***from the hong house*** *to the hotel leaving by 10:15...*" Question: "*where does the taxi leave from?*" (taxi.semi.departure)	lan hong house

Table 3.24: Zero-shot examples to MultiWOZ 2.1 by span-based reading comprehension model trained on MRQA dataset. The predicted span by the span-based MRC model are bolded.

all the slots.

Table 3.24 illustrates the zero shot examples for the span-based MRC model. We can see that although the span-based MRC model does not directly point to the state value itself, it usually points to a span that *contains* the ground truth state value and the canonicalization procedure then turns the span into the actual slot value. Such predicted spans can be viewed as *evidence* for getting the ground-truth dialogue state, which makes dialogue state tracking more explainable.

3.2.6 Summary

Task-oriented dialogue systems aim to help users to achieve a variety of tasks. It is not unusual to have hundreds of different domains in modern task-oriented virtual assistants. How can we ensure the dialogue system is robust enough to scale to different tasks given a limited amount of data? Some approaches focus on domain expansion by training on several source domains and then adapting to the target domain. While such methods can be successful in certain cases, it is hard for them to generalize to other completely different out-of-domain tasks.

Machine reading comprehension provides us a clear and general basis for understanding the context given a wide variety of questions. By formulating the dialogue state tracking as reading comprehension, we can utilize the recent advances in reading comprehension models. More importantly, we can utilize reading comprehension datasets to mitigate some of the resource issues in task-oriented dialogue systems. As a result, we achieve much higher accuracy in dialogue state tracking across different domains given a limited amount of data, compared to the existing methods. As the variety of tasks and functionalities in a dialogue system continues to grow, general methods for tracking dialogue state across all tasks will become increasingly necessary. We hope that the developments suggested here will help to address this need.

3.3 Dual Adversarial Neural Transfer for Low Resource Named Entity Recognition

3.3.1 Introduction

Named entity recognition (NER) is an important step in most natural language processing (NLP) applications. It detects not only the type of named entity, but also the entity boundaries, which requires deep understanding of the contextual semantics to disambiguate the different entity types of same tokens. To tackle this challenging problem, most early studies were based on hand-crafted rules, which suffered from limited performance in practice. Current methods are devoted to developing learning based algorithms, especially neural network based methods, and have been advancing the state-of-the-art [Chiu and Nichols, 2016; Collobert et al., 2011; Huang et al., 2015; Lample et al., 2016; Ma and Hovy, 2016]. These end-to-end models generalize well on new entities based on features automatically learned from the data. However, when the annotated corpora are small, the performance of these methods degrades significantly since the hidden feature representations cannot be learned adequately [Zhang et al., 2016].

Recently, more and more approaches have been proposed to address low-resource NER. Early works [Chen et al., 2010; Li et al., 2012] primarily assumed the access to a large parallel corpus and focused on exploiting them to project information from this high-resource dataset to those target low-resource datasets. Unfortunately, such a large parallel corpus may not be available for many low-resource languages (i.e., those languages where large annotated data does not exist). More recently, cross-resource word embedding [Adams et al., 2017; Fang and Cohn, 2017; Yang et al., 2017b] was proposed to bridge the low and high resources and enable knowledge transfer. Although the aforementioned transfer-based methods show promising performance in low-resource NER, there are two issues deserving to be further investigated: 1) *Representation Difference*: they did not consider the representation difference across resources and forced the feature representation to be shared across languages/domains; 2) *Resource Data Imbalance*: the training size in high-resource languages is usually much larger than that in low-resource ones. The existing methods neglect such a difference in their models, resulting in poor generalization.

In this section, we present a general neural transfer framework termed **Dual Adversarial Transfer Network (DATNet)** to address the above issues in a unified framework for low-resource NER. Specifically, to handle the representation difference, we first investigate two architectures of hidden layers (we use a bi-directional long-short term memory (BiLSTM) model as the hidden layer) for transfer. The first one is that all the units in hidden layers are common units shared across languages/domains. The second one is composed of both private and common units, where the private part preserves the independent language/domain information. Extensive experiments are conducted to show that there is not always a winner and the two transfer strategies have their own advantages over each other in different situations, which is largely ignored by existing research. On top of common units, the adversarial discriminator (AD) loss is introduced to encourage a resource-agnostic representation so that the knowledge from the high resource setting can be more compatible with that from the low resource one.

To handle the resource data imbalance issue, we further propose a variant of the AD loss, termed *Generalized Resource-Adversarial Discriminator (GRAD)*, to impose the resource weight during training so that more attention can be paid to low-resource and hard samples. In addition, we create adversarial samples to conduct the *Adversarial Training (AT)*, further improving the generalization and alleviating the over-fitting problem. We unify two kinds of adversarial learning, i.e., GRAD and AT, into one transfer learning model, termed Dual Adversarial Transfer Network (DATNet), to achieve end-to-end training and obtain the state-of-the-art performance on a series of NER tasks: 88.16% F1 for CoNLL-2002 Spanish, 53.43% and 42.83% F1 for WNUT-2016 and 2017. Different from prior works, we do *not* use additional hand-crafted features and do *not* use cross-lingual word embeddings while addressing the cross-language tasks.

3.3.2 Related Work

Named Entity Recognition NER is typically framed as a sequence labeling task which aims at automatic detection of named entities (e.g., person, organization, location, etc.) from free text [Marrero et al., 2013]. The early works applied CRF, SVM, and perceptron models with hand-crafted features [Luo et al., 2015; Passos et al., 2014; Ratinov and Roth, 2009]. With the advent of deep learning, research focus has been shifting towards deep neural networks (DNN), which require little feature engineering and domain knowledge [Lample et al., 2016; Zukov Gregoric et al., 2018]. Collobert et al. [2011] propose a feed-forward neural network with a fixed sized window for each word, which failed to consider useful long-distance relations betwee words. To overcome this limitation, Chiu and Nichols [2016] present a bidirectional LSTM-CNNs architecture that automatically detects word- and character-level features. Ma and Hovy [2016] further extend it into a bidirectional LSTM-CNNs-CRF architecture, where the CRF module was added to optimize the output label sequence. Liu et al. [2018] propose a task-aware neural language model termed LM-LSTM-CRF, where character-aware neural language models were incorporated to extract character-level embeddings under a multi-task framework.

Transfer Learning for NER Transfer learning can be a powerful tool to low resource NER tasks. To bridge high and low resource, transfer learning methods for NER can be divided into two types: the parallel corpora based transfer and the shared representation based transfer. Early works mainly focused on exploiting parallel corpora to project information between the high- and low-resource language [Chen et al., 2010; Feng et al., 2018b; Li et al., 2012; Yarowsky et al., 2001]. For example, Chen et al. [2010] and Feng et al. [2018b] propose to jointly identify and align bilingual named entities. Ni and Florian [2016]; Ni et al. [2017] utilize the Wikipedia entity type mappings to improve low-resource NER. Mayhew et al. [2017] create a cross-language NER system, which works well for very minimal resources by translate annotated data of high-resource into low-resource. On the other hand, the shared representation methods do not require the parallel correspondence [Rei and Søgaard, 2018]. For instance, Fang and Cohn [2017] propose cross-lingual word embeddings to transfer knowledge across resources. Yang et al. [2017b] present a transfer learning approach based on a deep hierarchical recurrent neural network (RNN), where full/partial hidden features between source and target tasks are shared.

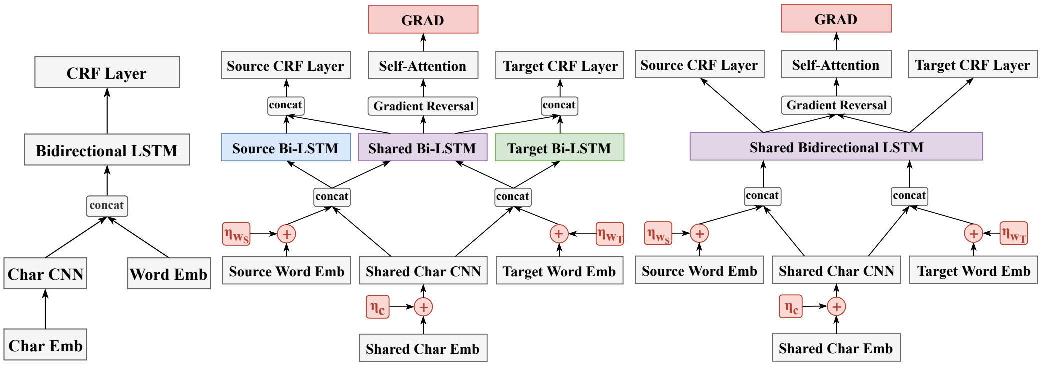
Al-Rfou' et al. [2015] build massive multilingual annotators with minimal human expertise by using language agnostic techniques. Cotterell and Duh [2017] propose character-level neural CRFs to jointly train and predict low- and high-resource languages. Pan et al. [2017] propose a large-scale cross-lingual named entity dataset which contains 282 languages for evaluation. In addition, multi-task learning [Aguilar et al., 2017; Hashimoto et al., 2017; Lin et al., 2018; Luong et al., 2016; Rei, 2017a; Yang et al., 2016] shows that jointly training on multiple tasks/languages helps improve performance. Different from transfer learning methods, multi-task learning aims at improving the performance of all the resources instead of low resource only.

Adversarial Learning Adversarial learning originates from Generative Adversarial Nets (GAN) [Goodfellow et al., 2014], which shows impressive results in computer vision. Recently, many papers have tried to apply adversarial learning to

NLP tasks. Liu et al. [2017a] present an adversarial multi-task learning framework for text classification. Gui et al. [2017] apply the adversarial discriminator to POS tagging for Twitter. Kim et al. [2017a] propose a language discriminator to enable language-adversarial training for cross-language POS tagging. Adversarial training is another concept originally introduced by Szegedy et al. [2013] and Goodfellow et al. [2015] to improve the robustness of image classification models by injecting malicious perturbations into input images. Recently, Miy propose a semi-supervised text classification method by applying adversarial training, where for the first time adversarial perturbations were added onto word embeddings. Yasunaga et al. [2018] apply adversarial training to POS tagging. Different from all these adversarial learning methods, our method is more general and integrates both the adversarial discriminator and adversarial training in a unified framework to enable end-to-end training.

3.3.3 Dual Adversarial Transfer Network (DATNet)

In this section, we introduce DATNet in more detail. We first describe a base model for NER, and then discuss two proposed transfer architectures for DATNet.



(a) Base Model (b) DATNet-P (c) DATNet-F

Figure 3-8: The general architecture of proposed models.

3.3.3.1 Basic Architecture

We follow state-of-the-art models for the NER task [Chiu and Nichols, 2016; Huang et al., 2015; Lample et al., 2016; Ma and Hovy, 2016], i.e., a LSTM-CNNs-CRF based structure, to build the base model. It consists of the following pieces: character-level embedding, word-level embedding, BiLSTM for feature representation, and CRF as the decoder. The character-level embedding takes a sequence of characters in the word as atomic inputs to derive the word representation that encodes the morphological information, such as root, prefix, and suffix. These character features are usually encoded by character-level CNN or BiLSTM, then concatenated with word-level embeddings to form the final word vectors. On top of them, the network further incorporates the contextual information using BiLSTM to output new feature representations, which is subsequently fed into the CRF layer to predict label sequence. Although both the word-level layer and the character-level layer can be implemented using CNNs or RNNs, we use CNNs for extracting character-level and RNNs for extracting word-level representations. Fig. 3-8a shows the the architecture of the base model.

3.3.3.2 Dual Adversarial Transfer Architecture

Character-level Encoder Previous works have shown that character features can boost sequence labeling performance by capturing morphological and semantic information [Dernoncourt et al., 2017; dos Santos and Guimarães, 2015]. For a low-resource dataset to obtain high-quality word features, character features learned from other languages or domains may provide crucial information for labeling, especially for rare and out-of-vocabulary words. The character-level encoder usually uses BiLSTM [Lample et al., 2016] or CNN [Chiu and Nichols, 2016; Ma and Hovy, 2016] approaches. In practice, Reimers and Gurevych [2017] observe that the difference between the two approaches is statistically insignificant in sequence labeling tasks, but character-level CNN is more efficient and has fewer parameters. Thus, we use character-level CNN and share character features between high- and low-resource tasks to enhance the representations of low-resource tasks.

Word-level Encoder To learn a better word-level representation, we concatenate character-level features of each word with a latent word embedding as $\mathbf{w}_i$ =

$[\mathbf{w}_i^{char}, \mathbf{w}_i^{emb}]$, where the latent word embedding $\mathbf{w}_i^{emb}$ is initialized with pre-trained embeddings and fixed during training. One unique characteristic of NER is that the left and right input for a given time step could be useful for label inference. To exploit such a characteristic, we use a bidirectional LSTM architecture [Hochreiter and Schmidhuber, 1997] to extract contextualized word-level features. In this way, we can gather the information from the past and future for a particular time frame t as follows, $\overrightarrow{\mathbf{h}}_t = \texttt{lstm}(\overrightarrow{\mathbf{h}}_{t-1}, \mathbf{w}_t)$, $\overleftarrow{\mathbf{h}}_t = \texttt{lstm}(\overleftarrow{\mathbf{h}}_{t+1}, \mathbf{w}_t)$. After the LSTM layer, the representation of a word is obtained by concatenating its left and right context representations as follows, $\mathbf{h}_t = [\overrightarrow{\mathbf{h}}_t, \overleftarrow{\mathbf{h}}_t]$.

To mitigate the representation difference between domains, we introduce two kinds of transferable word-level encoders in our model, namely DATNet-Full Share (DATNet-F) and DATNet-Part Share (DATNet-P). In DATNet-F, all the BiLSTM units are shared by both resources while word embeddings for different resources are disparate. The illustrative figure is depicted in the Figure 3-8c. Different from DATNet-F, the DATNet-P decomposes the BiLSTM units into the shared component and the resource-related one, which is shown in the Figure 3-8b. Differently from existing works [Cao et al., 2018; Cotterell and Duh, 2017; Fang and Cohn, 2017; Yang et al., 2017b], in this work, we investigate the performance of two different shared representation architectures on different tasks.

Generalized Resource-Adversarial Discriminator In order to make the feature representation extracted from the source domain more compatible with that from the target domain, we encourage the outputs of the shared BiLSTM part to be resource-agnostic by constructing a resource-adversarial discriminator, which is inspired by the Language-Adversarial Discriminator proposed by Kim et al. [2017a]. Unfortunately, previous works did not consider the imbalance of training size for two resources. Specifically, the target domain consists of very limited labeled training data, e.g., 10 sentences. In contrast, labeled training data in the source domain are much richer, e.g., 10k sentences. If such imbalance was not considered during training, the stochastic gradient descent (SGD) optimization would make the model more biased to the high resource data [Lin et al., 2017b]. To address this imbalance problem, we impose a weight α on two resources to balance their influences. However, in the experiment we also observe that the easily classified samples from the high resource data comprise the majority of the loss and dominate the gradient. To overcome this issue, we further propose the Generalized Resource-Adversarial Discriminator (GRAD) to enable adaptive weights for each sample (note that the sample here means each sentence of a resource), which focuses the model training on hard samples.

To compute the loss of GRAD, the output sequence of the shared BiLSTM is firstly encoded into a single vector via a self-attention module [Bahdanau et al., 2015], and then projected into a scalar r via a linear transformation. The loss function of the resource classifier is formulated as:

$$\begin{aligned} \ell_{GRAD} \;\; = \;\; & -\sum_i \{\mathbf{I}_{i\in\mathcal{D}_S}\alpha(1-r_i)^\gamma \log r_i \\ & +\mathbf{I}_{i\in\mathcal{D}_T}(1-\alpha)r_i^\gamma \log(1-r_i)\} \end{aligned} \tag{3.3}$$

where $\mathbf{I}_{i\in\mathcal{D}_S}, \mathbf{I}_{i\in\mathcal{D}_T}$ are the identity functions to denote whether a sentence is from the high resource source or the low resource target, respectively; α is a weighting factor to balance the loss contribution from high and low resource data; the parameter $(1-r_i)^\gamma$ (or r_i^γ) controls the loss contribution from individual samples by measuring the discrepancy between prediction and true label (easy samples have a smaller contribution); and γ scales the contrast of loss contribution from hard and easy samples. In practice, the value of γ does not need to be tuned much and is usually set as 2 in our experiment. Intuitively, the weighting factors α and $(1-r_i)^\gamma$ reduce the loss contribution from high resource and easy samples, respectively. Note that though the resource classifier is optimized to minimize the resource classification error, when the gradients originating from the resource classification loss are back-propagated to the model parts other than the resource classifier, they are negated for parameter updates so that these bottom layers are trained to be resource-agnostic.

Label Decoder The label decoder induces a probability distribution over sequences of labels, conditioned on the word-level encoder features. In this paper, we use a linear chain model based on the first-order Markov chain structure, termed the chain conditional random field (CRF) [Lafferty et al., 2001], as the decoder. In this decoder, there are two kinds of cliques: local cliques and transition cliques. Specifically, local cliques correspond to the individual elements in the sequence. Transition cliques, on the other hand, reflect the evolution of states between two neighboring elements at time $t-1$ and t and we define the transition distribution as θ. Formally, a linear-chain CRF can be written as $p(\mathbf{y}|\mathbf{h}_{1:T}) = \frac{1}{Z(\mathbf{h}_{1:T})} \exp\left\{\sum_{t=2}^{T} \theta_{y_{t-1},y_t} + \sum_{t=1}^{T} \mathbf{W}_{y_t}\mathbf{h}_t\right\}$, where $Z(\mathbf{h}_{1:T})$ is a normalization term and $\mathbf{y}$ is the sequence of predicted labels as follows: $\mathbf{y} = y_{1:T}$. Model parameters are optimized to maximize this conditional log likelihood, which acts as the objective function of the model. We define the loss function for source and target resources as follows, $\ell_S = -\sum_i \log p(\mathbf{y}|\mathbf{h}_{1:T})$, $\ell_T = -\sum_i \log p(\mathbf{y}|\mathbf{h}_{1:T})$.

Adversarial Training So far our model can be trained end-to-end with standard back-propagation by minimizing the following loss:

$$\ell = \ell_{GRAD} + \ell_S + \ell_T \tag{3.4}$$

Recent works have demonstrated that deep learning models are fragile to *adversarial examples* [Goodfellow et al., 2015]. In computer vision, those adversarial examples can be constructed by changing a very small number of pixels, which are virtually indistinguishable to human perception [Pin-Yu et al., 2018]. Recently, adversarial samples are widely incorporated into training to improve the generalization and robustness of the model, which is called adversarial training (AT) [Miy]. It emerges as a powerful regularization tool to stabilize training and prevent the model from being stuck in local minima. In this paper, we explore AT in the context of NER. To be specific, we prepare an adversarial sample by adding the original sample with a perturbation bounded by a small norm ϵ to maximize the loss function as follows:

$$\eta_{\mathbf{x}} = \arg \max_{\eta:\|\eta\|_2 \leq \epsilon} \ell(\Theta; \mathbf{x} + \eta) \tag{3.5}$$

where Θ is the current model parameters set. However, we cannot calculate the value of η exactly in general, because the exact optimization with respect to η is intractable in neural networks. Following the strategy in Goodfellow et al. [2015], this value can be approximated by linearizing it as follows,

$$\eta_{\mathbf{x}} = \epsilon \frac{\mathbf{g}}{\|\mathbf{g}\|_2}, \quad \text{where} \;\; \mathbf{g} = \nabla \ell(\Theta; \mathbf{x}) \tag{3.6}$$

where ϵ can be determined on the validation set. In this way, adversarial examples are generated by adding small perturbations to the inputs in the direction that most significantly increases the loss function of the model. We find such η against the current model parameterized by Θ at each training step, and construct an adversarial example by $\mathbf{x}_{adv} = \mathbf{x} + \eta_{\mathbf{x}}$. Note that we generate this adversarial example on the word and character embedding layer, respectively, as shown in Fig. 3-8b and 3-8c. Then, the classifier is trained on the mixture of original and adversarial examples to improve its generalization. To this end, we augment the loss in Eqn. 3.4 and define the loss function for adversarial training as:

$$\ell_{AT} = \ell(\Theta; \mathbf{x}) + \ell(\Theta; \mathbf{x}_{adv}) \tag{3.7}$$

where $\ell(\Theta; \mathbf{x}), \ell(\Theta; \mathbf{x}_{adv})$ represents the loss from an original example and its adversarial counterpart, respectively. Note that we present the AT in a general form for the convenience of presentation. For different samples, the loss and parameters should correspond to their counterparts. For example, for the source data with word embedding $\mathbf{w}_S$, the loss for AT can be defined as follows, $\ell_{AT} = \ell(\Theta; \mathbf{w}_S) + \ell(\Theta; \mathbf{w}_{S,adv})$ with $\mathbf{w}_{S,adv} = \mathbf{w}_S + \eta_{\mathbf{w}_S}$ and $\ell = \ell_{GRAD} + \ell_S$. Similarly, we can compute the perturbations $\eta_{\mathbf{c}}$ for char-embedding and $\eta_{\mathbf{w}_T}$ for target word embedding.

3.3.4 Experiments

3.3.4.1 Datasets

In order to evaluate the performance of DATNet, we conduct the experiments on the following widely used NER datasets: CoNLL-2003 English NER [Kim and De, 2003], CoNLL-2002 Spanish & Dutch NER [Kim, 2002], WNUT-2016 & 2017 English Twitter NER [Zeman, 2017]. The statistics of these datasets are described in Table 3.25. We use the official split of training/validation/test sets. Since our goal is to study the effects of transferring knowledge from a high-resource dataset to a low-resource dataset, unlike previous works [Chiu and Nichols, 2016; Collobert et al., 2011; Yang et al., 2017b] that append one-hot gazetteer features to the input of the CRF layer, and the works [Aguilar et al., 2017; Limsopatham and Collier, 2016; Partalas et al., 2016] that introduce orthographic feature as additional input for learning social media NER in tweets, we do *not* experiment with hand-crafted features and only consider word and character embeddings as the inputs of our model. We used only the train set for model training for all datasets except the WNUT-2016 NER dataset. Since in this dataset, all the previous studies merged the training and validation sets together for training, we followed the same procedure, for fair comparison. Specifically, we use the CoNLL-2003 English NER dataset as high-resource (i.e., source) for all the experiments on CoNLL and WNUT datasets, while CoNLL-2002 Spanish & Dutch NER datasets and WNUT-2016 & 2017 Twitter NER datasets as low-resource (i.e., target) in cross-language and cross-domain NER settings, respectively.

In addition to the CoNLL and WNUT datasets, we also experiment on the cross-language named entity dataset described in Pan et al. [2017], which contains datasets for 282 languages including more low-resource languages such as *Galician (gl)*, *West Frisian (fy)*, *Ukrainian (uk)* and *Marathi (mr)*, etc., to evaluate our methods and investigate the transferability of different linguistic families and branches in both low- and high-resource scenarios. We choose 9 languages in our experiment, where *Galician (gl)*, *West Frisian (fy)*, *Ukrainian (uk)* and *Marathi (mr)* are target languages, the corresponding source languages are *Spanish (es)*, *Dutch (nl)*, *Russian (ru)* and

Hindi (hi), and *Arabic (ar)* is also a source language, which is from different linguistic family. Following the setting in Cotterell and Duh [2017], we also simulate the low- and high-resource scenarios by creating 100 and 10,000 sentences split for training target language datasets, respectively. Then we create 1,000 sentences split for validation and test, respectively. For source languages, we create 10,000 sentence split for training only. For high-resource scenario, we only conduct experiments on *Galician (gl-high)* and *Ukrainian (uk-high)*. The list of selected datasets are described in Table 3.26.

Benchmark	Resource	Language	# Training Tokens (# Entities)	# Dev Tokens (# Entities)	# Test Tokens (# Entities)
CoNLL-2003	Source	English	204,567 (23,499)	51,578 (5,942)	46,666 (5,648)
			Cross-language NER		
CoNLL-2002	Target	Spanish	207,484 (18,797)	51,645 (4,351)	52,098 (3,558)
CoNLL-2002	Target	Dutch	202,931 (13,344)	37,761 (2,616)	68,994 (3,941)
			Cross-domain NER		
WNUT-2016	Target	English	46,469 (2,462)	16,261 (1,128)	61,908 (5,955)
WNUT-2017	Target	English	62,730 (3,160)	15,733 (1,250)	23,394 (1,740)

Table 3.25: Statistics of CoNLL and WNUT Named Entity Recognition Datasets.

3.3.4.2 Experimental Setup

We use 50-dimensional publicly available pre-trained word embeddings for English, Spanish and Dutch languages of the CoNLL and WNUT datasets in our experiments,

Language	Resource	Linguistic Family	Linguistic Branch	# Training Sentences	# Dev Sentences	# Test Sentences
Spanish (es)	Source	Indo-European	Romance	10,000	-	-
Galician (gl / gl-high)	Target	Indo-European	Romance	100 / 10,000	1,000	1,000
Dutch (nl)	Source	Indo-European	Germanic	10,000	-	-
West Frisian (fy)	Target	Indo-European	Germanic	100	1,000	1,000
Russian (ru)	Source	Indo-European	Slavic	10,000	-	-
Ukrainian (uk / uk-high)	Target	Indo-European	Slavic	100 / 10,000	1,000	1,000
Hindi (hi)	Source	Indo-European	Indo-Aryan	10,000	-	-
Marathi (mr)	Target	Indo-European	Indo-Aryan	100	1,000	1,000
Arabic (ar)	Source	Afro-Asiatic	Semitic	10,000	-	-

Table 3.26: List of Named Entity Recognition Datasets in Pan et al. [2017].

which are trained by the word2vec package[10] on the corresponding Wikipedia articles (2017-12-20 dumps) [Lin et al., 2018]. For the named entity datasets selected from Pan et al. [2017], we use 300-dimensional pre-trained word embeddings trained by the fastText package[11] on Wikipedia [Bojanowski et al., 2017], and 30-dimensional randomly initialized character embeddings are used for all the datasets. We set the filter number as 20 for char-level CNN and the dimension of hidden states of the word-level LSTM as 200 for both base model and DATNet-F. For DATNet-P, we set 100 for source, share, and target LSTM dimension, respectively. Parameter optimization is performed by the Adam optimizer [Kingma and Ba, 2014] with gradient clipping of 5.0 and the learning rate decay strategy. We set the initial learning rate as $\beta_0 = 0.001$ for all experiments. At each epoch t, learning rate β_t is updated using $\beta_t = \beta_0/(1+\rho \times t)$, where ρ is decay rate and set as 0.05. To reduce over-fitting, we also apply Dropout [Srivastava et al., 2014] to the embedding layer and the output of the LSTM layer.

3.3.5 Results

3.3.5.1 Comparison with State-of-The-Art Results

In this section, we compare our approach with state-of-the-art (SOTA) methods on the CoNLL and WNUT benchmark datasets. In the experiment, we exploit all the source data (i.e., CoNLL-2003 English NER) and target data to improve performance on target tasks. The averaged results with standard deviation over 10 repetitive runs are summarized in Table 3.27, and we also report the best results on each task for fair comparison with other SOTA methods. From results, we observe that incorporating the additional resource is helpful to improve performance. The DATNet-P model achieves the highest F1 score on CoNLL-2002 Spanish and second F1 score on CoNLL-2002 Dutch, while the DATNet-F model beats others on WNUT-2016 and WNUT-2017 English Twitter datasets. Differently from other state-of-the-art models, DATNets do *not* use any additional features[12].

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

[11] https://github.com/facebookresearch/fastText

[12] Although our model performance on CONLL-2002 Dutch NER dataset is only comparable to the SOTA result, on the one hand, we do not use any additional features that the SOTA method

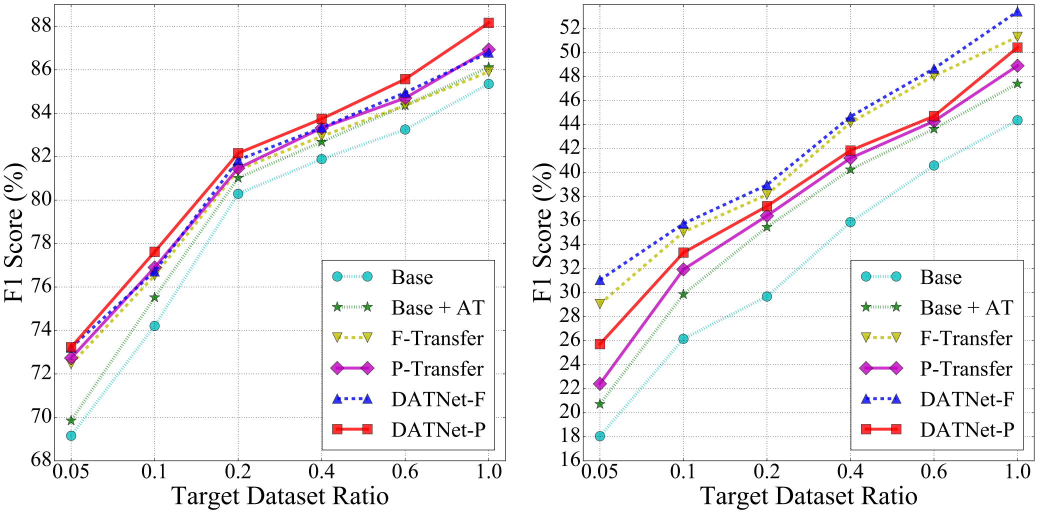
Mode	Methods		Additional Features			CoNLL Datasets		WNUT Datasets	
			POS	Gazetteers	Orthographic	Spanish	Dutch	WNUT-2016	WNUT-2017
Mono-language /domain	Gillick et al. [2016]		×	×	×	82.59	82.84	-	-
	Lample et al. [2016]		×	√	×	85.75	81.74	-	-
	Partalas et al. [2016]		√	√	√	-	-	46.16	-
	Limsopatham and Collier [2016]		×	×	√	-	-	52.41	-
	Lin et al. [2017a]		√	√	×	-	-	-	40.42
	Our Base Model	Best	×	×	×	85.53	85.55	44.96	35.20
		Mean & Std				85.35±0.15	85.24±0.21	44.37±0.31	34.67±0.34
Cross-language /domain	Yang et al. [2017b]		×	√	×	85.77	85.19	-	-
	Lin et al. [2018]		×	√	×	85.88	86.55	-	-
	Feng et al. [2018b]		√	×	×	86.42	**88.39**	-	-
	von Däniken and Cieliebak [2017]		×	√	×	-	-	-	40.78
	Aguilar et al. [2017]		√	×	√	-	-	-	41.86
	DATNet-P	Best	×	×	×	**88.16**	88.32	50.85	41.12
		Mean & Std				87.89±0.18	88.09±0.13	50.41±0.32	40.52±0.38
	DATNet-F	Best	×	×	×	87.04	87.77	**53.43**	**42.83**
		Mean & Std				86.79±0.20	87.52±0.19	53.03±0.24	42.32±0.32

Table 3.27: Comparison with State-of-the-art Results in CoNLL and WNUT datasets (F1-score).

3.3.5.2 Transfer Learning Performance

In this section, we investigate the improvements with transfer learning under multiple low-resource settings with partial target data. To simulate a low-resource setting, we randomly select subsets of target data with varying data ratio at 0.05, 0.1, 0.2, 0.4, 0.6, and 1.0. For example, 20,748 training tokens are sampled from the training set under a data ratio of $r = 0.1$ for the dataset CoNLL-2002 Spanish NER (Cf. Table 3.25). The results for cross-language and cross-domain transfer are shown in Fig. 3-9a and 3-9b, respectively, where we compare the results with each part of DATNet under various data ratios. From those figures, we have the following observations: 1) both adversarial training and the adversarial discriminator in DATNet consistently contribute to the performance improvement; 2) the transfer learning component in the DATNet consistently improves over the base model results and the improvement margin is more substantial when the target data ratio is lower. For example, when the data ratio is 0.05, the DATNet-P model outperforms the base model by more than 4%

did use; on the other, we are not sure if the SOTA method has incorporated the validation set into training. And if we merge training and validation sets, we can push the F1 score to **88.71**, which outperforms the SOTA method.



(a) CoNLL-2002 Spanish NER

(b) WNUT-2016 Twitter NER

Figure 3-9: Comparison with Different Target Data Ratio, where AT stands for adversarial training, F(P)-Transfer denotes the DATNet-F(P) without AT.

absolutely in F1-score on Spanish NER and the DATNet-F model improves around 13% absolutely in F1-score compared to the base model on WNUT-2016 NER.

In the second experiment, we further investigate DATNet in the extremely low resource cases, e.g., the number of training target sentences is 10, 50, 100, 200, 500 and 1,000. The setting is quite challenging and fewer previous works have studied this before. The results are summarized in Table 3.28. We have two interesting observations[13]: 1) DATNet-F outperforms DATNet-P on cross-language transfer when the target resource is extremely low; however, this situation is reversed when the target dataset size is large enough (here for this specific dataset, the threshold is 100 sentences); 2) DATNet-F is always superior to DATNet-P on cross-domain transfer. For the first observation, it is because DATNet-F with more shared hidden units is more efficient to transfer knowledge than DATNet-P when data size is extremely small. For the second observation, because cross-domain transfers are in the same language, more knowledge is in common between the source and target domains, requiring more shared hidden features to carry over this knowledge compared to cross-language transfer. Therefore, for cross-language transfer with extremely low resources

[13] For other tasks/languages we have the similar observation, we only report CoNLL-2002 Spanish and WNUT-2016 Twitter results as illustration.

and cross-domain transfer, we suggest using the DATNet-F model to achieve better performance. As for cross-language transfer with relatively more training data, the DATNet-P model is preferred.

Tasks	CoNLL-2002 Spanish NER						WNUT-2016 Twitter NER					
# Target train sentences	10	50	100	200	500	1000	10	50	100	200	500	1000
Base	21.53	42.18	48.35	63.66	68.83	76.69	3.80	14.07	17.99	26.20	31.78	36.99
+ AT	19.23	41.01	50.46	64.83	70.85	77.91	4.34	16.87	18.43	26.32	35.68	41.69
+ P-Transfer	29.78	61.09	64.78	66.54	72.94	78.49	7.71	16.17	20.43	29.20	34.90	41.20
+ F-Transfer	39.72	63.00	63.36	66.39	72.88	78.04	15.26	20.04	26.60	32.22	38.35	44.81
DATNet-P	39.52	62.57	64.05	**68.95**	**75.19**	**79.46**	9.94	17.09	25.39	30.71	36.05	42.30
DATNet-F	**44.52**	**63.89**	**66.67**	68.35	74.24	78.56	**17.14**	**22.59**	**28.41**	**32.48**	**39.20**	**45.25**

Table 3.28: Experiments on Extremely Low Resource (F1-score).

3.3.5.3 Cross-language Transfer Learning

Table 3.29 summarizes the results of our methods under different cross-language transfer settings as well as the comparison with Cotterell and Duh [2017]. In this experiment, we study the transferability between languages not only from same linguistic family and branch, but also from different linguistic families or branches. According to the results, DATNets outperform the transfer method of Cotterell and Duh [2017] for both low- and high-resource scenarios within the same linguistic family and branch (i.e., in-family in-branch) transfer case. We also observe that: 1) For the low-resource scenario, transfer learning is significantly helpful for improving the performance of target datasets within both same and different linguistic family or branch (i.e., in/cross-family in/cross-branch) transfer cases, while the improvements are more prominent under the in-family in-branch case. 2) For the high-resource scenario, say, when the target language data is sufficient, the improvements of transfer learning are not very distinct compared with that for low-resource scenario under in-family in-branch case. We also find that there is no effect by transferring knowledge from *Arabic* to *Galician* and *Ukrainian*. We suspect that it is caused by the great linguistic differences between source and target languages, since, for example, *Arabic* and *Galician* are from totally different linguistic families.

Language		Transferring Strategy	Cotterell and Duh [2017]		Our Methods		
Source	Target		Base Model	Transfer	Base Model	DATNet-P	DATNet-F
nl	fy	In-Family In-Branch	58.43	72.12	57.47	75.08	76.05
hi	fy	In-Family Cross-Branch	-	-	57.47	69.25	68.44
ar	fy	Cross-Family Cross-Branch	-	-	57.47	67.89	66.05
hi	mr	In-Family In-Branch	39.02	60.92	43.55	68.55	64.87
nl	mr	In-Family Cross-Branch	-	-	43.55	63.83	60.50
ar	mr	Cross-Family Cross-Branch	-	-	43.55	63.28	59.76
es	gl	In-Family In-Branch	49.19	76.40	49.94	79.60	86.01
hi	gl	In-Family Cross-Branch	-	-	49.94	60.57	61.68
ar	gl	Cross-Family Cross-Branch	-	-	49.94	59.18	60.43
es	gl-high	In-Family In-Branch	89.42	89.46	92.78	93.14	93.02
ar	gl-high	Cross-Family Cross-Branch	-	-	92.78	92.63	92.21
ru	uk	In-Family In-Branch	60.65	76.74	61.48	79.02	80.76
hi	uk	In-Family Cross-Branch	-	-	61.48	72.73	73.84
ar	uk	Cross-Family Cross-Branch	-	-	61.48	71.55	72.24
ru	uk-high	In-Family In-Branch	87.39	87.42	93.29	93.62	93.51
ar	uk-high	Cross-Family Cross-Branch	-	-	93.29	92.83	92.42

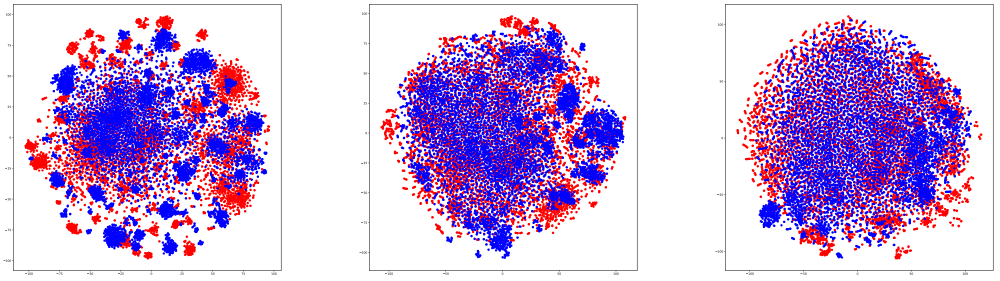
Table 3.29: Results of Varying Cross-language Transfer Settings in Pan et al. [2017] Datasets (F1-score). Base model means the model is trained by using target language dataset only.

3.3.5.4 Ablation Study of DATNet

In the proposed DATNet, both GRAD and AT play important roles in low resource NER. In this experiment, we further investigate how GRAD and AT help transfer knowledge across language/domain. In the first experiment[14], we used t-SNE [Maaten and Hinton, 2008] to visualize the feature distribution of BiLSTM outputs without AD, with normal AD (GRAD without considering data imbalance), and with the proposed GRAD in Figure 3-10. From this figure, we can see that the GRAD in DATNet makes the distribution of extracted features from the source and target datasets much more similar by considering the data imbalance, which indicates that the outputs of BiLSTM are resource-invariant.

To better understand the working mechanism, Table 3.30 further reports the quantitative performance comparison between models with different components. We observe that GRAD shows stable superiority over the normal AD regardless of other

[14] We used data ratio $\rho = 0.5$ for training model and randomly selected 10k testing data for visualization.



(a) no AD (b) AD (c) GRAD

Figure 3-10: The visualization of extracted features from shared bidirectional-LSTM layer. The left, middle, and right figures show the results when no Adversarial Discriminator (AD), AD, and GRAD is performed, respectively. Red points correspond to the source CoNLL-2003 English examples, and blue points correspond to the target CoNLL-2002 Spanish examples.

components. There is no consistent winner between DATNet-P and DATNet-F on different settings. The DATNet-P architecture is more suitable to cross-language transfer while DATNet-F is more suitable to cross-domain transfer.

CoNLL-2002 Spanish NER				WNUT-2016 Twitter NER			
Model	F1-score	Model	F1-score	Model	F1-score	Model	F1-score
Base	85.35	+AT	86.12	Base	44.37	+AT	47.41
+P-T (no AD)	86.15	+AT +P-T (no AD)	86.90	+P-T (no AD)	47.66	+AT +P-T (no AD)	48.44
+F-T (no AD)	85.46	+AT +F-T (no AD)	86.17	+F-T (no AD)	49.79	+AT +F-T (no AD)	50.93
+P-T (AD)	86.32	+AT +P-T (AD)	87.19	+P-T (AD)	48.14	+AT +P-T (AD)	49.41
+F-T (AD)	85.58	+AT +F-T (AD)	86.38	+F-T (AD)	50.48	+AT +F-T (AD)	51.84
+P-T (GRAD)	86.93	+AT +P-T (GRAD) (***DATNet-P***)	**88.16**	+P-T (GRAD)	48.91	+AT +P-T (GRAD) (***DATNet-P***)	50.85
+F-T (GRAD)	85.91	+AT +F-T (GRAD) (***DATNet-F***)	87.04	+F-T (GRAD)	51.31	+AT +F-T (GRAD) (***DATNet-F***)	**53.43**

Table 3.30: Quantitative Performance Comparison between Models with Different Components. AT: Adversarial Training; P-T: P-Transfer; F-T: F-Transfer; AD: Adversarial Discriminator; GRAD: Generalized Resource-Adversarial Discriminator.

From the previous results, we know that AT helps enhance the overall performance by adding perturbations to inputs with the limit of $\epsilon = 5$, i.e., $\|\eta\|_2 \leq 5$. In this experiment, we further investigate how target perturbation $\epsilon_{\mathbf{w}_T}$ with fixed source perturbation $\epsilon_{\mathbf{w}_S} = 5$ in AT affects knowledge transfer and the results on Spanish NER are summarized in Table 3.31. The results generally indicate that less training data requires a larger ϵ to prevent over-fitting, which further validates the necessity of AT in the case of low resource data.

$\epsilon_{\mathbf{w}_T}$	1.0	3.0	5.0	7.0	9.0
Ratio	CoNLL-2002 Spanish NER				
$\rho = 0.1$	75.90	76.23	77.38	77.77	**78.13**
$\rho = 0.2$	81.54	81.65	81.32	**81.81**	81.68
$\rho = 0.4$	83.62	83.83	83.43	**83.99**	83.40
$\rho = 0.6$	84.44	84.47	**84.72**	84.04	84.05

Table 3.31: Analysis of Maximum Perturbation $\epsilon_{\mathbf{w}_T}$ in AT with Varying Data Ratio ρ (F1-score).

Finally, we analyze the discriminator weight α in GRAD and results are summarized in Table 3.32. From the results, it is interesting to find that α is directly proportional to the data ratio ρ, which means that more target training data requires larger α (i.e., smaller $1 - \alpha$ to reduce training emphasis on the target domain) to achieve better performance.

α	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8
Ratio	CoNLL-2002 Spanish NER														
$\rho = 0.1$	78.37	78.63	**78.70**	78.32	77.96	77.92	77.88	77.78	77.85	77.90	77.65	77.57	77.38	77.49	77.29
$\rho = 0.2$	80.99	81.71	**82.18**	81.57	81.53	81.55	81.44	81.25	81.32	81.16	81.02	81.16	80.63	80.79	80.54
$\rho = 0.4$	83.76	83.73	84.18	**84.48**	84.26	84.12	83.54	83.40	83.52	84.18	83.42	83.47	83.28	83.33	83.19
$\rho = 0.6$	85.18	85.24	85.85	85.68	85.84	**86.10**	85.71	85.74	85.42	85.60	85.20	85.40	85.26	85.24	84.98

Table 3.32: Analysis of Discriminator Weight α in GRAD with Varying Data Ratio ρ (F1-score).

3.3.6 Summary

In this section we develop a transfer learning model DATNet for low-resource NER, which aims at addressing two problems remained in existing work, namely representation difference and resource data imbalance. We introduce two variants of DATNet, DATNet-F and DATNet-P, which can be chosen for use according to the cross-language/domain user case and the target dataset size. To improve model generalization, we propose dual adversarial learning strategies, i.e., AT and GRAD. Extensive experiments show the superiority of DATNet over existing models and it achieves new state-of-the-art performance on CoNLL NER and WNUT NER benchmark datasets.

3.4 Conclusions

In this chapter, we have explored various transfer learning methods on three types of natural language understanding tasks: multi-choice question answering, dialogue state tracking, and named entity recognition. For the MCQA task, we present a transfer learning framework that combines multi-task learning and sequential transfer learning, which achieves significant improvements over either alone. To leverage the success of question answering models and abundant data from this task, we adapt the dialogue state tracking task into a combination of multi-choice and span-based question answering tasks and achieve decent performance on the few-shot and zero-shot settings via sequential transfer learning. Finally, for the task of NER, we propose a dual adversarial transfer learning framework for domain adaptation so that the low-resource domains can be boosted by a large margin by the high-resource source domain data. Overall, we have made novel contributions to all three of the most popular paradigms of transfer learning, as mentioned in Section 2.2.2.

Chapter 4

Transfer Learning for Natural Language Generation

In Chapter 2, we have reviewed the previous methods for transfer learning; however, the majority of them are targeted for NLU tasks. And in Chapter 3, we introduced our newly proposed transfer learning methods specially for NLU. Now in this chapter we will describe our work on transfer learning for Natural Language Generation (NLG) (this chapter is based on this work: [Jin et al., 2020c]), which is fundamentally different from those for NLU due to the difference in nature between NLU and NLG. Specifically, we will focus on the machine translation task, which is the most important and widely studied NLG task. Although it is easy to obtain millions of translation pairs as supervised data for several most widely spoken languages such as English, French, Chinese, etc., there are still hundreds of languages that are used by small populations and thus it is difficult to collect enough data to train a decent neural model for them. Even for those languages spoken by large populations, we are still facing the low-resource problem for those specialized domains such as education, legislation, technology, and medicine, to name a few. Moreover, in this era of deep learning, we are compelled to adopt larger and larger neural models due to their proved unparalleled performance, which further amplifies the problem of data insufficiency. One of the most effective solutions to this issue is transfer learning, which distills the knowledge from high-resource languages/domains as much as possi-

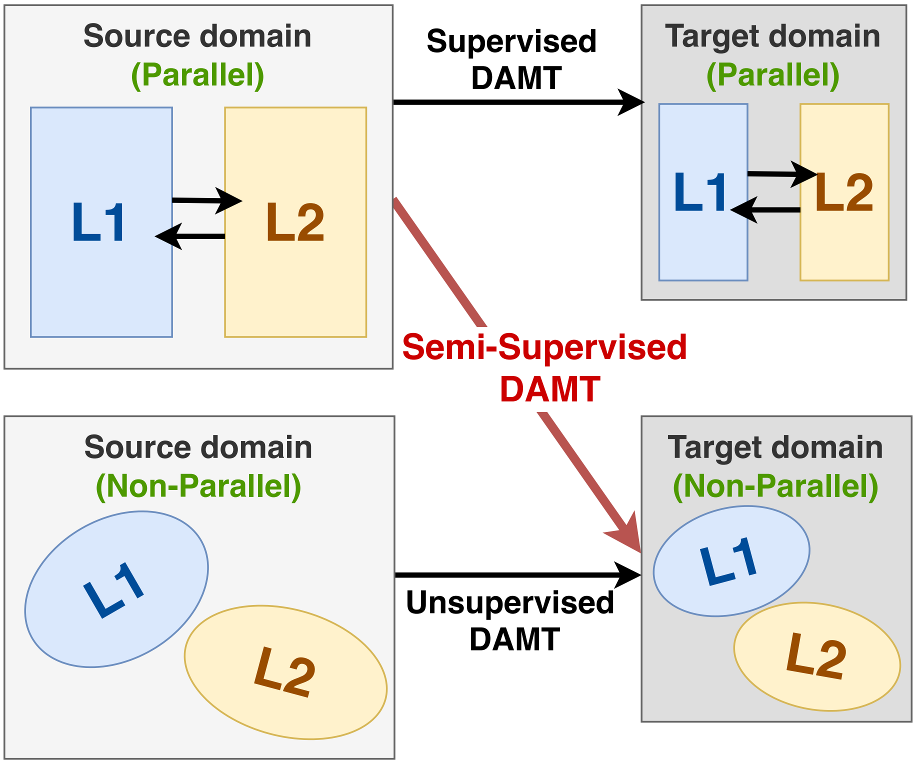


Figure 4-1: Three types of DAMT: supervised, unsupervised, and semi-supervised (the focus of our paper). L1 is the source language and L2 is the target language of MT.

ble to help improve their low-resource counterparts. This chapter will propose a new method in this direction, more specifically on domain adaptation, which focuses on generalizing a translation model trained on a source domain with a large amount of supervised data to the target domain with only unsupervised data.

4.1 Introduction

Machine Translation (MT) is an attractive and successful research field. For many general domains, millions of parallel data are annotated. For example, the WMT14 dataset alone has 4M supervised sentence pairs. However, for more specific domains such as law, medicine, and technology [Nakov and Tiedemann, 2012], there is very little supervised data. In practice, collecting supervised data in specialized fields is expensive and in some cases even impractical.

To obtain a good translation model on these specialized domains (our target domains), semi-supervised domain adaptation for machine translation (DAMT) has become an active research field. It aims to generalize the MT models trained on the source domain with large-scale supervised data to the target domain that has no supervised data at all, as illustrated in Figure 4-1.

Existing approaches for semi-supervised DAMT can be categorized into two lines: *model-centric* and *data-centric* methods. Model-centric methods focus on multi-task learning of the translation task on the source domain parallel data and the language modeling task on the target domain target-language data [Domhan and Hieber, 2017; Dou et al., 2019]. By contrast, data-centric methods rely on a pseudo-parallel corpus constructed either by simply copying the target-language sentences to the source side in the target domain, termed *Copy* [Currey et al., 2017], or by pairing the target-language sentences with their back-translated counterparts by a well-trained MT model, termed *back-translation* [Sennrich et al., 2015]. Specifically, back-translation first generates translations from the target language to the source language, and then trains the translation model to map the generated sentences back to the target language. Despite its simplicity, the *back-translation* strategy has been demonstrated to be most effective in many cases.

Inspired by the success of back-translation, we propose a simple but much stronger approach as illustrated by Figure 4-2. We first initialize both encoder and decoder of the sequence-to-sequence model with pre-trained parameters as a good starting point. The pre-training process is implemented via language modeling over large-scale monolingual corpora from Wikipedia. Afterwards, we implement iterative *back-translation* (in both L1→L2 and L2→L1 directions) and *language modeling* training over the target domain non-parallel data, which serves as our base method. We further augment this base method by incorporating the supervised translation training step over the source domain data into each iteration, which leads to more significant performance gains.

Despite the simple nature of this method, we call attention to our approach because it demonstrates *significant* improvements over all previous state-of-the-art

DAMT models on all experiments. We conduct experiments with two different domain adaptation settings and on two language pairs. First, for domain adaptation between two specific domains, our base method achieves up to +9.48 BLEU score improvement over the strongest out of four baseline models and +27.77 BLEU over the unadapted baseline. Second, for domain adaptation from a general domain with large-scale supervised data (WMT) to specific domains, our model combined with data augmentation by supervised source domain data achieves up to +19.31 BLEU improvement over the best previous method and +47.69 BLEU improvement over the unadapted model. Source code is provided at: https://github.com/jind11/DAMT.

4.2 Related Work

There are three scenarios for domain adaptation for machine translation, as illustrated in Figure 4-1:

1. **Supervised:** Both source and target domains have supervised parallel data, although the amount of source domain data is much larger than that of the target.

2. **Unsupervised:** Neither of the source or target domains has parallel data.

3. **Semi-supervised:** Only the source domain has parallel data while the target does not.

Supervised DAMT Most previous works for DAMT focus on the supervised setting [Chu et al., 2017; Fadaee and Monz, 2018; Guzmán et al., 2019]. For example, sequential fine-tuning [Freitag and Al-Onaizan, 2016; Luong and Manning, 2015] first trains a neural machine translation (NMT) model on source domain data and subsequently fine-tunes it on the target domain data. Britz et al. [2017] proposes to jointly train the translation task and the domain discrimination task to mitigate the domain shift. Kobus et al. [2016] uses the domain tokens and domain embeddings to force the NMT model to take into account the domain information. Joty et al. [2015] assigns

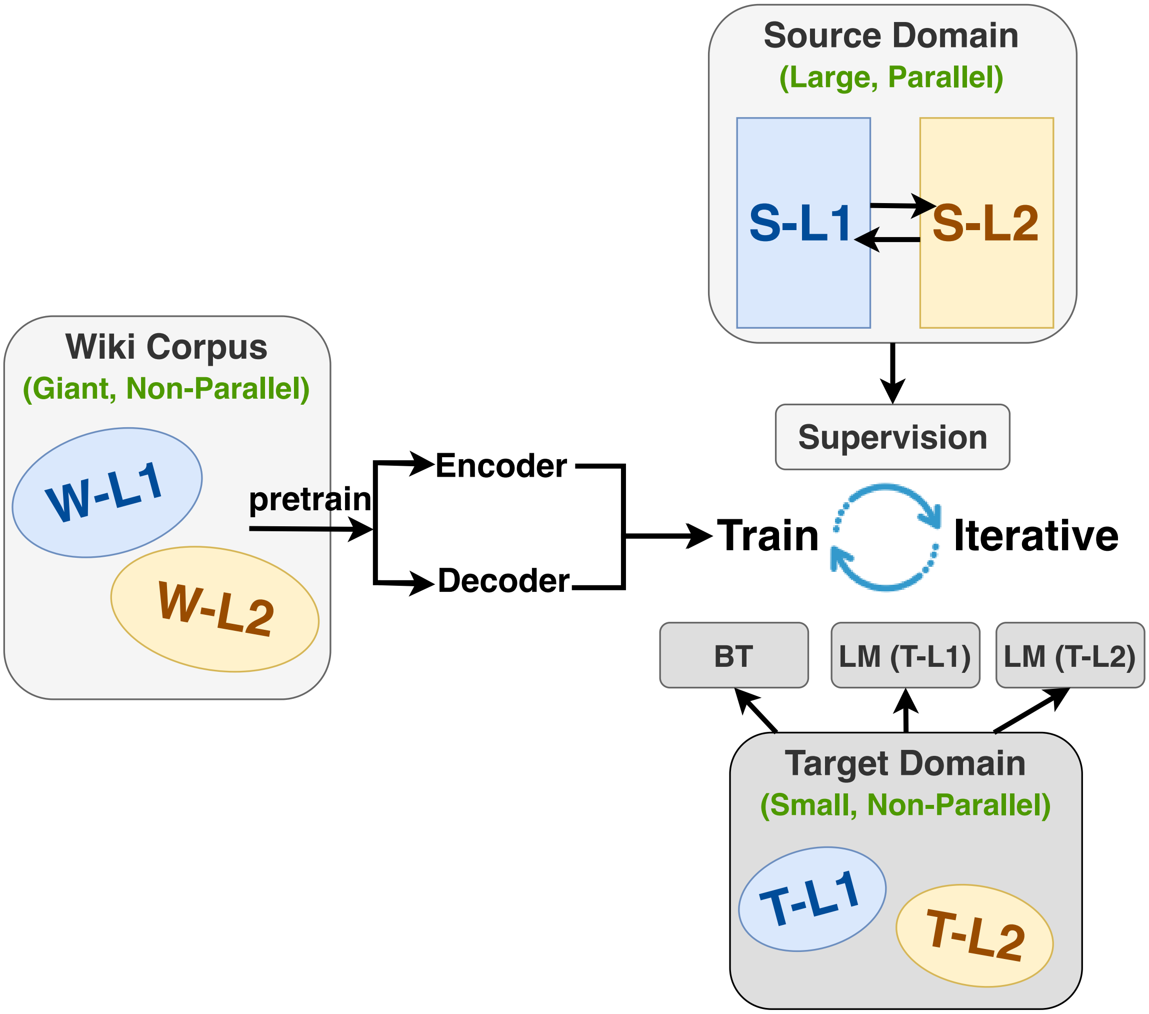


Figure 4-2: The schema of our method. We first initialize the encoder and decoder via language modeling pre-training over the wiki monolingual corpora for both the source language (W-L1) and target language (W-L2). Then we train the model via iteratively optimizing the back-translation loss (BT in both T-L1→T-L2 and T-L2→T-L1 directions) and two language modeling losses (LM (T-L1) and LM (T-L2)) on the non-parallel target domain data, as the base method. We further enhance this base by adding one more optimization step on the supervised translation loss using the source domain data (e.g., S-L1→S-L2).

higher weights to those source domain data that more resemble the target domain so as to remove unwanted noise.

Unsupervised DAMT DAMT in the unsupervised setting has started only recently, where Sun et al. [2019a] defines several scenarios for it and proposes modified domain adaptation methods to improve the performance of adaptation in these scenarios.

Semi-Supervised DAMT Our proposed baseline falls under the semi-supervised scenario, where related works can be divided into two threads: data-centric and model-centric. Data-centric methods mainly propose to select or generate domain-related pseudo-parallel data for model training. For data selection, Duh et al. [2013] uses language models to rank the source domain data and select the top-ranked parallel sentences as synthetic data. More representative methods are back translation-based [Sennrich et al., 2015] and copy-based [Currey et al., 2017], which are simple yet have been widely demonstrated to be effective. On the other hand, model-based methods propose to change model architectures to leverage the monolingual corpus by introducing a new learning objective, such as auto-encoding [Cheng et al., 2016] and language modeling [Dou et al., 2019; Ramachandran et al., 2017] on the target-side sentences.

In contrast, we would like to re-visit the classic *back-translation* method and propose extending it to the online iterative version so as to make better use of the target domain data in an unsupervised manner. The iterative back-translation scheme we adopt has achieved great success in unsupervised NMT and text style transfer in the past two years [Artetxe et al., 2017; He et al., 2016; Jin et al., 2019c; Lample et al., 2018]. In this paper, we propose a novel adaptation of it to the specific setting of semi-supervised DAMT and achieve profound improvements over all previous state-of-the-art methods.

4.3 Method

In this section, we first introduce the architecture of our method, and then formulate the overall training strategy.

4.3.1 Model Architecture

We adopt the Transformer [Vaswani et al., 2017] with the encoder-decoder structure as the sequence-to-sequence translation model, as shown in Figure 4-3. Following the practice in [Lample and Conneau, 2019], we add the language embeddings to the standard token and position embeddings via the element-wise summation operation. This language embedding can inform both encoder and decoder which language it is processing. For instance, when translating from German to English, we set the language embedding of the encoder to German (through a look-up table) while setting that of the decoder to English. For the reversed direction of translation (i.e., from English to German), we just need to reverse the language embedding settings for the encoder and decoder without changing the model architecture. In this way, the same model can be used to translate any language pair.

Three key properties of our model are introduced in the following paragraphs:

Shared Sub-Word Vocabulary In our experiments, we process all languages with the same shared vocabulary created through Byte Pair Encoding (BPE) [Sennrich et al., 2016]. This not only enables us to translate between any language pair with the same model, but improves the alignment of embedding spaces across languages that share either the same alphabet or anchor tokens such as digits. The BPE splits are learned on the concatenation of sentences from the monolingual corpora.

Shared Latent Representations All encoder parameters (including the embedding matrices, since we perform joint tokenization) are shared across the source and target languages so that the encoder can map the input of any source language into a shared latent representation space, which is then translated to the target language by the decoder. Furthermore, we share the decoder parameters across the two languages to reduce parameter size. We also share the encoder and decoder between the translation and language modeling tasks (will be introduced in the Section 4.3.2), which ensures that the benefits of language modeling, implemented via the denoising auto-encoder objective, nicely transfer to translation and helps the NMT model translate more fluently.

Pre-Training Both the encoder and decoder are initialized by pre-trained parameters from Lample and Conneau [2019], which are obtained by pre-training a transformer based language model with both the conditional language modeling and masked language modeling (MLM) objectives on large-scale monolingual corpora of both languages in the language pair (the corpora are extracted from the Wikipedia dump). We refer readers to the original paper for more technical details on the pre-training process. Such initialization not only accelerates the model convergence but also improves the adaptation performance, which will be discussed in Section 4.6.1.

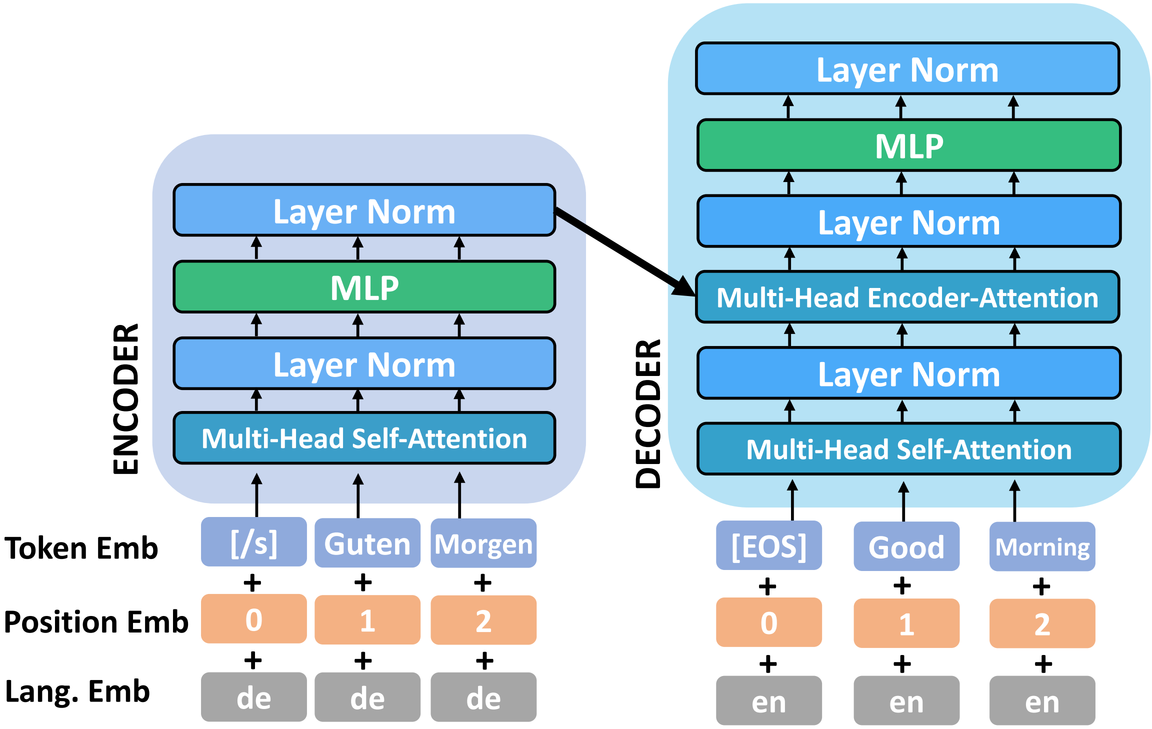


Figure 4-3: Transformer-based model architecture.

4.3.2 Training Objectives

In the semi-supervised domain adaptation setting, we assume access to *parallel* translation pairs as the training corpus $(X_{\text{src}}, Y_{\text{src}})$ in the source domain, and *non-parallel* data X_{tgt} and Y_{tgt} in the target domain. As illustrated by Figure 4-2, we train our model with the following three objectives:

Language Modeling (LM) The language modeling objective is implemented via denoising auto-encoding, by minimizing

$$\begin{aligned}\mathcal{L}_{\text{lm}} =& \mathbb{E}_{\boldsymbol{x}\in X_{\text{tgt}}}[-\log P_{\text{s}\to\text{s}}(\boldsymbol{x}|C(\boldsymbol{x});\boldsymbol{\theta})] + \\ & \mathbb{E}_{\boldsymbol{y}\in Y_{\text{tgt}}}[-\log P_{\text{t}\to\text{t}}(\boldsymbol{y}|C(\boldsymbol{y});\boldsymbol{\theta})],\end{aligned} \tag{4.1}$$

where C is a word corruption function with some words randomly dropped, blanked, and swapped; $P_{\text{s}\to\text{s}}$ and $P_{\text{t}\to\text{t}}$ are the composition of encoder and decoder both operating on the source language s and target language t, respectively; θ denotes the model parameters.

Iterative Back-Translation We have two NMT models, $\text{Model}_{\text{s2t}}(\cdot)$ which translates from the source language s to the target language t, and $\text{Model}_{\text{t2s}}(\cdot)$ vice versa (they are implemented by one model architecture). In each iteration, we translate on the fly from each source language sentence in the target domain $\boldsymbol{x} \in X_{\text{tgt}}$ to the target language sentence $\text{Model}_{\text{s2t}}(\boldsymbol{x})$. Similarly, we translate from every target sentence $\boldsymbol{y} \in Y_{\text{tgt}}$ to its counterpart in the source language $\text{Model}_{\text{t2s}}(\boldsymbol{y})$. Then the pairs of $(\boldsymbol{x}, \text{Model}_{\text{s2t}}(\boldsymbol{x}))$ and $(\text{Model}_{\text{t2s}}(\boldsymbol{y}), \boldsymbol{y})$ can be used as synthetic parallel data to train the NMT model in two directions by minimizing the following loss:

$$\begin{aligned}\mathcal{L}_{\text{back}} =& \mathbb{E}_{\boldsymbol{x}\in X_{\text{tgt}}}[-\log P_{\text{t}\to\text{s}}(\boldsymbol{x}|\text{Model}_{\text{s2t}}(\boldsymbol{x});\boldsymbol{\theta})] + \\ & \mathbb{E}_{\boldsymbol{y}\in Y_{\text{tgt}}}[-\log P_{\text{s}\to\text{t}}(\boldsymbol{y}|\text{Model}_{\text{t2s}}(\boldsymbol{y});\boldsymbol{\theta})].\end{aligned} \tag{4.2}$$

Note that, when minimizing this objective function, we do not back-propagate

Algorithm 1 Training Strategy

```
Require: Non-parallel data $X_{\text{tgt}}$ and $Y_{\text{tgt}}$ in the target domain, parallel data $(X_{\text{para}}, Y_{\text{para}})$, and model parameters $\boldsymbol{\theta}$
while $\boldsymbol{\theta}$ has not converged do
    Sample $\boldsymbol{x}$ from $X_{\text{tgt}}$ and $\boldsymbol{y}$ from $Y_{\text{tgt}}$
    Create pairs $(C(\boldsymbol{x}), \boldsymbol{x})$ and $(C(\boldsymbol{y}), \boldsymbol{y})$ via word corruption
    Update $\boldsymbol{\theta}$ by minimizing Eq. (4.1)
    Sample $\boldsymbol{x}$ from $X_{\text{tgt}}$ and $\boldsymbol{y}$ from $Y_{\text{tgt}}$
    Create $(\boldsymbol{x}, \text{Model}_{\text{s2t}}(\boldsymbol{x}))$ and $(\text{Model}_{\text{t2s}}(\boldsymbol{y}), \boldsymbol{y})$ via back-translation
    Update $\boldsymbol{\theta}$ by minimizing Eq. (4.2)
    Sample $(\boldsymbol{x}, \boldsymbol{y})$ from $(X_{\text{para}}, Y_{\text{para}})$
    Update $\boldsymbol{\theta}$ by minimizing Eq. (4.3)
end while
```

through the models that are used to generate translations.

Supervised Machine Translation When given parallel data, denoted as $(X_{\text{para}}, Y_{\text{para}})$, we can also minimize the supervised translation loss:

$$\mathcal{L}_{\text{sup}} = \mathbb{E}_{\boldsymbol{x} \in X_{\text{para}}, \boldsymbol{y} \in Y_{\text{para}}}[-\log P_{\text{s}\to\text{t}}(\boldsymbol{y}|\boldsymbol{x}; \boldsymbol{\theta})]. \tag{4.3}$$

The parallel data can be the source domain supervised data $(X_{\text{src}}, Y_{\text{src}})$ or the back-translated synthetic pairs by an NMT model trained on the source domain data.

4.3.3 Training Strategy

As shown in Algorithm 1, in each iteration, we randomly draw a batch of data to minimize the aforementioned three loss equations 4.1, 4.2 and 4.3. The training will continue until the validation set BLEU score does not increase for a certain number of iterations.

Language Pair	**Corpus**	**Words**	**Sentences**	**W/S**
De-En	MED	14,533,613	1,104,752	13.2
	LAW	18,461,140	715,372	25.8
	IT	3,212,130	337,817	9.5
	TED	3,110,970	151,627	20.5
	WMT-14	126,735,962	4,468,840	28.4
Ro-En	MED	13,142,512	990,220	13.3
	LAW	10,631,517	450,715	23.6
	TED	3,328,621	161,291	20.6
	WMT-16	10,796,138	399,375	27.0

Table 4.1: Statistics of the corpora used for training (target side).

4.4 Experiments

4.4.1 Datasets

We validate our model under two different adaptation settings. For the first setting, we test the domain adaptation ability of our method for adapting from one specific domain to another specific one on every pair of the following specific domains: law (LAW), medical (MED), and Information Technology (IT). The other setting is to adapt models trained on the general-domain WMT datasets to specific domains: TED [Duh, 2018] (TED talks),[1] LAW, and MED datasets [Tiedemann, 2012]. Two language pairs are tested, German-English (DE-EN) and Romanian-English (RO-EN). The general-domain WMT datasets for DE-EN and RO-EN come from the WMT-14[2] and WMT-16[3] tasks, respectively. Data statistics for the train sets are in Table 4.1. The size of validation and test sets for WMT-14 are both 3K, and all the other domains are 2K.

[1] https://www.ted.com/

[2] https://www.statmt.org/wmt14/translation-task.html

[3] https://www.statmt.org/wmt16/translation-task.html

For fair comparison, we follow previous works [Dou et al., 2019; Hu et al., 2019a] to build the unaligned corpus for each domain. Specifically, we randomly shuffle the original parallel corpus and split it into two equal halves. We then use the first half of the source-side sentences and the second half of the target-side ones, which form the non-parallel corpus for the target domain. In this way, we assure that there are no parallel sentences in the target domain.

4.4.2 Baselines

We compare our models with the following baselines described below.

Unadapted We train the NMT model on the supervised source domain data and directly test its performance on the target domain.

Copy [Currey et al., 2017] The target-side sentences in the target domain are copied to the source-side, and then they are combined with the parallel source domain data as the train data to train an NMT model.

Back [Sennrich et al., 2015] This is the one time *back-translation* baseline. A target-to-source NMT model is first trained on the parallel source domain data and then used to generate pseudo parallel data in the target domain for model training by translating the target domain target-language sentences to the source language.

DALI [Hu et al., 2019a] Lexicon induction is first performed to extract a lexicon in the target domain, and then a pseudo-parallel target domain corpus is constructed by performing word-to-word back-translation of monolingual sentences of the target language, which is used for fine-tuning a pre-trained source domain NMT model.

DAFE [Dou et al., 2019] It performs multi-task learning on a translation model on source domain parallel data and a language model on target domain target-side monolingual data, while inserting domain and task embedding learners into the transformer-based model.

4.4.3 Settings

Both encoder and decoder in the transformer model have 6 layers, 8 heads, and a dimension of 1024. For the word corruption function, word dropping and blanking adopt a uniform distribution with a probability of 0.1, and word shuffling is implemented with a window of 3 tokens. The Adam optimizer uses a learning rate of 0.0001.

Our implemented methods involve three variants:

IBT It serves as the base of our method. For this variant, we do not use any supervised data. And we train our model by optimizing Equation 4.1 and 4.2 only with the target domain non-parallel data.

IBT+SRC Based on the variant of IBT, besides optimizing Equation 4.1 and 4.2, we additionally optimize Equation 4.3 using the supervised data from the source domain.

IBT+BACK Similar to the variant of IBT+SRC, instead of using the source domain data for solving Equation 4.3, we use the pseudo parallel data provided by the aforementioned baseline BACK.

4.5 Results

4.5.1 Main Results

Adapting between Specific Domains Our main results are shown in Table 4.2, with the left six columns showing the adaptation setting where models are adapted between specific domains. In this table, the second row lists the source domains whereas the third row shows the target domains. From this table, we see that the unadapted baseline model, UNADAPTED, performs very poorly, verifying the previous statement that current NMT models cannot generalize well to test data from a new domain. In contrast, the copy method, COPY, and back-translation method, BACK,

	Methods	DE to EN									RO to EN		
		LAW		MED		IT		WMT-14			WMT-16		
		MED	IT	LAW	IT	LAW	MED	TED	LAW	MED	TED	LAW	MED
(1)	Unadapted	18.76	6.62	7.92	5.94	6.19	10.90	23.36	23.77	24.42	23.59	33.26	18.39
(2)	Copy	23.57	10.58	11.44	12.83	9.39	18.19	24.32	25.25	27.67	29.29	38.23	27.37
(3)	Back	33.94	22.21	23.74	23.56	22.43	31.00	31.02	31.27	35.69	36.98	49.28	43.70
(4)	DALI	11.32	8.75	26.98	19.49	11.65	10.99	–	–	–	–	–	–
(5)	DAFE	26.96	15.41	14.28	13.03	11.67	21.30	**34.89**	31.46	38.79	37.05†	49.63†	46.77†
(6)	IBT	38.67	31.69	27.89	31.69	27.89	38.67	30.88	27.89	38.67	34.48	49.45	61.55
(7)	IBT+Src	**41.22**	34.33	29.54	32.47	30.20	39.77	33.23	32.81	41.40	38.68	53.49	60.98
(8)	IBT+Back	40.40	**35.41**	**30.27**	**35.76**	**30.49**	**40.28**	34.15	**33.35**	**42.08**	**38.90**	**54.39**	**66.08**
(9)	MT (Sup.)	48.95	59.38	37.72	59.38	37.72	48.95	38.97	37.72	48.95	42.22	61.69	80.32

Table 4.2: Translation accuracy (BLEU) under different settings. The second and third columns are source and target domains respectively. "DE", "EN", and "RO" denote German, English, and Romanian, respectively. DALI and DAFE results are the best results from the original papers, except that numbers marked by † are from our re-implementation. Settings (7) IBT+Src and (8) IBT+Back uses the out-of-domain data and back-translated data to minimize the supervised machine translation loss, respectively. (9) "MT (Sup.)" results are obtained by training an NMT model on the supervised target domain data.

can significantly improve the adaptation performance, with Back showing much superior performance. Note that, Back even outperforms the other two baselines: DALI and DAFE, by a large margin in the majority of cases, although it was proposed earlier and is much simpler.

Our method variant IBT, shown in row (6) of Table 4.2, achieves higher performance than all baselines, with absolute gains of +0.91 to +9.48 BLEU scores over the strongest baseline, and +19.91 to +27.77 BLEU scores over the Unadapted baseline. Notably, IBT only needs the target domain non-parallel data but can still substantially outperform those baselines that rely on the parallel source domain data (e.g., Back, DALI, DAFE), indicating that previous methods have not exhausted the potential contained within the target domain data.

Adapting from a General to a Specific Domain In the second adaptation setting, the right six columns of Table 4.2 show the results by adapting a NMT model trained on the general domain corpus (WMT) to specific domains (TED, LAW, and MED) for two language pairs: from German to English and from Romanian to English. In this setting, both Copy and Back achieve better performance compared to the previous setting where the source domain is specific. Our method variant IBT surpasses Unadapted by a large margin but it does not outperform the baselines Back and DAFE in some cases. To complement this gap, we next augment IBT with supervised data either directly from the source domain or from the back-translated data using a NMT model trained on the source domain.

Combining IBT with Source Domain or Back-Translated Data IBT is trained purely on the non-parallel target domain data and can be augmented by adding supervised data, which leads to the two variants: IBT+Src and IBT+Back. In row (7) of Table 4.2, for IBT+Src, we insert the supervised translation task using the source domain data, which can bring in around +1 to +4 BLEU improvements consistently compared with IBT (except for the MED target domain in the ro-en language pair). For IBT+Back, we replace the source domain data with the back-translated data provided by the baseline Back, and it achieves even better performance, as shown in row (8) of Table 4.2. The superior performance of both variants demonstrates the benefit from the supervised data. And by comparing IBT+Src and IBT+Back, we see that although the back-translated data is obtained by performing inference of a model trained on the source domain data, the back-translated data is still a better option for domain adaptation than the latter one. Overall, our best setting can harvest up to +19.31 BLEU improvement over the best baseline model and +47.69 BLEU improvement over the Unadapted model. Notably, we have also tried combining the source domain parallel data with the back-translated data for supervised translation training but found that it performs worse than current settings.

4.5.2 Ablation Study

To check the importance of the each component in our model, we conduct an ablation study on the domain adaptation performance of the best performing model, IBT+Back. We report the validation set BLEU scores by adapting from the LAW domain to two target domains, MED and IT, in Table 4.3.

Model	Target Domain BLEU	
	MED	IT
IBT+Back	**42.13**	**47.64**
– Pre-training	31.80 (↓10.33)	27.71 (↓19.93)
– $\mathcal{L}_{\text{bt}}$	33.75 (↓8.38)	25.37 (↓22.27)
– $\mathcal{L}_{\text{lm}}$	40.97 (↓1.16)	40.82 (↓6.82)
– Source-side $\mathcal{L}_{\text{lm}}$	40.04 (↓2.09)	42.66 (↓4.98)
– $\mathcal{L}_{\text{bt}}$ – Source-side $\mathcal{L}_{\text{lm}}$	37.29 (↓4.84)	35.06 (↓12.58)

Table 4.3: Ablation study on the domain adaptation performance of IBT+Back. The source domain is LAW, and target domains are MED and IT. "– Source-side $\mathcal{L}_{\text{lm}}$" means no language modeling on the source-side sentences.

The most important components of our model are *Pre-training* and $\mathcal{L}_{\text{bt}}$. If the model is not initialized with pre-trained parameters ("– Pre-training"), the model suffers from a substantial performance decrease by 10 to 20 BLEU scores. If we remove the iterative back translation objective ($\mathcal{L}_{\text{bt}}$), the performance also drops significantly, ↓8.38 on MED and ↓22.27 on IT. We also find that $\mathcal{L}_{\text{lm}}$ and source-side $\mathcal{L}_{\text{lm}}$ are also important to the BLEU scores but not as crucial as the previous two components. However, an interesting finding is that if back translation $\mathcal{L}_{\text{bt}}$ and the source-side language modeling $\mathcal{L}_{\text{lm}}$ are removed together ("– $\mathcal{L}_{\text{bt}}$ – Source-side $\mathcal{L}_{\text{lm}}$"), the domain adaptation performance is better than mere "– $\mathcal{L}_{\text{bt}}$". The reason is that if back translation is removed, the decoder just needs to learn the target language, and language modeling on the source-side will impose a negative effect.

4.6 Discussion

4.6.1 Is Pre-Training Always Helpful?

Through the experiments, we find that initializing the translation model with pre-trained parameters can not only benefit our method but also the baselines. In Table 4.4, we compare two settings: with and without pre-training, for three baselines: Unadapted, Copy, Back, where we adapt from the LAW domain to MED and IT domains. For all three baselines, pre-training consistently brings in substantial improvements, although the pre-training process is performed via unsupervised language modeling training on Wikipedia text that is irrelevant to the target domain data we used here. This shows that proper initialization of models is crucial to the domain adaptation problem to circumvent the lack of supervised data in the target domain.

Target	**Unadapted**		**Copy**		**Back**	
	w/	**w/o**	**w/**	**w/o**	**w/**	**w/o**
MED	**17.44**	9.69	**24.38**	17.42	**36.61**	26.75
IT	**5.35**	3.87	**8.73**	5.07	**28.32**	9.21

Table 4.4: The comparison of three baselines: unadapted, copy, and back, between cases with and without pre-training when adapting them from the domain of LAW to MED and IT. Validation set BLEU scores are reported.

4.6.2 Do More Non-parallel Data Help?

One advantage of our method is that it keeps gaining improvement if the non-parallel data get larger. To verify this statement, we collected additional non-parallel data, combined them with the original target domain data,[4] and analyzed the performance difference before and after adding these extra data, as shown in Table 4.5. Specifically, we studied the adaptation from the WMT data to TED for both DE-EN and RO-EN

[4] When combining the extra non-parallel data with the original target domain data, we always up-sample the original data via replication so that it can have the same size as the additional data.

language pairs. We consider two sources of extra non-parallel data: source domain and target domain. The source domain data source can be the WMT data itself, whereas for the target domain, we collected an additional dataset of TED talks. After scraping all the TED talk web-pages[5] until early January 2020, we extracted the transcripts in three languages, English, German and Romanian, and kept the unique TED talk identifier of the transcript. Note that for any language pair, we made sure that the transcript of a TED talk only appeared once in either the source or the target side to avoid any parallel sentences.

From Table 4.5, we see that in general adding extra non-parallel data to our method can always lead to better performance, and choosing those extra data that have the same data distribution as the target domain is optimal. However, if we could not get more non-parallel data from the target domain, those in the source domain that are more readily to be obtained may also potentially improve the adaptation performance. By adding extra data, our best setting achieved even higher BLEU scores that are very close to supervised translation performance, as shown in Table 4.5. This set of experiments have shown the great potential of our method: we can always seek to collect more non-parallel data to keep improving the adaptation performance.

Model	**+ Data**	**WMT14→ TED (DE-EN)**	**WMT16→ TED (RO-EN)**
IBT	–	30.88	34.48
IBT	WMT	32.45	37.03
IBT	TED	**33.34**	**39.01**
IBT+Back	–	34.15	38.90
IBT+Back	WMT	34.74	38.79
IBT+Back	TED	**36.45**	**40.92**
MT (Sup.)	–	38.97	42.22

Table 4.5: Test set BLEU scores after adding extra non-parallel data ("+Data") from the source WMT domain ("WMT") or from the target TED domain ("TED").

[5] https://www.ted.com/

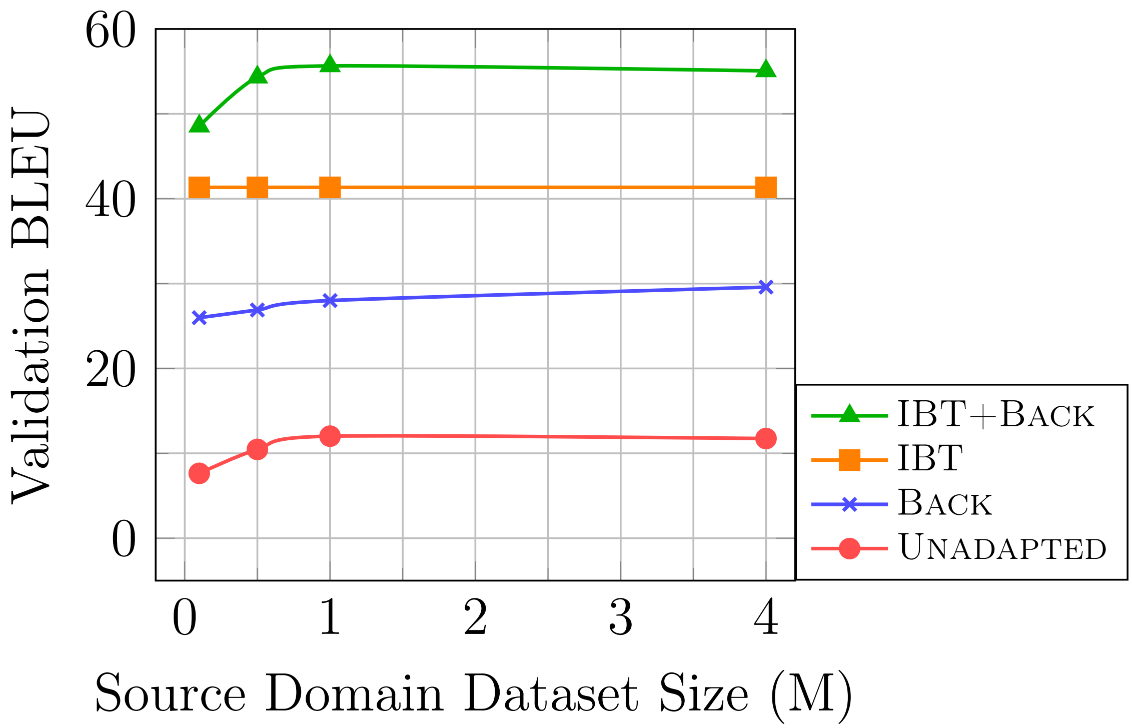


Figure 4-4: Effects of source domain dataset size on four adaptation methods: Unadapted, Back, IBT, and IBT+Back. We adapt from the general domain (WMT14) to the IT domain for the German-English language pair. All the target domain monolingual data, and sub-sampled 0.1, 0.5, 1, and 4 million source domain parallel pairs are used, respectively.

4.6.3 How do Different Sizes of Source Domain Data Influence the Performance?

In this section, we want to examine the effects of source domain dataset size on various adaptation methods. To this end, we sub-sample the source domain parallel data at different sampling ratios, and report the validation set BLEU scores on four adaptation methods: Unadapted, Back, IBT, and IBT+Back, as shown in Figure 4-4. Specifically, we adapt from the general domain (WMT14) to the IT domain for the German-English language pair. We use all the target domain non-parallel data, and sub-sample 0.1, 0.5, 1, and 4 million source domain parallel data. In Figure 4-4, IBT does not use any source domain data so it stays unchanged, while all the other settings demonstrate improved performance with the increasing number of source domain data, and the performance gradually saturates when the number of source domain data exceeds 1 million. Notably, IBT+Back consistently outperforms all others by a large margin, and its performance also increases at a higher rate, indicating that our method makes better use of the source domain supervised data.

4.7 Conclusion

In this chapter, we empirically identify that the iterative back-translation training scheme can yield large improvements to semi-supervised domain adaptation for NMT. On three low-resource domains, this basic approach demonstrates improvements of up to +9.48 BLEU scores over the strongest of four previous models, and up to +27.77 BLEU over the unadapted baseline. By further combining with popular data augmentation methods and utilizing supervised data from the source domain, our model shows a substantial improvement of up to +19.31 BLEU higher than the strongest baseline model, and up to +47.69 over the unadapted model. We put forward this method as a simple but strong baseline for semi-supervised domain adaptation for MT and future works in this direction should be compared with it.

Chapter 5

Robustness of Natural Language Processing Models

In the previous chapters, we have been extensively investigating various approaches to improving model performance on low-resource datasets/tasks through transfer learning, and have shown that these methods are incredibly effective and can even push some models to human-level performance. Albeit achieving such high numbers of performance metrics on the official released test sets, we still cannot guarantee that these models can generalize well to unseen data and are robust to natural noise that may exist in real-word applications. It is possible that the models are just overfitted to the given static test set and would catastrophically degrade when given new data even with exactly the same data distribution. Such robustness issues are indeed very common in deep learning models due to their sheer number of parameters and strong expressive power. Negative examples revealing to researchers that deep learning models are indeed not robust but surprisingly susceptible to human-imperceptible perturbations were first found in computer vision [Szegedy et al., 2013] and later in NLP [Jia and Liang, 2017]. Later on, this field has been attracting more and more attention from the community with one side proposing new methods to test the robustness of models and the other side introducing effective ways of improving robustness.

This chapter will introduce our two works on probing the robustness of NLP models: one is to propose an adversarial attack method to reveal the weakness of NLP models on text classification and NLI tasks (in Section 5.1 based on this work: Jin et al. [2020b]); and the other is to propose a new challenge set to test the robustness of models for the aspect-level sentiment analysis task (in Section 5.2 based on this work: Xiang et al. [2020]).[1] The heart of our proposed methods lies in crafting adversarial examples based on the original data samples so that the newly created data would be perceived to be semantically the same as the original one by people but can fool models to make wrong predictions. They are described in detail as follows.

5.1 Is BERT Really Robust?

5.1.1 Introduction

In the last decade, Machine Learning (ML) models have achieved remarkable success in various tasks such as classification, regression and decision making. However, recently they have been found vulnerable to adversarial examples that are legitimate inputs altered by small and often imperceptible perturbations [Kurakin et al., 2016a,b; Papernot et al., 2017; Zhao et al., 2017]. These carefully curated examples are correctly classified by a human observer but can fool a target model, raising serious concerns regarding the security and integrity of existing ML algorithms. On the other hand, it is shown that robustness and generalization of ML models can be improved by crafting high-quality adversaries and including them in the training data [Goodfellow et al., 2015].

While existing works on adversarial examples have obtained success in the image and speech domains [Carlini and Wagner, 2018; Szegedy et al., 2013], it is still challenging to deal with text data due to its discrete nature. Formally, besides the ability to fool the target models, outputs of a natural language attacking system should also meet three key utility-preserving properties: (1) human prediction consistency—prediction by humans should remain unchanged, (2) semantic similarity—the crafted

[1] Zhijing and I together proposed the idea of this work and I provided guidance for all experiments and helped with paper writing.

example should bear the same meaning as the source, as judged by people, and (3) language fluency—generated examples should look natural and grammatical. Previous works barely conform to all three requirements. For example, methods such as word misspelling [Gao et al., 2018; Li et al., 2018a], single-word erasure [Li et al., 2016], and phrase insertion and removal [Liang et al., 2017] result in unnatural sentences. Moreover, there is almost no work that attacks the newly-risen BERT model on text classification.

In this work, we present TextFooler, a simple but strong baseline for natural language attack in the black-box setting, a common case where no model architecture or parameters are accessible. We design a more comprehensive paradigm to create both semantically and syntactically similar adversarial examples that meet the aforementioned three desiderata. Specifically, we first identify the important words for the target model and then prioritize to replace them with the most semantically similar and grammatically correct words until the prediction is altered. We successfully applied this framework to attack three state-of-the-art models in five text classification tasks and two textual entailment tasks, respectively. On the adversarial examples, we can reduce the accuracy of almost all target models in all tasks to below 10% with only less than 20% of the original words perturbed. In addition, we validate that the generated examples are (1) correctly classified by human evaluators, (2) semantically similar to the original text, and (3) grammatically acceptable by human judges.

Our main contributions are summarized as follows:

- We propose a simple but strong baseline, TextFooler, to quickly generate high-profile utility-preserving adversarial examples that force the target models to make wrong predictions under the black-box setting.

- We evaluate TextFooler on three state-of-the-art deep learning models over five popular text classification tasks and two textual entailment tasks, and it achieves the state-of-the-art attack success rate and perturbation rate.

- We propose a comprehensive four-way automatic and three-way human evaluation of language adversarial attacks to evaluate the effectiveness, efficiency, and utility-preserving properties of our system.

- We open-source the code, pre-trained target models, and test samples for the convenience of future benchmarking at: https://github.com/jind11/TextFooler.

5.1.2 Method

5.1.2.1 Problem Formulation

Given a corpus of N sentences $\mathcal{X} = \{X_1, X_2, \ldots, X_N\}$, and a corresponding set of N labels $\mathcal{Y} = \{Y_1, Y_2, \ldots, Y_N\}$, we have a pre-trained model $F : \mathcal{X} \to \mathcal{Y}$, which maps the input text space $\mathcal{X}$ to the label space $\mathcal{Y}$.

For a sentence $X \in \mathcal{X}$, a valid adversarial example X_{adv} should conform to the following requirements:

$$F(X_{\text{adv}}) \neq F(X), \text{and } \text{Sim}(X_{\text{adv}}, X) \geq \epsilon, \tag{5.1}$$

where $\text{Sim} : \mathcal{X} \times \mathcal{X} \to (0, 1)$ is a similarity function and ϵ is the minimum similarity between the original and adversarial examples. In the natural language domain, $\text{Sim}(\cdot)$ is often a semantic and syntactic similarity function.

5.1.2.2 Threat Model

Under the black-box setting, the attacker is *not* aware of the model architecture, parameters, or training data. It can only query the target model with supplied inputs, getting as results the predictions and corresponding confidence scores.

The proposed approach for adversarial text generation is shown in Algorithm 2, and consists of the two main steps:

Step 1: Word Importance Ranking (line 1-6) Given a sentence of n words $X = \{w_1, w_2, \ldots, w_n\}$, we observe that only some key words act as influential signals for the prediction model F, echoing the discovery of [Niven and Kao, 2019] that BERT attends to the statistical cues of some words. Therefore, we create a selection mechanism to choose the words that most significantly influence the final prediction results. Using this selection process, we minimize the alterations, and thus maintain the semantic similarity as much as possible.

Note that the selection of important words is trivial in a white-box scenario, as it can be easily solved by inspecting the gradients of the model F, while most other words are irrelevant. However, under the more common black-box set up in our work, the model gradients are unavailable. Therefore, we create a selection mechanism as follows. We use the score I_{w_i} to measure the influence of a word $w_i \in X$ towards the classification result $F(X) = Y$. We denote the sentence after deleting the word w_i as $X_{\backslash w_i} = X \setminus \{w_i\} = \{w_1, \ldots, w_{i-1}, w_{i+1}, \ldots w_n\}$, and use $F_Y(\cdot)$ to represent the prediction score for the Y label.

The importance score I_{w_i} is therefore calculated as the prediction change before and after deleting the word w_i, which is formally defined as follows,

$$I_{w_i} = \begin{cases} F_Y(X) - F_Y(X_{\backslash w_i}), \qquad \text{if } F(X) = F(X_{\backslash w_i}) = Y \\ (F_Y(X) - F_Y(X_{\backslash w_i})) + (F_{\bar{Y}}(X_{\backslash w_i}) - F_{\bar{Y}}(X)), \\ \qquad \text{if } F(X) = Y, F(X_{\backslash w_i}) = \bar{Y}, \text{ and } Y \neq \bar{Y}. \end{cases} \tag{5.2}$$

After ranking the words by their importance score, we further filter out stop words derived from NLTK[2] and spaCy[3] libraries such as "the", "when", and "none". This simple step of filtering is important to avoid grammar destruction.

Step 2: Word Transformer (lines 7-30) For a given word $w_i \in X$ with a high importance score obtained in Step 1, we need to design a word replacement mechanism. A suitable replacement word needs to fulfill the following criteria: it should (1) have similar meaning to the original one, (2) fit within the surrounding context, and (3) force the target model to make wrong predictions. In order to select replacement words that meet such criteria, we propose the following workflow.

Synonym Extraction: We gather a candidate set CANDIDATES for all possible replacements of the selected word w_i. CANDIDATES is initiated with N closest

[2] https://www.nltk.org/

[3] https://spacy.io/

Algorithm 2 Adversarial Attack by TextFooler

```
Require: Sentence example $X = \{w_1, w_2, ..., w_n\}$, the corresponding ground truth label $Y$, target model $F$, sentence similarity function $\mathrm{Sim}(\cdot)$, sentence similarity threshold $\epsilon$, word embeddings Emb over the vocabulary Vocab.
Ensure: Adversarial example $X_{\text{adv}}$
Initialization: $X_{\text{adv}} \leftarrow X$
for each word $w_i$ in $X$ do
    Compute the importance score $I_{w_i}$ via Eq. (5.2)
end for

Create a set $W$ of all words $w_i \in X$ sorted by the descending order of their importance score $I_{w_i}$.
Filter out the stop words in $W$.
for each word $w_j$ in $W$ do
    Initiate the set of candidates Candidates by extracting the top $N$ synonyms using CosSim(Emb$_{w_j}$, Emb$_{\text{word}}$) for each word in Vocab.
    Candidates $\leftarrow$ POSFilter(Candidates)
    FinCandidates $\leftarrow \{\ \}$
    for $c_k$ in Candidates do
        $X' \leftarrow$ Replace $w_j$ with $c_k$ in $X_{\text{adv}}$
        if $\mathrm{Sim}(X', X_{\text{adv}}) > \epsilon$ then
            Add $c_k$ to the set FinCandidates
            $Y_k \leftarrow F(X')$
            $P_k \leftarrow F_{Y_k}(X')$
        end if
    end for
    if there exists $c_k$ whose prediction result $Y_k \neq Y$ then
        In FinCandidates, only keep the candidates $c_k$ whose prediction result $Y_k \neq Y$
        $c^* \leftarrow \underset{c \in \text{FinCandidates}}{\operatorname{argmax}} \mathrm{Sim}(X, X'_{w_j \to c})$
        $X_{\text{adv}} \leftarrow$ Replace $w_j$ with $c^*$ in $X_{\text{adv}}$
        return $X_{\text{adv}}$
    else if $P_{Y_k}(X_{\text{adv}}) > \min_{c_k \in \text{FinCandidates}} P_k$ then
        $c^* \leftarrow \underset{c_k \in \text{FinCandidates}}{\operatorname{argmin}} P_k$
        $X_{\text{adv}} \leftarrow$ Replace $w_j$ with $c^*$ in $X_{\text{adv}}$
    end if
end for
return None
```

synonyms according to the cosine similarity between the embeddings of w_i and every other word in the vocabulary.

To represent the words, we use word embeddings from [Mrkšić et al., 2016a]. These word vectors are specially curated for finding synonyms, as they achieve the state-of-the-art performance on SimLex-999, a dataset designed to measure how well different models judge semantic similarity between words [Hill et al., 2015].

Using this set of embedding vectors, we identify the top N synonyms whose cosine similarities with w are greater than δ. Note that enlarging N or lowering δ would both generate more diverse synonym candidates; however, the semantic similarity between the adversary and the original sentence would decrease. In our experiments, empirically setting N to be 50 and δ to be 0.7 strikes a balance between diversity and semantic similarity.

POS Checking: In the set $\textsc{Candidates}$ of the word w_i, we only keep the ones with the same part-of-speech (POS)[4] as w_i. This step is to assure that the grammar of the text is mostly maintained (line 10 in Algorithm 2).

Semantic Similarity Checking: For each remaining word $c \in \textsc{Candidates}$, we substitute it for w_i in the sentence X, and obtain the adversarial example $X_{\text{adv}} = \{w_1, \dots, w_{i-1}, c, w_{i+1}, \dots, w_n\}$. We use the target model F to compute the corresponding prediction scores $F(X_{\text{adv}})$. We also calculate the sentence semantic similarity between the source X and adversarial counterpart X_{adv}. Specifically, we use the Universal Sentence Encoder (USE) [Cer et al., 2018] to encode the two sentences into high dimensional vectors and use their cosine similarity score as an approximation of semantic similarity. The words resulting in similarity scores above a preset threshold ϵ are placed into the final candidate pool $\textsc{FinCandidates}$ (lines 11-19 in Algorithm 2).

Finalization of Adversarial Examples: If there exists any candidate in the final candidate pool $\textsc{FinCandidates}$ that can already alter the prediction of the target model, then we select the word with the highest semantic similarity score among these winning candidates. But if not, then we select the word with the least

[4] We used the off-the-shelf spaCy tagger, available at https://spacy.io/api/tagger

confidence score of label y as the best replacement word for w_i, and repeat Step 2 to transform the next selected word (lines 20-30 in Algorithm 2).

Overall, the algorithm first uses Step 1 to rank the words by their importance scores, and then repeats Step 2 to find replacements for words in the sentence X until the prediction of the target model is altered.

5.1.3 Experiments

5.1.3.1 Tasks

We study the effectiveness of our adversarial attack on two important NLP tasks, text classification and textual entailment. The dataset statistics are summarized in Table 5.1. Following the practice by Alzantot et al. [2018], we evaluate our algorithm on a set of 1,000 examples randomly selected from the test set.

Task	**Dataset**	**Train**	**Test**	**Avg Len**
Classification	AG's News	30K	1.9K	43
	Fake News	18.8K	2K	885
	MR	9K	1K	20
	IMDB	25K	25K	215
	Yelp	560K	38K	152
Entailment	SNLI	570K	3K	8
	MultiNLI	433K	10K	11

Table 5.1: Overview of the datasets.

Text Classification To study the robustness of our model, we use text classification datasets with various properties, including news topic classification, fake news detection, and sentence- and document-level sentiment analysis, with average text length ranging from tens to hundreds of words.

- **AG's News (AG)**: Sentence-level classification with regard to four news topics: World, Sports, Business, and Science/Technology. Following the practice of

Zhang et al. [2015], we concatenate the title and description fields for each news article.

- **Fake News Detection (Fake)**: Document-level classification on whether a news article is fake or not. The dataset comes from the Kaggle Fake News Challenge.[5]

- **MR**: Sentence-level sentiment classification on positive and negative movie reviews [Pang and Lee, 2005]. We use 90% of the data as the training set and 10% as the test set, following the practice in [Li et al., 2018a].

- **IMDB**: Document-level sentiment classification on positive and negative movie reviews.[6]

- **Yelp Polarity (Yelp)**: Document-level sentiment classification on positive and negative reviews [Zhang et al., 2015]. Reviews with a rating of 1 and 2 are labeled negative and 4 and 5 positive with the rating of 3 omitted.

Textual Entailment

- **SNLI**: A dataset of 570K sentence pairs derived from image captions. The task is to judge the relationship between two sentences: whether the second sentence can be derived from entailment, contradiction, or neutral relationship with the first sentence [Bowman et al., 2015].

- **MultiNLI**: A multi-genre entailment dataset with a coverage of transcribed speech, popular fiction, and government reports [Williams et al., 2017]. Compared to SNLI, it contains more linguistic complexity with various written and spoken English texts.

5.1.3.2 Attacking Target Models

For each dataset, we train three state-of-the-art models on the training set, and achieved test set accuracy scores similar to the original implementation, as shown in

[5] https://www.kaggle.com/c/fake-news/data

[6] https://datasets.imdbws.com/

Table 5.2. We then generate adversarial examples which are semantically similar to the test set to attack the trained models and make them generate different results.

	WordCNN	**WordLSTM**	**BERT**
AG	92.5	93.1	94.6
Fake	99.9	99.9	99.9
MR	79.9	82.2	85.8
IMDB	89.7	91.2	92.2
Yelp	95.2	96.6	96.1
	InferSent	**ESIM**	**BERT**
SNLI	84.6	88.0	90.7
MultiNLI	71.1/71.5	76.9/76.5	83.9/84.1

Table 5.2: Original accuracy of target models on standard test sets.

On the sentence classification task, we target three models: word-based convolutional neural network (WordCNN) [Kim, 2014], word-based long-short term memory (WordLSTM) [Hochreiter and Schmidhuber, 1997], and the state-of-the-art Bidirectional Encoder Representations from Transformers (BERT) [Devlin et al., 2019].

For the WordCNN model, we used three window sizes of 3, 4, and 5, and 100 filters for each window size with dropout of 0.3. For the WordLSTM, we used a 1-layer bidirectional LSTM with 150 hidden units and a dropout of 0.3.[7] For both models, we used the 200 dimensional Glove word embeddings pre-trained on 6B tokens from Wikipedia and Gigawords [Pennington et al., 2014]. We used the 12-layer BERT model with 768 hidden units and 12 heads, with 110M parameters, which is called the base-uncased version.[8]

We also implemented three target models on the textual entailment task: standard InferSent[9] [Conneau et al., 2017], ESIM[10] [Chen et al., 2016], and fine-tuned BERT.

[7] All these hyper-parameters are tuned on the development set of the MR dataset and they work generally well for all datasets we used.

[8] https://github.com/huggingface/pytorch-pretrained-BERT

[9] https://github.com/facebookresearch/InferSent

[10] https://github.com/coetaur0/ESIM

5.1.3.3 Setup of Automatic Evaluation

We first report the accuracy of the target models on the original test samples before attack as the original accuracy. Then we measure the accuracy of the target models against the adversarial samples crafted from the test samples, denoted as *after-attack accuracy*. By comparing these two accuracy scores, we can evaluate how successful the attack is — a larger gap between the original and after-attack accuracy signals the more successful our attack is. Apart from these accuracies, we also report the perturbed word percentage as the ratio of the number of perturbed words to the text length. Furthermore, we apply USE[11] to measure the semantic similarity between the original and adversarial texts. These two metrics, the perturbed words percentage and the semantic similarity score, together evaluate how semantically similar the original and adversarial texts are. We finally report the number of queries the attack system made to the target model to fetch the output probability scores. This metric can reveal the efficiency of the attack model.

5.1.3.4 Setup of Human Evaluation

We conduct a human evaluation on three criteria: semantic similarity, grammaticality, and classification accuracy. We randomly select 100 test sentences of each task to generate adversarial examples, one targeting WordLSTM on the MR dataset and another targeting BERT on SNLI. We first shuffle a mix of original and adversarial texts and asked human judges to rate their grammaticality on a Likert scale of $1-5$, similar to the practice of [Gagnon-Marchand et al., 2018]. Next, we evaluate the classification consistency by asking human judges to classify each example in the shuffled mix of the original and adversarial sentences and then calculate the consistency rate of both classification results. Lastly, we evaluate the semantic similarity of the original and adversarial sentences by asking humans to judge whether the generated adversarial sentence is similar, ambiguous, or dissimilar to the source sentence. Each task is completed by two independent human judges who are native English speakers.

[11] https://tfhub.dev/google/ universal-sentence-encoder

	WordCNN					WordLSTM					BERT				
	MR	**IMDB**	**Yelp**	**AG**	**Fake**	**MR**	**IMDB**	**Yelp**	**AG**	**Fake**	**MR**	**IMDB**	**Yelp**	**AG**	**Fake**
Original Accuracy	78.0	89.2	93.8	91.5	96.7	80.7	89.8	96.0	91.3	94.0	86.0	90.9	97.0	94.2	97.8
After-Attack Accuracy	2.8	0.0	1.1	1.5	15.9	3.1	0.3	2.1	3.8	16.4	11.5	13.6	6.6	12.5	19.3
% Perturbed Words	14.3	3.5	8.3	15.2	11.0	14.9	5.1	10.6	18.6	10.1	16.7	6.1	13.9	22.0	11.7
Semantic Similarity	0.68	0.89	0.82	0.76	0.82	0.67	0.87	0.79	0.63	0.80	0.65	0.86	0.74	0.57	0.76
Query Number	123	524	487	228	3367	126	666	629	273	3343	166	1134	827	357	4403
Average Text Length	20	215	152	43	885	20	215	152	43	885	20	215	152	43	885

Table 5.3: Automatic evaluation results of the attack system on text classification datasets, including the original model prediction accuracy before being attacked ("Original Accuracy"), the model accuracy after the adversarial attack ("After-Attack Accuracy"), the percentage of perturbed words with respect to the original sentence length ("% Perturbed Words"), and the semantic similarity between original and adversarial samples ("Semantic Similarity").

	InferSent		ESIM		BERT	
	SNLI	**MultiNLI (m/mm)**	**SNLI**	**MultiNLI (m/mm)**	**SNLI**	**MultiNLI (m/mm)**
Original Accuracy	84.3	70.9/69.6	86.5	77.6/75.8	89.4	85.1/82.1
After-Attack Accuracy	3.5	6.7/6.9	5.1	7.7/7.3	4.0	9.6/8.3
% Perturbed Words	18.0	13.8/14.6	18.1	14.5/14.6	18.5	15.2/14.6
Semantic Similarity	0.50	0.61/0.59	0.47	0.59/0.59	0.45	0.57/0.58
Query Number	57	70/83	58	72/87	60	78/86
Average Text Length	8	11/12	8	11/12	8	11/12

Table 5.4: Automatic evaluation results of the attack system on textual entailment datasets. "m" means matched, and "mm" means mismatched, which are the two variants of the MultiNLI development set.

The volunteers have university-level education backgrounds and passed a test batch before they started annotation.

5.1.4 Results

5.1.4.1 Automatic Evaluation

The main results of black-box attacks in terms of automatic evaluation on five text classification and two textual entailment tasks are summarized in Table 5.3 and 5.4, respectively. Overall, as can be seen from our results, TextFooler achieves a high success rate when attacking with a limited number of modifications on both tasks. No matter how long the text sequence is, and no matter how accurate the target model is, TextFooler can generally reduce the accuracy from the state-of-the-art values to below 15% (except on the Fake dataset) with less than 20% word perturbation ratio (except the AG dataset under the BERT target model). For instance, it only perturbs 5.1% of the words on average when reducing the accuracy from 89.8% to only 0.3% on the IMDB dataset against the WordLSTM model. Notably, our attack system makes the WordCNN model on the IMDB dataset totally wrong (reaching an accuracy of 0%) with only 3.5% word perturbation rate. In the IMDB dataset, which has an average length of 215 words, the system only perturbed 10 words or fewer per sample to conduct successful attacks. This means that our attack system can successfully mislead the classifiers into assigning wrong predictions via subtle manipulation.

Even for BERT, which has achieved seemingly "robust" performance compared with the non-pretrained models such as WordLSTM and WordCNN, our attack model can still reduce its prediction accuracy by about 5–7 times on the classification task (e.g., from 95.6% to 6.8% for the Yelp dataset) and about 9-22 times on the NLI task (e.g., from 89.4% to 4.0% for the SNLI dataset), which is unprecedented. Our curated adversarial examples can contribute to the study of the interpretability of the BERT model [Feng et al., 2018a].

Another two observations can be drawn from Tables 5.3 and 5.4. (1) Models with higher original accuracy are, in general, more difficult to attack. For instance, the after-attack accuracy and perturbed word ratio are both higher for the BERT model compared with WordCNN on all datasets. (2) The after-attack accuracy of the Fake dataset is much higher than all other classification datasets for all three target models. We found in experiments that it is easy for the attack system to convert a real news article to a fake one, whereas the reverse process is much harder, which is in line with intuition.

Comparing the semantic similarity scores and the perturbed word ratios in both Tables 5.3 and 5.4, we find that the two results have a high positive correlation. Empirically, when the text length is longer than 10 words, the semantic similarity measurement becomes more stable. Since the average text lengths of text classification datasets are all above 20 words and those of textual entailment datasets are around or below 10 words, we need to treat the semantic similarity scores of these two tasks individually. Therefore, we performed a linear regression analysis between the word perturbation ratio and semantic similarity for each task and obtained r-squared values of 0.94 and 0.97 for text classification and textual entailment tasks, respectively. Such high values of r-squared reveal that our proposed semantic similarity has a high correlation (negative) with the perturbed words ratio, which can both be good automatic measurements to evaluate the degree of alteration of the original text.

We include the average text length of each dataset in the last row of Tables 5.3 and 5.4 so that it can be conveniently compared against the number of queries. That number is almost linear in the text length, with a ratio in $(2, 8)$. Longer text correlates with a smaller ratio, which validates the efficiency of TextFooler.

Benchmark Comparison We compared TextFooler with the previous state-of-the-art adversarial attack systems against the same target model and dataset. Our baselines include Li et al. [2018a] that generates misspelled words by character- and word-level perturbation, Alzantot et al. [2018] that iterates through every word in the sentence and find its perturbation, and Kuleshov et al. [2018] that uses word replacement by greedy heuristics. From the results in Table 5.5, we can see that our system beats the previous state-of-the-art models in both the attack success rate (calculated by dividing the number of wrong predictions by the total number of adversarial examples) and perturbed word ratio.

5.1.4.2 Human Evaluation

We sampled 100 adversarial examples on the MR dataset with the WordLSTM and 100 examples on SNLI with BERT. We verified the quality of our examples via three experiments. First, we ask human judges to give a grammaticality score of a shuffled mix of original and adversarial text.

As shown in Table 5.7, the grammaticality of the adversarial texts are close to that of the original texts on both datasets. By sensibly substituting synonyms, TextFooler generates smooth outputs such as “the big metaphorical wave” in Ta-

Dataset	Model	Success Rate	% Perturbed Words
IMDB	[Li et al., 2018a]	86.7	6.9
	[Alzantot et al., 2018]	97.0	14.7
	Ours	**99.7**	**5.1**
SNLI	[Alzantot et al., 2018]	70.0	23.0
	Ours	**95.8**	**18.0**
Yelp	[Kuleshov et al., 2018]	74.8	-
	Ours	**97.8**	**10.6**

Table 5.5: Comparison of our attack system against other published systems. The target model for IMDB and Yelp is LSTM and SNLI is InferSent.

Movie Review (Positive (POS) ↔ Negative (NEG))	
Orig. (Label: NEG)	The characters, cast in impossibly ***contrived situations***, are ***totally*** estranged from reality.
Adv. (Label: POS)	The characters, cast in impossibly ***engineered circumstances***, are ***fully*** estranged from reality.
Orig. (Label: POS)	It cuts to the ***knot*** of what it actually means to face your ***scares***, and to ride the ***overwhelming*** **metaphorical wave** that life wherever it takes you.
Adv. (Label: NEG)	It cuts to the ***core*** of what it actually means to face your ***fears***, and to ride the ***big*** **metaphorical wave** that life wherever it takes you.
SNLI (Entailment (ENT), Neutral (NEU), Contradiction (CON))	
Premise	Two small boys in blue soccer uniforms use a wooden set of steps to wash their hands.
Orig. (Label: CON)	The boys are in band ***uniforms***.
Adv. (Label: ENT)	The boys are in band ***garment***.
Premise	A child with wet hair is holding a butterfly decorated beach ball.
Orig. (Label: NEU)	The ***child*** is at the ***beach***.
Adv. (Label: ENT)	The ***youngster*** is at the ***shore***.

Table 5.6: Examples of original and adversarial sentences from MR (WordLSTM) and SNLI (BERT) datasets. Orig.: original; Adv.: adversary.

Source Text	MR (WordLSTM)	SNLI (BERT)
Original	4.22	4.50
Adversarial	4.01	4.27

Table 5.7: Grammaticality of original and adversarial examples for MR (WordLSTM) and SNLI (BERT) on $1-5$ scale.

ble 5.6.

We then asked the human raters to assign classification labels to a shuffled set of original and adversarial samples. The overall agreement between the labels of the original sentence and the adversarial sentence is relatively high, with 92% on MR and 85% on SNLI. Though our adversarial examples are not perfect in every case, this shows that majorities of adversarial sentences have the same attribute as the original sentences from a human perspective. Table 5.6 shows typical examples of sentences with almost the same meanings that result in contradictory classifications by the target model.

Lastly, we asked the judges to decide whether each adversarial sample retains the meaning of the original sentence. They need to decide whether the synthesized adversarial example is similar, ambiguous, or dissimilar to the provided original sentence. We regard similar as 1, ambiguous as 0.5, and dissimilar as 0, and obtained sentence similarity scores of 0.91 on MR and 0.86 on SNLI, which shows the perceived difference between original and adversarial text is small.

5.1.5 Discussion

5.1.5.1 Ablation Study

Word Importance Ranking To validate the effectiveness of Step 1 in Algorithm 2, i.e., the word importance ranking, we remove this step and instead randomly select the words in text to perturb. We keep the perturbed word ratio and Step 2 the same. We use BERT as the target model and test on three datasets: MR, AG, and SNLI. The results are summarized in Table 5.8. After removing Step 1 and instead randomly selecting the words to perturb, the after-attack accuracy increases by more than 45% on all three datasets, which reveals that the attack becomes ineffective without the word importance ranking step. The word importance ranking process is crucial to the algorithm in that it can accurately and efficiently locate the words which cast the most significant effect on the predictions of the target model. This strategy can also reduce the number of perturbed words so as to maintain the semantic similarity as much as possible.

	MR	**AG**	**SNLI**
% Perturbed Words	16.7	22.0	18.5
Original Accuracy	86.0	94.2	89.4
After-Attack Accuracy	11.5	12.5	4.0
After-Attack Accuracy (Random)	**68.3**	**80.8**	**59.2**

Table 5.8: Comparison of the after-attack accuracies before and after removing the word importance ranking of Algorithm 2. For control, Step 2 and the perturbed words ratio are kept the same. BERT model is used as the target model.

Semantic Similarity Constraint In Step 2 of Algorithm 2, for every possible word replacement, we check the semantic similarity between the newly generated sample and the original text, and adopt this replacement only when the similarity is above a preset threshold ϵ. We found that this strategy can effectively filter out irrelevant synonyms to the selected word. As we can see from the examples in Table 5.9, the synonyms extracted by word embeddings are noisy, so directly injecting them into the text as adversarial samples would probably shift the semantics significantly. By applying the sentence-level semantic similarity constraint, we can obtain more related synonyms as good replacements. To be noted, applying the semantic similarity constraint will also increase the difficulty in finding successful attacks, as indicated by the increased after-attack accuracy and percentage of perturbed words in Table 5.10.

Original	like a south of the border melrose ***place***
Adversarial	like a south of the border melrose ***spot***
- Sim.	like a south of the border melrose ***mise***
Original	their computer animated faces are very ***expressive***
Adversarial	their computer animated face are very ***affective***
- Sim.	their computer animated faces are very ***diction***

Table 5.9: Qualitative comparison of adversarial attacks with and without the semantic similarity constraint ("-Sim."). We highlight the original word, TextFooler's replacement, and the replacement without semantic constraint.

	MR	**IMDB**	**SNLI**	**MNLI(m)**
After-Attack Accuracy	11.5/6.2	13.6/11.2	4.0/3.6	9.6/7.9
% Perturbed Words	16.7/14.8	6.1/4.0	18.5/18.3	15.2/14.5
Query Number	166/131	1134/884	60/57	78/70
Semantic Similarity	0.65/0.58	0.86/0.82	0.45/0.44	0.57/0.56

Table 5.10: Comparison of automatic evaluation metrics with and without the semantic similarity constraint (numbers in the left and right of the symbol "/" represent results with and without the constraint, respectively). The target model is BERT-Base.

5.1.5.2 Transferability

We examined transferability of adversarial text, that is, whether adversarial samples curated based on one model can also fool another. For this, we collected the adversarial examples from IMDB and SNLI test sets that are wrongly predicted by one target model and then measured the prediction accuracy of them against the other two target models. As we can see from the results in the Table 5.11, there is a moderate degree of transferability between models, and the transferability is higher in the textual entailment task than in the text classification task. Moreover, the adversarial samples generated based on the model with higher prediction accuracy, i.e. the BERT model here, show higher transferability.

		WordCNN	**WordLSTM**	**BERT**
IMDB	**WordCNN**	—	84.9	90.2
	WordLSTM	74.9	—	87.9
	BERT	84.1	85.1	—
		InferSent	**ESIM**	**BERT**
SNLI	**InferSent**	—	62.7	67.7
	ESIM	49.4	—	59.3
	BERT	58.2	54.6	—

Table 5.11: Transferability of adversarial examples on IMDB and SNLI dataset. Row i and column j is the accuracy of adversaries generated for model i evaluated on model j.

5.1.5.3 Adversarial Training

Our work casts insights on how to better improve the original models through these adversarial examples. We conducted a preliminary experiment on adversarial training, by feeding the models both the original data and the adversarial examples (adversarial examples share the same labels as the original counterparts), to see whether the original models can gain more robustness. We collected the adversarial examples curated from the MR and SNLI training sets that fooled BERT and added them to the original training set. We then used the expanded data to fine-tune BERT and attacked this adversarially-trained model. As is seen in the attack results in Table 5.12, both the after-attack accuracy and perturbed words ratio after adversarial re-training get higher, indicating the greater difficulty to attack. This reveals one of the potential benefits of our attack system — we can enhance the robustness of a model to future attacks by training it with the generated adversarial examples.

5.1.5.4 Error Analysis

Our adversarial samples are susceptible to two types of errors: word sense ambiguity, and grammatical error. It is difficult for our model to disambiguate the word senses. For example, the sentence “One man *shows* the ransom money to the other” has the

	MR		SNLI	
	Af. Acc.	**Pert.**	**Af. Acc.**	**Pert.**
Original	11.5	16.7	4.0	18.5
+ Adv. Training	**18.7**	**21.0**	**8.3**	**20.1**

Table 5.12: Comparison of the after-attack accuracy ("Af. Acc.") and percentage of perturbed words ("Pert.") of original training ("Original") and adversarial training ("+ Adv. Train") of BERT model on MR and SNLI dataset.

adversary "One man *testifies* the ransom money to the other", where "testify" in this context is not the appropriate synonym of "show" though it fits in other cases. There also exist grammatical errors. For example, in the adversarial example "A man with headphones is *motorcycle*", "motorcycle" is similar to the original word "biking" but is not grammatical here. As future work, we will improve the heuristic design, and use a pre-trained language model over large-scale corpora such as GPT-2 [Radford et al., 2019].

Content shift can be seen in a task-specific situation. In the sentiment classification task, a change of words might not affect the overall sentiment, whereas in the task of textual entailment, the substitution of words might result in a fundamental difference. For example, if the premise is "a *kid* with a red hat is running", and the original hypothesis is "a *kid* is running (Entailment)", then if the adversarial example becomes "a *girl* is running", the sensible result turns into Neutral instead.

5.1.5.5 Limitations and Future Work

In this section, we will discuss some limitations in this work and cast light onto possible future work to resolve these limitations:

- Although we can rely on human evaluation to verify the validity of generated adversaries as a compliment to automatic evaluation for three aspects of label preservation, semantic similarity, and fluency, human evaluation is still more or less subject to the subjectivity of annotators. And it is resource- and time-consuming, thus only a small number of samples can be evaluated. For benchmarking comparison, every new work needs to conduct human evaluation again and again for all previous works, which severely hinders the development of this field. Future work should put more efforts on proposing better automatic evaluation metrics that can highly align with human evaluation so that it can serve as a good replacement.

- Although we replace the tokens by their synonyms, we cannot guarantee that every replacement can strictly conform to grammar rules and every new token can well fit into the sentence to form a fluent one. To alleviate this issue, a strong and well-trained language model can be used to rank the candidate tokens so that the one that can best fit the sentence is selected. However, the use of such a language model will also add tremendous computation and thus make the attack process very slow. Some techniques to accelerate the inference process of the language model added should be developed.
- Our proposed adversarial attack method can be used to create a challenge set to any existing text classification and entailment dataset, however, further removal of unqualified and in-fluent sentences by human annotation is still needed. Future work is needed to validate this process and demonstrate that such a strategy can effectively and efficiently create a challenge set by showcasing on several important datasets such as NLI, IMDB, etc..

5.1.6 Related Work

Adversarial attack has been extensively studied in computer vision [Kurakin et al., 2016a; Moosavi-Dezfooli et al., 2017]. Most works make gradient-based perturbations on continuous input spaces [Goodfellow et al., 2015; Szegedy et al., 2013].

Adversarial attack on discrete data such as text is more challenging. Inspired by the approaches in computer vision, early work in language adversarial attack focuses on variations of gradient-based methods. For example, Zhao et al. [2017] transform input data into a latent representation by generative adversarial networks (GANs), and then retrieve adversaries close to the original instance in the latent space.

Other works observed the intractability of GAN-based models on text and the shift in semantics in the latent representations, so heuristic methods such as scrambling, misspelling, or removing words were proposed [Alzantot et al., 2018; Li et al., 2018a, 2016]. Ribeiro et al. [2018] automatically craft the semantically equivalent adversarial rules from the machine generated paraphrases using back-translation techniques.

5.1.7 Summary

Overall, we study adversarial attacks against state-of-the-art text classification and textual entailment models under the black-box setting. Extensive experiments demonstrate the effectiveness of our proposed system, TextFooler, at generating targeted adversarial texts. Human studies validated that the generated adversarial texts are legible, grammatical, and similar in meaning to the original texts.

5.2 Probing Aspect Robustness in Aspect-Based Sentiment Analysis

5.2.1 Introduction

Aspect-based sentiment analysis (ABSA) is an advanced sentiment analysis task that aims to classify the sentiment towards a specific aspect (e.g., *burgers* or *fries* in the review "Tasty *burgers*, and crispy *fries*."). The key to a strong ABSA model is that it should be sensitive to *only* the sentiment words of the target aspect, and therefore not be interfered with by the sentiment of any non-target aspect. Although state-of-the-art models have shown high accuracy on existing test sets, we still question their robustness. Specifically, given the *prerequisite* that a model outputs correct sentiment polarity for the test sentence, we have the following questions:

(Q1) If we reverse the sentiment polarity of the target aspect, can the model change its prediction accordingly?

(Q2) If the sentiments of all non-target aspects become opposite to the target one,

SubQ.	Generation Strategy	Example
Prereq.	Source: The original sample from the test set	Tasty **burgers**, and crispy fries. (Tgt: burgers)
Q1	RevTgt: Reverse the sentiment of the *target* aspect	Terrible **burgers**, but crispy fries.
Q2	RevNon: Reverse the sentiment of the *non-target* aspects with originally the same sentiment as target	Tasty **burgers**, but soggy fries.
Q3	AddDiff: Add aspects with the *opposite* sentiment from the target aspect	Tasty **burgers**, crispy fries, but poorest service ever!

Table 5.13: The generation strategies and examples of the prerequisite (Prereq) and three questions (Q1)-(Q3). Each example is annotated with the **target aspect (Tgt)**, and altered sentence parts.

can the model still make the correct prediction?

(Q3) If we add more non-target aspects with sentiments opposite to the target one, can the model still make the correct prediction?

A robust ABSA model should both meet the prerequisite and have affirmative answers to all the questions above. For example, if a model makes the correct sentiment classification (i.e., positive) for *burgers* in the original sentence "Tasty *burgers*, and crispy fries", it should flip its prediction (to negative) when seeing the new context "Terrible *burgers*, but crispy fries". Hence, these questions together form a probe to verify if an ABSA model has high ***aspect robustness***.

Unfortunately, existing ABSA datasets have very limited capability to probe *aspect robustness*. For example, the Twitter dataset [Dong et al., 2014] has only one aspect per sentence, so the model does not need to discriminate against non-target aspects. In the most widely used SemEval 2014 Laptop and Restaurant datasets [Pontiki et al., 2014], for 83.9% and 79.6% samples in the test sets, the sentiments of the target aspect and all non-target aspects are all the same. Hence, we cannot decide whether models that make correct classifications attend only to the target aspect, because they may also wrongly look at the non-target aspects, which are *confounding factors*. Only a small portion of the test set can be used to answer our target questions proposed in the beginning. Moreover, when we test on the subset of the test set (59 samples in Laptop, and 122 samples in Restaurant) where the target aspect sentiment differs from all non-target aspect sentiments (so that the confounding factor is disentangled), the best model [Xu et al., 2019a] drops from 78.53% to 59.32% on Laptop and from 86.70% to 63.93% on Restaurant. This implies that the success of previous models may over-rely on the confounding non-target aspects, but not necessarily on the target aspect only. However, *no* existing datasets can be used to analyze the aspect robustness in greater depth.

We develop an automatic generation framework that takes as input the original test sentences from SemEval 2014, and applies three generation strategies shown in Table 5.13. Samples generated by RevTgt, RevNon, and AddDiff can be used to answer the questions (Q1)-(Q3), respectively. The generated new sentences largely overlap with the content and aspect terms of the original sentence, but manage to disentangle the confounding sentiment polarity of non-target aspects from the target, as shown in the examples in Table 5.13. In this way, we produce an "all-rounded" test set that can test whether a model robustly captures the target sentiment instead of using other irrelevant clues.

We enriched the laptop dataset by 294% from 638 to 1,877 samples and the restaurant dataset by 315% from 1,120 to 3,530 samples. By human evaluation, more than 92% of the new aspect robustness test set (ARTS) shows high fluency and desired sentiment on all aspects. Using our new test set, we analyze the aspect robustness of nine existing models. Experiment results show that their performance degrades by up to 55.64% on Laptop and 69.73% on Restaurant.

The contributions are as follows:

1. We develop simple but effective automatic generation methods that generate new test samples (with over 92% accuracy by human evaluation) to challenge the aspect robustness.
2. We construct ARTS, a new test set targeting aspect robustness for ABSA models, and propose a new metric, Aspect Robustness Score.
3. We probe the aspect robustness of nine models, and reveal up to 69.73% performance drop on ARTS compared with the original test set.

4. We provide solutions to enhance aspect robustness for ABSA models (Section 5.2.5).

5.2.2 Data Generation

As shown in Table 5.13, we aim to build a systematic method to generate all possible aspect-related alternations, in order to remove the confounding factors in the existing ABSA data. In the following, we will introduce three generation strategies.

5.2.2.1 RevTgt

The first strategy is to generate sentences that reverse the original sentiment of the target aspect. The word spans of each aspect's sentiment in SemEval 2014 data are provided by [Fan et al., 2019]. We design two methods to reverse the sentiment, and one additional step of conjunction adjustment on top of the two methods to polish the resulting sentence.

Strategy	**Example**
Flip Opinion	It's **light** and **easy** to transport. → It's **heavy** and **difficult** to transport.
Add Negation	The menu **changes** seasonally. → The menu **does not change** seasonally.
Adjust Conjunctions	The food is good, **and** the decor is **nice**. → The food is good, **but** the decor is **nasty**.

Table 5.14: Three strategies and examples of RevTgt.

Strategy	**Example**			
Original sentence & sentiment	It has great food and a reasonable price, but the service is poor.			
	(Tgt) food:+	**price:**+	**service:**−	**overall:**○
RevNon				
Flip same-sentiment non-target aspects (and adjust conjunctions)	It has great food **but** an **unreasonable** price, **and** the service is poor.			
Exaggerate opposite-sentiment non-target aspects	It has great food **but** an **unreasonable** price, **and** the service is **extremely** poor.			
	(Tgt) food:+	**price:**−	**service:**−−	**overall:**−

Table 5.15: The generation process of RevNon. The target aspect (Tgt), and sentiments of all aspects are annotated.

Flip Opinion Words Suppose we have the sentence "*Tasty* **burgers** and crispy fries," where the sentiment term for the target aspect is *Tasty.* We aim to generate a new sentence that flips the sentiment *Tasty.* A baseline approach is antonym replacement by looking up WordNet [Miller, 1995]. However, due to polysemy, the simple lookup is very likely to derive an inappropriate antonym and cause incompatibility with the context. Among the retrieved set of antonyms, we only keep words with the same Part-of-Speech (POS) tag as original, using the stanza package[12] which takes the context into account by the state-of-the-art neural network-based model.[13] Lastly, in the case of multiple antonyms, we prioritize the words that are already in the existing vocabulary, and then randomly select an antonym from the candidate set.

Add Negation The above strategy of flipping by the antonym is constrained by the availability of antonyms. For those cases without suitable antonyms, including long phrases, we add negation according to the linguistic features. In most cases, the sentiment expression is an adjective or verb term, so we simply add negation (i.e., "not") in front of it. If the sentiment term is not an adjective or verb, we add negation to its closest verb. For example, in Table 5.14, there are no available antonyms for "change" in the original sentence "The menu **changes** seasonally.", so we simply negate it as "The menu **does not change** seasonally."

Adjust Conjunctions As pinpointed in Section 5.2.1, 79.6∼83.9% of the test sentences have the same sentiment for all aspects. A possible result of reversing one aspect's sentiment is that the other aspects' sentiments will be opposite to the altered one. So we need to adjust the conjunctions for language fluency. If the two closest surrounding sentiments of a conjunction word have the same polarity, then cumulative conjunctions such as "and" should be applied; otherwise, we should adopt adversative conjunctions such as "but". In the example in Table 5.14, after flipping the sentiment, we derive the sentence "The food is good, **and** the decor is nasty"

[12] `https://stanfordnlp.github.io/stanza/`

[13] For the candidate filter, we do not use GPT-2 perplexity because of its low accuracy, e.g., 38.4% on a random sample set. And its output is also less interpretable than the POS filter.

which is very unnatural, so we replace the conjunction “and” with “but”, and thus generate the sentence “The food is good, **but** the decor is nasty.”

5.2.2.2 RevNon

Changing the target sentiment by RevTgt can test if a model is sensitive enough towards the target-aspect sentiment, but we need to further complement this probe by perturbing the sentiments of the non-target aspects (RevNon). As shown in Table 5.15, for all the non-target aspects with the same sentiment as the target aspect’s, we reverse their sentiments using the same method as RevTgt. And for all the remaining non-target aspects, whose sentiments are already opposite from the target sentiment, we exaggerate the extent by randomly adding an adverb (e.g., “very”, “really” and “extremely”) from a dictionary of adverbs of degree that is collected based on the training set. The resulting test sentence will be a solid proof of the ABSA quality, because only the target aspect has the desired sentiment, and all non-target aspects have been flipped to or exaggerated with the opposite sentiment.

5.2.2.3 AddDiff

The first two strategies, RevTgt and RevNon, have explored how the sentiment changes of existing aspects will challenge an ABSA model, and AddDiff further investigates if adding more non-target aspects can confuse the model. Moreover, the existing SemEval 2014 test sets have only on average of 2 aspects per sentence, but the real-world applications can have more aspects. With these motivations, we develop AddDiff as follows.

We first form a set of aspect expressions AspectSet by extracting all aspect expressions from the entire dataset. Specifically, for each sentence in the dataset, we first identify each sentiment term (e.g., “reasonable” in “Food at a reasonable price”) and then extract its linguistic branch as the aspect expression (e.g., “at a reasonable price”) by pretrained constituency parsing [Joshi et al., 2018]. Table 5.16 shows several examples of AspectSet in the restaurant domain.

Using the AspectSet, we randomly sample 1-3 aspects that are not mentioned in

Sentiment	Aspect Expression
Positive	staff is *friendly* and *knowledgeable* desserts are *out of this world* texture is *velvety*
Negative	service is *severely slow* dining experience is *miserable* tables are *uncomfortably close*

Table 5.16: Example aspect expressions from AspectSet of the restaurant domain.

the original test sample and whose sentiments are different from the target aspect's, and then append these to the end of the original sentence. For example, "Great food and best of all GREAT beer!" $\xrightarrow{\text{AddDiff}}$ "Great food and best of all GREAT beer, *but management is less than accommodating, music is too heavy, and service is severely slow.*"

5.2.3 ARTS Dataset

5.2.3.1 Overview

Our source data is the most widely used ABSA dataset, SemEval 2014 Laptop and Restaurant Reviews [Pontiki et al., 2014].[14] We follow Wang et al. [2016], Ma et al. [2017], and Xu et al. [2019a] to remove samples with conflicting polarity and only keep positive, negative, and neutral labels. We use the train-dev split as in [Xu et al., 2019a]. The resulting Laptop dataset has 2,163 training, 150 validation, and 638 test data, and Restaurant has 3,452 training, 150 validation, and 1,120 test data.

Building upon the original SemEval 2014 data, we generate enriched test sets of 1,877 samples (294% of the original size) in the laptop domain, and 3,530 samples (315%) in the restaurant domain using generation method introduced in Section 5.2.2. The statistics of our ARTS test set are in Table 5.17.

[14] http://alt.qcri.org/semeval2014/task4/

	Laptop	**Restaurant**
Original Test Set	638	1,120
Enriched Test Set	1,877	3,530
Relative Size	294.20%	315.17%
	Fluency Check	
Accepted Samples	1,732	3,260
Fixed Samples	145	270
Acceptance Rate	92.27%	92.35 %
Inter-Agreement	91.10%	92.69%
	Sentiment Check	
Accepted Samples	1,763	3,362
Fixed Samples	114	168
Acceptance Rate	93.93%	95.24%
Inter-Agreement	94.14%	95.61%

Table 5.17: Overall statistics of the ARTS test set and results of fluency and sentiment checks.

5.2.3.2 Quality Inspection

We conduct human evaluation to validate the generation quality of our ARTS dataset on two criteria:

1. **Fluency**: Does the generated sentence maintain the fluency of the source sentence?
2. **Sentiment Correctness**: Does the sentiment of each aspect have the desired polarity?

- RevTgt: Is the target sentiment reversed?
- RevNon: For non-target aspects with originally the same sentiment as the target, is it reversed? For the rest, are they exaggerated?
- AddDiff: Is the target sentiment unchanged?

Each task is completed by two native-speaker judges. We first calculate the inter-agreement rate of the human annotators, and then resolve the divergent opinions on samples where they disagree. We accept the samples that both judges considered as correct or are resolved to be correct after our check. Finally, we ask the annotators to fix the rejected samples by some minimal edit that does not change the aspect term or the sentence meaning, but satisfies both criteria.

Fluency Check The evaluation results on fluency are shown in Table 5.17. Most samples (92.27% of Laptop and 92.35% of Restaurant test sets) are accepted as fluent text. The inter-rater agreement between the two human judges is also high, 91.10% and 92.69% on the two datasets.

Sentiment Check We also evaluate the sentiment correctness of the generated text. Note that for RevNon, we count the samples with all "yes" answers as accepted samples. Overall, the acceptance rate of the generated samples is 93.93% on Laptop and 95.24% on Restaurant, along with inter-rater agreement of over 94.14% on both datasets.

5.2.3.3 Dataset Analysis

After checking the quality of our enriched ARTS test set, we analyze the dataset characteristics and make comparisons with the original test sets.

	Laptop		**Restaurant**	
	Ori	ARTS	Ori	ARTS
#Words/Sent	18.56	22.27	19.37	23.15
Vocab Size	1565	1746	2197	2451
Labels				
Positive	341	883	728	1953
Negative	128	587	196	1104
Neutral	169	407	196	473
#Positive/#Negative	2.66	1.5	3.71	1.77
Aspect-Related Challenge				
#Aspects/Sent	2.05	2.75	2.57	3.28
Opp. Nontgt ≥ 1	16%	59%	20%	67%
Opp. Nontgt = All	9%	38%	11%	42%
#Opp. Nontgt/Sent	0.23	1.16	0.27	1.39

Table 5.18: Characteristics of the New (“New”) test sets in comparison to the Original (“Ori”) Laptop and Restaurant test sets.

For general statistics, we can see from Table 5.18 that the sentence length in the new test set is on average 4 words more than the original, and the vocabulary size is also larger by around two hundred. *For the label distribution*, we can see that the new test set has an increasing number of all labels, and especially balances the ratio of positive-to-negative labels from the original 2.66 to 1.5 on Laptop, and from 3.71 to 1.77 on Restaurant.

For the aspect-related challenge in the test set, the new test set has a larger number of aspects per sentence than the original. Our test set also features a higher disentanglement of the target aspect from the non-target aspects that have the same sentiment as the target: the portion of samples with at least one non-target aspect with sentiments different from the target is 59-67%, and on average 45% higher than the original test sets. And the portion of the most challenging samples where all non-target aspects have sentiments different from the target one on the new test set is on average 30% more than that of the original test set. The average number of non-target aspects with opposite sentiments per sample in the new test set is on average 5 times that of the original set.

5.2.3.4 Aspect Robustness Score (ARS)

As mentioned in Section 5.2.1, a model is considered to have high aspect robustness if it satisfies both the prerequisite and all three questions (Q1)-(Q3). So we propose a novel metric, Aspect Robustness Score (ARS), that counts the correct classification of the source sentence and all its variations (RevTgt, RevNon, and AddDiff) as one unit of correctness. Then we apply the standard calculation of accuracy. Note that the three variations correspond to questions (Q1)-(Q3), respectively.

5.2.4 Evaluating ABSA Models

We use our enriched test set as a comprehensive test on the aspect robustness of ABSA models.

5.2.4.1 Models

For a comprehensive overview of the ABSA field, we conduct extensive experiments on models with a variety of neural network architectures.

TD-LSTM: [Tang et al., 2016a] uses two Long Short-Term Memory Networks (LSTM) to encode the preceding and following contexts of the target aspect (inclusive) and concatenates the last hidden states of the two LSTMs to make the sentiment classification.

AttLSTM: Wang et al. [2016] apply an Attention-based LSTM on the concatenatation of the aspect and word embeddings of each token.

GatedCNN: Xue and Li [2018] use a Gated Convolutional Neural Networks (CNN) that applies a Tanh-ReLU gating mechanism to the CNN-encoded text with aspect embeddings.

MemNet: Tang et al. [2016b] use memory networks to store the sentence as external memory and calculate the attention with the target aspect.

GCN: Aspect-specific Graph Convolutional Networks (GCN) [Zhang et al., 2019a] first applies GCN over the syntax tree of the sentence and then imposes an aspect-specific masking layer on its top.

BERT: Xu et al. [2019a] uses a BERT-based baseline [Devlin et al., 2019] and takes as input the concatenation of the aspect term and the sentence.

BERT-PT: Xu et al. [2019a] post-train BERT on other review datasets such as Amazon laptop reviews [He and McAuley, 2016] and Yelp Dataset Challenge reviews, and finetune on ABSA tasks.

CapsBERT: [Jiang et al., 2019] encode the sentence and the aspect term with BERT, and then feed it into Capsule Networks to predict the polarity.

BERT-Sent: For more in-depth analysis, we also implement a sentence classification baseline that only feeds the sentence without aspect information into BERT, and directly predicts the sentiment.

5.2.4.2 Implementation Details

For all existing models, we use the authors' official implementation. For our self-proposed BERT-Sent, we use the Adam optimizer with a learning rate of 5e-5, weight decay of 0.01, batch size of 32, apply the l_2 regularization with $\lambda = 10^{-4}$, and train 50 epochs.

Model	Entire Test	RevTgt Subset	RevNon Subset	AddDiff Subset
	Ori → New (Change)	Ori → New (Change)	Ori → New (Change)	Ori → New (Change)
Laptop Dataset				
MemNet	64.42 → 16.93 (↓47.49)*	72.10 → 28.33 (↓43.77)*	82.22 → 79.26 (↓02.96)	64.42 → 56.58 (↓07.84)*
GatedCNN	65.67 → 10.34 (↓55.33)*	75.11 → 24.03 (↓51.08)*	83.70 → 78.52 (↓05.18)	65.67 → 45.14 (↓20.53)*
AttLSTM	67.55 → 09.87 (↓57.68)*	72.96 → 27.04 (↓45.92)*	85.93 → 75.56 (↓10.37)*	67.55 → 39.66 (↓27.89)*
TD-LSTM	68.03 → 22.57 (↓45.46)*	73.39 → 29.83 (↓43.56)*	83.70 → 77.04 (↓06.66)	68.03 → 60.66 (↓07.37)*
GCN	72.41 → 19.91 (↓52.50)*	78.33 → 35.62 (↓42.71)*	88.89 → 74.81 (↓14.08)*	72.41 → 52.51 (↓19.90)*
BERT-Sent	73.04 → 17.40 (↓55.64)*	78.76 → 59.44 (↓19.32)*	88.15 → 42.22 (↓45.93)*	73.04 → 34.64 (↓38.40)*
CapsBERT	77.12 → 25.86[16] (↓51.26)*	80.69 → 57.73 (↓22.96)*	88.89 → 49.63 (↓39.26)*	77.12 → 45.14 (↓31.98)*
BERT	77.59 → 50.94 (↓26.65)*	83.05 → 65.02 (↓18.03)*	93.33 → 71.85 (↓21.48)*	77.59 → 71.00 (↓06.59)*
BERT-PT	78.53 → 53.29 (↓25.24)*	82.40 → 60.09 (↓22.31)*	93.33 → 83.70 (↓09.63)*	78.53 → 75.71 (↓02.82)
Average	71.60 → 25.23 (↓46.37)*	77.42 → 43.01 (↓34.41)*	87.57 → 70.29 (↓17.28)*	71.60 → 53.45 (↓18.15)*
Restaurant Dataset				
MemNet	75.18 → 21.52 (↓53.66)*	80.73 → 27.54 (↓53.19)*	84.46 → 73.65 (↓10.81)*	75.18 → 60.71 (↓14.47)*
GatedCNN	76.96 → 13.12 (↓63.84)*	85.11 → 23.17 (↓61.94)*	88.06 → 72.97 (↓15.09)*	76.96 → 54.91 (↓22.05)*
AttLSTM	75.98 → 14.64 (↓61.34)*	82.98 → 28.96 (↓54.02)*	86.26 → 61.26 (↓25.00)*	75.98 → 52.32 (↓23.66)*
TD-LSTM	78.12 → 30.18 (↓47.94)*	85.34 → 34.99 (↓50.35)*	88.51 → 75.68 (↓12.83)*	78.12 → 70.18 (↓07.94)*
GCN	77.86 → 24.73 (↓53.13)*	86.76 → 35.58 (↓51.18)*	88.51 → 79.50 (↓09.01)*	77.86 → 65.00 (↓12.86)*
BERT-Sent	80.62 → 10.89 (↓69.73)*	89.60 → 44.80 (↓44.80)*	89.86 → 57.21 (↓32.65)*	80.62 → 30.89 (↓49.73)*
CapsBERT	83.48 → 55.36 (↓28.12)*	89.48 → 71.87 (↓17.61)*	90.99 → 74.55 (↓16.44)*	83.48 → 77.86 (↓05.62)*
BERT	83.04 → 54.82 (↓28.22)*	90.07 → 63.00 (↓27.07)*	91.44 → 83.33 (↓08.11)*	83.04 → 79.20 (↓03.84)*
BERT-PT	86.70 → 59.29 (↓27.41)*	92.20 → 72.81 (↓19.39)*	92.57 → 81.76 (↓10.81)*	86.70 → 80.27 (↓06.43)*
Average	79.77 → 31.62 (↓48.15)*	86.92 → 44.75 (↓42.17)*	88.96 → 73.32 (↓15.64)*	79.77 → 63.48 (↓16.29)*

Table 5.19: Model accuracy on Laptop and Restaurant data. We compare the accuracy on the **Ori**ginal and our **New** test sets (*Ori* → *New*), and calculate the *change* of accuracy. Besides the Entire Test Set, we also list accuracy on subsets where the generation strategies RevTgt, RevNon and AddDiff can be applied. The accuracy of *Entire Test-New* is calculated using ARS. $\star$ indicates whether the performance drop is statistically significant (with p-value ≤ 0.05 by Welch's t-test).

5.2.4.3 Results on ARTS

We list the accuracy[15] of the nine models on the Laptop and Restaurant test sets in Table 5.19.

Overall Performance On the entire test set, we can see that the accuracy of all models on the original test set is very high, achieving up to 78.53% on Laptop and 86.70% on Restaurant, but it drops drastically (↓69%–↓25%) on our new test sets.

[15] For ABSA, accuracy is the standard metric to be reported [Tang et al., 2016b; Wang et al., 2016; Xue and Li, 2018]. For *Entire Test-New* in Table 5.19, accuracy is calculated using ARS.

Performance of Different Models From the overall performance on our new test set, we can see that BERT models on average are more robust to the aspect-targeted challenges that our new test set poses. The most effective model BERT-PT scores the best on both original accuracy and robustness. It has 53.29% ARS on Laptop and 59.29% on Restaurant. However, the accuracy of non-BERT models on average drops drastically to under 30% by over ↓50%.

Performance on Different Subsets We list in detail the performance of each model on the three subsets of our new test set: RevTgt, RevNon, and AddDiff. They correspond to the three questions (Q1)-(Q3). RevTgt on average induces the most performance drop, as it requires the model to pay precise attention to the target sentiment words. RevNon makes the performance of the sentence classifier BERT-Sent drop the most, by up to ↓45.93%, and the model CapsBERT also drops by up to ↓39.26%. The last subset AddDiff causes most non-BERT models to drop significantly, indicating that these models are not robust enough against an increased number of non-target aspects, which should have been irrelevant.

5.2.5 Analysis

5.2.5.1 Variations of Generation Strategies

Combining Multiple Strategies Each sample in the ARTS test set is generated by one of the three strategies. However, it is also worth exploring whether combining several strategies can make a more challenging probe on the aspect robustness of ABSA models. As a case study, we analyze the model robustness against test samples generated by the combination of RevNon+AddDiff. By comparing the performance decrease caused by RevNon+AddDiff in Table 5.20 and by only RevNon and AddDiff in Table 5.19, we can see that the accuracy of each model decreases by a much larger extent on RevNon+AddDiff than either of RevNon or AddDiff.

Model	Laptop	Restaurant
	Ori → New (Change)	Ori → New (Change)
MemNet	82.22 → 72.59 (↓09.63)	84.46 → 50.90 (↓33.56)*
GatedCNN	84.44 → 59.26 (↓25.18)*	87.84 → 53.83 (↓34.01)*
AttLSTM	85.93 → 51.85 (↓34.08)*	86.26 → 38.06 (↓48.20)*
TD-LSTM	83.70 → 68.89 (↓14.81)*	88.51 → 65.99 (↓22.52)*
GCN	88.89 → 60.74 (↓28.15)*	88.51 → 72.52 (↓15.99)*
BERT-Sent	88.15 → 11.85 (↓76.30)*	89.86 → 11.94 (↓77.92)*
CapsBERT	90.37 → 24.44 (↓65.93)*	90.99 → 66.89 (↓24.10)*
BERT	93.33 → 68.15 (↓25.18)*	91.44 → 76.58 (↓14.86)*
BERT-PT	93.33 → 78.52 (↓14.81)*	92.57 → 78.60 (↓13.97)*
Average	87.57 → 55.14 (↓32.43) *	88.96 → 57.26 (↓31.70)*

Table 5.20: The accuracy of each model on the original test set and the new test set generated by RevNon+AddDiff in laptop and restaurant domains.

AddDiff with More Aspects Some strategies such as AddDiff can be parameterized by k, where k is the number of additional non-target aspects to be added. We select three models (the best, the worst, and an average-performing one), and plot their accuracy on test samples generated by AddDiff(k) on Laptop in Figure 5-1. As k gets larger, the test samples become more difficult. The sentence classification baseline BERT-Sent drops drastically, BERT-PT remains high, and GCN lies in the middle.

5.2.5.2 How to Effectively Model the Aspect?

An important usage of our ARTS is to understand what model components are key to aspect robustness. We list the aspect-specific mechanisms of all models according to the descending order of their ARS on the Laptop dataset in Table 5.21. We can see that for BERT-based models, BERT-PT, which is further trained on large review

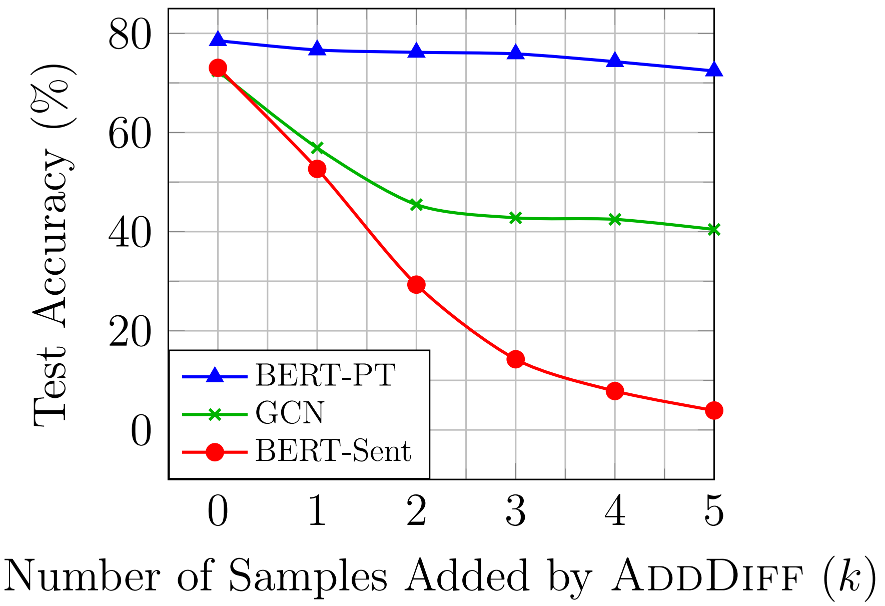


Figure 5-1: Accuracy of BERT-PT, GCN, and BERT-Sent on the test samples in the laptop domain generated by AddDiff(k) where k varies from 1 to 5.

corpora, gets the best accuracy and aspect robustness. More complicated structures like CapsBERT underperforms the basic BERT by 25.08%.

Among the non-BERT models, the aspect position-aware models TD-LSTM and GCN are the most robust, as they have a stronger sense of the location of the target aspect in a sentence. On the contrary, the other models with poorer robustness (9.87%–16.93% in Table 5.21) only use mechanisms such as aspect-based attention, or concatenating the aspect embedding to the word embedding.

To summarize, the main takeaways are

- For BERT models, additional pretraining is the most effective.
- For non-BERT models, explicit position-aware designs lead to more aspect robustness.

5.2.5.3 Does Better Training Help?

The following three settings explore whether better training can improve the aspect robustness.

Model	ARS	Asp+W Emb	Posi-Aware	Asp Att
AttLSTM	9.87	✓	✗	✓
GatedCNN	10.34	✓	✗	✓
MemNet	16.93	✗	✗	✓
GCN	19.91	✗	✓	✓
TD-LSTM	22.57	✗	✓	✗
CapsBERT	25.86	✗	✗	✓
BERT	50.94	✗	✗	✗
BERT-PT	53.29	✗	✗	✗

Table 5.21: Models in the descending order of their ARS on Laptop. We list their aspect-specific mechanisms, including concatenating the aspect and word embeddings (Asp+W Emb), position-aware mechanism for aspects (Posi-Aware), and attention using the aspect (Asp Att). We highlight ✓ for Posi-Aware as it is the most related to aspect robustness for non-BERT models.

Training and Testing on MAMS A recent dataset, Multi-Aspect Multi-Sentiment (MAMS) [Jiang et al., 2019], is collected from the same data source as the SemEval 2014 Restaurant dataset [Ganu et al., 2009]. However, the sentences are more complicated, each having at least two aspects with different sentiment polarities. Table 5.22a checks the aspect robustness of models trained on MAMS using the original MAMS test set (O→O) and the new test set that we produced by applying the same generation strategies to its test set (O→N). Models trained and tested on MAMS have a smaller performance decrease than those on the Restaurant dataset. This shows that a more challenging training set can make models more robust.

Training on MAMS and Testing on Restaurant As MAMS and Restaurant are collected from the same source data, we test whether MAMS-trained models perform well on the new test set of Restaurant (in the column "MAMS→N" of Table 5.22b). We can see that all models trained on MAMS are more robust than those trained on

Model	MAMS	
	O→O	O→N
MemNet	70.51	37.80
GatedCNN	66.02	32.93
AttLSTM	67.14	39.67
TD-LSTM	77.62	49.25
GCN	76.95	47.98
BERT-Sent	49.25	10.48
CapsBERT	83.38	60.18
BERT	84.51	61.38
BERT-PT	85.10	64.37

(a) Accuracy of each model trained on the MAMS **O**riginal training data and evaluated on the **O**riginal test data (O→O), as well as the **N**ew test set generated by our models (O→N).

Restaurant				Laptop		
O→O	O→N	MAMS→N	Adv→N	O→O	O→N	Adv→N
75.18	21.52	24.02	37.95	64.42	16.93	31.82
76.96	13.13	18.48	37.50	65.67	10.34	41.85
75.98	14.64	22.32	48.66	67.55	9.87	42.63
78.12	30.18	41.60	62.76	68.03	22.57	54.86
77.86	24.73	46.51	61.52	72.41	19.91	56.43
80.62	10.89	12.95	45.80	73.04	17.40	53.92
83.66	55.36	61.43	75.80	76.80	25.86	61.23
83.04	54.82	62.77	74.82	77.59	50.94	65.67
86.70	59.29	62.77	74.64	78.53	53.29	66.93

(b) Accuracy of each model trained on the **O**riginal data and evaluated on the **O**riginal test set (O→O), and the **N**ew test set (O→N), as well as that trained on the **Adv**ersarial data and evaluated on the **N**ew test set (Adv→N). For Restaurant, we also test models trained on **MAMS** dataset and tested on the **N**ew test set of Restaurant (MAMS→N).

Table 5.22: Improvements on the new test set MAMS using different training data.

the Restaurant dataset. For example, the accuracy of BERT and BERT-PT on the new test set is lifted up to 62.77%.

Training on Adversarial Samples Adversarial training is also a good way to enhance models' aspect robustness. We conducted adversarial training on the Laptop and Restaurant datasets, and analyze its effect in Table 5.22b. In both domains, adversarial training (Adv→N) leads to significant performance improvement over only training on the original datasets (O→N). On the Restaurant datasets, adversarial training is even more effective than training on MAMS, because our generated samples comprehensively covered all possible perturbations of the non-target aspects, and naturally collected datasets might not be comparable.

5.2.6 Error Analysis

We analyze the error types in the subset that was fixed by human judges. Two most significant error types are wrong antonyms (∼2%), such as "the weight of the laptop is light→dark", and negation which causes grammatical errors (∼1.1%). In future work, we can fix the latter by applying a grammatical error correction system on top of our generation. Also, RevTgt and RevNon cannot be applied to 1.4–6.6% samples with complicated sentiment expressions which rely on commonsense. For example, "a 2-hour wait" is negative but too difficult to alter in our current generation framework. It needs more advanced models such as text style transfer [Jin et al., 2019c; Shen et al., 2017].

5.2.7 Related Work

Robustness in NLP Robustness in NLP has attracted extensive attention in recent works [Hsieh et al., 2019; Li et al., 2016]. As a popular method to probe the robustness of models, adversarial text generation becomes an emerging research field in NLP. Techniques include adding extraneous text to the input [Jia and Liang, 2016], character-level noise [Belinkov and Bisk, 2018; Ebrahimi et al., 2018], and word replacement [Alzantot et al., 2018; Jin et al., 2019b]. Using the adversarial generation techniques, new adversarial test sets are proposed for several tasks such as paraphrasing [Zhang et al., 2019c] and entailment [Glockner et al., 2018].

Aspect-Based Sentiment Analysis ABSA has emerged as an active research area recently. Early works hand-craft sentiment lexicons and syntactic features for rule-based classifiers [Kiritchenko et al., 2014; Vo and Zhang, 2015]. Recent neural network-based models use architectures such as LSTM [Tang et al., 2016a], CNN [Xue and Li, 2018], Attention mechanisms [Wang et al., 2016], Capsule Networks [Jiang et al., 2019], and the pretrained model BERT [Xu et al., 2019a]. Similar to the motivation in our paper, some work shows preliminary speculation that the current ABSA datasets might be downgraded to sentence-level sentiment classification [Xu et al., 2019b].

5.2.8 Summary

In this section, we proposed a simple but effective mechanism to generate test samples to probe the aspect robustness of sentiment analysis models. We enhanced the original SemEval 2014 test sets by 294% and 315% in the laptop and restaurant domains. Using our new test set, we probed the aspect robustness of nine ABSA models, and discussed model designs and better training that can improve their robustness.

5.3 Conclusions

In this chapter, we have proposed two approaches to probing the robustness of NLP models for text classification, natural language inference, and aspect-based sentiment analysis tasks, based on adversarial attacking. On one hand, we successfully demonstrate that even the best-performing BERT models are vulnerable to small perturbations to the original data although these crafted alterations are not perceivable by people. On the other hand, such a probing method can be used to automatically create a new test set as a challenge set to more comprehensively evaluate the current state-of-the-art and future models. Overall, in order for deep learning models to be used in real-world scenarios, we need to evaluate and enhance their robustness and the works introduced in this chapter can be good contributions to this direction.

Chapter 6

Conclusions

6.1 Synopsis

In this dissertation, we first proposed a series of transfer learning methods to help improve the performance of deep learning models on those low-resource tasks/datasets and prevent them from over-fitting due to the scarce training data. Besides pursuing high performance on the benchmark test sets, we further explored the robustness of state-of-the-art deep learning models on NLP tasks and evaluated their performance on noisy data that more commonly exist in real-world applications.

In Chapter 2, we first gave a brief introduction of the NLP tasks that have been tackled in this dissertation. Then we summarized in detail the definition, the taxonomy, and the four divisions of transfer learning, i.e., multi-task learning, sequential transfer learning, domain adaptation, and cross-lingual learning. Last, we introduced the background for understanding robustness, especially focusing on adversarial robustness, including its origin, theories behind it, its development, and general methods. This chapter lays the knowledge foundation for the following chapters and helps readers to have a better technical understanding.

In Chapter 3, we proposed three different transfer learning methods for three NLP tasks: multi-choice question answering, dialogue state tracking, and named entity recognition. First of all, we introduced a multi-stage and multi-task transfer learning strategy to boost the performance of four low-resource MCQA datasets by at least

9%, close to human-level performance. Secondly, we proposed casting a dialogue state tracking task into a combination task of multi-choice question answering and span-based question answering so that we can leverage the advances and large datasets from the question answering field via sequential transfer learning. Lastly, we proposed a dual adversarial domain adaptation method to bring close the latent representations across different domains and languages for the NER task so that models can better learn the common knowledge between high- and low-resource domains/languages. Overall, all these proposed methods can help boost the performance of state-of-the-art deep learning models on low-resource natural language understanding tasks.

In Chapter 4, we focused on improving the low-resource natural language generation task, particularly the machine translation task. Specifically, we proposed an iterative back-translation based approach, where the translation model is iteratively trained to map the translated sentences back to the source language so that the model can make use of non-parallel low-resource domain data. We multi-task train this model on the parallel high-resource domain data via a supervised translation objective and on the non-parallel low-resource domain data via the mentioned iterative back-translation. In this way, this model can not only be regularized by source domain data but also be customized to the language characteristics of target domain data. Based on experiments on two adaptation settings and two language pairs, we demonstrated that this method can improve the semi-supervised domain adaptation of the machine translation task significantly.

In chapter 5, we raised the concern that although current best deep learning models can achieve impressive performance on various clean benchmark datasets, they are still susceptible to natural noise/artifacts that exist in the data or to adversarial perturbations that are small and human-imperceptible distortions of the input, yet can easily fool neural networks. To study the robustness of NLP models, we proposed a method to create adversarial samples that are semantically similar to the original text but can force models to make wrong predictions. Such a method can degrade the best text classification and entailment models from a high accuracy of around 90% to around 10% with less than 20% of words in the text changed to their synonyms. We also introduced a new challenge set for the aspect-based sentiment analysis task so as to comprehensively evaluate the robustness of state-of-the-art methods for this task. Such a challenge set is created by our proposed four kinds of adversarial strategies. Overall, with all findings mentioned in this chapter, we revealed the weakness of current NLP models and urge future works to enhance their robustness before applying them into the wild.

All together, we are really excited about the progress that has been made in this field for the past 3 years and have been glad to be able to contribute to improving and evaluating the generalization of NLP models. At the same time, we also deeply believe that there is still a long way to go towards genuine human-level natural language understanding, and we are still facing enormous challenges and a lot of open questions that we will need to address in the future.

6.2 Limitations and Future Works

Understanding task relationships Through massive experiments, we can find out the best transfer learning strategy for a specific task and dataset. However, such a trial and error strategy is not scalable and time- and resource-consuming. It would be very helpful to come up with a metric to calculate the task similarity between two tasks and relate this metric to the transfer performance from one to the other. In this way, we can have a prior judgment of whether one task would be beneficial to another one before conducting any transfer learning experiments, which can save much effort and many resources. Alternatively, we can have a large-scale study on enormous existing NLP datasets, similar to the one done by Zamir et al. [2018] in computer vision, to obtain the empirical transfer learning performance between every pair of datasets among all investigated candidates and then summarize the performance results into a taxonomy, which can be used by future works as a reference. Another interesting direction is that we can possibly leverage techniques from AutoML [He et al., 2019] to search the best transfer learning strategy by algorithms instead of human efforts for a certain target task.

Challenge sets As we become more cognizant of the brittleness of our current methods, there is an emerging research trend that various challenge problems and data sets are proposed for some existing important tasks, such as adversarial NLI [Nie et al., 2020], contrast sets on ten popular datasets [Gardner et al., 2020], and our work on ABSA datasets [Xiang et al., 2020]. Such challenges probe particular aspects, such as certain linguistic phenomena on which current models fail, and can thus teach us what current models are still not good at. Challenge sets have been mainly created for tasks such as text classification, question answering, and machine translation and mostly for English. In the future, we expect the creation of challenge sets for other tasks and non-English languages. Ideally, when new datasets are created, in addition to a random test set, dedicated out-of-domain test sets will become common practice. This will enable us to test whether our models truly generalize.

Robustness to out-of-distribution and adversarial data Although we now have been aware of models' potential degradation in performance on those adversarially created data or out-of-distribution challenge sets, we hope that more work will focus on making them more robust, which is still a hot and open question. Adversarial training can be a good method to enhance the robustness of models by training them on augmented data obtained by combining the original data with newly adversarially generated data. However, such a straightforward method has only achieved a limited improvement of robustness and also worse performance on clean test sets. More advanced methods such as certified robustness [Jia et al., 2019] should be developed.

Transfer Learning On a broader note, in the long-term we expect transfer learning to be an integral part of NLP systems. Language is a reflection of our context and experience. Training from a blank slate deprives our models of experience and the ability to interpret context. Ultimately, in order to come closer to the elusive goal of true natural language understanding, we need to equip our models with as much relevant knowledge and experience as possible. For example, common sense knowledge is a valuable knowledge source that has stored human experience over thousands of years, and thus it would be extremely beneficial to develop components that can acquire and integrate common sense and world knowledge from disparate sources into our models. And transfer learning can surely play an important role in this integration process.

Bibliography

Oliver Adams, Adam Makarucha, Graham Neubig, Steven Bird, and Trevor Cohn. Cross-lingual word embeddings for low-resource language modeling. In *EACL*, pages 937–947. Association for Computational Linguistics, 2017.

Gustavo Aguilar, Suraj Maharjan, Adrian Pastor López Monroy, and Thamar Solorio. A multi-task approach for named entity recognition in social media data. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 148–153, 2017.

Rami Al-Rfou', Vivek Kulkarni, Bryan Perozzi, and Steven Skiena. Polyglot-ner: Massive multilingual named entity recognition. In *SDM*, 2015.

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. Generating natural language adversarial examples. *arXiv preprint arXiv:1804.07998*, 2018.

Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, 2017.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. Unsupervised neural machine translation. *arXiv preprint arXiv:1710.11041*, 2017.

Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *International conference on machine learning*, pages 284–293, 2018.

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/D11-1033`.

Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.

Yonatan Belinkov and Yonatan Bisk. Synthetic and natural noise both break neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL `https://openreview.net/forum?id=BJ8vJebC-`.

Shai Ben-David and Reba Schuller. Exploiting task relatedness for multiple task learning. In *Learning Theory and Kernel Machines*, pages 567–580. Springer, 2003.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Advances in neural information processing systems*, pages 137–144, 2007.

Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100, 1998.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In *Advances in neural information processing systems*, pages 343–351, 2016.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2015.

Denny Britz, Quoc Le, and Reid Pryzant. Effective domain mixing for neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 118–126, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4712. URL `https://www.aclweb.org/anthology/W17-4712`.

Peter F Brown, Vincent J Della Pietra, Stephen A Della Pietra, and Robert L Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311, 1993.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Inigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *arXiv preprint arXiv:1810.00278*, 2018.

Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, and Shengping Liu. Adversarial transfer learning for chinese named entity recognition with self-attention mechanism. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 182–192, 2018.

Nicholas Carlini and David Wagner. Audio adversarial examples: Targeted attacks on speech-to-text. *arXiv preprint arXiv:1801.01944*, 2018.

Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*, 2018.

Soravit Changpinyo, Hexiang Hu, and Fei Sha. Multi-task learning for sequence tagging: An empirical study. *arXiv preprint arXiv:1808.04151*, 2018.

Danqi Chen. *Neural Reading Comprehension and Beyond*. PhD thesis, Stanford University, 2018.

Minmin Chen, Zhixiang Eddie Xu, Kilian Q. Weinberger, and Fei Sha. Marginalized denoising autoencoders for domain adaptation. *ArXiv*, abs/1206.4683, 2012.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. Enhanced lstm for natural language inference. *arXiv preprint arXiv:1609.06038*, 2016.

Shang-Tse Chen, Cory Cornelius, Jason Martin, and Duen Horng Chau. Robust physical adversarial attack on faster r-cnn object detector. In *ECML/PKDD*, 2018.

Yufeng Chen, Chengqing Zong, and Keh-Yih Su. On jointly recognizing and aligning bilingual named entities. In *ACL*, pages 631–639, 2010.

Yong Cheng, Wei Xu, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. Semi-supervised learning for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1965–1974, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1185. URL `https://www.aclweb.org/anthology/P16-1185`.

Jason Chiu and Eric Nichols. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370, 2016.

Noam Chomsky and David W Lightfoot. *Syntactic structures*. Walter de Gruyter, 1957.

Sauhaarda Chowdhuri, Tushar Pankaj, and Karl Zipser. Multinet: Multi-modal multi-task learning for autonomous driving. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1496–1504. IEEE, 2019.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. An empirical comparison of domain adaptation methods for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 385–391, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-2061. URL https://www.aclweb.org/anthology/P17-2061.

Yu-An Chung, Hung-Yi Lee, and James Glass. Supervised and unsupervised transfer learning for question answering. *arXiv preprint arXiv:1711.05345*, 2017.

Ming-Chien Chyu, Tony Austin, Fethi Calisir, Samuel Chanjaplammootil, Mark J Davis, Jesus Favela, Heng Gan, Amit Gefen, Ram Haddas, Shoshana Hahn-Goldberg, et al. Healthcare engineering defined: a white paper. *Journal of healthcare engineering*, 6(4):635–648, 2015.

Massimiliano Ciaramita and Olivier Chapelle. Adaptive parameters for entity recognition with perceptron hmms. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, pages 1–7, 2010.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *JMLR*, pages 2493–2537, 2011.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1070. URL https://www.aclweb.org/anthology/D17-1070.

Ryan Cotterell and Kevin Duh. Low-resource named entity recognition with cross-lingual, character-level neural conditional random fields. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 91–96. Asian Federation of Natural Language Processing, 2017.

Anna Currey, Antonio Valerio Miceli Barone, and Kenneth Heafield. Copied monolingual data improves low-resource neural machine translation. In *WMT*, 2017.

Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. In *Advances in neural information processing systems*, pages 3079–3087, 2015.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

Franck Dernoncourt, Ji Young Lee, Özlem Uzuner, and Peter Szolovits. De-identification of patient notes with recurrent neural networks. *Journal of the American Medical Informatics Association*, 24:596–606, 2017.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *2019*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/N19-1423`.

Emily Dinan, Varvara Logacheva, Valentin Malykh, Alexander Miller, Kurt Shuster, Jack Urbanek, Douwe Kiela, Arthur Szlam, Iulian Serban, Ryan Lowe, et al. The second conversational intelligence challenge (convai2). *arXiv preprint arXiv:1902.00098*, 2019.

Tobias Domhan and Felix Hieber. Using target-side monolingual data for neural machine translation through multi-task learning. In *EMNLP*, 2017.

Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 49–54, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-2009. URL `https://www.aclweb.org/anthology/P14-2009`.

Cícero dos Santos and Victor Guimarães. Boosting named entity recognition with neural character embeddings. In *Proceedings of the Fifth Named Entity Workshop*, pages 25–33, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.18653/v1/W15-3904. URL `https://www.aclweb.org/anthology/W15-3904`.

Zi-Yi Dou, Junjie Hu, Antonios Anastasopoulos, and Graham Neubig. Unsupervised domain adaptation for neural machine translation with domain-aware feature embeddings. In *EMNLP/IJCNLP*, 2019.

Kevin Duh. The multitarget ted talks task. `http://www.cs.jhu.edu/~kevinduh/a/multitarget-tedtalks/`, 2018.

Kevin Duh, Graham Neubig, Katsuhito Sudoh, and Hajime Tsukada. Adaptation data selection using neural language models: Experiments in machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 678–683, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/P13-2119`.

Matthew Dunn, Levent Sagun, Mike Higgins, V Ugur Guney, Volkan Cirik, and Kyunghyun Cho. Searchqa: A new q&a dataset augmented with context from a search engine. *arXiv preprint arXiv:1704.05179*, 2017.

Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. 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 845–850, 2015.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. Hotflip: White-box adversarial examples for text classification. In *ACL*, 2017.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. Hotflip: White-box adversarial examples for text classification. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 31–36. Association for Computational Linguistics, 2018. doi: 10.18653/v1/P18-2006. URL `https://www.aclweb.org/anthology/P18-2006/`.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*, 2018.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, and Dilek Hakkani-Tur. Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines. *arXiv preprint arXiv:1907.01669*, 2019.

Marzieh Fadaee and Christof Monz. Back-translation sampling by targeting difficult words in neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 436–446, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1040. URL `https://www.aclweb.org/anthology/D18-1040`.

Zhifang Fan, Zhen Wu, Xin-Yu Dai, Shujian Huang, and Jiajun Chen. Target-oriented opinion words extraction with target-fused neural sequence labeling. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2509–2518. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1259. URL `https://doi.org/10.18653/v1/n19-1259`.

Meng Fang and Trevor Cohn. Model transfer for tagging low-resource languages using a bilingual dictionary. In *ACL*, pages 587–593, 2017.

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1615–1625, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1169. URL `https://www.aclweb.org/anthology/D17-1169`.

Vitaly Feldman. Does learning require memorization? A short tale about a long tail. *CoRR*, abs/1906.05271, 2019. URL `http://arxiv.org/abs/1906.05271`.

Shi Feng, Eric Wallace, II Grissom, Mohit Iyyer, Pedro Rodriguez, Jordan Boyd-Graber, et al. Pathologies of neural models make interpretations difficult. *arXiv preprint arXiv:1804.07781*, 2018a.

Xiaocheng Feng, Xiachong Feng, Bing Qin, Zhangyin Feng, and Ting Liu. Improving low resource named entity recognition using cross-lingual knowledge transfer. In *IJCAI*, pages 4071–4077, 2018b.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. Mrqa 2019 shared task: Evaluating generalization in reading comprehension. *arXiv preprint arXiv:1910.09753*, 2019.

Markus Freitag and Yaser Al-Onaizan. Fast domain adaptation for neural machine translation. *arXiv preprint arXiv:1612.06897*, 2016.

Jules Gagnon-Marchand, Hamed Sadeghi, Md Haidar, Mehdi Rezagholizadeh, et al. Salsa-text: self attentive latent space based adversarial text generation. *arXiv preprint arXiv:1809.11155*, 2018.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189, 2015.

Gayatree Ganu, Noemie Elhadad, and Amélie Marian. Beyond the stars: Improving rating predictions using review text content. In *12th International Workshop on the Web and Databases, WebDB 2009, Providence, Rhode Island, USA, June 28, 2009*, 2009. URL `http://webdb09.cse.buffalo.edu/papers/Paper9/WebDB.pdf`.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. Black-box generation of adversarial text sequences to evade deep learning classifiers. *arXiv preprint arXiv:1801.04354*, 2018.

Shuyang Gao, Abhishek Sethi, Sanchit Agarwal, Tagyoung Chung, and Dilek Hakkani-Tur. Dialog state tracking: A neural reading comprehension approach. *arXiv preprint arXiv:1908.01946*, 2019.

Shuyang Gao, Sanchit Agarwal, Di Jin, Tagyoung Chung, and Dilek Hakkani-Tur. From machine reading comprehension to dialogue state tracking: Bridging the gap. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 79–89, Online, July 2020. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/2020.nlp4convai-1.10`.

Matt Gardner, Yoav Artzi, Victoria Basmova, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, et al. Evaluating nlp models via contrast sets. *arXiv preprint arXiv:2004.02709*, 2020.

Dan Gillick, Cliff Brunk, Oriol Vinyals, and Amarnag Subramanya. Multilingual language processing from bytes. In *NAACL HLT*, pages 1296–1306, 2016.

Max Glockner, Vered Shwartz, and Yoav Goldberg. Breaking NLI systems with sentences that require simple lexical inferences. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 650–655. Association for Computational Linguistics, 2018. doi: 10.18653/v1/P18-2103. URL `https://www.aclweb.org/anthology/P18-2103/`.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *ICML*, 2011.

Rahul Goel, Shachi Paul, Tagyoung Chung, Jeremie Lecomte, Arindam Mandal, and Dilek Hakkani-Tur. Flexible and scalable state tracking framework for goal-oriented dialogue systems. *arXiv preprint arXiv:1811.12891*, 2018.

Rahul Goel, Shachi Paul, and Dilek Hakkani-Tür. Hyst: A hybrid approach for flexible and accurate dialogue state tracking. *arXiv preprint arXiv:1907.00883*, 2019.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, 2014.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning.* MIT press, 2016.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1412.6572`.

Tao Gui, Qi Zhang, Haoran Huang, Minlong Peng, and Xuanjing Huang. Part-of-speech tagging for twitter with adversarial neural networks. In *EMNLP*, pages 2411–2420, 2017.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. The FLORES evaluation datasets for low-resource machine translation: Nepali–English and Sinhala–English. 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 6098–6111, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1632. URL `https://www.aclweb.org/anthology/D19-1632`.

Hrayr Harutyunyan, Hrant Khachatrian, David C Kale, Greg Ver Steeg, and Aram Galstyan. Multitask learning and benchmarking with clinical time series data. *Scientific data*, 6(1):1–18, 2019.

Kazuma Hashimoto, caiming xiong, Yoshimasa Tsuruoka, and Richard Socher. A joint many-task model: Growing a neural network for multiple nlp tasks. In *EMNLP*, pages 1923–1933, 2017.

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. Dual learning for machine translation. In *Advances in neural information processing systems*, pages 820–828, 2016.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. Adaptive semi-supervised learning for cross-domain sentiment classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3467–3476, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1383. URL `https://www.aclweb.org/anthology/D18-1383`.

Ruining He and Julian J. McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In Jacqueline Bourdeau, Jim Hendler, Roger Nkambou, Ian Horrocks, and Ben Y. Zhao, editors, *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pages 507–517. ACM, 2016. doi: 10.1145/2872427.2883037. URL `https://doi.org/10.1145/2872427.2883037`.

Xin He, Kaiyong Zhao, and Xiaowen Chu. Automl: A survey of the state-of-the-art. *arXiv preprint arXiv:1908.00709*, 2019.

Yulan He and Deyu Zhou. Self-training from labeled features for sentiment analysis. *Information Processing & Management*, 47(4):606–616, 2011.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *arXiv preprint arXiv:1907.07174*, 2019.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701, 2015.

Felix Hill, Roi Reichart, and Anna Korhonen. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695, 2015.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1031. URL `https://www.aclweb.org/anthology/P18-1031`.

Yu-Lun Hsieh, Minhao Cheng, Da-Cheng Juan, Wei Wei, Wen-Lian Hsu, and Cho-Jui Hsieh. On the robustness of self-attentive models. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1520–1529. Association for Computational Linguistics, 2019. doi: 10.18653/v1/p19-1147. URL `https://doi.org/10.18653/v1/p19-1147`.

Junjie Hu, Mengzhou Xia, Graham Neubig, and Jaime Carbonell. Domain adaptation of neural machine translation by lexicon induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2989–3001, Florence, Italy, July 2019a. Association for Computational Linguistics. doi: 10.18653/v1/P19-1286. URL `https://www.aclweb.org/anthology/P19-1286`.

Minghao Hu, Furu Wei, Yuxing Peng, Zhen Huang, Nan Yang, and Dongsheng Li. Read+ verify: Machine reading comprehension with unanswerable questions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6529–6537, 2019b.

Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *CoRR*, abs/1508.01991, 2015.

John Hutchins. The first public demonstration of machine translation: the georgetown-ibm system, 7th january 1954. *noviembre de*, 2005.

Robin Jia and Percy Liang. Data recombination for neural semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12–22, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1002. URL `https://www.aclweb.org/anthology/P16-1002`.

Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1215. URL `https://www.aclweb.org/anthology/D17-1215`.

Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. Certified robustness to adversarial word substitutions. 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 4129–4142, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1423. URL `https://www.aclweb.org/anthology/D19-1423`.

Qingnan Jiang, Lei Chen, Ruifeng Xu, Xiang Ao, and Min Yang. A challenge dataset and effective models for aspect-based sentiment analysis. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the*

2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, pages 6279–6284. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1654. URL `https://doi.org/10.18653/v1/D19-1654`.

Di Jin, Shuyang Gao, Jiun-Yu Kao, Tagyoung Chung, and Dilek Hakkani-tur. Mmm: Multi-stage multi-task learning for multi-choice reading comprehension. *arXiv preprint arXiv:1910.00458*, 2019a.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is BERT really robust? natural language attack on text classification and entailment. *CoRR*, abs/1907.11932, 2019b. URL `http://arxiv.org/abs/1907.11932`.

Di Jin, Shuyang Gao, Jiun-Yu Kao, Tagyoung Chung, and Dilek Z. Hakkani-Tür. Mmm: Multi-stage multi-task learning for multi-choice reading comprehension. In *AAAI*, 2020a.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *AAAI*, 2020b.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. A simple baseline to semi-supervised domain adaptation for machine translation, 2020c.

Zhijing Jin, Di Jin, Jonas Mueller, Nicholas Matthews, and Enrico Santus. Imat: Unsupervised text attribute transfer via iterative matching and translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3088–3100, 2019c.

Vidur Joshi, Matthew E. Peters, and Mark Hopkins. Extending a parser to distant domains using a few dozen partially annotated examples. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1190–1199. Association for Computational Linguistics, 2018. doi: 10.18653/v1/P18-1110. URL `https://www.aclweb.org/anthology/P18-1110/`.

Shafiq Joty, Hassan Sajjad, Nadir Durrani, Kamla Al-Mannai, Ahmed Abdelali, and Stephan Vogel. How to avoid unwanted pregnancies: Domain adaptation using neural network models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1259–1270, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1147. URL `https://www.aclweb.org/anthology/D15-1147`.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *NAACL*, pages 252–262, 2018.

Douwe Kiela, Alexis Conneau, Allan Jabri, and Maximilian Nickel. Learning visually grounded sentence representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 408–418, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1038. URL `https://www.aclweb.org/anthology/N18-1038`.

Joo-Kyung Kim, Young-Bum Kim, Ruhi Sarikaya, and Eric Fosler-Lussier. Cross-lingual transfer learning for pos tagging without cross-lingual resources. In *EMNLP*, pages 2832–2838, 2017a.

Sang Erik F. Tjong Kim. Introduction to the conll-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*, 2002.

Sang Erik F. Tjong Kim and Meulder Fien De. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *NAACL HLT*, 2003.

Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

Young-Bum Kim, Karl Stratos, and Dongchan Kim. Adversarial adaptation of synthetic or stale data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1297–1307, Vancouver, Canada, July 2017b. Association for Computational Linguistics. doi: 10.18653/v1/P17-1119. URL `https://www.aclweb.org/anthology/P17-1119`.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, 2014.

Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, and Saif Mohammad. Nrc-canada-2014: Detecting aspects and sentiment in customer reviews. In Preslav Nakov and Torsten Zesch, editors, *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014*, pages 437–442. The Association for Computer Linguistics, 2014. doi: 10.3115/v1/s14-2076. URL `https://doi.org/10.3115/v1/s14-2076`.

Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302, 2015.

Julie Thompson Klein and William H Newell. Advancing interdisciplinary studies. *Handbook of the undergraduate curriculum: A comprehensive guide to purposes, structures, practices, and change*, pages 393–415, 1997.

Catherine Kobus, Josep Maria Crego, and Jean Senellart. Domain control for neural machine translation. In *RANLP*, 2016.

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328, 2018.

Volodymyr Kuleshov, Shantanu Thakoor, Tingfung Lau, and Stefano Ermon. Adversarial examples for natural language classification problems. 2018.

Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. In *International conference on machine learning*, pages 1378–1387, 2016.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016a.

Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *CoRR*, abs/1611.01236, 2016b. URL `http://arxiv.org/abs/1611.01236`.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, 2001.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*, 2017.

Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*, 2019.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *NAACL HLT*, pages 260–270, 2016.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*, 2017.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. Phrase-based & neural unsupervised machine translation. In *EMNLP*, 2018.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *ArXiv*, abs/1909.11942, 2019.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553): 436, 2015.

Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. Sumbt: Slot-utterance matching for universal and scalable belief tracking. *arXiv preprint arXiv:1907.07421*, 2019.

Omer Levy, Yoav Goldberg, and Ido Dagan. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225, 2015. doi: 10.1162/tacl_a_00134. URL `https://www.aclweb.org/anthology/Q15-1016`.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. Textbugger: Generating adversarial text against real-world applications. *arXiv preprint arXiv:1812.05271*, 2018a.

Jiwei Li, Will Monroe, and Dan Jurafsky. Understanding neural networks through representation erasure. *CoRR*, abs/1612.08220, 2016. URL `http://arxiv.org/abs/1612.08220`.

Qi Li, Haibo Li, Heng Ji, Wen Wang, Jing Zheng, and Fei Huang. Joint bilingual name tagging for parallel corpora. In *CIKM '12*, pages 1727–1731, 2012.

Yitong Li, Timothy Baldwin, and Trevor Cohn. Towards robust and privacy-preserving text representations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 25–30, Melbourne, Australia, July 2018b. Association for Computational Linguistics. doi: 10.18653/v1/P18-2005. URL `https://www.aclweb.org/anthology/P18-2005`.

Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. Deep text classification can be fooled. *arXiv preprint arXiv:1704.08006*, 2017.

Nut Limsopatham and Nigel Collier. Bidirectional lstm for named entity recognition in twitter messages. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 145–152, 2016.

Bill Y. Lin, Frank Xu, Zhiyi Luo, and Kenny Zhu. Multi-channel bilstm-crf model for emerging named entity recognition in social media. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 160–165, 2017a.

T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017b.

Ying Lin, Shengqi Yang, Veselin Stoyanov, and Heng Ji. A multi-lingual multi-task architecture for low-resource sequence labeling. In *ACL*, 2018.

Bing Liu and Ian Lane. An end-to-end trainable neural network model with belief tracking for task-oriented dialog. *arXiv preprint arXiv:1708.05956*, 2017.

L. Liu, J. Shang, F. Xu, X. Ren, H. Gui, J. Peng, and J. Han. Empower sequence labeling with task-aware neural language model. In *AAAI*, 2018.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Adversarial multi-task learning for text classification. In *ACL*, 2017a.

Qi Liu, Yue Zhang, and Jiangming Liu. Learning domain representation for multi-domain sentiment classification. 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 541–550, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1050. URL `https://www.aclweb.org/anthology/N18-1050`.

Xiaodong Liu, Yelong Shen, Kevin Duh, and Jianfeng Gao. Stochastic answer networks for machine reading comprehension. *arXiv preprint arXiv:1712.03556*, 2017b.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Improving multi-task deep neural networks via knowledge distillation for natural language understanding. *arXiv preprint arXiv:1904.09482*, 2019a.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*, 2019b.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019c.

Bin Lu, Myle Ott, Claire Cardie, and Benjamin K Tsou. Multi-aspect sentiment analysis with topic models. In *2011 IEEE 11th international conference on data mining workshops*, pages 81–88. IEEE, 2011.

Gang Luo, Xiaojiang Huang, Chin-Yew Lin, and Zaiqing Nie. Joint entity recognition and disambiguation. In *EMNLP*, pages 879–888, 2015.

Minh-Thang Luong and Christopher D Manning. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 76–79, 2015.

Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. Multi-task sequence to sequence learning. In *ICLR*, 2016.

Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. Interactive attention networks for aspect-level sentiment classification. In Carles Sierra, editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 4068–4074. ijcai.org,

2017. doi: 10.24963/ijcai.2017/568. URL `https://doi.org/10.24963/ijcai.2017/568`.

Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *ACL*, pages 1064–1074, 2016.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 9:2579–2605, 2008.

Christopher D Manning, Christopher D Manning, and Hinrich Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.

Mónica Marrero, Julián Urbano, Sonia Sánchez-Cuadrado, Jorge Morato, and Juan Miguel Gómez-Berbís. Named entity recognition: Fallacies, challenges and opportunities. *Computer Standards & Interfaces*, (5):482–489, 2013.

Stephen Mayhew, Chen-Tse Tsai, and Dan Roth. Cheap translation for cross-lingual named entity recognition. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2536–2545. Association for Computational Linguistics, 2017.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6294–6305, 2017.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*, 2018.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11): 39–41, 1995. doi: 10.1145/219717.219748. URL `http://doi.acm.org/10.1145/219717.219748`.

Sewon Min, Minjoon Seo, and Hannaneh Hajishirzi. Question answering through transfer learning from large fine-grained supervision data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 510–517, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-2081. URL `https://www.aclweb.org/anthology/P17-2081`.

Marvin Minsky. A framework for representing knowledge. 1974.

Saif Mohammad. #emotional tweets. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main* *conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 246–255, Montréal, Canada, 7-8 June 2012. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/S12-1033`.

John E Moody. The effective number of parameters: An analysis of generalization and regularization in nonlinear learning systems. In *Advances in neural information processing systems*, pages 847–854, 1992.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1765–1773, 2017.

Nikola Mrkšić, Diarmuid O Séaghdha, Blaise Thomson, Milica Gašić, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. Counter-fitting word vectors to linguistic constraints. *arXiv preprint arXiv:1603.00892*, 2016a.

Nikola Mrkšić, Diarmuid O Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. Neural belief tracker: Data-driven dialogue state tracking. *arXiv preprint arXiv:1606.03777*, 2016b.

Preslav Nakov and Jörg Tiedemann. Combining word-level and character-level models for machine translation between closely-related languages. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 301–305, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/P12-2059`.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. Ms marco: A human-generated machine reading comprehension dataset. 2016.

Jian Ni and Radu Florian. Improving multilingual named entity recognition with wikipedia entity type mapping. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1275–1284, 2016.

Jian Ni, Georgiana Dinu, and Radu Florian. Weakly supervised cross-lingual named entity recognition via effective annotation and representation projection. In *ACL*, pages 1470–1480, 2017.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.441. URL `https://www.aclweb.org/anthology/2020.acl-main.441`.

Timothy Niven and Hung-Yu Kao. Probing neural network comprehension of natural language arguments. *arXiv preprint arXiv:1907.07355*, 2019.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, 2016.

Elnaz Nouri and Ehsan Hosseini-Asl. Toward scalable neural dialogue state tracking model. *arXiv preprint arXiv:1812.00899*, 2018.

Simon Ostermann, Michael Roth, Ashutosh Modi, Stefan Thater, and Manfred Pinkal. Semeval-2018 task 11: Machine comprehension using commonsense knowledge. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 747–757, 2018.

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

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958. Association for Computational Linguistics, 2017.

Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 115–124, 2005.

Nicolas Papernot, Patrick D. McDaniel, and Ian J. Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *ArXiv*, abs/1605.07277, 2016.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pages 506–519. ACM, 2017.

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

Ioannis Partalas, Cédric Lopez, Nadia Derbas, and Ruslan Kalitvianski. Learning to search for recognizing named entities in twitter. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 171–177, 2016.

Alexandre Passos, Vineet Kumar, and Andrew McCallum. Lexicon infused phrase embeddings for named entity resolution. *arXiv preprint arXiv:1404.5367*, 2014.

Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL `https://www.aclweb.org/anthology/D14-1162`.

Christian S. Perone, Roberto Silveira, and Thomas S. Paula. Evaluation of sentence embeddings in downstream and linguistic probing tasks. *ArXiv*, abs/1806.06259, 2018.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL `https://www.aclweb.org/anthology/N18-1202`.

Jason Phang, Thibault Févry, and Samuel R Bowman. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv preprint arXiv:1811.01088*, 2018.

Chen Pin-Yu, Sharma Yash, Zhang Huan, Yi Jinfeng, and Cho-Jui Hsieh. Ead: Elastic-net attacks to deep neural networks via adversarial examples. In *AAAI*, 2018.

Steven Pinker. *The language instinct: How the mind creates language*. Penguin UK, 2003.

Barbara Plank and Gertjan van Noord. Effective measures of domain similarity for parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1566–1576, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/P11-1157`.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, 2014.

Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. Semeval-2016 task 5: Aspect based sentiment analysis. In *10th International Workshop on Semantic Evaluation (SemEval 2016)*, 2016.

Danish Pruthi, Bhuwan Dhingra, and Zachary C. Lipton. Combating adversarial misspellings with robust word recognition, 2019.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*, 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8), 2019.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP*, 2016.

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*, 2018.

Prajit Ramachandran, Peter Liu, and Quoc Le. Unsupervised pretraining for sequence to sequence learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 383–391, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1039. URL `https://www.aclweb.org/anthology/D17-1039`.

Bharath Ramsundar, Steven M. Kearnes, Patrick Riley, Dale Webster, David E. Konerding, and Vijay S. Pande. Massively multitask networks for drug discovery. *ArXiv*, abs/1502.02072, 2015.

Abhinav Rastogi, Dilek Hakkani-Tür, and Larry Heck. Scalable multi-domain dialogue state tracking. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 561–568. IEEE, 2017.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. *arXiv preprint arXiv:1909.05855*, 2019.

Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155, 2009.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems*, pages 506–516, 2017.

Siva Reddy, Danqi Chen, and Christopher D Manning. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266, 2019.

Marek Rei. Semi-supervised multitask learning for sequence labeling. In *ACL*, pages 2121–2130, 2017a.

Marek Rei. Semi-supervised multitask learning for sequence labeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2121–2130, Vancouver, Canada, July 2017b. Association for Computational Linguistics. doi: 10.18653/v1/P17-1194. URL https://www.aclweb.org/anthology/P17-1194.

Marek Rei and Anders Søgaard. Zero-shot sequence labeling: Transferring knowledge from sentences to tokens. In *NAACL HLT*, pages 293–302, 2018.

Roi Reichart and Ari Rappoport. Self-training for enhancement and domain adaptation of statistical parsers trained on small datasets. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 616–623, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P07-1078.

Nils Reimers and Iryna Gurevych. Reporting score distributions makes a difference: Performance study of lstm-networks for sequence tagging. In *EMNLP*, pages 338–348, 2017.

Robert Remus. Domain adaptation using domain similarity-and domain complexity-based instance selection for cross-domain sentiment analysis. In *2012 IEEE 12th international conference on data mining workshops*, pages 717–723. IEEE, 2012.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Semantically equivalent adversarial rules for debugging nlp models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 856–865, 2018.

Matthew Richardson, Christopher JC Burges, and Erin Renshaw. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *EMNLP*, pages 193–203, 2013.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how bert works. *arXiv preprint arXiv:2002.12327*, 2020.

Sebastian Ruder and Barbara Plank. Strong baselines for neural semi-supervised learning under domain shift. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1044–1054, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1096. URL https://www.aclweb.org/anthology/P18-1096.

Sebastian Ruder, Parsa Ghaffari, and John G Breslin. Data selection strategies for multi-domain sentiment analysis. *arXiv preprint arXiv:1702.02426*, 2017a.

Sebastian Ruder, Parsa Ghaffari, and John G Breslin. Knowledge adaptation: Teaching to adapt. *arXiv preprint arXiv:1702.02052*, 2017b.

Stuart Russell and Peter Norvig. Artificial intelligence: a modern approach. 2002.

Kenji Sagae. Self-training without reranking for parser domain adaptation and its impact on semantic role labeling. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, pages 37–44, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W10-2606`.

Oana Sandu, Giuseppe Carenini, Gabriel Murray, and Raymond Ng. Domain adaptation to summarize human conversations. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, pages 16–22, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W10-2603`.

Erik F Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*, 2003.

Victor Sanh, Thomas Wolf, and Sebastian Ruder. A hierarchical multi-task approach for learning embeddings from semantic tasks. In *AAAI 2019*, volume 33, pages 6949–6956, 2019.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*, 2015.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL `https://www.aclweb.org/anthology/P16-1162`.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*, 2016.

Aliaksei Severyn and Alessandro Moschitti. UNITN: Training deep convolutional neural network for twitter sentiment classification. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 464–469, Denver, Colorado, June 2015. Association for Computational Linguistics. doi: 10.18653/v1/S15-2079. URL `https://www.aclweb.org/anthology/S15-2079`.

Pararth Shah, Dilek Hakkani-Tur, Bing Liu, and Gokhan Tur. Bootstrapping a neural conversational agent with dialogue self-play, crowdsourcing and on-line reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 41–51, 2018.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems*, pages 6833–6844, 2017.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*, pages 1929–1958, 2014.

Sandeep Subramanian, Adam Trischler, Yoshua Bengio, and Christopher J Pal. Learning general purpose distributed sentence representations via large scale multi-task learning. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=B18WgG-CZ`.

Haipeng Sun, Rui Wang, Kehai Chen, Masao Utiyama, Eiichiro Sumita, and Tiejun Zhao. An empirical study of domain adaptation for unsupervised neural machine translation. *arXiv preprint arXiv:1908.09605*, 2019a.

Kai Sun, Dian Yu, Dong Yu, and Claire Cardie. Improving machine reading comprehension with general reading strategies. *arXiv preprint arXiv:1810.13441*, 2018.

Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. Dream: A challenge data set and models for dialogue-based reading comprehension. *Transactions of the Association for Computational Linguistics*, 7:217–231, 2019b.

Jared Suttles and Nancy Ide. Distant supervision for emotion classification with discrete binary values. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 121–136. Springer, 2013.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Christian Szegedy, Wojciech Zaremba, Dumitru Erhan Ian Goodfellow Ilya Sutskever, Joan Bruna, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014.

Alon Talmor and Jonathan Berant. Multiqa: An empirical investigation of generalization and transfer in reading comprehension. *arXiv preprint arXiv:1905.13453*, 2019.

Ben Tan, Yu Zhang, Sinno Jialin Pan, and Qiang Yang. Distant domain transfer learning. In *AAAI*, 2017.

Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. Effective lstms for target-dependent sentiment classification. In Nicoletta Calzolari, Yuji Matsumoto, and Rashmi Prasad, editors, *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 3298–3307. ACL, 2016a. URL `https://www.aclweb.org/anthology/C16-1311/`.

Duyu Tang, Bing Qin, and Ting Liu. Aspect level sentiment classification with deep memory network. In Jian Su, Xavier Carreras, and Kevin Duh, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing,* *EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 214–224. The Association for Computational Linguistics, 2016b. doi: 10.18653/v1/d16-1021. URL `https://doi.org/10.18653/v1/d16-1021`.

Jörg Tiedemann. Parallel data, tools and interfaces in opus. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Ugur Dogan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may 2012. European Language Resources Association (ELRA). ISBN 978-2-9517408-7-7.

Bo-Hsiang Tseng, Sheng-Syun Shen, Hung-Yi Lee, and Lin-Shan Lee. Towards machine comprehension of spoken content: Initial toefl listening comprehension test by machine. *arXiv preprint arXiv:1608.06378*, 2016.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

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

Marlies van der Wees, Arianna Bisazza, and Christof Monz. Dynamic data selection for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1410, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1147. URL `https://www.aclweb.org/anthology/D17-1147`.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

Duy-Tin Vo and Yue Zhang. Target-dependent twitter sentiment classification with rich automatic features. In Qiang Yang and Michael J. Wooldridge, editors, *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 1347–1353. AAAI Press, 2015. URL `http://ijcai.org/Abstract/15/194`.

Pius von Däniken and Mark Cieliebak. Transfer learning and sentence level features for named entity recognition on tweets. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 166–171, 2017.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018a.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. Attention-based LSTM for aspect-level sentiment classification. In Jian Su, Xavier Carreras, and Kevin Duh, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 606–615. The Association for Computational Linguistics, 2016. doi: 10.18653/v1/d16-1058. URL https://doi.org/10.18653/v1/d16-1058.

Zhenghui Wang, Yanru Qu, Liheng Chen, Jian Shen, Weinan Zhang, Shaodian Zhang, Yimei Gao, Gen Gu, Ken Chen, and Yong Yu. Label-aware double transfer learning for cross-specialty medical named entity recognition. 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 1–15, New Orleans, Louisiana, June 2018b. Association for Computational Linguistics. doi: 10.18653/v1/N18-1001. URL https://www.aclweb.org/anthology/N18-1001.

Sean Welleck, Jason Weston, Arthur Szlam, and Kyunghyun Cho. Dialogue natural language inference. *CoRR*, abs/1811.00671, 2018. URL http://arxiv.org/abs/1811.00671.

TH Wen, D Vandyke, N Mrkšíc, M Gašíc, LM Rojas-Barahona, PH Su, S Ultes, and S Young. A network-based end-to-end trainable task-oriented dialogue system. In *15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017-Proceedings of Conference*, volume 1, pages 438–449, 2017.

Georg Wiese, Dirk Weissenborn, and Mariana Neves. Neural domain adaptation for biomedical question answering. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 281–289, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/K17-1029. URL https://www.aclweb.org/anthology/K17-1029.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. Towards universal paraphrastic sentence embeddings. *CoRR*, abs/1511.08198, 2016.

Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017.

Terry Winograd. Understanding natural language. *Cognitive psychology*, 3(1):1–191, 1972.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. Transferable multi-domain state generator for task-oriented dialogue systems. *arXiv preprint arXiv:1905.08743*, 2019.

Fangzhao Wu and Yongfeng Huang. Sentiment domain adaptation with multiple sources. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 301–310, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1029. URL https://www.aclweb.org/anthology/P16-1029.

Yingce Xia, Tao Qin, Wei Chen, Jiang Bian, Nenghai Yu, and Tie-Yan Liu. Dual supervised learning. In *ICML*, 2017.

Xiaoyu Xiang, Zhijing Jin, Di Jin, Bingning Wang, Qi Zhang, and Xuanjing Huang. Tasty burgers, soggy fries: Probing aspect robustness in aspect-based sentiment analysis. *Submitted to EMNLP*, 2020.

Yunxuan Xiao, Yanru Qu, Lin Qiu, Hao Zhou, Lei Li, Weinan Zhang, and Yong Yu. Dynamically fused graph network for multi-hop reasoning. *arXiv preprint arXiv:1905.06933*, 2019.

Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. BERT post-training for review reading comprehension and aspect-based sentiment analysis. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2324–2335. Association for Computational Linguistics, 2019a. doi: 10.18653/v1/n19-1242. URL https://doi.org/10.18653/v1/n19-1242.

Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. A failure of aspect sentiment classifiers and an adaptive re-weighting solution. *CoRR*, abs/1911.01460, 2019b. URL http://arxiv.org/abs/1911.01460.

Puyang Xu and Qi Hu. An end-to-end approach for handling unknown slot values in dialogue state tracking. *arXiv preprint arXiv:1805.01555*, 2018.

Wei Xue and Tao Li. Aspect based sentiment analysis with gated convolutional networks. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2514–2523. Association for Computational Linguistics, 2018. doi: 10.18653/v1/P18-1234. URL https://www.aclweb.org/anthology/P18-1234/.

Ya Xue, Xuejun Liao, Lawrence Carin, and Balaji Krishnapuram. Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research*, 8(Jan):35–63, 2007.

Jie Yang, Yue Zhang, and Fei Dong. Neural word segmentation with rich pretraining. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 839–849, Vancouver, Canada, July 2017a. Association for Computational Linguistics. doi: 10.18653/v1/P17-1078. URL https://www.aclweb.org/anthology/P17-1078.

Yongxin Yang and Timothy M. Hospedales. Trace norm regularised deep multi-task learning. *ArXiv*, abs/1606.04038, 2017.

Zhilin Yang, Ruslan Salakhutdinov, and William W. Cohen. Multi-task cross-lingual sequence tagging from scratch. *CoRR*, abs/1603.06270, 2016.

Zhilin Yang, Ruslan Salakhutdinov, and William W. Cohen. Transfer learning for sequence tagging with hierarchical recurrent networks. In *ICLR*, 2017b.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*, 2018.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019.

David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196, Cambridge, Massachusetts, USA, June 1995. Association for Computational Linguistics. doi: 10.3115/981658.981684. URL `https://www.aclweb.org/anthology/P95-1026`.

David Yarowsky, Grace Ngai, and Richard Wicentowski. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the first international conference on Human language technology research*, pages 1–8, 2001.

Michihiro Yasunaga, Jungo Kasai, and Dragomir Radev. Robust multilingual part-of-speech tagging via adversarial training. In *NAACL HLT*, pages 976–986, 2018.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014.

Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3712–3722, 2018.

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. Word-level textual adversarial attacking as combinatorial optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6066–6080, Online, July 2020. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/2020.acl-main.540`.

Daniel et al. Zeman. Conll 2017 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, 2017.

Boliang Zhang, Xiaoman Pan, Tianlu Wang, Ashish Vaswani, Heng Ji, Kevin Knight, and Daniel Marcu. Name tagging for low-resource incident languages based on expectation-driven learning. In *NAACL HLT*, pages 249–259, 2016.

Chen Zhang, Qiuchi Li, and Dawei Song. Aspect-based sentiment classification with aspect-specific graph convolutional networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4567–4577. Association for Computational Linguistics, 2019a. doi: 10.18653/v1/D19-1464. URL `https://doi.org/10.18653/v1/D19-1464`.

Jian-Guo Zhang, Kazuma Hashimoto, Chien-Sheng Wu, Yao Wan, Philip S Yu, Richard Socher, and Caiming Xiong. Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking. *arXiv preprint arXiv:1910.03544*, 2019b.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657, 2015.

Yuan Zhang, Jason Baldridge, and Luheng He. PAWS: paraphrase adversaries from word scrambling. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1298–1308. Association for Computational Linguistics, 2019c. doi: 10.18653/v1/n19-1131. URL `https://doi.org/10.18653/v1/n19-1131`.

Zhengli Zhao, Dheeru Dua, and Sameer Singh. Generating natural adversarial examples. *arXiv preprint arXiv:1710.11342*, 2017.

Victor Zhong, Caiming Xiong, and Richard Socher. Global-locally self-attentive dialogue state tracker. *arXiv preprint arXiv:1805.09655*, 2018.

Guangyou Zhou, Zhiwen Xie, Jimmy Xiangji Huang, and Tingting He. Bi-transferring deep neural networks for domain adaptation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 322–332, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1031. URL `https://www.aclweb.org/anthology/P16-1031`.

Joey Tianyi Zhou, Hao Zhang, Di Jin, Hongyuan Zhu, Meng Fang, Rick Siow Mong Goh, and Kenneth Kwok. Dual adversarial neural transfer for low-resource named entity recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3461–3471, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1336. URL `https://www.aclweb.org/anthology/P19-1336`.

Li Zhou and Kevin Small. Multi-domain dialogue state tracking as dynamic knowledge graph enhanced question answering. *arXiv preprint arXiv:1911.06192*, 2019.

Zhi-Hua Zhou and Ming Li. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on knowledge and Data Engineering*, 17(11):1529–1541, 2005.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. Freelb: Enhanced adversarial training for natural language understanding. In *International Conference on Learning Representations*, 2019.

Fuzhen Zhuang, Xiaohu Cheng, Ping Luo, Sinno Jialin Pan, and Qing He. Supervised representation learning: Transfer learning with deep autoencoders. In *IJCAI*, 2015.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1163. URL `https://www.aclweb.org/anthology/D16-1163`.

Andrej Zukov Gregoric, Yoram Bachrach, and Sam Coope. Named entity recognition with parallel recurrent neural networks. In *ACL*, pages 69–74, 2018.