Author/s:
Li, Yitong

Title:
Towards Robust Representation of Natural Language Processing

Date:
2019

Persistent Link:
https://hdl.handle.net/11343/240992

Towards Robust Representation of Natural Language Processing

Yitong Li

School of Computing and Information Systems

The University of Melbourne

Submitted in total fulfilment for the degree of

Doctor of Philosophy

June 2020

Abstract

There are many challenges in building robust natural language applications. Machine learning based methods require large volumes of annotated text data, and variations over text can lead to problems, namely: (1) language can be highly variable and expressed with different variations, such as lexical and syntactic. Robust models should be able to handle these variations. (2) A text corpus is heterogeneous, often making language systems domain-brittle. Solutions for domain adaptation and training with corpora comprised of multiple domains are required for language applications in the real world. (3) Many language applications tend to be biased to the demographic of the authors of documents the system is trained on, and lack model fairness. Demographic bias also causes privacy issues when a model is made available to others.

In this thesis, I aim to build robust natural language models to tackle these problems, focusing on deep learning approaches which have shown great success in language processing via representation learning. I pose three basic research questions: how to learn representations that are robust to language variation, robust to domain variation, and robust to demographic variables.

Each of these research questions is tackled using different approaches, including data augmentation, adversarial learning, and variational inference. For learning robust representations to language variation, I study lexical variation and syntactic variation. To be specific, a regularisation method is proposed to tackle lexical variation, and a data augmentation method is proposed to build robust models, using a range of language generation methods from both linguistic and machine learning perspectives. For domain robustness, I focus on multi-domain learning and investigate domain supervised and unsupervised learning, where domain labels

may or may not be available. Two types of models are proposed, via adversarial learning and latent domain gating, to build robust models for heterogeneous text. For robustness to demographics, I show that demographic bias in the training corpus leads to model fairness problems with respect to the demographic of the authors, as well as privacy issues under inference attacks. Adversarial learning is adopted to mitigate bias in representation learning, to improve model fairness and privacy-preservation.

To demonstrate the proposed approaches, a range of tasks are considered, including text classification and POS tagging. To evaluate the generalisation and robustness, both in-domain and out-of-domain experiments are conducted with two classes of language tasks: text classification and part-of-speech tagging. For multi-domain learning, multi-domain language identification and multi-domain sentiment classification are conducted, and I simulate domain supervised learning and domain unsupervised learning to evaluate domain robustness. I evaluate model fairness with different demographic attributes and apply inference attacks to test model privacy. The experiments show the advantages and the robustness of the proposed methods.

Finally, I discuss the relations between the different forms of robustness, including their commonalities and differences. The limitations of this thesis are discussed in detail, including potential methods to address these shortcomings in future work, and potential opportunities to generalise the proposed methods to other language tasks. Above all, these methods of learning robust representations can contribute towards progress in natural language processing.

Declaration

This page declares that:

1. the thesis comprises only my original work towards the Ph.D except where indicated in the preface;
2. due acknowledgement has been made in the text to all other material used; and
3. the thesis is fewer than 100,000 in length, exclusive of tables, maps, bibliographies and appendices.

Yitong Li

June 2020

Preface

All the papers presented in Chapter 3 have been published, in collaboration with my Ph.D supervisors, Tim and Trevor. I claim all the work is completed originally during my candidature. This applies to:

- Li, Yitong, Trevor Cohn and Timothy Baldwin (2016) Learning Robust Representations of Text, In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, Austin, Texas, USA, pp. 1979–1985.

- Li, Yitong, Trevor Cohn and Timothy Baldwin (2017) Robust Training under Linguistic Adversity, In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers (EACL 2017)*, Valencia, Spain, pp. 21–27.

- Li, Yitong, Trevor Cohn and Timothy Baldwin (2017) BIBI System Description: Building with CNNs and Breaking with Deep Reinforcement Learning, In *Proceedings of the EMNLP 2017 Workshop on Building Linguistically Generalizable NLP Systems (BLGNLP 2017)*, Copenhagen, Denmark, pp. 27–32.

- Li, Yitong, Timothy Baldwin and Trevor Cohn (2018) What's in a Domain? Learning Domain-Robust Text Representations Using Adversarial Training, In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (NAACL HLT 2018, Short Papers)*, New Orleans, USA, pp. 474–479.

- Li, Yitong, Timothy Baldwin and Trevor Cohn (2019) Semi-supervised Stochastic Multi-Domain Learning using Variational Inference, In *Proceedings of the 57th Annual*

Meeting of the Association for Computational Linguistics (ACL 2019), Florence, Italy, pp. 1923–1934.

- Li, Yitong, Timothy Baldwin and Trevor Cohn (2018) Towards Robust and Privacy-preserving Text Representations, In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018, Volume 2: Short Papers)*, Melbourne, Australia, pp. 25–30.

Acknowledgements

First of all, I would like to thank my beloved parents for encouraging me to pursue a Ph.D degree overseas for these years. It was quite an experience, and I was able to make it because of you, always trusting and supporting me both physically and mentally. Thank you!

I particularly want to express my gratitude and respect to my supervisors, Prof. Tim Baldwin and Prof. Trevor Cohn, who both led and encouraged me during my studies. Tim is very knowledgeable, always illuminating the research direction for me. Trevor is full of interesting ideas that have been a constant inspiration for my research. I am really appreciative of being able to become their student, and I learned a lot under their supervision. I also say thanks to the teaching staff in the school, especially to A/Prof. Benjamin Rubinstein, Prof. Karin Verspoor and Prof. Rao Kotagiri, for sharing their teaching knowledge and research experience, which really helped me to polish my skills. I also would like to thank the administrators of the school, Julie and Rhonda, for always being so patient and helpful.

I would also like thank all my colleagues in the NLP group – Fei Liu, Elly Liang, Julian Brooke, Meng Fang, Yuan Li, Qingyu Chen, Doris Hoogeveen, Michal Lukasik, Long Duong, Oliver Adams, Afshin Rahimi, Ned Letcher, Ekaterina Vylomova, Cong Duy Vu Hoang, Aili Shen, Shivashankar Subramanian, Nitika Mathur, Shraey Bhatia, Dat Nguyen, Minghao Wu, Anirudh Joshi, Haonan Li, Dalin Wang, Brian Hur, Tatsuya Aoki, Zenan Zhai, Yuxia Wang, Biaoyan Fang, Chunhua Liu, Xudong Han, Jinghui Liu, Daniel Beck, Jey Han Lau, Fajri Koto – and colleagues in the School – Wenxi Wang, Weihao Cheng, Bowen Zhou, Chenhao Qu, Xunyun Liu, Yiqing Zhang, Zeyi Wen, Jianzhong Qi, Ang Li, Yunxiang Zhao, Lingjuan Lyu, Alex Vedernikov, Ziad Al Bkhetan – for having lunch, discussing ideas, sharing happiness, and everything else every day. I really had a great time with you.

I would also like to thank my friends in Melbourne – Ye Wang, Hui Zheng, Jiahua Zhu, Wei Wu, Fan Liu, Limeng Zhang, Alice Kou and Geordie Zhang, Kate Wang, Lili Sun, and Hua Wang, Jinli Cao and Yanchun Zhang, Jiao Yin and Mingshang – for making my life super colorful with mountain white, ocean blue, and flower red. Because of you, I felt as warm as if I were at home. Specially, I express my appreciation to Ye, who inspires and rebuilds me, to have a great time. Also, I would like to thank my friends, Tianfan and Qi, who cheered me up every day.

I acknowledge the Australian Research Council for supporting my research, and Google and Amazon for providing funding for conference travel.

Lastly, I appreciate every person I met during my Ph.D and my life: you have all made me who I am now.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Natural Language Processing (NLP), the understanding and generation of human language, is an important subfield of Artificial Intelligence (AI). There are many challenges for building natural language tools, as understanding natural language requires semantic understanding to capture the underlying structure and composition of language. To tackle these problems, researchers have proposed many approaches, from rule-based to statistical methods, among which Machine Learning (ML) and Deep Learning (DL) in particular have been highly successful.

Many factors have contributed to the success of DL, including data volume, better optimization algorithms, and the increasing power of computing resources. But two factors are particularly important: data quality and model complexity. The community has developed an increasingly better understanding of the impact of the quantity and quality of data on trained models. However, there are many questions related to data. For example, dealing with text data can be challenging, and applying machine learning models over such data is challenging from both training and test perspectives. Text data is often heterogeneous, requiring models to perform well across several different types of text, often referred to as "domains". Further, models are prone to bias towards majority patterns in training, such as over-represented author demographics (Hardt et al., 2016). Another contributing factor is the depth of models. Technically, deep learning models introduce much higher model complexity to learn a high-level data representation, compared with other machine learning models, and

this can cause models to become over-parameterised and prone to overfitting. In order to perform language understanding, training algorithms and model architectures are required to be robust to all these problems.

Text is highly variable, that is, a sentence can be expressed in many different ways without changing the meaning. For example, we have a sentence *The cat sat on the mat*. We can modify the sentence only on the lexical level, like *The cat sat on the rug*, where only the underlined word is replaced with its synonym but the meaning of the sentence remains largely unchanged. The sentence can also be changed at the syntactic level, for a instance to *On the mat the cat sat*, where the phrase order of the sentence has changed but the semantics are largely unchanged. These modifications are grammatical and a robust NLP model for a task such as sentiment analysis, should produce the same output regardless of the variations at these level. On the other hand, not all the text is so clean or formal. Social media data can be more informal and variable in terms of language usage. For example, a user could caption a picture simply as *Kittty on mat !!!!! LOL!!!*, with misspellings, nonstandard punctuation, syntactic omissions, and abbreviations. Such language phenomena are commonly used in digital communication and daily chat, which are usually more informal in nature and do not follow prescriptive grammar of English, but can still be readily understood by people. Relatedly, text can be corrupted during copying or transcription.

Machine learning models or deep models can be brittle to such inputs, when applied in real-world scenarios, during inference. Lexical token substitutions, misspellings, lengthening, and other lexical transformation, can cause out-of-vocabulary issues for word-level models, both for traditional and deep learning approaches. For example, a review *I am mad for the movie* usually expresses positive sentiment towards the movie, however, it can confuse many machine learning systems because they associate *mad* with negative sentiment. Syntactic transformations can also be problematic, leading to performance instability as they have not been seen in training. For example, the review *It is impossible not to like such a wicked movie* which consists of many negative words, can fool a model, as it is hard to understand the underlying logic of the sentence. However, humans are remarkably resilient to these

examples. As such, algorithms for training robust models are required, such that models can be robust to these text variations.

Besides the variability of text, a second challenge is that deep models are brittle to domain shifts. For example, considering the case of training a model based on *book* reviews and then applying it to a new domain such as *radio* reviews. The target domain usually follows a different data distribution, but shares some knowledge with the source domain. This happens particularly when the target domain lacks training data, i.e. in low-resource settings, or because we overfit to the target domain. The model should be capable of learning relevant information from the source domain and forgetting irrelevant information. In real-world settings it is inevitable that our model will be applied to out-of-domain data instances, and we would like it to be robust against arbitrary inputs. Therefore, in this thesis, I assume no knowledge of the target domain, and focus on building domain-invariant off-the-shelf models that are robust to a variety of inputs. To address this challenge, we require deep models that are robust to all manner of domain changes.

Text data is heterogeneous, consisting of latent information or domains in terms of where the data was sourced from, who wrote it, and what its content is. Large corpora often comprise text from many different domains. However, given different domain contexts, the same word can mean different things. For examples, one review may say *This pillow is soft and comfortable* and another review may say *Seems to me the table surface is quite soft*. Both reviews mention the word *soft*, both in the context of *Homeware*. However, the first review describes a *pillow*, where *soft* is generally a good attribute, while the second review is complaining that the materials of the table are deficient in hardness. Training over such a corpus comprised of different domains is referred to as multi-domain learning. This also requires deep model to be robust to the comprised domains during training, instead of just memorising general feature weights.

Furthermore, corpora sometimes contain text with varying authorship, resulting in heterogeneous datasets. Authorship traits like gender, age, and ethnicity can cause fairness issues, which is a third factor, and machine learning algorithms should be robust to them. Turing proposed that computer science should be concerned with demonstrating intelligent

behavior (Turing, 1950). Therefore, we should attempt to build state-of-the-art deep models to be as fair as human beings are. Fairness is a well-accepted concept whereby individuals should be treated equally by law and society, regardless of their background of any personal, physical, or mental trait. A recent study by Rudinger et al. (2018) confirmed gender bias in coreference systems using the Winogender schema, in which a pair of sentences differs only in pronoun gender. For example, given the sentence *The surgeon couldn't operate on the patient: it was [his/her] son!*, the system should resolve *his* and *her* as coreferent with *The surgeon*, but in the second case, due to gender bias associated with the word *surgeon*, systems tend to resolve *her* to the *patient*. Another example is related to word embeddings, where *man* is more connected with *doctor*, while *woman* is more related with *nurse* (Bolukbasi et al., 2016). Note that there are several definitions of fairness in the community, and this remains as an open question (Barocas et al., 2019; Chouldechova and Diaz, 2019). In this thesis, I focus on the fairness in the form of the bias problem relating to demographic authors.

While such bias is likely caused by data bias in the training data, machine learning models tend to amplify such biases. For example, Zhao et al. (2017) showed that the activity of cooking is much more likely to be associated with females than males in the training set of a visual semantic role labeling dataset, and a trained model further amplifies the disparity at test time. A fair model should not learn that bias. From another point of view, statistical methods do a good job of modelling the majority class, because the majority contributes most to the training objective. However for the same reason they could easily ignore minority voices. These factors lead to the idea of fairness in machine learning, including learning unbiased representations, and supporting minority voices.

The concept of fairness is closely related to privacy, and thus solving one might have benefits for the other. This is very much a fledgling area of ML, which we discuss in more detail in Section 2.4. Nowadays, due to the development of the Internet, individuals and companies can readily share data. However, malicious attackers can reconstruct the data associated with individual users from models trained over their data, via inference attacks (Hamm, 2017). This can take the form of the source data associated with a given representation of demographic variables, like gender, age, nationality, or behaviour in the

real-world. This can be worse if the published representation has learned to amplify these demographic variables. These privacy problem are quite serious if we would like to apply AI systems broadly, and to be able to distribute trained models without compromising the privacy of individual users. Learning privacy-preserving models also requires robust and unbiased model, emphasising the importance of learning robust, fair text representations.

1.1 Scope of the Thesis

In this thesis, I aim to build machine learning systems that can learn robust representations of natural language. There are many languages in the world, each of which has its own characteristics. I will focus on *English* for most of our experiments due to data availability, but will look in part at applying our multi-domain adaptation techniques to a multi-lingual task, namely language identification. Most of the proposed methods are language-independent and applicable to any language, given suitable resources.

I mostly focus on tasks involving short documents or sentence-level classification tasks. This includes sentiment analysis and language identification. In a few cases I consider sequence classification tasks. I use document tasks to evaluate the models, but the underlying algorithms can be applied to other tasks, such as machine translation (Bahdanau et al., 2015) and machine reading (Cheng et al., 2016).

In term of models, I focus primarily on convolutional neural networks, which have proven to be efficient and effective across a range of tasks. I use word-level inputs in most cases. To exemplify our ideas, I will experiment with different scenarios, including single-domain, multi-domain, and in- and out-of domain cases to demonstrate the robustness of our proposed models. When applying out-of-domain evaluations, I assume no prior knowledge of the target domains. For sequence classification tasks, I apply recurrent neural networks in the form of long-short-term-memories. Nevertheless, I could easily extend our methods to other model architectures.

1.2 Research Questions

The research questions I address in this thesis are the followings:

1. *To what extent can we learn text representations that are robust to the variability of language?*

   We address the question by asking the following sub-questions. Firstly, to what extent can we learn text representation that are robust to lexical variability? Secondly, we ask how we can automatically generate variations given a sentence, and to what extent our deep models can be robust to these examples, as well as other human generated examples? Building on this question, we ask to what extent we can learn robust representation of text by training over these generated variations of text?

2. *To what extent can we learn domain-robust representations?*

   For inference, we ask: If we assume no knowledge of the target domains, to what extent can we learn robust text representations when applied to the target domains?

   For training, we ask: To what extent can we learn text representations that are robust to multiple domains during training? Additionally, to what extent can we learn robust text representations if we have little or no knowledge of the training domains?

3. *To what extent can we learn unbiased representations towards demographic variables and improve model fairness?*

   As a secondary measure, how does this help deep models learn privacy-preserving representations?

These research questions are organised by the relatedness of the robust representation learning, and will be revisited in Chapter 2, where we will return to discuss each question in greater detail and reveal the relations between them.

1.3 Contributions

The remainder of this thesis is organised as follows: I first conduct a literature review in Chapter 2, and introduce related methods and techniques. I then address the three research questions in Chapter 3, which is comprised of six published research papers. The last part of this thesis, Chapter 4, will contextualise and reflect the contributions of the thesis, and discuss possible future work.

The contributions can be summarised as follows, according to our three research questions:

Robustness to Language Variation For the first research question, learning robust text representations against language variation, I answer it through a series of papers.

The first one answers how to build robust models through better regularisation, in the form of a publication at EMNLP 2016 (see Section 3.1). In this work, I assume that when dealing with short texts such as Twitter posts, it is common to see unknown words due to typos, abbreviations, and sociolinguistic markers of different types. Such phenomena introduce unknown tokens into the word-level system. To simulate this, I apply word-dropout to each document. In order to handle these kinds of inputs, I propose a robust regularisation method by introducing a method that learns representations which are less sensitive to perturbations in the data. This is equivalent to minimising the magnitude of the partial derivatives of the output with respect to the inputs. I demonstrate that our method is robust over both synthetic data and in a cross-domain setup.

To investigate generating text with greater variability, I propose methods from both machine learning and linguistic points of view.

Firstly, from the linguistic point of view, Section 3.2 presentes a paper from EACL 2017, which considers two linguistic approaches to generating variations of text: syntactic and semantic. The generated text is used to augment the training data. I demonstrate that our proposed methods can deliver better robustness in a deep learning model.

Secondly, I address the problem from a machine learning point of view, as presented in Section 3.3, based on a published paper at the BLGNLP workshop 2017. This workshop

involves a "build it break it" competition based on a text classification task, where breakers try to generate text that can "break" builders' systems, while builders try to train models robust to breakers' examples. In this work, I played both roles, as a builder and a breaker. On the breaker side, I take inspiration from adversarial learning on vision (Szegedy et al., 2014), parameterising the generation task as an optimisation problem. Specifically, I jointly minimise the cross-entropy loss with the wrong label and the edit-distance between the original and generated sentences. Unlike the vision setting, text data takes the form of discrete tokens, and thus cannot be easily tackled by continuous methods. To handle the discrete optimisation, I applied reinforcement learning. For the builder part, I retrained a convolution model with our generated examples, based on both phrase-level and sentence-level adversarial examples.

Robustness to Domain The second research question of learning domain-robust representations is addressed using two approaches. In Section 3.4, I present a method published at NAACL 2018, using adversarial learning to address multi-domain learning. In this work, methods are proposed to model domains, based on a domain-adversarial and domain-generative supervision. The idea is to learn robust domain-invariant representations by obstructing the domain information during training. Similarly, domain-specific representation is encouraged to learn domain knowledge during training.

Section 3.5 presents a method published at ACL 2019 on robust multi-domain training. This work is inspired by variational inference. We proposed stochastic domain adaptation, to study the scenario where domain information is incomplete, i.e. we only have the domain knowledge of a few training instances, or none of them in extreme cases. We refer to this as domain semi-supervised and unsupervised multi-domain adaptation, respectively. I model the discriminative learning problem by both discrete and continuous domain identifiers, which learn a joint distribution of domain parameterised variational distributions. I parameterise the prior using a range of distributions and solve the intractable optimisation problem using variational inference.

To evaluate the domain-robustness of the representation learned during the inference, I perform cross-domain evaluation in Sections 3.1 and 3.2. I perform real-world out-of-domain evaluation in Sections 3.4 and 3.5, including seven language identification benchmarks, and Amazon multi-domain sentiment data.

Robustness, Fairness, and Privacy For the third research question of fairness and privacy, I answer it in Section 3.6, based on work published at ACL 2018. I again adapt adversarial learning to force the model to obstruct the bias signals. On the tasks of POS-tagging and sentiment classification, I demonstrate that models trained with our proposed methods are less biased in their predictions based on texts authored by individuals of different gender, age, and race. Additionally, the privacy-perservation of the learned model is empirically evaluated under inference attacks.

To summarise, we propose a range of methods to learn robust representations of text under different scenarios, including variability of text, multi-domain learning, domain changing, and how this can be extended to encourage fairness and privacy-preserving representations. To achieve this, we take inspiration from state-of-the-art methods, such as adversarial neural networks, and latent variable neural networks, such as variational autoencoder. Based on a range of different tasks and datasets, our experimental results demonstrate that our methods are robust, fair and high performance.

Chapter 2

Background

In this chapter, I review the research literature on the following topics: robustness of machine learning models to language variation, robustness to domain, robustness to demography of author, and fairness and privacy in machine learning. This chapter is purely a review of the literature and I will return to discuss the relations with my research contributions in Chapter 4.

2.1 Representation Learning of Text

I provide a basic overview of deep learning for text applications, as deep learning is used extensively in this thesis Deep learning has been applied to many NLP applications, from text classification (Kim, 2014), POS tagging (Huang et al., 2015), and semantic parsing (Yih et al., 2014) to machine translation (Bahdanau et al., 2015), question answering (Xiong et al., 2017), and natural language understanding (Devlin et al., 2019).

Generally, building a neural model for text requires the following steps. Let us consider an example: *A cat sat on the mat* . First, a pre-processing step, including tokenisation and case normalisation, is responsible for cleaning the raw text dataset into tokens. Now the processed sentence is *a cat sat on the mat* . In word-based models, a vocabulary set is fixed based on the training data. Tokens with low frequency are filtered out and replaced with the UNK token.

Second, a lookup table maps each token id into a word embedding (Mikolov et al., 2013). A word embedding is a distributed representation (i.e. a vector of real numbers) capturing word semantics. A range of word embeddings pretraining approaches have been proposed, which are reviewed in the Sections 2.1.1 and 2.1.2.

Thirdly, a neural architecture is applied over the word embeddings. Several neural architectures have been proposed, however, there are two main classes for text applications: Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). These neural models generate a hidden representation of the sentence, which is then processed by output layer to produce a prediction, such as softmax for classification probabilities.

To learn the best parameters, the training process is formalised as an optimisation problem, for example, minimising the cross-entropy loss ($\mathscr{H}$) between the predictions and the true labels. This can be done by many optimisation algorithms, like Stochastic Gradient Descent (SGD) (Bottou, 2010) or its variants (Kingma and Ba, 2015). In the remainder of this section, I review some of the key points in this learning procedure, including word embeddings, pretraining methods, regularisation methods, and the CNN and RNN models.

2.1.1 Word Embeddings

Word embeddings, or word representations, are a distributed semantic representation of the lexicon, learned by an unsupervised learning method. The word embedding has become a basic concept in neural models for text.

Let us first look at word embeddings in detail. From the perspective of machine learning, conventionally text is represented as a one-hot vector or bag-of-words vector, where each word is a discrete variable in a vocabulary-sized space. For example, let us return to our earlier example of *A cat sat on the mat*. After pre-processing the dataset, we map each word to its corresponding id, such as $\Phi(cat) \rightarrow \mathrm{id}_{cat}$. And the word is encoded using a "one-hot" vector $\mathbb{O}^{\mathrm{cat}} \in \{0,1\}^n$, defined as

$$\mathbb{O}_i^{\mathrm{cat}} = \begin{cases} 1 & i = \mathrm{id}_{cat} \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

where n is the vocabulary size. Clearly, the dimensionality of $\mathbb{O}$ increases linearly with vocabulary size n.

Different from one-hot vectors, word embeddings encode each word with a continuous vector with a pre-defined fixed dimension size $d \ll n$, to represent its semantics, for example

$$\mathbf{w}^{cat} \in \mathbb{R}^d. \tag{2.2}$$

2.1.1.1 Distributional Semantics

One way of learning word representations which capture their semantics is via distributional semantics. Distributional semantics is based on the hypothesis that words that occur in similar contexts tend to share similar meanings (Harris, 1954). Distributional semantic methods like latent semantic analysis (Deerwester et al., 1990) take a document-word co-occurrence matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$, where m is the total number of collected documents, and reduce it into a lower-dimensional representation $M' \in \mathbb{R}^{m \times k}$, where $k \ll n$.

Word embeddings are one way of capturing the distributional semantics of words in a dense space of dimension $d \ll n$, where the distance between $\mathbf{w}^{cat}$ to $\mathbf{w}^{dog}$ is small. Some dimensions of word embeddings can be aligned with linguistic features (Tsvetkov et al., 2015). I present more details, as well as other advantages, in Section 2.1.2.2.

2.1.2 Pretraining Approaches

One approach of training word embeddings is treating them as parameters of the model, trained end-to-end with the objective function. However, pre-trained word embeddings, usually learned in an unsupervised fashion over large corpora, have shown consistent improvement over a range of NLP tasks (Dauphin et al., 2017; Lai et al., 2015). Many approaches have been proposed for learning pre-trained word embeddings, and I start with WORD2VEC (Mikolov et al., 2013).

2.1.2.1 Word2Vec

Word2Vec uses either of two model architectures to learn distributed representations of words: continuous bag-of-words (CBOW) or continuous skip-gram. In the CBOW architecture, the model is trained to predict the target word from a window of surrounding context words. Formally, the objective of CBOW is to maximise log probability

$$\frac{1}{T}\sum_{t=1}^{T}\log p(\mathbf{w}_t|\mathbf{w}_{context}), \tag{2.3}$$

where, $\mathbf{w}_{context}$ is the average of the context word embeddings

$$\mathbf{w}_{context} = \sum_{-c\leq j\leq c, j\neq 0}\mathbf{w}_{t+j}, \tag{2.4}$$

and here c is the size the context window. Under the bag-of-words assumption, the order of context words does not influence prediction. In contrast, in the continuous skip-gram architecture, the model uses the current word to predict the surrounding window of context words

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(\mathbf{w}_{t+j}|\mathbf{w}_t). \tag{2.5}$$

Despite the similarity in model formulation, according to Mikolov et al. (2013), CBOW is faster while skip-gram does a better job for infrequent words by sub-sampling frequent words.

Building on this research, Pennington et al. (2014) proposed Global Vectors for Word Representation (GLOVE), which is similar in terms of training methods but considers global statistics of the corpus' word occurrence. They assumed that word vector learning should consider ratios of co-occurrence probabilities rather than the probabilities themselves. And thus, they minimised difference between the ratio of embeddings of each pair of words and the log word-word co-occurrence counts of them.

By maximising the likelihood of Equation 2.3 and Equation 2.5 over large corpora, the algorithm generates an embedding $\mathbf{w}$ for each word. A downstream model can initialise the

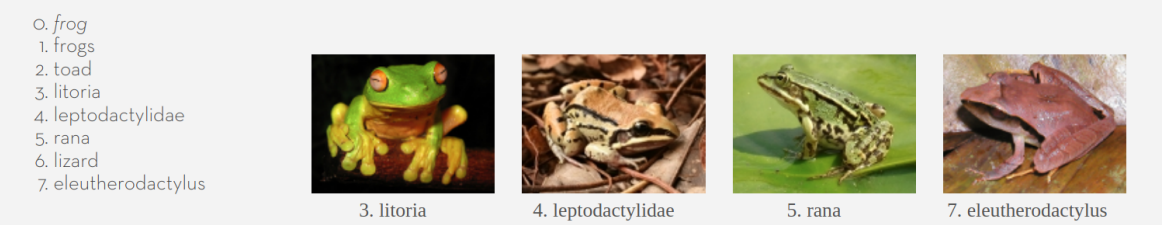


Figure 2.1: The nearest neighbours of *frog* in GloVe. On the right, figures illustrate some obscure terms of the candidates, which are still relevant words with *frog*. Figures are originally from the GloVe website https://nlp.stanford.edu/projects/glove/.

embeddings using these pre-trained embeddings, keeping them either fixed or updated during training.

2.1.2.2 Advantages of Word Embeddings

There are many advantages to learning word semantics via a distributed semantic representation.

One direct advantage is that it can reduce the model complexity for downstream machine learning algorithms, reducing overfitting, as well as improving the model robustness.

From the perspective of computational linguistics, in the Euclidean space of word embeddings, words with similar semantics are placed closer. For example, *dog* and *cat* have close distributional semantics in a corpus, and their word embeddings are close. This is because the left and right contexts of *dog* and *cat* are similar, and in many cases one can be replaced by the other without substantially changing the overall sentence semantics (but clearly changing the denotational meaning). This can be also examined by checking the nearest neighbours. Figure 2.1 gives a example of the nearest neighbours of *frog*. Some relevant words even lie outside an average human's vocabulary but can be identified through the word embedding training.

Second, word embeddings can capture linear substructures between different concepts. For example, Figure 2.2 illustrates some kinds of linear relationships between groups of concepts in Word2Vec. A famous example of this is the following:

$$\mathbf{w}^{king} - \mathbf{w}^{man} + \mathbf{w}^{woman} \approx \mathbf{w}^{queen}. \tag{2.6}$$

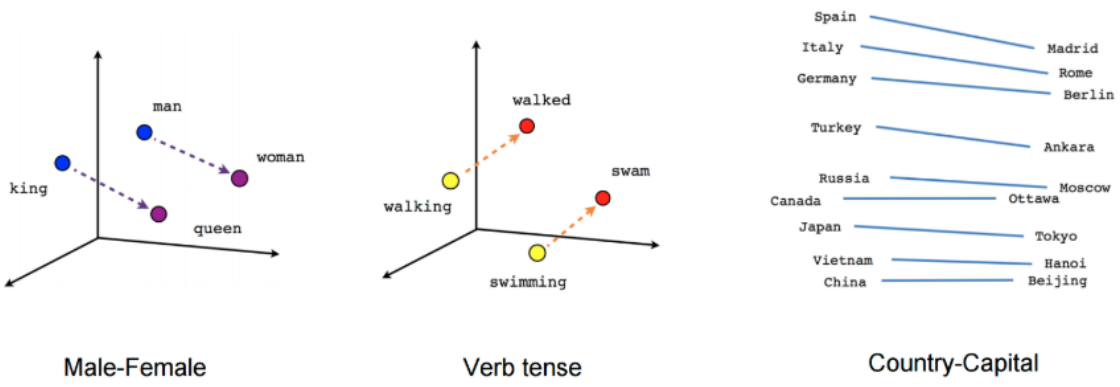


Figure 2.2: Illustration of some linear substructures discovered by Word2Vec.

This holds due to the likelihood ratio $\frac{p(\mathbf{w}^{king}|\mathbf{w}^{king}_{context})}{p(\mathbf{w}^{queen}|\mathbf{w}^{queen}_{context})}$ (using CBOW from Equation 2.3) being equal to the difference in contexts, $\mathbf{w}^{king}_{context} - \mathbf{w}^{queen}_{context} \approx \mathbf{w}^{man} - \mathbf{w}^{woman}$, which is true in most context. Similarly for different tenses of a verb, the relations holds:

$$\mathbf{w}^{walked} - \mathbf{w}^{walking} \approx \mathbf{w}^{swam} - \mathbf{w}^{swimming}. \tag{2.7}$$

There are also some linear relations, like country vs capital (also shown in Figure 2.2), city vs zip code, comparative vs superlative, etc (Pennington et al., 2014).

Third, from a machine learning point of view, distributed semantics support greater flexibility in terms of the capacity of the learned representation. One can select the dimensionality depending on the requirements of the downstream task and model, achieving a trade-off between performance and time. For example, smaller size, eg. $d \approx 100$, is good for training efficiency and reducing overfitting, whereas a larger size, such as $d \approx 1000$, can give better model performance over a larger pre-training corpus.

2.1.2.3 Improving Word2Vec

Word2Vec has achieved great success as a semantic representation, however, there are still some problems with the original method. Various approaches have been proposed to improve the method.

The first problem is that Word2Vec can not distinguish sentiment of words, which is very useful for downstream tasks like sentiment classification (Pang and Lee, 2008). For example, the word embeddings of *happy* and *sad* are close, $\mathbf{w}^{happy} \approx \mathbf{w}^{sad}$, as they have similar distributional contexts, despite their very different sentiments. Therefore, if one replaces *good* with *bad* in a sentence, a sentiment model using pre-trained Word2Vec embeddings is unable to change its predictions. One idea to tackle this is to encode antonymy and synonymy into the word embedding during the training. Mrkšić et al. (2016) proposed a COUNTER-FITTING word embedding approach, by injecting antonymy and synonymy constraints into the model as a post-processing step, to improve the robustness of word representations in terms of sentiment. Their training approach pushes antonymous word vectors away from each other and synonymous word pairs closer together, based on the cosine similarity of antonymous or synonymous pairs taken from semantic lexicons. Various other approaches have been proposed to incorporate language resources into word embeddings training (Liu et al., 2015a; Ono et al., 2015). Generally speaking, approaches to tackle this issue have been based on constraining or fine-tuning the learning based on external resources.

The second problem is context independence. Word2Vec learns a single word representation aggregated across all contexts of occurrence of a given word, which is limiting for polysemous words. For example, consider the two sentences: *A child runs on the street* and *We repeat the experiment using 5 runs*. Obviously, these *run*s have different semantics given these two contexts, and thus they should have different semantic representations, however in Word2Vec they share the same embedding. To tackle this problem, one idea is to jointly learn word embeddings with language models. Peters et al. (2018) proposed Embeddings from Language Models (ELMO), where contextualised word embeddings are derived from the internal states of a deep bidirectional language model trained over a large corpus. The language model is parameterised by a multi-layer RNN (Jozefowicz et al., 2016). The word embedding is output by combining the forward and backward representations from all layers. More recently, Devlin et al. (2019) proposed Bidirectional Encoder Representations from Transformers (BERT) where the language model is trained by transformers (Vaswani et al., 2017), which attends to both the left and right contexts, is much deeper than ELMo (and also

based on multi-headed attention), and leads to better performance over a range of NLP tasks. Different from BERT pretraining using an auto-encoder language model, Yang et al. (2019) proposed XLNET, which uses a generalized auto-regressive pretraining language model based on Transformer-XL (Dai et al., 2019b). Nowadays, such pre-trained embeddings, including Word2Vec, have achieved great success (Devlin et al., 2019; Liu et al., 2019; Peters et al., 2018; Yang et al., 2019) and become the de facto standard for NLP.

A third problem is that Word2Vec introduces bias, such as gender bias, in the representation, through capturing bias originally in the corpus, and potentially amplifying this bias (Zhao et al., 2017). Note that, I only discuss the bias of word embeddings in the context of Word2Vec, excluding BERT or ELMo. This lead to the third research question on fairness in machine learning, which I will discuss in Section 2.4.

2.1.3 Neural Architectures

Building on word embeddings, a range of deep learning architectures have been proposed for text. Among them, RNN based methods are particularly good for modelling language tasks, such as language modeling (Mikolov et al., 2010) and machine translation (Bahdanau et al., 2015). For text classification tasks, CNNs are more suitable and achieve high performance (Kim, 2014; Zhang et al., 2015). More recently, attention-based models, like transformers (Vaswani et al., 2017), have achieved great success in many language applications.

In this thesis, CNNs are used in all of my papers, while RNNs are only used in only one case, in the context of POS tagging in Section 3.6. In this section, I review the basics of RNNs and CNNs, and come back to provide the technical details of CNNs as used in this thesis in Section 2.5.1.

2.1.3.1 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are powerful sequence models, which suit the sequential nature of language. Generally, RNNs consider a sequence of inputs $\mathbf{x} = \{x_1, x_2, \cdots, x_n\}$. At each time step t, a hidden representation encodes the current input and the previous step,

except the initial step $\mathbf{h}_0$

$$\mathbf{h}_t = \mathrm{f}(\mathbf{h}_{t-1}, x_t; \theta) \tag{2.8}$$

where f represents an encoding function. Most commonly, the update f is defined as

$$\mathrm{f}(\mathbf{h}_{t-1}, x_t; \theta) = \mathrm{g}(\mathbf{W}x_t + \mathbf{U}\mathbf{h}_{t-1}), \tag{2.9}$$

where g is a smooth, bounded function such as the logistic sigmoid or a hyperbolic tangent, and the bias term is omited here. However, simply computing RNNs using Equation 2.9 does not work well in practise due to the gradient vanishing or explosion problem (Hochreiter, 1998; Pascanu et al., 2012), where the gradient is vanishingly small, or explosively big, due to iteratively calculating using g and $\mathbf{U}$ for too many steps. To tackle this, gated RNNs are proposed with the introduction of forget gates, such as the long-short-term-memory (LSTM) (Hochreiter and Schmidhuber, 1997) and gated recurrent unit (GRU) (Cho et al., 2014). For example, in a GRU (Cho et al., 2014), one update function is defined as

$$\mathbf{z}_t = \mathrm{g}(\mathbf{W}_\mathbf{z}\mathbf{x}_t + \mathbf{U}_\mathbf{z}\mathbf{h}_{t-1}) \tag{2.10}$$

$$\mathbf{r}_t = \mathrm{g}(\mathbf{W}_\mathbf{r}\mathbf{x}_t + \mathbf{U}_\mathbf{r}\mathbf{h}_{t-1}) \tag{2.11}$$

$$\mathbf{h}_t = (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \mathrm{g}(\mathbf{W}_\mathbf{h}\mathbf{x}_t + \mathbf{U}_\mathbf{h}(\mathbf{r}_t \odot \mathbf{h}_{t-1})) \tag{2.12}$$

where $\mathbf{z}_t$ is the update gate, $\mathbf{r}_t$ is the reset gate, both learned by the model, and $\odot$ is the Hadamard product.

For sequence learning tasks, each $\mathbf{h}_t$ is responsible for each output at time t

$$\mathbf{o}_t = \mathrm{NN}(\mathbf{h}_t; \theta) \tag{2.13}$$

where NN denotes a neural network, which commonly is a fully-connected layer and maps the hidden representation $\mathbf{h}_t$ to the prediction $\mathbf{o}_t$.

2.1.3.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are originally designed for two-dimensional image data (Krizhevsky et al., 2012), however, CNNs have been adopted to text classification (Kim, 2014; Zhang et al., 2015). CNNs use multiple convolutional filters over the embeddings matrix of the sentence, where each filter is learned to capture a distributional feature, such as the tri-gram phrase *on the mat*. The hidden representation $\mathbf{h}$ consists of all the learned features, and this forms the input to the final layer of a network, from which the prediction is made. A detailed exposition of CNNs for text is given in Section 2.5.1.

2.1.4 Regularisation for Deep Models

Deep models have achieved great success, however, they introduce the problem of much larger models compared to linear statisical machine learning methods. Large models can lead to overfitting, making a model less robust. To tackle this problem, one technique is regularisation, which is fundamental for training deep learning models (Goodfellow et al., 2016).

Conventional methods include L_1 and L_2 regularisation (Ng, 2004), which constrain the objective functions,

$$\mathscr{L}^{new} = \mathscr{L} + \sum_{\theta} |\theta_i|^p, \tag{2.14}$$

where $p = 1$ or $p = 2$.

Another widely used method is dropout (Srivastava et al., 2014), which applies element-wise random masks $\mathbf{m}$ over the hidden representations

$$\mathbf{h}^{new} = \mathbf{m} \odot \mathbf{h}, \tag{2.15}$$

where $\odot$ is the Hadamard product, and each dimension of $\mathbf{m}_i$ is either 0 or 1 randomly with a pre-defined probability. In particular, dropout randomly masks some nodes or neurons in the model with a probability of p, during training. In fact, Wager et al. (2013) showed that the dropout regulariser is first-order equivalent to an L_2 regulariser applied after scaling

the features. Dropout is equivalent to "Follow the Perturbed Leader" (FPL) which perturbs exponential numbers of experts by noise and then predicts with the expert of minimum perturbed loss for online learning robustness (Van Erven et al., 2014).

Early-stopping (Caruana et al., 2001) is also commonly used, in which training is stopped when the validation loss is observed to have converged. To avoid peeking at the test data during training, this observation is usally based on a validation dataset. When the validation loss stops decreasing, indicating that the model is starting to overfit the training set, the training process should be stopped.

In this thesis, all these regularisation methods are applied. Additionally, dropout regularisation is also considered as a baseline for robust representation learning in Section 3.1.

2.1.5 Robustness

One question that is core to this thesis is what is the definition of robustness for machine learning models. The robustness of an algorithm means it does not deteriorate too much when training and testing with slightly different data, either by injecting noise or using other dataset (Xu and Mannor, 2012). A formal definition is as follows: Given a dataset $\mathbf{x}$ and a small amount of noise ϵ, such that $\mathbf{x} \approx \mathbf{x} + \epsilon$, the model M is robust if

$$\mathscr{L}_{\mathrm{M}}(\mathbf{x} + \epsilon) \approx \mathscr{L}_{\mathrm{M}}(\mathbf{x}), \tag{2.16}$$

where $\mathscr{L}$ is the model loss. Based on this, a robust optimisation (Xu and Mannor, 2012) can be defined, which relates to the adversarial example in Section 2.2.3:

$$\theta^{(*)} = \arg\min_{\theta} \max_{\epsilon} \mathscr{L}(\mathbf{x} + \epsilon), \tag{2.17}$$

where θ is the model parameter.

For language applications, the noise ϵ can be language variation, domain variation, or demographic authorship variation, depending on each research question of this thesis. In

Language Phenomena	Expression 1	Expression 2
deletion of characters	heterogenous	heterogeneous
phonetic substitution	cat	kat
abbreviation	approximately	approx.
dialectal variation	how are you doing	ow ya goin
semantic equivalence	I am ok	I am fine

Table 2.1: Example of different language phenomena relating to lexical variation.

the following sections, I review the literature related to these research questions of machine learning robustness for text applications.

2.2 Robustness to Language Variation

As languages change over time and vary according to social scenarios, many variations in language can be observed. As mentioned, there are different kinds of language variations, of which I focus on two – lexical variation and syntactic variation. Lexical variation means variations are words and expressions. For example, *football* in British English is semantically equivalent to *soccer* in American English. Syntactic variation refers to differences in sentence structures, such as grammar. For example, in spoken language under certain conditions, pronouns can be omitted without changing the meaning (Kashima and Kashima, 1998).

In this section, I first review approaches to robust NLP based on language variation. Then, I review how language variation can help model robustness, primarily in terms of data augmentation.

2.2.1 Robustness to Lexical Variation

Let us first look at methods that are robust to lexical variation. Lexical variation relates to differences in the choice of words and multi-word expressions. It can be caused by many language phenomena, such as deletion of characters, phonetic substitution, abbreviation, dialectal, semantic equivalent, as well as other informal usages. Table 2.1 shows different examples of lexical variation. Additionally, people tend to borrow words from foreign

languages, especially for proper names (Haspelmath, 2009; Pfaff, 1979). For example, one can say *Ping Pong* in English, which is borrowed from the Chinese word for *table tennis*. People arbitrarily make up new words, such as *catdog*, which can be understood readily by natural speakers. These phenomena often occur in casual language, like social media or daily communication. It can also arise in other ways, like transcription mistakes. One motivation of this thesis is to build NLP models that are robust to these variations.

Many traditional NLP systems are based on word-level features, such as n-grams, or bag-of-words. Recently, deep learning based NLP systems have become more widespread, and many deep learning models are built upon word embeddings (as reviewed in Section 2.1.1). Lexical variation can lead to irregular words or expressions, which cause challenges to the robustness of NLP systems.

One problem caused by lexical variation is that it can easily result in out-of-vocabulary tokens. As mentioned in Section 2.1, for a range of word-level NLP systems, including both conventional machine learning and deep learning models, pre-processing filters out low-frequency words and treats them as special unknown words. When an important token, for example, *excellent* in a sentiment task, is treated as UNK by a model, the model is unable to capture any useful information relating to this token. Consequently, this limits the ability to fully understand the original semantics of the text.

Another problem is the black-box nature of deep learning models, where lexical variation can cause models to give unexpected and unexplainable predictions. For example, simply modifying a few trivial tokens in the sentence, such as punctuation, can flip model predictions (Alzantot et al., 2018). The idea of adversarial examples, which are carefully chosen, exposes the brittleness of these models. I review adversarial examples of text in Section 2.2.3.

In this section, I review the literature on lexical variation. Generally, many methods and resources have been proposed, suitable for conventional and deep learning approaches.

2.2.1.1 Tokenisation

Tokenisation is the task of segmenting a character sequence or a document into pieces, called tokens (Manning et al., 2014), which is a typical pre-processing step in NLP. So far I have

assumed a word tokeniser and word-level models, which cause the problems mentioned with rare and out-of-vocabulary tokens above. Enlarging the vocabulary size can mitigate the problem to some degree, but cannot totally solve it, and meanwhile, it also exacerbates problems associated with the model complexity.

One method to tackle this problem is to tokenise at the sub-word level, the character level, or even the byte level, which is sub-character for non-European scripts in encodings such as UTF-8. Consequently, one can learn distributed semantic representations built upon sub-words or characters. This has the benefit that out-of-vocabulary tokens can be built using sub-word or character embeddings. Intuitively, such models can learn more meaningful morphemes, like prefixes, suffixes, and other derivations, leading to greater robustness in deep models.

The effectiveness and robustness of character-level models to unknown words was demonstrated by Zhang et al. (2015), who developed character-level CNNs for text classification. The model showed better performance than both traditional methods and word-level RNN and LSTM methods over large corpora. Kim et al. (2016a) proposed an approach for language modelling. In their model, a convolutional neural net is adopted over character inputs, whose output is fed to an RNN. They showed that on languages with rich morphology, the model outperformed the word-level LSTM baselines.

There have also been methods proposed to segment words at the sub-word level Sennrich et al. (2016) address the translation of out-of-vocabulary words by encoding rare and unknown words as sequences of sub-word units. Returning to BERT (see Section 2.1.2.3), the model is trained at a sub-word level (Devlin et al., 2019) using a sub-word tokeniser.

2.2.1.2 Lexical Normalisation

Let us come back to word-level methods, as most of our work assumes word-level models in this thesis. To tackle the problems arising from lexical variation, another idea is to *clean* word features for word-level approaches. This attempts to remove or minimise the number of the unknown words, through a pre-processing step for both training and test data. One method is rule-based lexical normalisation, which has also become a standard pre-processing step for

language data, which includes stemming, or lemmatisation (Piskorski et al., 2007; Plisson et al., 2004; Porter, 1980). Through rule-based lexical normalisation, many misspellings can be corrected, and the vocabulary size can be reduced, reducing model complexity and achieving better model efficiency and robustness.

However, there are certain limitations of rule-based word normalisation methods. First, language variation can be irregular and manually designed algorithms can be problematic, causing common mistakes like overstemming and understemming. For example, widely used stemmers stem *universal*, *university*, and *universe* to *univers*, which have very different meanings (Kraaij and Pohlmann, 1994; Tordai and De Rijke, 2005). Second, many lexical normalisation methods are highly language-dependent, meaning that experts need to create different rules for different languages, which is very expensive.

As a result, machine learning based automatic normalisation methods have been proposed. Here, methods rely on a training corpus, rather than expert rules (Baldwin et al., 2015; Han and Baldwin, 2011). Many approaches have been proposed. Jongejan and Dalianis (2009) proposed to use a machine learning model that automatically learns lemmatisation rules to learn prefix, infix and suffix changes to generate the lemma from the word. They used different graph and tree structures to store the rules and measure the probability of the rules with training pairs from different languages. Gesmundo and Samardžić (2012) formalised the lemmatisation problem as a sequence classification task, in the form of a series of category tagging tasks. They model the task as one of learning word-to-lemma transformation rules, by encoding rule as labels in a sequential classification model (e.g. conditional random field).

More recently, deep learning methods have also been proposed for automatic lexical normalisation. One kind of model is based on an unsupervised encoder-decoder architecture. Liou et al. (2014) proposed denoising autoencoders, which learns a hidden compressed representation of the given sentence and later reconstructs the sentence with bag-of-words features from the hidden representation. A Sequence-to-sequence encoder-decoder model was also introduced by Leeman-Munk et al. (2015), in which a sentence is parameterised using a recurrent neural network, maintaining the word sequence. These autoencoder models learn to remember the original text and filter out the noise during the compression-decompression

process, and work well for situations like spelling correction. However, when a given token is too divergent from the original to match, such as *lolllllllllllll!*, a token from social media, unsupervised learning is not sufficient. Instead, supervised lexical normalisation approaches are required (Baldwin et al., 2015; Han and Baldwin, 2011). These methods are based on character-level encoder-decoder models and use annotations of paired words, much like neural machine translation approaches (Bahdanau et al., 2015). Additionally, Vincent et al. (2008) introduced a method where they jointly learn robust features using stacked autoencoders. They integrated the normalisation step into an end-to-end model, rather than viewing lexical normalisation as a pre-processing step.

2.2.1.3 Data Augmentation with Lexical Substitution

As most deep models are data-driven, having more training data often leads to better model performance and robustness. Therefore, when faced with limited training data, one idea is to generate large volumes of synthetic data. This is the idea of data augmentation and it has been broadly used in many applications of machine learning (Tanner and Wong, 1987).

A range of data augmentation approaches have been adopted to learn robust NLP models. Lexical substitution (McCarthy and Navigli, 2007) is one method to augment language data on the lexical level (Granville, 1984). This usually involves replacing a word with a synonymy, using a language resource like WORDNET (Miller, 1995). In this thesis, I also apply WordNet as a resource for augmentation (Section 3.2), and thus I review WordNet-like semantic networks below.

Semantic Network WordNet is a semantic network which is structured based on lexical relations between sets of synonyms (Cruse, 1986). There are several different lexical relations, including hyponymy, hypernymy, synonymy, antonymy, and derivation.

In the semantic network, an item, or a word, is linked with its related concepts. Figure 2.3 illustrates the example of the word *motorcar* and related words in WordNet.

The overall idea of augmentation using WordNet is that the algorithm randomly takes a training sentence, replaces a word with a synonym, and adds the new instance to the training

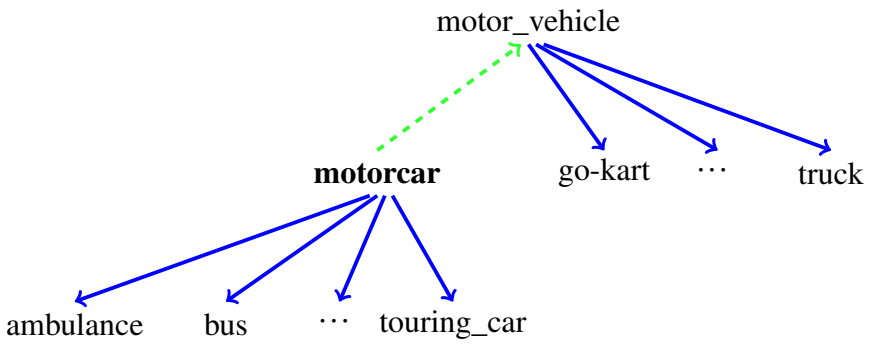


Figure 2.3: The example of *motorcar* in WORDNET. Green dashed lines denote the hypernyms and blue solid lines denote the hyponyms.

set with the same label. Zhang et al. (2015) adopted lexical substitutions, that exactly replace words with synonyms randomly for text classification. Their method enlarged the vocabulary size for learning robust character level model. Fadaee et al. (2017) proposed a method by augmenting rare words for machine translation. They applied a pre-trained language model to suggest rare substitutions in the parallel corpus, augmenting the corpus with new sentences. They loop over the original corpus multiple times, sampling substitution positions for each sentence and making sure that each rare word gets augmented at most N times. They demonstrate the method can help machine translation quality in a simulated low-resource scenario using English data.

Aina et al. (2019) proposed a method using lexical substitution data (Kremer et al., 2014) and averaging the substitution embeddings to augment the hidden representation in an LSTM model. They showed that the augmentation approach can improve language modelling and reduce lexical ambiguity under the given context.

However, it is important to note that more data does not always mean better performance, and the quality of the substitutions for data augmentation is important for robustness. Therefore, lexical substitution identification (Dagan et al., 2006; McCarthy and Navigli, 2007; Mihalcea et al., 2010) is necessary as a means of determining whether a given synonym is appropriate as a lexical substitute in a given context. Many approaches have been proposed for this task, including methods which combine different knowledge systems (Hassan et al.,

2007), use semantic similarity based on word embedding (Melamud et al., 2015), or use language model based on BERT (Zhou et al., 2019).

2.2.1.4 Lexical Substitution vs. Regularization

From the deep learning perspective, lexical substitution can be viewed as a form of regularisation (Section 2.1.4). Many language applications use word-level dropout regularisation (Gal and Ghahramani, 2016), where a word is randomly replaced with a non-informative token (eg. UNK) during training. Accordingly, lexical substitution behaves similarly to word dropout functionally, with the difference that replacing with meaningful substitutions provides linguistically-motivated semantic variability. Zhang et al. (2015) adopted lexical substitutions for training character-level CNNs, and in their paper, they also claimed the augmentation training methods behaviour like word-level dropout.

2.2.2 Robustness to Syntactic Variation and Paraphrasing

Language variation is more than lexical variation, and also includes syntactic variation (Sankoff, 1988), document structure variation (Namboodiri and Jain, 2007), and semantic and pragmatic variation (Van Dijk, 1977). In this section, I focus on syntactic variation and paraphrasing, which is also considered in Section 3.2. Comparing with lexical variation, syntactic variation is more complicated, which involves changes in sentence structure. In this section, I still focus on data augmentation methods, but at the syntactic level, using syntactic methods to generate new training data.

The first class of method is to generate variations based on a given reference text, such as a sentence or a document.

2.2.2.1 Paraphrasing with Syntactic Variation

Paraphrasing is one way of generating syntactic variation, in changing the structure of a sentence using different constructions while maintaining the meaning of the sentence. For

Does Abrams competently manage Browne?
Abrams does manage Browne, competently?
Does Abrams manage Browne, competently?
Abrams manages Browne, competently?
Abrams does competently manage Browne?

Table 2.2: Examples of paraphrases generated by ERG using *Does Abrams competently manage Browne?*

example, *A cat sat on the mat* can be paraphrased as *On the mat sat a cat* or *There was a cat sitting on the mat*.

There are many approaches to paraphrase generation. Lexical substitution is a simple paraphrase generation method (Quirk et al., 2004).

Precision Grammar: English Resource Grammar A precision grammar can be used to generate paraphrases for data augmentation, by parsing a sentence into a formal semantic representation, and back-generating surface grammaticalisations. Specially for English, English Resource Grammar (ERG) is a precision Head-Driven Phrase Structure Grammar (HPSG) of English (Bender et al., 2002; Copestake and Flickinger, 2000). Bond et al. (2008) and Nichols et al. (2010) propose a method for expanding training data with the the ERG via this method. Using the augmented dataset, consistent improvements for English-Japanese machine translation were observed over a statistical machine translation baseline.

A precision grammar is designed to license only grammatical sentences in a given language (possibly with some facility to loosen the strictness of the requirements on grammaticality). ERG can be used to generate grammatical paraphrases, which gives high-quality grammatical variations. The ERG can be used to generate grammatical paraphrases, providing high-quality grammatical variations. Table 2.2 shows paraphrases examples generated by ERG.

Data-Driven Paraphrasing Machine learning based automatic paraphrasing approaches have also been studied. Different from precision grammar-based approaches, machine

learning approaches rely on training data, and paraphrases are often ungrammatical but of acceptable quality (Madnani and Dorr, 2010).

One idea is to formalise the problem as statistical paraphrase generation, adopting statistical machine translation approaches trained over a monolingual corpus (Wubben et al., 2010; Zhao et al., 2009, 2008a). In detail, a standard phrase-based machine translation framework can be trained with a large aligned monolingual corpus, and during the reranking phase, an additional dissimilarity metric can be used to select paraphrase candidates.

Apart from monolingual corpora, there has also been research on using machine translation over bilingual parallel corpora for paraphrasing (Bannard and Callison-Burch, 2005; Chen and Dolan, 2011; Zhao et al., 2008b). Quirk et al. (2004) proposed a word replacement method during the decoding phase in a statistic machine translation system to generate paraphrase. Bannard and Callison-Burch (2005) showed that paraphrases in one language can be identified using phrases in a second language as a pivot. Based on bilingual pivoting, they define a paraphrase probability which extracts paraphrases from a parallel corpus, ranked using translation probabilities. For example, based on the phrase table learned from an English-German bilingual corpus, both *under control* and *in check* are translated to *unter kontrolle* with high probability, and can thus be identified as paraphrases of one another. Chen and Dolan (2011) tried to address the data quality problem of monolingual corpora for paraphrase generation. They proposed a data collection procedure by asking different annotators to describe the same video segment, resulting in highly parallel data for paraphrasing. Neural machine translation models have also recently been adopted for paraphrase generation (Mallinson et al., 2017), through applying a sequence-to-sequence model to translate a source sentence into a pivot language and then translating the pivot back into the source language.

Additionally, Li et al. (2018) introduce a deep reinforcement learning method to generate paraphrases, based on adversarial learning, which I review in detail in Section 2.3.1.2. Briefly, their frameworks consists of a sequence-to-sequence generator and a deep matching evaluator, both of which are learned from data. The generator produces paraphrases of a given sentence, the evaluator judges whether two sentences are paraphrases of each other. They applied reinforcement learning to tackle the difficulty of gradient back-propagation (Section 2.3.1.2).

Feedback to NLP Models Augmentation of training corpora using paraphrases can help boost the robustness of a range of language applications. For example, paraphrasing can directly help statistical machine translation by enriching training data (He et al., 2011; Resnik et al., 2010; Salloum and Habash, 2011). Berant and Liang (2014) described an approach that paraphrases the input and selects the best candidates, to improve the performance of semantic parsing. Ray et al. (2018) proposed a robust spoken language understanding method via paraphrasing augmentation. In addition, Ling et al. (2013) proposed a method based on paraphrasing to help normalise micro-blog data.

Paraphrasing can also help automatic evaluation. Kauchak and Barzilay (2006) apply paraphrasing in the context of machine translation evaluation, finding that paraphrases of the reference translation can be closer in wording to the machine translation than the original reference, and that the use of a paraphrased synthetic reference improves the accuracy of automatic evaluation.

2.2.2.2 Sentence Compression

Sentence compression is an alternative approach for generating syntactic variation (Knight and Marcu, 2002), and is also considered as a data augmentation method in our work (Section 3.2). Sentence compression algorithms shorten the given sentence by removing unimportant or auxiliary information while preserving the key content of the sentence. For example, one can shorten the sentence *A white cat sat on the blue wool mat quietly, looking at the flower* simply into *A cat sat on the mat*, preserving the key propositional content of the original sentence.

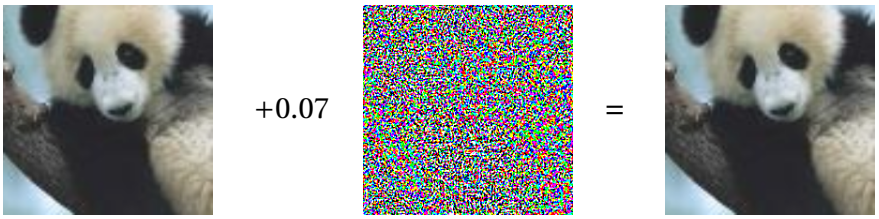
Many approaches to sentence compression have been proposed. Knight and Marcu (2000) proposed a noisy-channel model for sentence compression based on a standard probabilistic context-free grammar tree. They trained a probability model based on constituency trees using articles paired with human-written abstracts, and during decoding, drop expansion rules with low probability Turner and Charniak (2005) generalise Knight and Marcu (2000)'s methods to incorporate unsupervised and semi-supervised learning. Clarke and Lapata (2008) formalise the sentence compression task as integer linear programming. They introduced a decision

variable for each word in the source sentence and constrain it to be binary. A value of 0 represents a word being dropped in the target sentence, whereas a value of 1 indicates that the word is preserved. The objective is to maximise the probability of the token sequence subject to a number of constraints. Cohn and Lapata (2008, 2009) formalised sentence compression as a tree-to-tree transduction task, rather than simply deleting words or constituents. More recently, a neural-based word deletion model has been proposed (Filippova et al., 2015), based on similar ideas to paraphrasing using neural machine translation models (Mallinson et al., 2017). Briefly, a sequence-to-sequence model is trained over paired sentences, to learn word deletion probabilities given context.

2.2.2.3 Generating Syntactic Variations via Other Methods

There are also miscellaneous other methods for generating syntactic variation. Du and Black (2018) simply applied word permutation and word flipping operations over the training data to generate variations, resulting in ungrammatical and low quality data. They demonstrated that such data augmentation can improve model robustness in neural dialogue tasks. For robust training XLNET, Yang et al. (2019) also proposed a new objective called Permutation Language Modeling, which maximise the likelihood of the expectation of all permutations of the given sentence for the language model pre-training, and this permutation method has been provn to be a good choice. Similarly, Anastasopoulos et al. (2019) showed that augmenting training data using artificially-introduced grammatical errors, collected from non-native speakers, helps make a neural machine translation more robust to such errors in naturally-occurring test data.

So far, the augmentation approaches have all focused on generating new sentences based on a given reference sentence. There are also methods for generating entirely new training data, involving a language model trained or fine-tuned over a corpus, from which new instances are generated (Xie et al., 2019). Recently, deep generative models, such as generative adversarial nets (Goodfellow et al., 2014) and variational autoencoders (Bowman et al., 2016), have also been deployed for automatic augmentation (Antoniou et al., 2017). For example, Gupta et al. (2018) combine VAE and LSTM-based language model to generate more



“panda”	“nematode”	“gibbon”
57.7% confidence	8.2% confidence	99.3% confidence

Table 2.3: An adversarial example of image from (Szegedy et al., 2014).

suitable free text given the context. Li et al. (2019) proposed a model based on Transformer that can learn and generate paraphrases of a sentence at different levels of granularity in a disentangled way by composing of multiple encoders and decoders with different structures, each of which corresponds to the specific granularity. Our work also adopts adversarial learning and variation autoencoders for multi-domain learning, as described in Section 2.3.

2.2.3 Adversarial Text Examples

In this section, I review another language variation method, that is adversarial instance generation. This work originated in the vision literature. for example in Table 2.3, an adversarial example (on the right) is generated by applying adversarial perturbations (in the middle) to an image (on the left), which was originally labeled as “panda” by a state-of-the-art vision model. The difference between the original image and the adversarial image is not discerned to the human eye, however, the adversarial image easily fools the model. Such perturbation is cheap to generate by the *fast-gradient* method which computes the gradient of loss $\mathscr{L}$ with respect to the input image $\mathbf{x}$ (Goodfellow et al., 2015),

$$\mathbf{g}^* = \text{sign}\,\frac{\partial \mathscr{L}}{\partial \mathbf{x}}. \tag{2.18}$$

Recall the definition of robustness in Section 2.1.5: a robust model should be stable to adversarial perturbations. Therefore, adversarial instance generation is regarded as an attack method to the robustness of machine learning models.

In terms of adversarial instances when it comes to text, the definition is similar, in that slightly modifying the text of an input without changing its basic semantics is able to fool a machine learning system. For example, one can change a review from *Like such a wicked movie* into *It is impossible not to like such a wicked movie*. For sentiment tasks, both sentences express positive sentiment towards the *movie*, but the latter is misclassified by many machine learning models.

However, generating adversarial text examples is not an easy task. The fast-gradient method in Equation 2.18 is infeasible as $\mathbf{x}$ is discrete. Formally, the process of generating adversarial examples can be described as an optimisation problem

$$\begin{aligned} \min_{\epsilon} \quad & ||\epsilon||^2 \\ \text{s.t.} \quad & \mathbb{M}(\mathbf{x}) \neq \mathbb{M}(\mathbf{x}+\epsilon)\,, \end{aligned} \tag{2.19}$$

where $\mathbb{M}$ is the prediction given by some neural model, and ϵ represents the perturbations. It is not differentiable, as both the input $\mathbf{x}$ and ϵ are discrete tokens. Minimising the objective in Equation 2.19 becomes an integer programming problem, which is NP-hard (Papadimitriou, 1981). It is still an open question as to how to generate high-quality adversarial text. And how to defend against adversarial text remains an open question.

Alternatively, one method is adopting reinforcement learning to solve Equation 2.19, which estimates the expectation of the gradients instead of solving it in closed form. Based on reinforcement learning, many methods have been proposed for generating adversarial examples of text (Alzantot et al., 2018; Ebrahimi et al., 2018; Jia and Liang, 2017; Zhao et al., 2018c). Ribeiro et al. (2018) introduced semantically equivalent adversarial rules to generate semantic-preserving perturbations to detect bugs in black-box state-of-the-art models for different language tasks. However, the effectiveness and quality of these approaches are still low, and better generation algorithms are still needed.

2.3 Robustness to Domain

The second research question of this thesis is about robustness to domain variation. Domain usually refers to a grouping of data instances having some heterogeneity to them. For example, reviews within the *Book* domain will all relate to books, and have some consistency in terms of the aspects they touch on (the author, plot, etc) and the style of language they contain.

Domain adaptation is a large area in machine learning. There are many different criteria for categorising domain adaptation into sub-areas (Pan and Yang, 2009; Weiss et al., 2016). In this thesis, I particularly focus on two scenarios related to domain robustness. The first one is *transductive transfer learning*, where we need to train a model over the training data in one domain then apply it to test data in another domain. The domain for training is referred to as the source domain, and the domain for testing is the target domain. The second setting assumes that the training corpus is collected from multiple sources, and known as *multi-domain learning*, or *multi-domain adaptation* when the test domain is unknown.

2.3.1 Domain Adaptation

First, let us look at the adaptation problem with a single training domain, where only one source domain will be considered and the model will be tested over a different target domain. This setup is the most common for domain adaptation. One reason for doing domain adaptation is that although we know the target domain, we do not have enough target data for training. Depending on the quantity of annotated data of the target domain, we can categorise approaches into unsupervised, semi-supervised, and supervised domain adaptation, respectively (Pan and Yang, 2009). For example, when target data instances are not labeled, it is unsupervised adaptation, and based on unlabeled instances. When only a handful of instances are labeled, this is semi-supervised adaptation, and we can learn to align the source distribution with the target one using this data, in addition to unlabelled data.

Sometimes we have no knowledge of the target domain, i.e the model is required to make predictions for some unknown domain with no retraining. Under this scenario, the model's

robustness to an unknown domain is more important, which is especially true for commercial systems. For our purpose, the contributions of this thesis (Chapter 3) are in the latter case, i.e. unsupervised adaptation.

Domain adaptation has been applied to a range of NLP tasks, including machine translation (Chu and Wang, 2018), language modelling (Bellegarda, 2004), sentiment classification (Blitzer et al., 2007) and POS tagging (Collobert and Weston, 2008; Ruder and Plank, 2018). In this thesis, I focus on the text classification task (Joachims, 1999). Below, I review the literature on the topic.

2.3.1.1 Domain Adaptation for Text Classification

Text classification involves assigning a label to a sentence or document, usually with a model parameterised using a discriminative model.

Many different domain adaptation methods have been proposed for text classification. Blitzer et al. (2007) introduced an Amazon multi-domain product review dataset and use a structural correspondence learning algorithm to tackle the domain adaptation problem. The algorithm is based on word features, and they choose the pivot features of the target domain with the highest mutual information to the source label. Based on selected pivot features, they learn a predictor based on augmenting instances with selected pivot features. Tan et al. (2009) proposed a frequently co-occurring entropy method to leverage knowledge from the source domain, by picking out pivot features which occur frequently in both source and target domains and have similar co-occurring frequencies. To gather knowledge from the target data, they proposed an adapted naive Bayes classifier, which utilises both labeled and unlabeled target instances. Pan et al. (2010) proposed a spectral feature alignment algorithm to reduce the gap between domain-specific words of two domains, by aligning domain-specific words from different domains into unified clusters. In detail, the algorithm first selects domain-independent features based on feature frequency and mutual information, and then aligns them based on a bipartite feature graph between domain-specific and domain-independent features. They augment the domain-independent features with perfectly aligned domain-specific ones. Rai et al. (2010) proposed an algorithm that applies active learning in

the target domain, which leverages information of the source domain. They harnesses source domain data to learn a good initialiser hypothesis for doing active learning in the target domain (Cesa-Bianchi et al., 2006). The unlabeled target data is labeled by the hypothesis and is then used to update the hypothesis by querying the ground-truth from the oracle. Glorot et al. (2011) introduced a deep learning approach to study cross-domain sentiment analysis. They also assume access to unlabeled data from both domains and labels for the source domain only, and they adopt a stacked denoising auto-encoder (Vincent et al., 2008) to extract features from both domains. Bollegala et al. (2012) proposed a cross-domain learning method using as a sentiment-sensitive thesaurus. They first create a sentiment-sensitive distributional thesaurus using both labeled source data and unlabeled data from both the source and target domains, and then augment the features with the thesaurus during classification. Particularly relevant to this thesis is Daumé III (2007), who proposed a "frustratingly easy" domain adaptation method. I adopt this method to multi-domain adaptation (see Sections 3.4 and 3.5), and review the architecture in detail in Section 2.3.2.1.

Recently, Ganin et al. (2016) proposed to apply adversarial learning to domain adaptation to learn domain-invariant features, involving an auxiliary domain-adversarial loss. Li et al. (2017) encoded the adversarial loss into a memory network for cross-domain text classification. In this thesis, I adopt the domain-adversarial training to multi-domain adaptation, and thus I review the technical details of Ganin et al. (2016)'s domain-adversarial training in Section 2.3.1.2.

More recently, based on the success of pre-trained embedding over large corpora, fine-tuning based domain adaptation methods have been studied. Ziser and Reichart (2017) generalise Blitzer et al. (2007)'s method to a neural version using an autoencoder, injecting pre-trained word embeddings into the model to improve generalization with similar pivot features to improved domain adaptation. Howard and Ruder (2018) proposed a universal language model fine-tuning method, pretraining a language model on a large general-domain corpus, then fine-tuning it on the target task, and lastly transferring to the target classification task using concatenated pooling over the LSTM hidden states. Other pre-trained word embeddings, like ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019), have also been

shown to improve the performance and robustness of cross-domain text classification. These fine-tuning methods incorporate an unsupervised learning method using a large corpus, and then apply learned knowledge to other tasks, such as cross-domain text classification, which can also be considered as unsupervised domain adaptation.

2.3.1.2 Domain Adversarial Training

The first application of adversarial training in the context of deep learning was Generative Adversarial Nets (GANs), which were originally proposed as a generative model for image generation (Goodfellow et al., 2014). The idea also extends to text generation, such as language modelling (Yu et al., 2017) and machine translation (Wu et al., 2018; Yang et al., 2018).

The idea behind GANs can be interpreted from many perspectives. Literally, the original GAN consist of two adversarial components – a generator network G and a discriminator network D, which are in opposition to one another. For example in image generation, G takes the prior $\mathbf{z}$ and outputs images to the discriminator, which then attempts to identify whether a given image is an artificial one produced by the generator or a real one from the dataset. D and G play the following two-player minimax game with value function V (G, D):

$$\min_G \max_D V(D,G) = \mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}\left[\log D(\mathbf{x})\right] + \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}\left[\log(1-D(G(\mathbf{z})))\right], \tag{2.20}$$

where $p_{data}(\mathbf{x})$ is the data distribution, and $p_{\mathbf{z}}(\mathbf{z})$ is the prior distribution of the generator. As a result, they feed back to each other and both improve incrementally as a result of the adversarial setup. The discriminator is unable to distinguish between real and artificially generated images, it indicates that the generator can mimic real images in high quality, and that G and D have reached the global optimality of $p_{G(\mathbf{z})} = p_{data}$.

In the original paper of Goodfellow et al. (2014), training GANs was based on the Expectation–Maximization (EM) training algorithm (Dempster et al., 1977). In detail, the algorithm iteratively updates G and D. For each iteration, the algorithm first samples

$\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(m)}$, and noise priors $\mathbf{z}^{(1)}, \cdots, \mathbf{z}^{(m)}$. It updates D by ascending its stochastic gradient

$$\nabla_{\theta_D} \frac{1}{m} \sum_{i=1}^{m} \left[\log D(\mathbf{x}^{(i)}) + \log(1 - D(G(\mathbf{z}^{(i)}))) \right]. \tag{2.21}$$

Then, it updates the generator by descending its stochastic gradient based on the priors $\mathbf{z}^{(i)}$

$$\nabla_{\theta_D} \frac{1}{m} \sum_{i=1}^{m} \log \left(1 - D(G(\mathbf{z}^{(i)})) \right). \tag{2.22}$$

However, EM training can be unstable, and slow to converge. Therefore many approaches have been proposed to improve the training of GANs (Arjovsky et al., 2017; Salimans et al., 2016).

The generation of adversarial samples of text is hard due to the discrete nature of text (as outlined in Section 2.2.3). This applied equally here, as $D(\mathbf{x})$ is not differentiable when $\mathbf{x}$ is discrete. However, again, reinforcement learning can be applied (Yu et al., 2017), that is to sample $\mathbf{x} \sim p_G$ to estimate the expectation.

Domain-adversarial Learning Ganin et al. (2016) introduced adversarial training to domain adaptation. In domain adaptation, one essential consideration is how to learn features that can be shared by both source and target domains, that is to learn representations invariant to source and target. The overall procedure of domain-adversarial training and the gradient reversal layer are illustrated in Figure 2.4.

In Ganin et al. (2016), a feature extraction network, in the form of generator G, learns the feature vector $\mathbf{f}$ from the source $\mathbf{x}$,

$$\mathbf{f} = \mathrm{G}(\mathbf{x}), \tag{2.23}$$

and the discriminator, a domain classifier, aims to distinguish whether the feature $\mathbf{f}$ is learned through the source domain or the target domain. The discriminator can be modeled as a binary classification task, predicting the probability of $\mathbf{f}$ being from domain d,

$$\mathrm{D} = \mathscr{H}(d|\mathbf{f}), \tag{2.24}$$

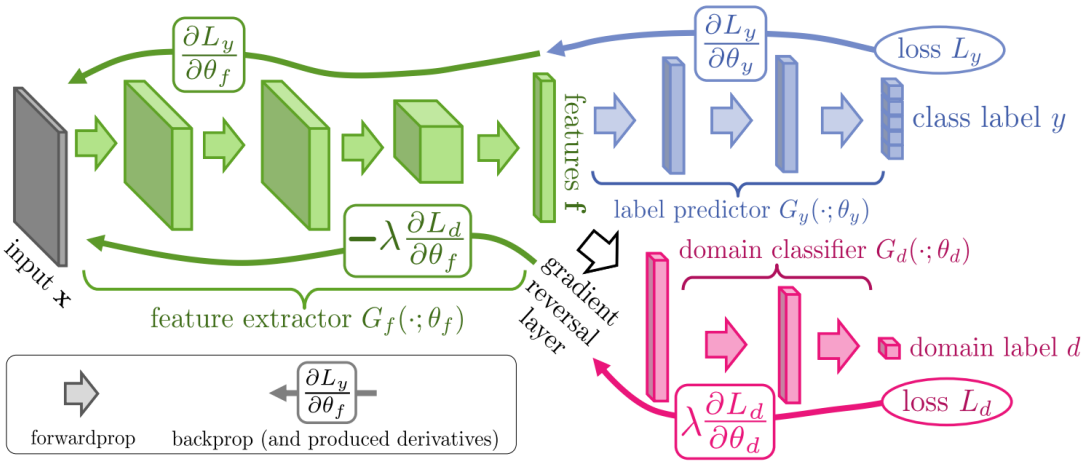


Figure 2.4: An illustration of domain adversarial training and gradient reversal layer from Ganin et al. (2016).

measured based on cross-entropy ($\mathscr{H}$) loss, where G and D are neural networks. The extracted feature vector **f** is also the inputs taken by the label predictor G_y to generate the class label predictions.

With the setup of G and D, now the adversarial process can be described as a min-max optimisation problem.

$$\mathscr{L} = \min_{\theta^{(G)},\theta^{(Gy)}} \max_{\theta^{(D)}} \mathscr{H}\left(\mathbf{y}|\,\mathrm{G_y}(\mathbf{x})\right) - \lambda\,\mathrm{D}\left(d|\,\mathrm{G}(\mathbf{x})\right), \tag{2.25}$$

where $\theta^{\{\cdot\}}$ denotes the parameters of G and D, respectfully. The first term denotes the task loss, and the second term is the domain-adversarial loss, both of which are balanced by a hyper-parameter λ.

To optimise Equation 2.25, Ganin and Lempitsky (2015) proposed a gradient reversal layer R_λ, which is defined as in following

$$\mathrm{R}_\lambda(x) = x \tag{2.26}$$

$$\frac{\partial \mathrm{R}_\lambda}{\partial \mathbf{x}} = -\lambda\mathbf{I}, \tag{2.27}$$

where, λ is the leveraging hyper-parameter in Equation 2.25. In detail, the gradient reversal layer is the identity function during the forward path, while it outputs the negative gradient, scaled by λ, during back-propagation. By introducing the gradient reversal layer, the min-max adversarial loss can be jointly trained with target objective $\mathscr{H}\big(\mathbf{y}|\,\mathrm{G}(\mathbf{x})\big)$, avoiding the EM algorithm and making the training more stable.

2.3.2 Multi-domain Adaptation

A second question is how to train using a corpus with multiple domains, or multiple sources, that is the multi-domain learning problem. This is a central contribution of this thesis as a technique for learning robust models.

One popular dataset is the Amazon multi-domain product review dataset (Blitzer et al., 2007),[1] consisting of reviews from two dozen categories, such as *Book*, *Electronic*, and *Music*. Each category has several hundreds or thousands of reviews of items from the given domain. Obviously, different items from different categories have different characteristics, and thus their corresponding reviews will focus on different features. Therefore, a model must learn different features for the different domains. Additionally, the concept of domain can be hierarchical. Taking a *Book* review as an instance, the category can be further categorised into some smaller sub-domains, such as *Fiction* and *Romance*. Most datasets are domain heterogeneous. Moreover, the "exact" domain identity is usually unknown. Therefore, most language applications are required to be robust over multiple unlabelled domains.

A range of multi-domain learning methods have been proposed in prior work. One popular setting is the supervised scenario of multi-domain learning, that is the domain of the test data has been seen during the training phase. For example, Mansour et al. (2009) presented a theoretical analysis of the problem of domain adaptation with multiple sources, and gave some empirical results based on their method of combined weights on several distributions.

Daumé III (2007) proposed a Frustratingly Easy Domain Adaptation ("FEDA") architecture for multi-domain learning. Two of the thesis contribution papers (Sections 3.4 and 3.5)

[1] From https://www.cs.jhu.edu/~mdredze/datasets/sentiment/.

are based on neural architectures inspired by FEDA, and thus I focus on these approaches and review the details of FEDA in the next section (Section 2.3.2.1). Generally, the method involves augmenting features, and thus it is also called a feature augmentation method. The method was originally framed as a single domain adaptation method, considering only learning from one source domain. However, Daumé III (2007) also outlined how it can be extended to multiple domains, and in fact, a number of subsequent papers extend the idea of FEDA for multi-domain learning. Recently, this has included applying different neural architectures to FEDA, including CNNs (Kim et al., 2016b) and RNNs (Liu et al., 2015b).

Besides the FEDA framework, recently, Wu and Huang (2016) considered a multi-task learning framework to extract global and domain-specific sentiment knowledge from multiple sources via auxiliary classifiers, and then transfer them to a target domain using lexical sentiment polarity relations extracted from the unlabeled target domain. They proposed sentiment graph extraction, wherein two words that frequently describe the same object in the same review, have a high probability of conveying the same sentiment polarity in that domain. Then, they measured domain similarity based on extracted sentiment features, composed by the similarity of features using Kullback-Leibler (KL) divergence. Based on domain similarity, they select a target domain from the most similar training domains.

In contrast to supervised adaptation, unsupervised multi-domain adaptation considers the case where the methods have no access to the labeled target data, but have access to the unlabeled data and meta-data identifying the domain. Yang and Eisenstein (2015) first proposed this unsupervised multi-domain adaptation challenge, and proposed a method for combining multiple word embeddings with metadata domain attributes to tackle this problem. They learned feature embeddings based on the skip-gram model for each domain, and domain-general features based on the union of the source and target data sets. They proposed to aggregate multiple embeddings from different domains to attain cross-domain generalisation.

2.3.2.1 Multi-domain Learning through Feature Augmentation

To tackle multi-domain learning, in this thesis, our approaches generalise the frustratingly easy domain adaptation architecture proposed by Daumé III (2007). Therefore, I go over the details of this model in details here.

In order to formally introduce the model architecture, let us assume a discriminative model and consider a multi-domain dataset $\mathbf{X} = \{\mathbf{x}_i, y_i, d_i\}|_{i=1}^{n}$ where $\mathbf{x}_i, y_i, d_i$ represent the input, the task label, and the domain label, respectively. The total number of domains is set to K.

For multi-domain learning, the feature augmentation method enlarges the original feature space into $K+1$ replicas, learning extra sets of features corresponding to specific domain d,

$$\Phi^d(\mathbf{x}) = \langle \Phi_0(\mathbf{x}), \Phi_1(\mathbf{x}), \cdots, \Phi_K(\mathbf{x}) \rangle \tag{2.28}$$

where

$$\Phi_t(\mathbf{x}) = \begin{cases} \mathbf{x} & t = 0 \; or \; t = d \\ \mathbf{0} & i \neq d \end{cases} \tag{2.29}$$

where input $\mathbf{x}$ is treated as a vector of features, and $t = 0$ denotes shared features across all domains.

After augmenting, one can apply a classifier C over $\Phi^d(\mathbf{x})$ to predict the label. Note that this model also enlarges the size of model parameters to $K+1$ times compared with non-augmenting methods. To be more specific, each Φ_i operates as a gating function, where Φ_0 is always on for all domains, while Φ_d is only turned on for instances from domain d. This process only involves the input features $\mathbf{x}$, and thus it can easily be performed as a pre-processing step.

We can also review feature augmentation from a different perspective. Instead of simply being a gating function, each channel $\Phi_i|_{i=0}^{K}$ can be parameterised by different parameters θ_i, learning different features. In this setup, the feature augmentation method consists of K different feature learning processes. For example, in the single-domain situation, a hidden

representation $\mathbf{h}$ is generated given the instance $\mathbf{x}$:

$$\mathbf{h} = \Phi(\mathbf{x}; \theta) \tag{2.30}$$

where Φ now represents a feature learning channel, and it can be parameterised by some neural architecture (θ), such as a convolutional network as described in Section 2.5.1. One of the contributions of this thesis is to parameterise the models based on this perspective in Sections 3.4 and 3.5.

Accordingly, in multi-domain cases, for each domain $d = 1 \cdots K$,

$$\mathbf{h}_d = \Phi_d(\mathbf{x}; \theta_d)\,. \tag{2.31}$$

Additionally, Φ_0 is a shared channel for all domains and generates a domain-general representation, denoted $\mathbf{h}^s$. Φ_d is a private channel for domain d, which is only turned on for instances from domain d, generating domain-specific representations,

$$\mathbf{h}^p_d = \Phi_d(\mathbf{x})\,. \tag{2.32}$$

Consequently, for each data instance, two representations will be learned, a domain-specific one $\mathbf{h}^p$ and a domain-general one $\mathbf{h}^s$. The concatenation of $\mathbf{h}^p$ and $\mathbf{h}^s$ is used to represent the input $\mathbf{x}$.

Feature augmentation introduces the idea of using multiple domain channels to learn from multiple domains, and has achieved great success in many domain adaptation tasks. However, there are some problems with its original version. Firstly, model complexity grows linearly with the number of domains, making it computational inefficient. This could be a problem when dealing with large numbers of domains, in which situation it introduces too many parameters, leading to the problem of overfitting. The second problem is that the shared $\mathbf{h}^s$ can be contaminated by domain-specific features in practice, as Φ_0 is active for all the data instances. For example, sometimes data is biased and unbalanced for some domains, and Φ_0 will learn such unbalanced priors and encode them in the model. Thirdly, in many

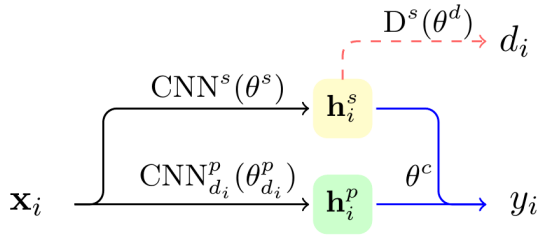


Figure 2.5: An illustration of multi-domain learning architecture with adversarial learning over the shared representation $\mathbf{h}^s$.

cases domain information is latent, that is domain labels are unknown, and the total number of domains is uncertain. I review some approaches to tackle these problems below.

2.3.2.2 Feature Augmentation with Domain Adversarial

To tackle the second problem of contaminating the shared $\mathbf{h}^s$, adversarial training is introduced (Liu et al., 2015b), to obtain a domain-invariant representation $\mathbf{h}^s$. Figure 2.5 illustrates the overall architecture. For each domain, two representations are learned: a shared $\mathbf{h}^s$ and a domain-private $\mathbf{h}^p$, and a predictor (c) takes the concatenation of $\mathbf{h}^s$ and $\mathbf{h}^p$ to make class predictions. Similar to the objective in Section 2.3.1.2, a domain-adversarial discriminator D^s is applied to $\mathbf{h}^s$ to predict the domain label. Therefore, the overall adversarial training objective is

$$\mathscr{L} = \min_{\theta^c, \theta^s, \{\theta^p_\cdot\}} \max_{\theta^d} \mathscr{H}(\mathbf{y}|\mathbf{h}^s, \mathbf{h}^p, \mathbf{d}; \theta^c) - \lambda_d \mathscr{H}(\mathbf{d}|\mathbf{h}^s; \theta^d) \tag{2.33}$$

However, note that Liu et al. (2015b) limit their evaluation to solely in-domain settings. As one of the aims of this thesis is robust out-of-domain evaluation, I develop this method further based on adversarial training, as described in Sections 3.4 and 3.5.

2.3.2.3 Domain learning with Latent Domains

The attentive reader will have noticed that all the work discussed until now is based on the assumption that all instances are annotated with domain labels. However, this assumption does not always hold in the real world, as text data is often heterogeneous, and annotations are

expensive. Therefore, we need to consider the task of multi-domain learning without domain labels or with overly general labels. In this thesis, these correspond to un- or semi-supervised learning with respect to domain, referred to as *domain unsupervised learning* and *domain semi-supervised learning*, which is one of the contributions of the thesis (Section 3.5).

One way to address this problem is to model domain as a latent variable $\mathbf{z}$, using a joint probability distribution of input and domain $p(\mathbf{x}, \mathbf{z})$. However, applying latent variables often make inference intractable, requiring approximate inference, such as *variational inference* (Blei et al., 2017). I review the details of variational inference methods below.

Variational Inference In Bayesian models, latent variables, in the form of domain vector $\mathbf{z}$, help govern the data distribution. A Bayesian model draws the latent variables from a prior distribution $p(\mathbf{z})$ and then relates them to the observations via the likelihood $p(\mathbf{x}|\mathbf{z})$.

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})\, p(\mathbf{x}|\mathbf{z}) \tag{2.34}$$

Inference in a Bayesian model amounts to conditioning on data and computing the posterior $p(\mathbf{z}|\mathbf{x})$. By Bayes rule,

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{z}, \mathbf{x})}{p(\mathbf{x})} \tag{2.35}$$

$$= \frac{p(\mathbf{z}, \mathbf{x})}{\int_{\mathbf{z}} p(\mathbf{z}, \mathbf{x})} \tag{2.36}$$

The denominator contains the marginal density of the observations, also called the evidence. However, the evidence is typically intractable, i.e. not computable in closed form or requires exponential time to compute. Therefore, approximate inference is needed.

To address this, an approximation probability $q(\mathbf{z})$ is introduced, and by minimising the Kullback-Leibler (KL) divergence between q and the true posterior, one can derive a good estimation of the posterior

$$q^*(\mathbf{z}) = \arg\min_{q(z) \in Q} \mathrm{KL}\big(q(\mathbf{z}) || p(\mathbf{z}|\mathbf{x})\big)\,, \tag{2.37}$$

where Q is a family of distributions for $q(z)$, chosen to lead to tractable computation.

However, it is not possible to directly compute the KL, as the true density is not accessible.

$$\mathrm{KL}\big(q(\mathbf{z}) \| p(\mathbf{z}|\mathbf{x})\big) = \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})}[\log q(\mathbf{z})] - \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})}[\log p(\mathbf{z}|\mathbf{x})] \tag{2.38}$$

$$= \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})}[\log q(\mathbf{z})] - \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})}[\log p(\mathbf{z},\mathbf{x})] + \log p(\mathbf{x})\,. \tag{2.39}$$

This reveals its dependence again on $\log p(\mathbf{x})$, which is the evidence in Equation 2.36.

Alternative, we can maximise an objective that bounds the KL,

$$\mathrm{ELBO}(q) = \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})}[\log p(\mathbf{z},\mathbf{x})] - \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})}[\log q(\mathbf{z})] \tag{2.40}$$

This function is called the evidence lower bound (ELBO).

The ELBO can be written as

$$\mathrm{ELBO}(q) = \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})}[\log p(\mathbf{z})] + \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})}[\log p(\mathbf{x}|\mathbf{z})] - \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})}[\log q(\mathbf{z})] \tag{2.41}$$

$$= \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})}[\log p(\mathbf{x}|\mathbf{z})] - \mathrm{KL}(q(\mathbf{z}) \| p(\mathbf{z})) \tag{2.42}$$

By maximising the ELBO, a good estimate of q can be obtained.

Variation inference can be used to optimise a variational autoencoder (Kingma and Welling, 2014). As one of the contributions, our work (Section 3.5) introduces a latent variable to model domain, and we also apply variational inference to make the model tractable.

2.4 Robustness, Fairness and Privacy-preserving

As machine learning has made rapid headway into social applications, the problem of fairness in machine learning has drawn the attention of not only the public but also the academic community, namely that machine learning model should treat people with different backgrounds equally (Barocas et al., 2019). A related problem is the privacy of users, as a

service supported by machine learning systems can expose the demographic information of trained users.

In this section, these two problems are compared and related to model fairness and user privacy. The problem of fairness in machine learning will be first contextualised, and we will see how fairness machine learning relates to privacy.

2.4.1 Fairness and Machine Learning

First, let us focus on the concept of fairness towards demographic variables, leaving the privacy problem aside for now. Broadly speaking, the term *fairness* can mean many things (Barocas et al., 2019; Francez, 1986; Rawls, 2001). However, in this thesis, I discuss it only as it pertains to demographic variables. A demographic variable is an attribute of a data producer, for instance, a user's gender, age, or educational background, all of which have impact on their use of language.

In socio-linguistics, these demographic variables are important factors in categorising people into different groups. Fairness in social science means people are required to be treated equally in human society, irrespective of their demographics. Similarly when it comes to machine learning, individuals should be treated equally by machine learning models. For example, two students with the same entrance scores but different ages, should be considered equally when applying for a scholarship. Such ideas have a long history but have only recently been studied in machine learning. As machine learning algorithms are increasingly playing a role in people's daily decisions, the necessity for fairness in machine learning is becoming more critical.

2.4.1.1 Definition of Fairness Towards Demographic Variables

An important question is how to define fairness in terms of demographic variables.

Let us consider the simplest case by assuming a binary classification model C, and a binary demographic variable b. We say C is a fair classification towards a demographic variable b, when

$$p(C = 1|b = 0) = p(C = 1|b = 1) \tag{2.43}$$

This means the classifier performs exactly the same when only the demographic variables is changed. The classifier does not depend on b when making decisions. Equation 2.43 can be generalised to the multi-variable case,

$$p(C = 1|\mathbf{b_1}) = p(C = 1|\mathbf{b_2}) \, \forall \mathbf{b_1}, \mathbf{b_2} \in \mathscr{B}, \tag{2.44}$$

where $\mathscr{B}$ is the space of all the demographic variables.

From Equation 2.44, recalling the concept of domain from the Section 2.3.1, one can see that demographic variables and domain actually play a similar role mathematically, in that they both govern the data distribution. However, one should notice that they are different concepts, which should not be confused with each other. Domain relates to groupings of data instances, while fairness with respect to a demographic variable is based on the individuals associated with the instances (i.e. the authors of the documents). In most settings, domains are not considered protected demographic attributes, although the sets of users in each domain may differ in their demographics, for example, *fashion* reviews may be bias for gender.

2.4.1.2 Demographic Bias

In natural language processing, such demographic variables often lead to bias in the data. These biases are often observed in the corpus (Hovy et al., 2015), and capturing this bias can improve model performance (Hovy, 2015; Michel and Neubig, 2018; Rabinovich et al., 2017). In machine learning, many approaches do not consider the demographic attributes, hence a consequent problem is that algorithms tend to learn and encode biased demographic attributes of the corpus into the model, potentially compromising model fairness. I review papers regarding this problem below.

A famous example regarding language is the gender bias learned by word embeddings. As mentioned in Section 2.1.2.2, one of the advantages of word embeddings is that words are represented by continuous vectors and linear relations between word embeddings can be

captured. Reusing an example from Section 2.1.2.2, in word embeddings one may observe:

$$\mathbf{w}^{king} - \mathbf{w}^{man} + \mathbf{w}^{woman} \approx \mathbf{w}^{queen} \tag{2.45}$$

where $\mathbf{w}^{(\cdot)}$ represents a learned word embedding. However, the same process of analogy can also be used to expose demography-biases. For example, Bolukbasi et al. (2016) reported the relation:

$$\mathbf{w}^{doctor} - \mathbf{w}^{man} + \mathbf{w}^{woman} = \mathbf{w}^{nurse}\,. \tag{2.46}$$

This linear relation is not semantically correct any more, as we notice that *doctor* and *nurse* are both occupations that can be associated with either gender. Often simple examples can be found, like *Man is to computer programmer as woman is to homemaker* (Bolukbasi et al., 2016; Zhao et al., 2019b). This is caused by demographic biases and word occurrences existing in the training corpus, because in the corpus, male doctors are mentioned more frequently than female doctors. More recently, Nissim et al. (2019) argued that this is overstated, and Gonen and Goldberg (2019) showed that such relations are not a good diagnostic for bias in embeddings, but mislead bias identification. However, for a robust word representation in many applications it is undesirable to learn such implicit biases, and violates the fairness definition in Equation 2.44.

Beyond word embeddings, gender bias can affect other language systems, for example, coreference resolution models (Rudinger et al., 2018; Zhao et al., 2018a). Here, gender bias was confirmed in coreference systems based on WINOGENDER schemas, in which a pair of sentences differs only in pronoun gender. For example, given the sentence *the surgeon couldn't operate on the patient: it was [his/her] son!*, the system should resolve *his* and *her* as coreferent with *the surgeon*, but in the second case, due to gender bias associated with the word *surgeon*, systems tend to resolve *her* to *the patient*. Also, recent work of Stanovsky et al. (2019) analyses gender bias in machine translation systems. They showed that all the test systems were significantly prone to gender-biased translation errors for all tested target languages.

Fairness can also be related to race (Chouldechova, 2017; Dressel and Farid, 2018). One example is applying machine learning algorithm to predict crime (Flores et al., 2016), e.g. given a portrait or the social media posts of a person, predicting whether the target is a criminal or not. Such an application can easily lead to racial-discrimination, in the system learning to consider groups of people as criminals largely because of their race, such as using skin colour (Flores et al., 2016). Besides bias in data, Sap et al. (2019) show that the annotators' insensitivity to differences in dialect can lead to racial bias in automatic hate speech detection models. Therefore, we need machine learning algorithms to be fair decision-makers, and must be particularly careful with the black-box of deep learning models, where we have less insight into what the model has truely learned.

There are also other demographic biases, such as age and education level, that have been observed by the research community (Rudinger et al., 2017). These demographic biases can lead to problems in real-life NLP systems, such as POS taggers (Hovy, 2015), and natural language inference models (Rudinger et al., 2017). Above all, building robustness in language applications requires fairness to demographic variables.

2.4.1.3 Mitigating Demographic Bias for NLP

As shown from the above examples, demographic bias in training corpora causes fairness problems for machine learning models. One way of building unbiased models is to construct unbiased datasets, for both training and evaluation. Zhao et al. (2018a) introduced a new benchmark, WINOBIAS, for a coreference resolution task focused on gender bias. Concurrently, Rudinger et al. (2018) evaluated and confirmed the gender bias in the coreference system by building the WINOGENDER dataset. Sakaguchi et al. (2019) extended this idea of WINOGENDER to a larger scale, WINOGRANDE, by combining data instances using an adversarial filtering algorithm with a careful crowdsourcing design.

Zhao et al. (2017) proposed a method based on corpus-level constraints to remove bias at the corpus level. They considered a vision activity recognition task, and defined identified gender bias toward *man* for each verb as $\frac{c(verb,man)}{c(verb,man)+c(verb,woman)}$, where c is the count

operation. Then, to reduce bias amplification, they set a constraint to ensure that the gender ratio for each activity prediction is bounded around the balanced ratio.

Another way of improving model fairness is to directly build unbiased models, for example, learning unbiased word embeddings or representations. Zhao et al. (2018b) proposed a training procedure for learning gender-neutral word embeddings, by preserving gender information in certain dimensions of word vectors. They adopted the GloVe framework (Pennington et al., 2014) using two gender constrains, one minimising the negative distances between the averaged word embeddings of the two gender groups (male vs. female), and the other one estimating the averaging differences between embeddings of predefined gendered word pairs. Similarly, Kaneko and Bollegala (2019) extended the de-biased embedding learning approach of Zhao et al. (2018b) with an additional reconstruction loss using an autoencoder network over each word.

More recently, approaches for learning unbiased models have been proposed for other language applications (Sun et al., 2019). Bordia and Bowman (2019) proposed methods for identifying and reducing bias in a word-level language model. Different from Zhao et al. (2017)'s method, they defined the probability of a word w occurring with context words $c(w, g)$ of gendered word g as $p(w|g) = \frac{c(w,g)/\sum_i c(w_i,g)}{c(g)/\sum_i c(w_i)}$, and defined the bias as $\text{bias}_{train}(w) = \log(\frac{p(w|f)}{p(w|m)})$. Based on this, they reduced the bias by estimating the gender amplification distribution and averaging the embeddings of gendered words. Stanovsky et al. (2019) studied the bias problem in machine translation systems. They built a WINOMT corpus for evaluate gender bias in machine translation, by combining the WINOGENDER and WINOBIAS, and observed gender bias, both commercial and academic machine translation systems. They confirmed their findings with human annotation.

Adversarial training has also been adopted to remove demographic biases in deep learning models. Here, the discriminator (Section 2.3.1.2) detects the demographic information from the learned hidden representation. This is a key contribution in this thesis, as presented in Section 3.6.

2.4.2 Privacy-preserving Deep Learning

The remarkable performance of deep learning and machine learning more generally requires large, representative data. These datasets often include sensitive data of users. As mentioned in Section 2.4.1.2, machine learning models can encode demographic biases in parameters. Once these models are published, malicious parties can detect and reconstruct sensitive information about the users whose data the model was trained on, or even the training datasets (Agrawal and Srikant, 2000; Gambs et al., 2012; Song et al., 2017). A good model should not expose this private information to third parties. In this section, I review related work on privacy-preserving approaches for deep learning.

2.4.2.1 Formal Privacy with Differential Privacy

One large class of methods is based on Differential Privacy (DP) (Dwork, 2011). This provides a formal privacy guarantee, that is a privacy-preserving mechanism can be quantified with a privacy budget.

One definition of DP is as follows. Let $\epsilon > 0$ be a real number, and $\mathscr{A}$ be a random mechanism. Mechanism $\mathscr{A}$ is said to provide (ϵ, δ)-differential privacy, if and only if for any input d,

$$p\left[\mathscr{A}(d) \in S\right] \leq e^{\epsilon} \cdot p[\mathscr{A}(d') \in S] + \delta\,, \tag{2.47}$$

where d' is the adjacent input of d, that they only different for one element. Generally, it means one cannot different the difference between any adjacent two inputs, or input datasets d and d' with a certain probability provided by the privacy budget, ϵ and δ, using mechanism $\mathscr{A}$.

Based on DP, a range of privacy-preserving deep learning methods have been proposed. Shokri and Shmatikov (2015) evaluated a distributed learning system that enables multiple parties to jointly learn an accurate neural-network model by sharing the gradients, instead of data, from all parties. In their architecture, a private model is trained on the private data of each party, and then they share DPed gradients with a central server, which averages the gradients from all parties. Each party can download the new parameters and re-train the

models. By only sharing DPed gradients, the distributed SGD approach can protect privacy from attackers. More recently, local differential privacy methods have also been introduced to deep learning (Chamikara et al., 2019), once again based on a distributed learning process, with the difference being that each party shares their intermediate hidden representation, rather than the gradients. Training is such that a data owner can add a randomization layer before sending data to a potentially untrusted machine learning service, which also provides differential privacy to the learning process.

From a different aspect, some approaches focus on publishing models without compromising the privacy of the training data. Abadi et al. (2016) proposed a deep learning training framework by adding DP noise into the gradient during training. Different from standard SGD (Bottou, 2010), they applied Gaussian noise to protect the gradient privacy before each gradient update. Fernandes et al. (2019) proposed a way of obfuscating text by adding differential privacy noise over bags-of-words representations and word embeddings, which can provide a formal privacy guarantee. Privacy-preserving deep learning has also been applied to biological and medical text applications, where the data is highly sensitive (Ching et al., 2018).

However, there are some limitations in this preliminary work. One problem is that these algorithms can sometimes overestimate the privacy budget. For example, Abadi et al. (2016)'s method applied DP noise to each dimension of the gradients of the parameters, for each gradient update. If one looks back to the definition of DP (see Equation 2.47) that a mechanism $\mathscr{A}$ is measured by two adjacent inputs, however, in Abadi et al. (2016)'s algorithm, they applied DP noise ($\mathscr{A}$) multiple times, and thus the privacy budget is linearly related with the number of update steps, as well as the model complexity. Therefore, the actual privacy budget is quite large, which makes the privacy meaningless in real world applications. Hitaj et al. (2017) introduced a GAN attack, which is adopted from adversarial generative methods, to leak user information from Shokri and Shmatikov (2015)'s model. In their attack framework, a malicious party can upload "malicious" gradients, which are learned by GANs, to the server and to cause specific users to upload more and more data, as

the overall objective for all parties is to gain good performance. Similarly here, once there are enough shared gradients, the data can be recovered (Gambs et al., 2012).

2.4.2.2 Privacy Under Inference Attacks

Besides formal privacy, sometimes an empirical privacy evaluation is also used to quantify privacy in machine learning, through creating an inference attack. Inference attacks evaluate how much privacy a model can leak information on the individual training data records (Krumm, 2007). Inference attacks focus on the scenario of providing black-box access to the model, and seeking to measure whether sensitive information of a data instance can be inferred from the model outputs. For example, in the cooperative learning system of Ching et al. (2018), when the locally learned features or representations are shared across different parties, other parties can use them to improve their models. If the shared representations are not protected, sensitive information like demographic information and other private information can be detected by the malicious party through inference attacks using a learning process. In this thesis, I focus on privacy under inference attacks.

A range of approaches have been proposed for studying inference attacks. Shokri et al. (2017) studied the membership inference attack of determining if a data instance was in the model's training dataset. In their method, the attacker creates k shadow models, where each shadow model is trained on a dataset distributed similarly to the target model's training dataset. All the shadow models are then ensembled to infer the membership of the training dataset.

To defend against inference attacks, Hamm (2017) proposed a method of applying a min-max filter, which is closely related to adversarial learning, and they also consider evaluation using an inference attack. As a contribution of this thesis, I also consider an adversarial learning method, as detailed in Section 3.6. Similar to my approach, Coavoux et al. (2018) applied adversarial learning to obtain privacy-preserving representations. Different from my model, they additionally consider reconstructing the training example and penalising pairs of examples with similar reconstructions or hidden representations to make the training more robust. The idea of adversarial learning is adopted for many NLP applications and

models. For example, Lang et al. (2019) generalise adversarial training to protect privacy for a transformer-based neural network, and Zhao et al. (2019a) also considered adversarial training to preserve privacy under an attribute inference attack, and provide a theoretical analysis of adversarial privacy-preservation. Mosallanezhad et al. (2019) further extend the adversarial idea to a deep reinforcement learning framework to anonymise the text against private-attribute inference, where reinforcement learning is used to handle the discreteness of the text (Section 2.3.1.2). Hu et al. (2019) investigate the problem of obfuscating sensitive information to perform privacy-preserving syntactic parsing. Their model operates at the word level, and learns to obfuscate each word separately by replacing it with a new word which has a similar syntactic role.

There are limitations of inference attack privacy, most notably it is based on empirical privacy, and lacks a formal bound of the privacy budget (Shokri et al., 2017). Considering this is still a growing area, I expect more research to be devoted to new approaches for privacy-preserving deep learning.

2.5 Model Technique and Tasks

In this thesis, most of my proposed approaches are based on convolutional neural networks, with one exception where I use a bi-directional LSTM neural network for POS tagging. Many convolutional neural network structures have many proposed for text classification, but here, I review the structure of the CNN proposed by Kim (2014), which is also used as a baseline.

2.5.1 Convolutional Neural Nets for Text

This section outlines how to build a word-level CNN for text classification from scratch, to help the reader better understand the foundations of Chapter 3.

Let us start with pre-processing before going through the details of CNNs. In language applications, as we mentioned in Section 2.1, raw text is processed using a pre-processing pipeline, including lowercasing, tokenisation, and normalisation, which help to harmonise from the text. After pre-processing, the raw documents will be represented as tokens. For

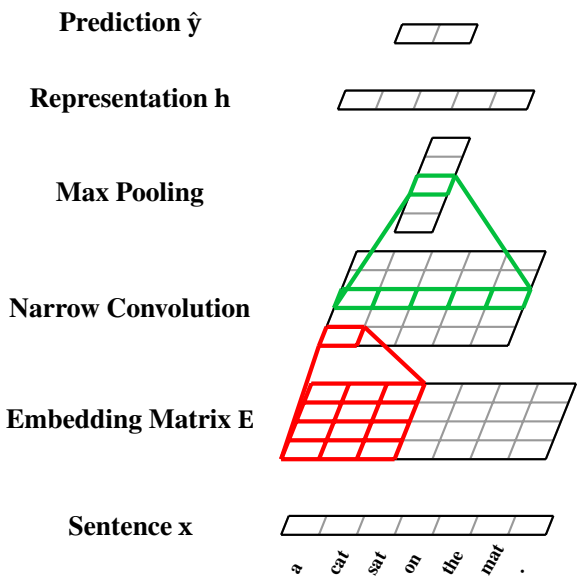


Figure 2.6: An illustration of the CNN structure with 7 input tokens, 5 convolution filters and max pooling. Note that the ReLU and the bias b are omitted here.

example, we may have the word-tokenised and lowercased input sequence

$$\mathbf{x} = \{\mathrm{id}_a \, \mathrm{id}_{cat} \, \mathrm{id}_{sat} \, \mathrm{id}_{on} \, \mathrm{id}_{the} \, \mathrm{id}_{mat} \, \mathrm{id}_.\} \tag{2.48}$$

with length l, where $l = 7$ in this cases. Each token is a discrete feature of the given document, and deep learning methods aim to learn a better representation in a continuous space.

Overall, a convolution network consists of a few components – a word embedding layer, convolution layer, pooling layer, non-linear activation, and output layer, which I explain in sequence below. Figure 2.6 illustrates the workflow of a TextCNN network, which is used through this thesis.

First is the word embedding layer. This layer first maps each token into a d-dimension word embedding (Section 2.1.1), and concatenates them into an embedding matrix.

$$\mathbf{E} = \mathrm{EMB}(\mathbf{x}) \in \mathbb{R}^{d \times l} \tag{2.49}$$

These word embeddings can be initialised randomly, or using pre-trained values, such as WORD2VEC (Mikolov et al., 2013), ELMO (Peters et al., 2018) or BERT (Devlin et al.,

2019). These word embeddings can be recognised as features of words, and can be trained based on a much larger corpus (see Section 2.1.1).

The next stage is the convolution layer, where **E** will be transformed into a convolution matrix **C**. The convolutional layer consists of multiple sets of one dimensional convolutional filters with a different window size for each set. For an instance, one filter with size k is parameterised by a matrix $\mathbf{f}_j \in \mathbb{R}^{d \times k}$ and a bias parameter $b \in \mathbb{R}^n$. A narrow convolutional operation can be formalised as following:

$$\mathbf{C}^j_{[i]} = \mathbf{f}_j^\top \mathbf{E}_{[i,\cdots,i+k-1]} + b_j, \quad i \in \{1, \cdots, l-k+1\} \tag{2.50}$$

where, [.] is the indexing operation and $\{.\}^\top$ indicates the matrix transpose. Note that the bias term b_j here is a different concept from demographic bias (see Section 2.4.1.1). Such an operation is referred to as narrow convolution, and a wide convolution will generate $l+k-1$ columns, accordingly, by concatenating PAD tokens to both sides of the sentence. One can easily induce that $\mathbf{C}^j \in \mathbb{R}^{1 \times (l-k+1)}$.

By combining all the filters, we have

$$\mathbf{C} \in \mathbb{R}^{(l-k+1) \times n} \tag{2.51}$$

where n is the total number of the filters. Each filter captures one type of feature, according to its filter size. For example, a filter $\mathbf{f_1}$ with size 2 could learn the feature of *a cat*, while another filter $\mathbf{f_2}$ with size 3 could capture trigrams like *on the mat*. These filters work like the n-grams, however, convolutional neural nets learn continuous features, and $\mathbf{f_1}$ could also generalise to other similar features such as *a feline*, if *cat* and *feline* have similar word embeddings.

Then, the activation function is applied to **C**. Activation functions are usually non-linear functions which play an important role in neural networks in changing the shape of the input distribution. By combining non-linear layers, neural networks are capable of modelling more complex signals to learn non-linear features of data. There have been many activation functions investigated in neural networks, such as sigmoid and hyperbolic tangent. In Kim

(2014), a rectified linear unit, or ReLU, is used, defined as

$$\text{ReLU}(x) = x^+ = \max(0, x)\,. \tag{2.52}$$

ReLU has been shown to keep the signal stable, and does not suffer from gradient vanishing or explosion in convolution networks. And applying the ReLU to $\mathbf{C}$, we have

$$\mathbf{C}' = \text{ReLU}(\mathbf{C})\,, \tag{2.53}$$

where ReLU is applied element wise to the matrix $\mathbf{C}$.

The next operation is pooling, which is usually applied after convolution filters. Convolution and pooling are a natural combination, as pooling can help convolution networks forget minor signals from unimportant features, for example, $\mathbf{f_1}$ can forget the phrase *mat* .

There are also many different types of pooling operations, such as max pooling, average pooling, and k-max pooling. Kim (2014) chose max-pooling over the filtered vectors, which is defined as the following,

$$\mathbf{p}_i = \max_j(\mathbf{C'}_{j,i}). \tag{2.54}$$

Again, the max operation will reduce the dimension, and $\mathbf{p} \in \mathbb{R}^n$. Note that this is invariant to input size l. After pooling, we see the dimension of learnt representation will be determined by the number of filters, which is a pre-defined hyper-parameter.

Now we have a fixed size $\mathbf{p}$, and this is further transformed to produce $\mathbf{h}$, leading to a hidden representation learned for the input $\mathbf{x}$. It is worth mentioning here that the learnt $\mathbf{h}$ can be used as a feature representation for other tasks, in a similar manner to word embeddings.

The last step is to transfer the learnt representation into the label or target distribution, via the output layer. This can be done by applying a simple fully-connected feed-forward network, or multi-layer perceptron. This step will change the dimension to match that of the target, such as the number of classes for classification. It is parameterised by a weight matrix $\mathbf{w}$, bias b, and the output activation:

$$\mathbf{o} = \text{softmax}(\mathbf{wh} + b) \tag{2.55}$$

Once $\mathbf{o}$ is generated, the loss ($\mathscr{L}$) can be computed, such as using cross-entropy, and optimised using gradient descent methods as described in Section 2.1.

2.6 Summary

In this chapter, I reviewed the literature related to robustness to language variation, domain variation, and demographic variables, and further extended to the problem of fairness in machine learning and privacy-preserving machine learning. Starting with the definition of robustness, the work reviewed here illustrates that many state-of-the-art NLP applications lack robustness to language variation and domain variation. More importantly, these NLP models are often biased and not privacy-preserving, lacking robustness to demographic variables. Therefore, to deal with these questions, algorithms for training robust NLP models are required. In the next chapter, I explain the details of my contributions, to tie each of the research questions.

Chapter 3

Research Contributions

In this chapter, I present the research contributions of this thesis in the form of a series of published papers. Each research question raised in Section 1.2 will be answered via these papers. I will present the papers following the same order as the research questions, that is: robustness to language variations, domain variation, and fairness. Each paper will be summarised by a cover page, in which the key points will be mentioned, including the relevance to the research question, the assumptions behind the paper, the main ideas, the experimental setup, as well as the limitations of the paper.

Following from the detailed research questions, the remainder of this chapter will be organised as below.

1. *To what extent can we learn text representations that are robust to the variability of language?*

   The first research question regarding robustness to language variation can be answered by the first three papers, including two conference papers (Section 3.1 and Section 3.2) and one workshop paper (Section 3.3).

   Section 3.1 tackles the question of learning text representations that are robust to lexical variability. It considers a regularisation method to deal with the out-of-vocabulary problem, obtaining better robustness for word-level deep models. Section 3.2 and Section 3.3 answer the question of how to generate high-quality text variations, and

augment the training data with generated variations to learn robust models. All the methods use in-domain and cross-domain experiments to demonstrate robustness.

2. *To what extent can we learn domain-robust representations?*

   To answer the second research question regarding robustness to domain variation, I will focus on the problem of multi-domain learning, in terms of two conference papers (Sections 3.4 and 3.5).

   These two papers make different assumptions about multi-domain learning, that is whether the model has access to domain labels (Section 3.4) or not (Section 3.5), during training. For models with knowledge of domain information, Section 3.4 uses adversarial learning to learn domain-invariant and domain-specific features, and Section 3.5 proposed a method that jointly models the label and domain. For models with no access to the knowledge of domain information, I parameterise the domain using a latent variable which is learned to gate the domain channels to learn robust multi-domain representations (Section 3.5). Again, both in-domain and cross-domain experiments are considered to demonstrate the robustness of the proposed models.

   Additionally, out-of-domain experiments are also considered in other works, including cross-domain classification (Sections 3.1 and 3.2) and cross-domain POS tagging (Section 3.6).

3. *To what extent can we learn unbiased representations and improve model fairness?*

   The last question will be addressed in a conference paper (Section 3.6). The same technique as proposed in Section 3.4 is used, but here demographic attributes, including the gender and age of users (the authors of text document), are focused on, rather than domain. Experiments show that the proposed model learns representations that are less informative with respect to these attributes, and that the model is more fair in terms of its predictive accuracy for different cohorts.

To evaluate privacy of models, an inference attack towards demographic attributes is applied, including gender, age, and location of the users. The proposed method is shown to be less sensitive to these attacks, thus improving privacy.

3.1 EMNLP 2016: Robustness to Synthetic Noise

Li, Yitong, Trevor Cohn and Timothy Baldwin (2016) Learning Robust Representations of Text, In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, Austin, Texas, USA, pp. 1979–1985.

3.1.1 Research Contributions

First of all, I consider model robustness to lexical variation and aim to investigate how lexical variation can affect deep text methods.

As this paper was conducted early in my Ph.D at a time when deep learning was relatively new to NLP, I assumed word-level neural models in this paper. I based my experiments on the TextCNN model (Kim, 2014) (see Section 2.5.1), which was recognised as a state-of-the-art method at the time. As such, I assume lexical variation can introduce out-of-vocabulary words during inference, which would be treated as unknown tokens. More recently, character level and subword level models are introduced, and I will discuss how to address this problem in Section 4.3.1.

To test the robustness of different approaches, text classification task is conducted under two settings: in-domain and cross-domain. For in-domain evaluation, word dropout noise is applied to synthesize the out-of-vocabulary phenomenon during inference. Cross-domain evaluation is performed to synthesize the real-world inference situation, where the model is trained on the *Movie Review* dataset and tested over the *Customer Review* dataset, and vice versa.

To improve robustness of the model, a new regularisation method was proposed in this paper. This is inspired by stabilizing predictions through minimisation of the ability of features to perturb predictions, which involves optimising over high-order derivatives. To

demonstrate the robustness of the proposed approach, I compare against other regularisation methods, including dropout training. The proposed method can improve the robustness of experimented models when the dropout noise increases, and it is complementary to the dropout regulariser.

Learning Robust Representations of Text

Yitong Li **Trevor Cohn** **Timothy Baldwin**
Department of Computing and Information Systems
The University of Melbourne, Australia
yitongl4@student.unimelb.edu.au, {tcohn,tbaldwin}@unimelb.edu.au

Abstract

Deep neural networks have achieved remarkable results across many language processing tasks, however these methods are highly sensitive to noise and adversarial attacks. We present a regularization based method for limiting network sensitivity to its inputs, inspired by ideas from computer vision, thus learning models that are more robust. Empirical evaluation over a range of sentiment datasets with a convolutional neural network shows that, compared to a baseline model and the dropout method, our method achieves superior performance over noisy inputs and out-of-domain data.[1]

1 Introduction

Deep learning has achieved state-of-the-art results across a range of computer vision (Krizhevsky et al., 2012), speech recognition (Graves et al., 2013) and natural language processing tasks (Bahdanau et al., 2015; Kalchbrenner et al., 2014; Yih et al., 2014; Bitvai and Cohn, 2015). However, deep models are often overconfident for noisy test instances, making them susceptible to adversarial attacks (Nguyen et al., 2015; Tabacof and Valle, 2016). Goodfellow et al. (2014) argued that the primary cause of neural networks' vulnerability to adversarial perturbation is their linear nature, due to neural models being intentionally designed to behave in a mostly linear manner to facilitate optimization. Fawzi et al. (2015) provided a theoretical framework for analyzing the robustness of classifiers to adversarial perturbations, and also showed linear models are usually not robust to adversarial noise.

In this work, we present a regularization method which makes deep learning models more robust to noise, inspired by Rifai et al. (2011). The intuition behind the approach is to stabilize predictions by minimizing the ability of features to perturb predictions, based on high-order derivatives. Rifai et al. (2011) introduced contractive auto-encoders based on similar ideas, using the Frobenius norm of the Jacobian matrix as a penalty term to extract robust features. Further, Gu and Rigazio (2014) introduced deep contractive networks, generalizing this idea to a feed-forward neural network. Also related, Martens (2010) investigated a second-order optimization method based on Hessian-free approach for training deep auto-encoders. Where our proposed approach differs is that we train models using first-order derivatives of the training loss as part of a regularization term, necessitating second-order derivatives for computing the gradient. We empirically demonstrate the effectiveness of the model over text corpora with increasing amounts of artificial masking noise, using a range of sentiment analysis datasets (Pang and Lee, 2008) with a convolutional neural network model (Kim, 2014). In this, we show that our method is superior to dropout (Srivastava et al., 2014) and a baseline method using MAP training.

2 Training for Robustness

Our method introduces a regularization term during training to ensure model robustness. We develop our approach based on a general class of parametric mod-

[1] Implementation available at https://github.com/lrank/Robust-Representation.

Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 1979–1985,
Austin, Texas, November 1-5, 2016.

els, with the following structure. Let $\mathbf{x}$ be the input, which is a sequence of (discrete) words, represented by a fixed-size vector of continuous values, $\mathbf{h}$. A transfer function takes $\mathbf{h}$ as input and produces an output distribution, $\mathbf{y}_{\text{pred}}$. Training proceeds using stochastic gradient descent to minimize a loss function L, measuring the difference between $\mathbf{y}_{\text{pred}}$ and the truth $\mathbf{y}_{\text{true}}$.

The purpose of our work is to learn neural models which are more robust to strange or invalid inputs. When small perturbations are applied on $\mathbf{x}$, we want the prediction $\mathbf{y}_{\text{pred}}$ to remain stable. Text can be highly variable, allowing for the same information to be conveyed with different word choice, different syntactic structures, typographical errors, stylistic changes, etc. This is a particular problem in transfer learning scenarios such as domain adaptation, where the inputs in distinct domains are drawn from related, but different, distributions. A good model should be robust to these kinds of small changes to the input, and produce reliable and stable predictions.

Next we discuss methods for learning models which are robust to variations in the input, before providing details of the neural network model used in our experimental evaluation.

2.1 Conventional Regularization and Dropout

Conventional methods for learning robust models include l_1 and l_2 regularization (Ng, 2004), and dropout (Srivastava et al., 2014). In fact, Wager et al. (2013) showed that the dropout regularizer is first-order equivalent to an l_2 regularizer applied after scaling the features. Dropout is also equivalent to "Follow the Perturbed Leader" (FPL) which perturbs exponential numbers of experts by noise and then predicts with the expert of minimum perturbed loss for online learning robustness (van Erven et al., 2014). Given its popularity in deep learning, we take dropout to be a strong baseline in our evaluation.

The key idea behind dropout is to randomly zero out units, along with their connections, from the network during training, thus limiting the extent of co-adaptation between units. We apply dropout on the representation vector $\mathbf{h}$, denoted $\hat{\mathbf{h}} = \text{dropout}_\beta(\mathbf{h})$, where β is the dropout rate. Similarly to our proposed method, training with dropout requires gradient based search for the minimizer of the loss L.

We also use dropout to generate noise in the test data as part of our experimental simulations, as we will discuss later.

2.2 Robust Regularization

Our method is inspired by the work on adversarial training in computer vision (Goodfellow et al., 2014). In image recognition tasks, small distortions that are indiscernible to humans can significantly distort the predictions of neural networks (Szegedy et al., 2014). An intuitive explanation of our regularization method is, when noise is applied to the data, the variation of the output is kept lower than the noise. We adapt this idea from Rifai et al. (2011) and develop the Jacobian regularization method.

The proposed regularization method works as follows. Conventional training seeks to minimise the difference between $\mathbf{y}_{\text{true}}$ and $\mathbf{y}_{\text{pred}}$. However, in order to make our model robust against noise, we also want to minimize the variation of the output when noise is applied to the input. This is to say, when perturbations are applied to the input, there should be as little perturbation in the output as possible. Formally, the perturbations of output can be written as $\mathbf{p}_{\text{y}} = \text{M}(\mathbf{x}+\mathbf{p}_{\text{x}}) - \text{M}(\mathbf{x})$, where $\mathbf{x}$ is the input, $\mathbf{p}_{\text{x}}$ is the vector of perturbations applied to $\mathbf{x}$, M expresses the trained model, $\mathbf{p}_{\text{y}}$ is the vector of perturbations generated by the model, and the output distribution $\mathbf{y} = \text{M}(\mathbf{x})$. Therefore

$$\lim_{\mathbf{p_x}\to\mathbf{0}} \mathbf{p_y} = \lim_{\mathbf{p_x}\to\mathbf{0}} \big(\text{M}(\mathbf{x}+\mathbf{p}_{\text{x}}) - \text{M}(\mathbf{x})\big) = \frac{\partial \mathbf{y}}{\partial \mathbf{x}} \cdot \mathbf{p_x},$$

$$\text{and} \quad \text{distance}\left(\lim_{\mathbf{p_x}\to\mathbf{0}} \mathbf{p}_{\text{y}}/\mathbf{p}_{\text{x}}, \mathbf{0}\right) = \left\| \frac{\partial \mathbf{y}}{\partial \mathbf{x}} \right\|_F .$$

In other words, minimising local noise sensitivity is equivalent to minimising the Frobenius norm of the Jacobean matrix of partial derivatives of the model outputs *wrt* its inputs.

To minimize the effect of perturbation noise, our method involves an additional term in the loss function, in the form of the derivative of loss L with respect to hidden layer $\mathbf{h}$. Note that while in principle we could consider robustness to perturbations in the input $\mathbf{x}$, the discrete nature of $\mathbf{x}$ adds additional mathematical complications, and thus we defer this setting for future work. Combining the elements, the new loss function can be expressed as

$$\mathscr{L} = L + \lambda \cdot \left\| \frac{\partial L}{\partial \mathbf{h}} \right\|_2 , \tag{1}$$

where λ is a weight term, and $\mathrm{distance}$ takes the form of the l_2 norm. The training objective in Equation (1) supports gradient optimization, but note that it requires the calculation of second-order derivatives of L during back propagation, arising from the $\partial L / \partial \mathbf{h}$ term. Henceforth we refer to this method as **robust regularization**.

2.3 Convolutional Network

For the purposes of this paper, we focus exclusively on convolutional neural networks (CNNs), but stress that the method is compatible with other neural architectures and other types of parametric models (not just deep neural networks). The CNN used in this research is based on the model proposed by Kim (2014), and is outlined below.

Let S be the sentence, consisting of n words $\{w_1, w_2, \cdots, w_n\}$. A look-up table is applied to S, made up of word vectors $\mathbf{e}_{\mathrm{i}} \in \mathbb{R}^m$ corresponding to each word w_i, where m is the word vector dimensionality. Thus, sentence S can be represented as a matrix $\mathbf{E}_{\mathrm{S}} \in \mathbb{R}^{m \times n}$ by concatenating the word vectors $\mathbf{E}_{\mathrm{S}} = \bigoplus_{i=1}^{n} \mathbf{e}_{\mathrm{w_i}}$.

A convolutional layer combined with a number of wide convolutional filters is applied to $\mathbf{E}_{\mathrm{S}}$. Specifically, the k-th convolutional filter operator filter_k involves a weight vector $\mathbf{w}_{\mathrm{k}} \in \mathbb{R}^{m \times t}$, which works on every t_k-sized window of $\mathbf{E}_{\mathrm{S}}$, and is accompanied by a bias term $b \in \mathbb{R}$. The filter operator is followed by the non-linear function $\mathscr{F}$, a rectified linear unit, ReLU, followed by a max-pooling operation, to generate a hidden activation $h_k = \mathrm{MaxPooling}(\mathscr{F}(\mathrm{filter}_k(\mathbf{E}_{\mathrm{S}}; \mathbf{w}_{\mathrm{k}}, b))$. Multiple filters with different window sizes are used to learn different local properties of the sentence. We concatenate all the hidden activations h_k to form a hidden layer $\mathbf{h}$, with size equal to the number of filters. Details of parameter settings can be found in Section 3.2.

The feature vector $\mathbf{h}$ is fed into a final $\mathrm{softmax}$ layer with a linear transform to generate a probability distribution over labels

$$\mathbf{y}_{\mathrm{pred}} = \mathrm{softmax}(\mathbf{w} \cdot \mathbf{h} + \mathbf{b}),$$

where $\mathbf{w}$ and $\mathbf{b}$ are parameters. Finally, the model minimizes the loss of the cross-entropy between the ground-truth and the model prediction, $L = \mathrm{CrossEntropy}(\mathbf{y}_{\mathrm{true}}, \mathbf{y}_{\mathrm{pred}})$, for which we use stochastic gradient descent.

3 Datasets and Experimental Setups

We experiment on the following datasets,[2] following Kim (2014):

- MR: Sentence polarity dataset (Pang and Lee, 2008)[3]
- Subj: Subjectivity dataset (Pang and Lee, 2005)[3]
- CR: Customer review dataset (Hu and Liu, 2004)[4]
- SST: Stanford Sentiment Treebank, using the 3-class configuration (Socher et al., 2013)[5]

In each case, we evaluate using classification accuracy.

3.1 Noisifying the Data

Different to conventional evaluation, we corrupt the test data with noise in order to evaluate the robustness of our model. We assume that when dealing with short text such as Twitter posts, it is common to see unknown words due to typos, abbreviations and sociolinguistic marking of different types (Han and Baldwin, 2011; Eisenstein, 2013). To simulate this, we apply word-level dropout noise to each document, by randomly replacing words by a unique sentinel symbol.[6] This is applied to each word with probability $\alpha \in \{0, 0.1, 0.2, 0.3\}$.

We also experimented with adding different levels of Gaussian noise to the sentence embeddings $\mathbf{E}_{\mathrm{S}}$, but found the results to be largely consistent with those for word dropout noise, and therefore we have omitted these results from the paper.

To directly test the robustness under a more realistic setting, we additionally perform cross-domain evaluation, where we train a model on one dataset

[2] For datasets where there is no pre-defined training/test split, we evaluate using 10-fold cross validation. Refer to Kim (2014) for more details on the datasets.

[3] https://www.cs.cornell.edu/people/pabo/movie-review-data/

[4] http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html

[5] http://nlp.stanford.edu/sentiment/

[6] This was to avoid creating new n-grams which would occur when symbols are deleted from the input. Masking tokens instead results in partially masked n-grams as input to the convolutional filters.

Dataset		MR				Subj			
Word dropout rate (α)		0	0.1	0.2	0.3	0	0.1	0.2	0.3
Baseline		80.5	79.4	77.9	76.5	**93.1**	92.0	90.9	89.8
Dropout (β)	0.3	80.3	79.5	78.1	76.7	92.7	92.0	90.9	89.5
	0.5	80.3	79.0	78.0	76.5	93.0	92.0	91.1	89.9
	0.7	80.3	79.3	78.3	76.8	92.8	91.9	90.9	89.8
Robust Regularization (λ)	10^{-3}	80.5	79.4	78.3	76.7	93.0	**92.2**	91.1	89.8
	10^{-2}	**80.8**	79.3	78.4	77.0	93.0	92.2	91.0	90.0
	10^{-1}	80.4	78.8	77.8	77.0	92.7	91.9	91.0	89.8
	1	79.3	77.1	76.1	75.5	91.7	91.1	90.1	89.3
Dropout + Robust	$\beta = 0.5, \lambda = 10^{-2}$	80.6	**79.9**	**78.6**	**77.3**	93.0	**92.2**	**91.2**	**90.1**

Dataset		CR				SST			
Word dropout rate (α)		0	0.1	0.2	0.3	0	0.1	0.2	0.3
Baseline		83.2	82.3	80.4	77.9	84.1	82.3	80.3	77.8
Dropout (β)	0.3	83.3	82.1	80.3	78.9	84.2	82.3	80.2	78.0
	0.5	83.2	82.4	81.0	79.3	84.2	82.4	80.5	78.2
	0.7	83.2	82.2	80.7	78.8	83.9	82.5	80.9	78.2
Robust Regularization (λ)	10^{-3}	83.3	82.6	81.4	79.5	**84.5**	**82.8**	**81.4**	78.8
	10^{-2}	**83.4**	82.5	81.6	79.3	84.2	82.4	80.7	78.6
	10^{-1}	83.3	**82.7**	**82.0**	79.6	82.5	81.5	79.7	77.6
	1	82.9	81.4	79.8	79.0	82.2	80.9	79.1	77.3
Dropout + Robust	$\beta = 0.5, \lambda = 10^{-2}$	83.3	82.5	81.5	**79.7**	84.3	82.6	80.8	**79.1**

Table 1: Accuracy (%) with increasing word-level dropout across the four datasets. For each dataset, we apply four levels of noise $\alpha = \{0, 0.1, 0.2, 0.3\}$; the best result for each combination of α and dataset is indicated in **bold**. The Baseline model is a simple CNN model without regularization. The last model combines dropout and our method with fixed parameters β and λ as indicated.

and apply it to another. For this, we use the pairing of MR and CR, where the first dataset is based on movie reviews and the second on product reviews, but both use the same label set. Note that there is a significant domain shift between these corpora, due to the very nature of the items reviewed.

3.2 Word Vectors and Hyper-parameters

To set the hyper-parameters of the CNN, we follow the guidelines of Zhang and Wallace (2015), setting word embeddings to $m = 300$ dimensions and initialising based on word2vec pre-training (Mikolov et al., 2013). Words not in the pre-trained vector table were initialized randomly by the uniform distribution $U([-0.25, 0.25)^m)$. The window sizes of filters (t) are set to $3, 4, 5$, with 128 filters for each size, resulting in a hidden layer dimensionality of $384 = 128 \times 3$. We use the Adam optimizer (Kingma and Ba, 2015) for training.

4 Results and Discussions

The results for word-level dropout noise are presented in Table 1. In general, increasing the word-level dropout noise leads to a drop in accuracy for all four datasets, however the relative dropoff in accuracy for Robust Regularization is less than for Word Dropout, and in 15 out of 16 cases (four noise levels across the four datasets), our method achieves the best result. Note that this includes the case of $\alpha = 0$, where the test data is left in its original form, which shows that Robust Regularization is also an effective means of preventing overfitting in the model.

For each dataset, we also evaluated based on the combination of Word Dropout and Robust Regularization using the fixed parameters $\beta = 0.5$ and $\lambda = 10^{-2}$, which are overall the best individual settings. The combined approach performs better than either individual method for the highest noise levels tested across all datasets. This indicates that Robust

Train/Test		MR/CR	CR/MR
Baseline		67.5	61.0
Dropout (β)	0.3	71.6	62.2
	0.5	71.0	62.1
	0.7	70.9	62.0
Robust Regularization (λ)	10^{-3}	70.8	61.6
	10^{-2}	71.1	**62.5**
	10^{-1}	**72.0**	62.2
	1	71.8	62.3
Dropout + Robust	$\beta = 0.5, \lambda = 10^{-2}$	**72.0**	62.4

Table 2: Accuracy under cross-domain evaluation; the best result for each dataset is indicated in **bold**.

Regularization acts in a complementary way to Word Dropout.

Table 2 presents the results of the cross-domain experiment, whereby we train a model on MR and test on CR, and vice versa, to measure the robustness of the different regularization methods in a more real-world setting. Once again, we see that our regularization method is superior to word-level dropout and the baseline CNN, and the techniques combined do very well, consistent with our findings for synthetic noise.

4.1 Running Time

Our method requires second-order derivatives, and thus is a little slower at training time. Figure 1 is a plot of the training and test accuracy at varying points during training over SST.

We can see that the runtime till convergence is only slightly slower for Robust Regularization than standard training, at roughly 30 minutes on a two-core CPU (one fold) with standard training vs. 35–40 minutes with Robust Regularization. The convergence time for Robust Regularization is comparable to that for Word Dropout.

5 Conclusions

In this paper, we present a robust regularization method which explicitly minimises a neural model's sensitivity to small changes in its hidden representation. Based on evaluation over four sentiment analysis datasets using convolutional neural networks, we found our method to be both superior and complementary to conventional word-level dropout under varying levels of noise, and in a cross-domain evaluation.

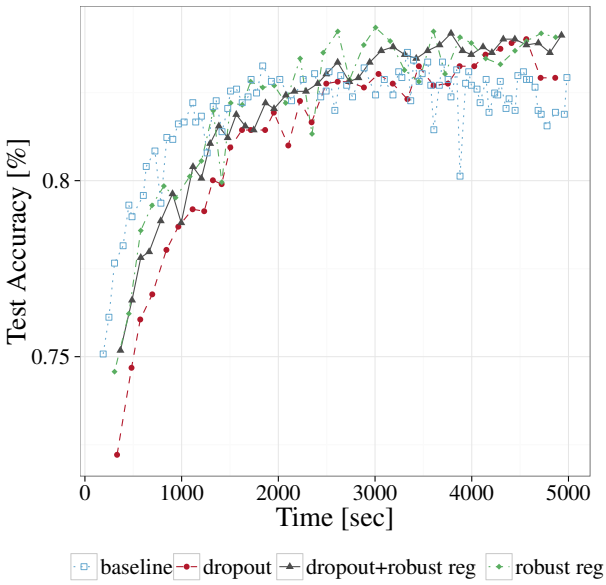


Figure 1: Time–accuracy evaluation over the different combinations of Word Dropout (dropout) and Robust Regularization (robust reg) over SST, without injecting noise.

For future work, we plan to apply our regularization method to other models and tasks to determine how generally applicable our method is. Also, we will explore methods for more realistic linguistic noise, such as lexical, syntactic and semantic noise, to develop models that are robust to the kinds of data often encountered at test time.

Acknowledgments

We are grateful to the anonymous reviewers for their helpful feedback and suggestions. This work was supported by the Australian Research Council (grant numbers FT130101105 and FT120100658). Also, we would like to thank the developers of Tensorflow (Abadi et al., 2015), which was used for the experiments in this paper.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg,

Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Technical report, Google Research.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*.

Zsolt Bitvai and Trevor Cohn. 2015. Non-linear text regression with a deep convolutional neural network. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers)*, pages 180–185.

Jacob Eisenstein. 2013. What to do about bad language on the internet. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 359–369.

Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. 2015. Analysis of classifiers' robustness to adversarial perturbations. *arXiv preprint arXiv:1502.02590*.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. In *Proceedings of the International Conference on Learning Representations*.

Alan Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649.

Shixiang Gu and Luca Rigazio. 2014. Towards deep neural network architectures robust to adversarial examples. In *Proceedings of the NIPS 2014 Deep Learning and Representation Learning Workshop*.

Bo Han and Timothy Baldwin. 2011. Lexical normalisation of short text messages: Makn sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 368–378.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 168–177.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 655–665.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105.

James Martens. 2010. Deep learning via Hessian-free optimization. In *Proceedings of the 27th International Conference on Machine Learning*, pages 735–742.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119.

Andrew Y. Ng. 2004. Feature selection, L1 vs. L2 regularization, and rotational invariance. In *Proceedings of the Twenty-first International Conference on Machine Learning*.

Anh Nguyen, Jason Yosinski, and Jeff Clune. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Bo Pang and Lillian Lee. 2005. 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.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135.

Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. 2011. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th International Conference on Machine Learning*, pages 833–840.

Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob

Fergus. 2014. Intriguing properties of neural networks. In *Proceedings of the International Conference on Learning Representations*.

Pedro Tabacof and Eduardo Valle. 2016. Exploring the space of adversarial images. In *Proceedings of the IEEE International Joint Conference on Neural Networks*.

Tim van Erven, Wojciech Kotłowski, and Manfred K. Warmuth. 2014. Follow the leader with dropout perturbations. In *Proceedings of the 27th Conference on Learning Theory*, pages 949–974.

Stefan Wager, Sida Wang, and Percy S. Liang. 2013. Dropout training as adaptive regularization. In *Advances in Neural Information Processing Systems 26*, pages 351–359.

Wen-tau Yih, Xiaodong He, and Christopher Meek. 2014. Semantic parsing for single-relation question answering. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers)*, pages 643–648.

Ye Zhang and Byron Wallace. 2015. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*.

3.2 EACL 2017: Generating Language Variations via Linguistic View

Li, Yitong, Trevor Cohn and Timothy Baldwin (2017) Robust Training under Linguistic Adversity, In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers (EACL 2017)*, Valencia, Spain, pp. 21–27.

3.2.1 Research Contributions

Following on from the simple assumption of lexical variation (Section 3.1), I extend the scope to other forms of language variation, such as syntactic variation. In this paper, I investigate a range of approaches to generating variations of a given sentence, including both semantic and syntactic approaches. In this, I aim to answer the question of learning robust representations by augmenting the training data using such approaches.

In this paper, I focus on linguistically-motivated methods to generate sentence variations, using both semantic and syntactic rules based on a range of computational linguistic resources, and subject to the constraint that the semantics of the given sentence does not change under such variations. I also applied a language model to ensure the fluency of the generated variations.

Based on these variations, an augmentation training method is proposed which leads to better robustness by learning from the generated sentences. To demonstrate the robustness of the proposed augmentation method, I conduct the same experimental setup as in the first paper (Section 3.1), where both in-domain and cross-domain experiments are considered. The proposed augmentation methods show better generalisation and robustness for both in-domain and cross-domain scenarios. For future work, I consider to apply the proposed approach to other language tasks and different models, and I give more details in Section 4.3.

Robust Training under Linguistic Adversity

Yitong Li and **Trevor Cohn** and **Timothy Baldwin**
Department of Computing and Information Systems
The University of Melbourne, Australia
yitongl4@student.unimelb.edu.au, {tcohn,tbaldwin}@unimelb.edu.au

Abstract

Deep neural networks have achieved remarkable results across many language processing tasks, however they have been shown to be susceptible to overfitting and highly sensitive to noise, including adversarial attacks. In this work, we propose a linguistically-motivated approach for training robust models based on exposing the model to corrupted text examples at training time. We consider several flavours of linguistically plausible corruption, include lexical semantic and syntactic methods. Empirically, we evaluate our method with a convolutional neural model across a range of sentiment analysis datasets. Compared with a baseline and the dropout method, our method achieves better overall performance.

1 Introduction

Deep learning has achieved state-of-the-art results across a range of computer vision (Krizhevsky et al., 2012), speech recognition (Graves et al., 2013), and natural language processing tasks (Bahdanau et al., 2015; Kalchbrenner et al., 2014; Bitvai and Cohn, 2015). However, deep models tend to be overconfident in their predictions over noisy test instances, including adversarial examples (Szegedy et al., 2014; Goodfellow et al., 2015). A range of methods have been proposed to train models to be more robust, such as injecting noise into the data and hidden layers (Jiang et al., 2009), dropout (Srivastava et al., 2014), and the incorporation of explicit regularization terms into the training objective (Ng, 2004; Li et al., 2016).

In this work, we propose a linguistically-motivated method customised to text applications, based on injecting different kinds of word- and sentence-level linguistic noise into the input text, inspired by adversarial examples (Goodfellow et al., 2015). Our method has its origins in computer vision, where it has been shown that small pixel perturbations indiscernible to humans can significantly distort the predictions of state-of-the-art deep models (Szegedy et al., 2014; Nguyen et al., 2015), an observation that has been harnessed in recent work on adversarial training (Goodfellow et al., 2015). This kind of noise is cheap to generate for images and is transferable between different models, but it is less clear how to generate analogous textual noise while preserving the fidelity of the training data, due to text being discrete and sequential in nature, with latent syntactic structure. Based on the same linguistic intuition, adversarial evaluation for natural language processing models was proposed by Smith (2012). Also, adversarial learning for text, such as perceptron learning (Søgaard, 2013) and unsupervised estimation methods (Smith and Eisner, 2005), have been studied in the language area.

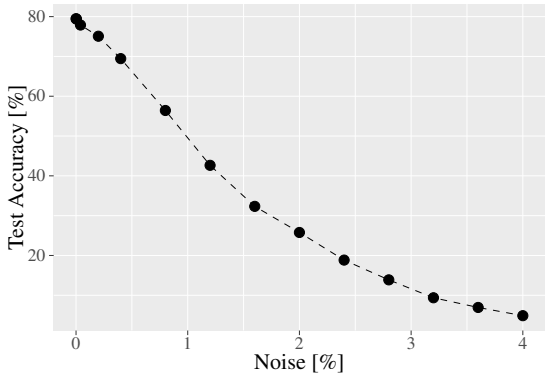


Figure 1: Accuracy (%) drops as we increase adversarial noise to word embeddings, as evaluated on binary classification dataset MR.

Word embeddings learned from WORD2VEC

Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 21–27,
Valencia, Spain, April 3-7, 2017.

(Mikolov et al., 2013) and GLOVE (Pennington et al., 2014) are now widely used as input to language processing models, however these representations are highly susceptible to noise. For example, Figure 1 shows that as we add adversarial noise $\eta = \epsilon \nabla_x Loss(x, y, \theta)$ to WORD2VEC representations, classification accuracy for a convolutional model (Kim, 2014) over a sentiment classification task (Pang and Lee, 2008) drops appreciably, such that with only 1% perturbations, a state-of-the-art model drops to the level of a random classifier.

Word embeddings are not an intuitive representation of human language, and it is not immediately clear how to generate adversarial noise over the raw text input without affecting the fidelity of the data. In human-to-human textual communication such as chat and microblogs, humans are remarkably resilient to "noise", in terms of typos, lexical and syntactic disfluencies, and the large variety of semantically-equivalent ways of expressing the same content (Han and Baldwin, 2011; Eisenstein, 2013; Baldwin et al., 2013; Pavlick and Callison-Burch, 2016). These observations are the inspiration for this work, in proposing a training strategy based on the explicit generation of linguistic corruption over the source training instances, to train robust text models. Empirically, we demonstrate the effectiveness of our method over a range of sentiment analysis datasets using a state-of-the-art convolutional neural network model (Kim, 2014). In this, we show that our method is superior to a baseline and dropout (Srivastava et al., 2014) using MAP training.[1]

2 Generating Text Noise

Our method involves the explicit generation of several kinds of linguistic corruption, to train more robust deep models. The first question is how to generate the linguistic noise, focusing on English for the purposes of this paper. We focus on the generation of two classes of text noise: (1) syntactic noise; and (2) semantic noise.[2]

Syntactic Noise The first class of linguistic noise is syntactic, focusing on the syntactic structure of the input, either through explicit parsing and generation using a deep linguistic parser, or sentence compression.

For the deep linguistic parser, we use the LinGO English Resource Grammar ("ERG": Copestake and Flickinger (2000)) with the ACE parser, based on pyDelphin.[3] The ERG supports both parsing and generation, via the semantic formalism of Minimal Recursion Semantics ("MRS": Copestake et al. (2005)). To generate paraphrases with the ERG, we simply parse a given input, select the preferred parse using a pretrained parse selection model (Oepen et al., 2002), and exhaustively generate from the resultant MRS. We then use uniform random sampling to select from the generator outputs, which potentially numbers in the thousands of variants. To handle unknown words during parsing and generation, we use POS mapping and introduce a unique relation for each unknown word, which we use to substitute the unknown word back in to the generation output. In practice, the primary sources of "noise" introduced by the ERG are due to topicalisation, adjective ordering, fronting of adverbial phrases, and relativisation of modifiers.

The second approach to syntactic noise is based on sentence compression ("COMP": Knight and Marcu (2000)), which aims to "trim" an input of peripheral content, while maintaining grammaticality, and also the syntax of the original as much as possible. While the state-of-the-art in sentence compression is based on deep learning methods such as recurrent neural networks (Filippova et al., 2015), we implement a simple parser-based model, due to the lack of large-scale annotated data for training and the fact that a relative lack of precision in the output may ultimately help our method. First, we parse the sentence using the Stanford CoreNLP constituency parser (Chen and Manning, 2014). Next, we model the conditional probability of deleting a sub-tree C with label S given its parent node with label R by $p(C|S,R) = \frac{p(C,S,R)}{\Sigma_C p(C,S,R)}$, trained on the sentence compression corpora of Clarke and Lapata (2006),[4] made up of a few hundred labelled instances.

Semantic Noise The second class of linguistic noise is semantic noise. Semantic noise is more subtle than syntactic noise, as we must be careful

[1] The implementation is freely available at `https://github.com/lrank/Linguistic_adversity`.

[2] We also experimented with a method which generates lexical noise, but for the purposes of our experiments here, as the vast majority of the generated candidates are OOV words, it is largely equivalent to word dropout, and omitted from this paper.

[3] `https://github.com/delph-in/pydelphin`

[4] `http://jamesclarke.net/research/resources/`

not to impact on the fidelity of the original labels, which can readily occur with full paraphrasing or abstractive summarisation. As such, we focus on lexical substitution of near-synonyms of words in the original text, and experiment with two methods for generating near-synonyms.

Our approach to generating semantic noise proceeds as follows. First, we apply filters to identify words which should not be candidates for lexical substitution, namely words which are parts of named entities or function words. As such, we use the Stanford CoreNLP POS tagger and named entity recogniser (Finkel et al., 2005; Chen and Manning, 2014), and identify "substitutable words" as those which are nouns, verbs, adjectives or adverbs, and not part of a named entity. For each substitutable word w, we generate the set of substitution candidates $s(w)$. For each candidate $w_i \in \{w\} \cup s(w)$ we allow the original word to be preserved with $p(w_i) = \alpha$, and share the remaining $1-\alpha$ proportional to the language model score based on substituting w_i into the original text. For this, we use the pre-trained US English language model from the CMU Sphinx Speech Recognition toolkit.[5] Finally, we sample from the probability distribution $\{p(w_i) : w_i \in \{w\} \cup s(w)\}$ for each substitutable word w to generate a semantically-corrupted version of the original.

We experiment with two approaches to generating the substitution candidates. The first is based on Princeton WordNet ("WN": Miller et al. (1990)), over all synsets that a given substitutable word occurs in, using the NLTK API (Bird, 2006). The second is based on the "counter-fitting" method of Mrkšić et al. (2016) ("CFIT"), whereby word embeddings from WORD2VEC are projected based on a supervised objective function which penalises similarity between antonym pairs, and rewards similarity between synonym pairs, as trained on 10k English news sentences from WMT14 (Bojar et al., 2014).

Word Dropout As a standard approach to training robust models, we use word dropout (Srivastava et al., 2014; Pham et al., 2014). Dropout can be viewed as a method for zeroing out noise, and is first-order equivalent to an ℓ_2 regularizer applied after feature scaling (Wager et al., 2013).

[5] https://sourceforge.net/projects/cmusphinx/

Method	Example
Original	The cat sat on the mat .
ERG	On the mat sat the cat .
COMP	The cat sat on ◇ mat ◇
WN	The kat sat on the flatness .
CFIT	The pet stood onto the mat .

Table 1: Examples of generated sentences across four proposed methods. Modified words are marked by "underwave" and omitted words are denoted with a "◇".

Table 1 shows an example sentence and sample corrupted outputs after applying each type of linguistic noise. The ERG seldom changes words, and instead tends to reorder the words based on syntactic alternation. COMP performs like word dropout in that it tends to remove tokens with low semantic content and to generate complete sentences. WN and CFIT both only modify the text at the word level, based on near-synonyms and words with similar semantic function, respectively.

3 Models and Training

We evaluate our methods on several sentence classification tasks, using a convolutional neural network ("CNN") model (Kim, 2014). Note that our method corrupts the input directly, and is thus easily transferrable to other classes of models (e.g., other deep learning or linear models).

Convolutional Neural Network The CNN operates at the sentence level by first embedding each word using a lookup table which is stacked into the sentence matrix $\mathbf{E}_{\mathrm{S}}$. A 1d convolutional layer is then applied to $\mathbf{E}_{\mathrm{S}}$, which applies a series of filters over each window of t words, with each filter employing a rectifier transform function. $\mathrm{MaxPooling}$ is applied over each set of filter outputs to result in a fixed-size sentence representation.[6] The sentence vector is fed into a final $\mathrm{Softmax}$ layer to generate a probability distribution over classification labels.

The model is trained to minimise the cross-entropy between the ground-truth and the model prediction, using the Adam Optimizer (Kingma and Ba, 2015) with learning rate 10^{-4} and a

[6] We use window widths of size $t \in \{3, 4, 5\}$, and 128 filters for each size. $\mathrm{MaxPooling}$ is applied to each of the three sizes separately, and the resulting vectors are concatenated to form the sentence representation.

batch size of 128. We initialise the embedding with dimension $m = 300$ Google pre-trained WORD2VEC word embeddings (Mikolov et al., 2013). Words not in the pre-trained vocabulary are initialized randomly using a uniform distribution $U([-0.25, 0.25)^m)$.

Injecting Noise during Training Our proposed method involves corrupting the training input with adversarial noise of various kinds. All the methods are non-deterministic, involving random sampling. They are applied afresh every epoch, such that each time an instance is processed, it will have a different input form.[7] The two semantic approaches (WN and CFIT) support configurable noise rates in terms of the proportion of substitutable words that are corrupted. Accordingly, we experiment with two thresholds on the random variable for substitution of each word: low ("lo"; $\alpha = 0.5$) and high ("hi"; $\alpha = 0$). Besides the above methods which employ a single type noise, we experiment with a combination (COMB) of the four different noise types (ERG + COMP + WN_{lo} + CFIT_{lo}), by uniformly randomly choosing one of the four methods for noise generation each time we process a training instance.

Datasets We experiment on the following datasets:

- MR: sentence polarity dataset from movie reviews (Pang and Lee, 2008)[8]
- CR: customer review dataset (Hu and Liu, 2004)[9]
- Subj: subjectivity dataset (Pang and Lee, 2005)[8]
- SST: Stanford Sentiment Treebank, using the 2-class configuration (Socher et al., 2013)[10]

We evaluate using classification accuracy, based on both in-domain evaluation[11] and a cross-domain setting, in which we evaluate a model trained on MR and tested on CR, and vice versa. This last setting characterises a realistic application scenario, where robustness to vocabulary shift and other differences in the input is paramount.

[7] Using a single application of noise is less effective, but still yields improvements over baseline methods including dropout.

[8] https://www.cs.cornell.edu/people/pabo/movie-review-data/

[9] http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html

[10] http://nlp.stanford.edu/sentiment/

[11] Where there is no pre-defined training/test split for a given dataset, we use 10-fold cross validation. See Kim (2014) for more details on the datasets and evaluation settings.

4 Experimental Results and Analysis

Table 2 presents the results of training with different sources of linguistic corruption in the in-domain and cross-domain settings. In general, the proposed methods perform better than the baseline and dropout, and semantic noise using WN achieves consistent improvements across all settings. The COMB method uniformly outperforms the other methods for all in-domain evaluations, indicating that the improvements from training with different types of noise are orthogonal. Note that improvements are smaller on SST and MR than CR and Subj for all methods. Almost every method improves over word dropout, except counter-fitting at a high noise level. Also surprising is the fact that dropout shows no improvement over standard training, and is overall mildly detrimental.

Our intuition behind why WN consistently outperforms the baseline methods and other single sources of noise is it sometimes performs similarity to dropout, in replacing common words with rare ones, and sometimes substitutes frequent words for frequent words, leading to better generalisation in the word embeddings. To test this hypothesis, we computed nearest neighbours in the word embedding space for both the baseline method and the WN method. For example, the top-3 nearest neighbours for *superior* in CR are *exceptional*, *excellent* and *unmatched* for WN, while for the baseline, they are *inferior*, *exceptional* and *excellent*. That is, similar to the intuition behind counter-fitting, the methods appears to learn to differentiate between synonyms and antonyms, in a manner which is sensitised to the target domain.

Although similar in function to WN, the counter-fitting based method performs unexpectedly poorly. This appears to be a consequence of the training of these embeddings, namely that the corpus was much smaller than that used for the WORD2VEC training, and consequently coverage on our corpora was substantially lower, leading to the approach making inappropriate substitutions and not aiding model robustness.

Sentence compression was found to be highly effective. To illustrate by example, the sentence *Player has a problem with dual-layer dvd's such*

Method	In domain				Cross domain	
	MR	CR	Subj	SST	MR/CR	CR/MR
baseline	80.4	82.6	92.4	84.5	67.0	67.2
dropout	80.1	82.4	92.6	84.5	67.7	67.4
ERG	80.0	82.8	92.9	84.4	68.1	67.3
COMP	79.5	83.1	93.2	84.3	68.1	**67.5**
WN_{lo}	80.9	83.2	93.1	84.3	68.5	67.3
WN_{hi}	81.2	83.8	92.9	84.6	67.9	**67.5**
$CFIT_{lo}$	79.8	82.7	92.6	84.1	**68.9**	67.3
$CFIT_{hi}$	76.2	78.9	91.0	80.3	67.4	64.2
COMB	**81.4**	**84.3**	**93.6**	**84.8**	68.4	67.4

Table 2: Accuracy (%) of the CNN, in four in-domain settings, and two cross-domain settings, with word dropout ("dropout"), or linguistic corruption based on different sources of syntactic and semantic corruption. The best result for each dataset is indicated in **bold**.

as Alias seasons 1 and season 2 is compressed into *has a problem with dual-layer dvd* which preserves the key information that we expect to be useful for model learning. This allows the model to better learn the components of the input that are predictive of sentiment.

Syntactic paraphrasing (ERG) tends to primarily corrupt the word order, with fewer lexical substitutions. Thus, the model is less prone to overfitting to local n-gram features, and focuses on learning words and phrases that are genuinely predictive of sentiment.

5 Conclusions

In this paper, we present a training method that corrupts training examples with linguistic noise, in order to learn more robust models. Based on evaluation over several sentiment analysis datasets with convolutional neural networks, we show that this method outperforms standard training and dropout, both for in-domain and out-of-domain application. Our approach has wide-spread potential to also benefit other types of discriminative model and in a range of other language processing tasks.

Acknowledgments

We are grateful to the anonymous reviewers for their helpful feedback and suggestions, and to Ned Letcher for assistance in running the ERG. This research was supported in part by the Australian Research Council.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*, San Diego, USA.

Timothy Baldwin, Paul Cook, Marco Lui, Andrew MacKinlay, and Li Wang. 2013. How noisy social media text, how diffrnt social media sources? In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 356–364, Nagoya, Japan.

Steven Bird. 2006. NLTK: The natural language toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 69–72, Sydney, Australia.

Zsolt Bitvai and Trevor Cohn. 2015. Non-linear text regression with a deep convolutional neural network. 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 180–185, Beijing, China.

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 Workshop on Statistical Machine Translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, USA.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 740–750, Doha, Qatar.

James Clarke and Mirella Lapata. 2006. Models for sentence compression: A comparison across domains, training requirements and evaluation measures. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 377–384, Sydney, Australia.

Ann Copestake and Dan Flickinger. 2000. An open source grammar development environment and broad-coverage english grammar using HPSG. In *Proceedings of the Second International Conference on Language Resources and Evaluation*, Athens, Greece.

Ann Copestake, Dan Flickinger, Ivan A. Sag, and Carl Pollard. 2005. Minimal recursion semantics: An introduction. *Journal of Research on Language and Computation*, 3(2–3):281–332.

Jacob Eisenstein. 2013. What to do about bad language on the internet. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 359–369, Atlanta, USA.

Katja Filippova, Enrique Alfonseca, A. Carlos Colmenares, Lukasz Kaiser, and Oriol Vinyals. 2015. Sentence compression by deletion with LSTMs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 360–368, Lisbon, Portugal.

Rose Jenny Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 363–370, Ann Arbor, USA.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *Proceedings of the International Conference on Learning Representations*, San Diego, USA.

Alan Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649, Vancouver, Canada.

Bo Han and Timothy Baldwin. 2011. Lexical normalisation of short text messages: Makn sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 368–378, Portland, USA.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 168–177, Seattle, USA.

Yulei Jiang, Richard M. Zur, Lorenzo L. Pesce, and Karen Drukker. 2009. A study of the effect of noise injection on the training of artificial neural networks. In *International Joint Conference on Neural Networks*, pages 1428–1432, Atlanta, USA.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665, Baltimore, USA.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751, Doha, Qatar.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*, San Diego, USA.

Kevin Knight and Daniel Marcu. 2000. Statistics-based summarization-step one: Sentence compression. In *Proceedings of the 18th Annual Conference on Artificial Intelligence*, pages 703–710, Austin, USA.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105, Lake Tahoe, USA.

Yitong Li, Trevor Cohn, and Timothy Baldwin. 2016. Learning robust representations of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1979–1985, Austin, USA.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119, Lake Tahoe, USA.

George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. 1990. Introduction to WordNet: an on-line lexical database. *International Journal of Lexicography*, 3(4):235–244.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 142–148, San Diego, USA.

Andrew Y. Ng. 2004. Feature selection, L1 vs. L2 regularization, and rotational invariance. In *Proceedings of the Twenty-first International Conference on Machine Learning*, Banff, Canada.

Anh Nguyen, Jason Yosinski, and Jeff Clune. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 427–436, Boston, USA.

Stephan Oepen, Kristina Toutanova, Stuart Shieber, Christopher Manning, Dan Flickinger, and Thorsten Brants. 2002. The LinGO Redwoods Treebank: Motivation and preliminary applications. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 1253–1257, Taipei, Taiwan.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 115–124, Ann Arbor, USA.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135.

Ellie Pavlick and Chris Callison-Burch. 2016. Simple PPDB: A paraphrase database for simplification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 143–148, Berlin, Germany.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, Doha, Qatar.

Vu Pham, Théodore Bluche, Christopher Kermorvant, and Jérôme Louradour. 2014. Dropout improves recurrent neural networks for handwriting recognition. In *14th International Conference on Frontiers in Handwriting Recognition*, pages 285–290, Crete, Greece.

Noah A. Smith and Jason Eisner. 2005. Contrastive estimation: Training log-linear models on unlabeled data. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 354–362, Ann Arbor, USA.

Noah A. Smith. 2012. Adversarial evaluation for models of natural language. *arXiv preprint arXiv:1207.0245*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, USA.

Anders Søgaard. 2013. Part-of-speech tagging with antagonistic adversaries. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 640–644, Sofia, Bulgaria.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *Proceedings of the International Conference on Learning Representations*, Banff, Canada.

Stefan Wager, Sida Wang, and Percy S. Liang. 2013. Dropout training as adaptive regularization. In *Advances in Neural Information Processing Systems 26*, pages 351–359, Lake Tahoe, USA.

3.3 BIBI 2017: Generating Language Variations using Machine Learning

Li, Yitong, Trevor Cohn and Timothy Baldwin (2017) BIBI System Description: Building with CNNs and Breaking with Deep Reinforcement Learning, In *Proceedings of the EMNLP 2017 Workshop on Building Linguistically Generalizable NLP Systems (BLGNLP 2017)*, Copenhagen, Denmark, pp. 27–32.

3.3.1 Research Contributions

This paper describes the system with which I participated in the *Build it Break it: the Language Edition* competition as part of the 2017 workshop attached to EMNLP 2017. Competitors were divided into two adversarial groups – builders and breakers – where builders were tasked with building models robust against instances crafted by breakers and breakers generating adversarial instances to fool builders' systems. In detail, a breaker is asked to construct minimal pairs that would fool a sentiment classification system, by generating a new sentence from altering a reference sentence and not changing its sentiment. For example, a breaker can construct a new sentence *I'm mad for this movie!* from *I love this movie!*, which may appear to have an opposite sentiment to a naive word based method. Builders aim to build robust sentiment system to these changes.

This paper proposed automatic approaches to both builder and breaker systems.

From the builder side, I investigate the question of how robust the deep model is to adversarial examples, generated by both humans and automatic approaches. In this paper, a TextCNN model is investigated, trained with both phrase-level and sentence-level instances. The full results demonstrate that the approach is not robust enough to such adversarial attacks.

From the breaker side, the task is to generate adversarial text examples, which are interpretable by humans but can fool machine learning models. Differently from Section 3.2 where I generated instances using linguistically motivated edits, and differently from other breaker teams which generate adversarial instances manually, I generated adversarial instances automatically using machine learning. Inspired by work on adversarial attacks in

computer vision (see Section 2.3.1.2), this task is formalised as an optimisation problem, and solves discrete optimisation problem using reinforcement learning.

There are a few limitations of this work. The quality of the generated sentences is poor, for example, some of them have grammar errors, or contain logical errors. The training process is hard, and the semantics of the sentence is hard to maintain. I return to discuss this in more detail in Chapter 4.

BIBI System Description: Building with CNNs and Breaking with Deep Reinforcement Learning

Yitong Li and **Trevor Cohn** and **Timothy Baldwin**
School of Computing and Information Systems
The University of Melbourne, Australia
yitongl4@student.unimelb.edu.au, {tcohn,tbaldwin}@unimelb.edu.au

Abstract

This paper describes our submission to the sentiment analysis sub-task of "Build It, Break It: The Language Edition (BIBI)", on both the builder and breaker sides. As a builder, we use convolutional neural nets, trained on both phrase and sentence data. As a breaker, we use Q-learning to learn minimal change pairs, and apply a token substitution method automatically. We analyse the results to gauge the robustness of NLP systems.

1 Introduction

Recently, deep learning models have made impressive gains over a range of NLP tasks (Bahdanau et al., 2015; Bitvai and Cohn, 2015). However, recent studies have exposed brittleness in the models, e.g. through adversarial examples (Szegedy et al., 2014; Goodfellow et al., 2015). In these papers, researchers construct cognitively implausible perturbations of raw image inputs to fool state-of-the-art deep learning models. These perturbations are cheap and easy to generate using a "fast-gradient" method, based on analysis of the derivative of the loss with respect to the input.

One issue with the generation of adversarial examples for NLP has been the fact that language data is discrete, and hence difficult to map the continuous outputs of "gradient" methods onto. Furthermore, the perturbations or mutations generated through adversarial methods may be nonsensical to humans. Given this background, the BIBI shared task was devised to study the reliability of NLP systems by generating adversarial test instances, and explicitly training systems to be robust against adversarial test instances. Specifically, the task is based on opposing sets of participants: *builders* aim to build systems robust to different inputs, and *breakers* try to construct instances which will cause the builders' systems to make incorrect predictions.

In this paper, we describe our builder and breaker submissions to the sentiment analysis subtask, which is a sentence-level binary classification task, to predict whether a given review sentence is positive or negative with respect to a given movie. The data set is derived from movie review data (Pang and Lee, 2005) and the Stanford Sentiment Treebank (Socher et al., 2013).

We participated both as a builder and breaker because we are interested in testing the robustness of state-of-the-art neural models, such as convolutional neural networks ("CNNs": Kim (2014)). Also, we were interested in the breaker task as an avenue for exploring how well we can automatically construct adversarial test instances. In the sentiment sub-task, the main job of breakers is to construct minimally-changed pairs that are able to fool the builders' sentiment analysers. For example, the following sentences can be considered to be a minimal pair, with positive (+1) sentiment:

(+1) I love this movie!
(+1) I'm mad for this movie!

2 Approach

Here, we describe the methods we used for both the builder and the breaker. Considering the expense of human judgements, especially for breakers, and the strong desire for the approaches to generalize, we decide to use automatic methods for both tasks.

2.1 Builder System Description

As a builder, we chose to use convolutional neural nets ("CNNs"), based on their strong performance over text classification tasks (Kim, 2014; Zhang et al., 2015). Specifically, we were interested in

Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems, pages 27–32
Copenhagen, Denmark, September 8, 2017.

testing the robustness of CNNs in NLP applications. We apply Kim (2014)'s model to this task, which is easy and fast to train. We train our models on both phrase-level labelled data (with neutral phrases removed), and sentence-level labelled data; we will refer to these as "phrased-based" and "sentence-based" CNNs, respectively. Below, we present a short outline of the CNN model.

2.1.1 Convolutional Neural Network

The CNN model first operates by embedding each word using a look-up table which is stacked into the sentence matrix $\mathbf{E}_{\mathrm{S}}$. Then, a 1d convolutional layer is applied to $\mathbf{E}_{\mathrm{S}}$, which applies a series of filters over each window of t words, with each filter employing a rectifier transform function. In practice, we use window widths of size $t \in \{3, 4, 5\}$, and 128 filters for each size. $\mathrm{MaxPooling}$ is applied to each of the three sizes separately, and the resulting vectors are concatenated to form a fixed-size representation of the given sentence or phrase. Finally, the representation vector is fed into a final $\mathrm{Softmax}$ layer to generate a probability distribution over classification labels.

The model is trained to minimize the loss — defined as the cross-entropy between the ground-truth and the prediction — using the Adam Optimizer (Kingma and Ba, 2015) with a learning rate of 10^{-4} and batch size of 64.

2.2 Breaker System Description

As our breaker, we borrow ideas from generating adversarial examples in computer vision (Szegedy et al., 2014).

Assuming the loss of the system s is accessible and the true label l of the sentence is known, then the given task can be seen as an optimization problem where we simultaneously minimize the loss between the perturbed sentence $h(x)$ and flipped label, and also the distance between them :

$$\min_{\theta_h} \mathcal{L}_s(h(x), \mathbf{1} - l) + \alpha \cdot \mathrm{distance}(h(x), x) \quad (1)$$

Here, system s maps the input sentence x into the label space, and h is the perturbing function.

In text applications, this can be seen as an integer programming problem. Generally, integer optimization is NP-hard (Cunningham et al., 1996), although estimations can be found using heuristic methods, such as simulated annealing. However, considering the complexity of language, solving the given optimization function in only a discrete text space could lead to nonsensical outputs to a human, according to the results of our preliminary experiments [1]. Empirically, this can be attributed to the difficulty of defining an order over a natural language token set, as well as the non-convex nature of the semantic space in natural language generation.

Therefore, instead of optimizing Equation (1) directly, we split the problem into two subtasks: first, we use a reinforcement learning method to learn which tokens or phrases should be changed; and second, we apply a substitution method to those selected tokens, ensuring the quality of the new sentence.

2.2.1 Reinforcement Learning Method

In order to learn the sentiment of a given text, most NLP systems use n-gram feature-based learning methods, including traditional bag-of-words methods (Pang and Lee, 2005) as well as deep learning models (Socher et al., 2013; Kim, 2014). Based on this observation, one intuitive method of fooling the system is to find the "important" tokens within a given sentence, and then modify these to trick a given system into making a wrong prediction.

In our method, we need a baseline system for our breaker method to attack. Here, we choose the sentence-based CNN model, as described above, as an imaginary enemy. For most black-box systems, it is impossible to access the internals of the model and parameters. Therefore, given an input instance, we only use the output of the system, such as the prediction and loss in our method.

To solve this discrete problem, we apply a Reinforcement Learning ("RL") method (Sutton and Barto, 1998), specifically Q-learning (Watkins and Dayan, 1992; Mnih et al., 2013), to model the probability of removing tokens or phrases from a given sentence. Given the token x and an instance of context sentence c, the RL system learns a policy function $\pi(x|c) \rightarrow \{\mathrm{remove}, \mathrm{keep}\}$. We consider each instance as one game, consisting of several rounds. In the first round, c is a randomly-selected sentence, x is the first token in c, and π is the decision process of removing x or not. In each round, π will be learned at the token level, and the resulting sentence will be taken as the new context. The game will be repeated iteratively un-

[1] We used simulated annealing method to solve the given constrain problem directly. The details of results are not presented in this paper.

Builder system	$\mathcal{F}_1$	% of broken examples					
		Total	Average	Breaker 1	Breaker 2	Breaker 3	Breaker 4
Builder 1	**0.528**	25.43	26.76	35.35	22.62	39.79	9.28
RNN (Socher et al., 2013)	0.457	25.96	27.45	34.34	27.38	36.73	11.34
DCNN (Kalchbrenner et al., 2014)	0.483	25.09	25.95	34.84	21.42	36.73	10.82
Bag-of-n-grams	0.510	24.74	25.51	38.38	20.23	36.73	6.70
Phrase-based CNN	0.518	**24.39**	**25.23**	35.35	22.62	33.67	9.28
Sentence-based CNN	0.490	28.57	31.42	39.39	39.28	39.79	7.21

Table 1: The results of builder systems for BIBI blind test set based on average $\mathcal{F}_1$ (higher is better) and percentage of broken cases (lower is better). Details of each builder's system against the breaker's examples are also listed. The best results are indicated in **bold**.

til a max-round limitation is hit. When the game is terminated, the reward of the game will be the loss difference between the original sentence and the residual sentence, where the loss is calculated relative to the baseline system s. Additionally, we use the number of removed tokens as a penalty item in the final reward. The procedure will then randomly select a new instance and start a new game.

For training, we use the standard Deep Q-Learning algorithm, as described in Mnih et al. (2013). For hyper-parameters, max-round is set to 100 and γ to 0.01. The feature extractor ϕ is a multi-layer perceptron over token embeddings, initialized by pre-trained word2vec vectors (Mikolov et al., 2013). The batch size is 128, the initial ϵ is set to 0.3, and the memory size is up to $10,000$. In order to change as few tokens as possible, we empirically set the distance penalty α to 2. The reward is calculated using pre-trained sentence-based CNN, as described above.

2.2.2 Token Substitution Method

Once the algorithm has decided which tokens should be changed, the next move is to find appropriate substitutions. As described above, most systems are based on n-grams, making them very sensitive to unknown tokens. Therefore, we came up with some heuristics.

The first approach draws on our earlier work on learning robust text representations (Li et al., 2017), and is based on synonyms of the given token, based on Princeton WordNet (Miller et al., 1990) using the NLTK API (Bird, 2006). Here, we test possible synonyms, considering their part-of-speech tag, asking the system s whether the loss is reduced after substitution. We also tried to find antonyms that cannot be recognized by the system, causing the predicted sentiment label to not flip. Finally, we add a small amount of human supervision to ensure the fluency of the output sentences, including removal of garbled examples and minor grammar corrections, and to ensure they have the correct sentiment label. To be specific, we discard the "bad-attacked" pairs with loss difference less than 1 empirically. These "bad-attacked" pairs might be able to fool the sentence-based CNN but with low confidence, such that we did not expect them to be good enough to fool other builders' systems. We also filter out sentences with wrong or ambiguous sentiment labels manually. For example, the system sometimes generates expressions with correct grammar but strange sentiment — e.g., *I don't like this lovely movie* which is contradictory and possibly interpretable as ironic — which remains a challenging problem for us to totally eliminate during generation. Last, we slightly modify the outputs to fix minor grammar errors, such as adding or removing the determiner *a* or *the*.

3 Results and Analysis

In this section, we detail the results of our methods, and perform error analysis.

3.1 Builder

The results for the builder systems over the test set are shown in Table 1. To evaluate the robustness of the builder systems, there are two evaluation criteria: average F-score ("$\mathcal{F}_1$") across all breaker test cases (higher is better), and the percentage of breaker test cases that break the system (lower is better). Having a builder fail over only one example in a given minimal pair is considered to have broken that system.

We observed that all the systems are very close over these two criteria. We also see that the phrase-based CNN achieved competitive performance, while the sentence-based CNN is not as

robust. This aligns with our intuition, as feeding phrase-labeled data is more precise for model training, and it is much easier for the sentence-based model to overfit the data, according to our analysis. For example, it might consider *the performance is* as a strong positive trigram feature instead of a neutral one, because the expression has higher frequency in the positive training set than that in the negative set. This also occurs for certain entity tokens, such as people's names and places.

To better understand the advantages and disadvantages of CNNs, we perform some error analysis.

One major class of breaker attack is modifying the polarity of a sentence, either syntactically (e.g. by adding/removing *not*) or morphologically (e.g. by adding the prefix *un-*), but actually the CNN is relatively robust to this. We believe the reason is that the n-gram features our CNN learns are more robust representations of words and short phrases. This also explains the performance of the bag-of-n-gram (BoN) system. However, CNN is still slightly better than BoN because CNN only learns the most important features through the $\mathrm{MaxPooling}$ operation, and using word embeddings appears to help the model deal with synonyms and antonyms at the word level.

It is almost the same situation when the systems encounter out-of-vocabulary words (OOV). Although OOVs are a significant challenge, we believe they can be overcome by training better sentiment-sensitized word embeddings (Mrkšić et al., 2016), or combining the system with character-level normalization methods (Han and Baldwin, 2011).

However, CNNs are not good at dealing with complex grammatical structures or long-distance dependencies. For instance, changing a comparative from *more than* to *less than* flips the sentiment and is something that humans are sensitized to, but CNNs tend not to capture this difference. Also, CNNs are not sensitive to tense, such as changing the present tense *is* to the past tense *was* to capture pragmatic/connotative effects. For these kinds of examples, we expected to see higher performance among models which better capture syntactic structure, such as recursive neural nets ("RNNs": Socher et al. (2013)) and dynamic convolutional neural networks ("DCNNs": Kalchbrenner et al. (2014)). In practice, however, this was not the case. We cannot conclude the exact reasons without further analysis of these models, however this might indicate that these perceptron-based deep models struggle to capture the logic in human langauge.

To conclude, among traditional models and state-of-the-art deep learning models for sentiment analysis, CNNs are relatively robust.

Test set	Average $\mathcal{F}_1$	Score
Breaker 1	0.79	28.64
Breaker 3	0.84	**31.17**
Breaker 4	0.83	7.48
Breaker 2 (our method)	0.75	19.28

Table 2: The final score of the breakers. The average $\mathcal{F}_1$ over the original sentences of all builders' systems is also listed for each break test set.

3.2 Breaker

Table 2 gives the final scores of the breaker teams. The final score is calculated by averaging the $\mathcal{F}_1$ of each builder's system on the original sentences, multiplied by the percentage of examples that break that system (shown in Table 1).

Overall, about one third of the breakers' examples were able to fool the builders' systems, which is not surprising. On the one hand this is encouraging, in that, without taking the untapped test cases into consideration, each builder can handle more than half of the break examples. On the other hand, still nearly one third of the break sentences cannot be handled by state-of-the-art statistical models.

For our breaking approach, we observe that all the builders' systems have lower $\mathcal{F}_1$ on our test set, indicating that our method tends to generate difficult sentences, where systems might have lower confidence. Actually, in our final submission, we only chose 42 pairs as the final break data from among the 521 test data instances provided by the organisers, as the rest of the generated pairs were removed due to low confidence or bad quality sentences. This unfortunately indicates that our automatic method cannot be applied to all examples, making our approach limited in application. Therefore, we can't really conclude that our automatic approach is a success, and we should explore more flexible approaches in the future. However, the approach itself still achieves a break rate higher than the error rate on the origi-

nal test set. Additionally, our method breaks the sentence-based CNN — which our RL model is built on — with 39.28% break rate.

Based on error analysis over the broken examples, we found that using tokens with opposite sentiment in an example worked across builder systems in most cases. For instance, *if you love to waste your time* could confuse most systems because of the contrast between *love* and *waste time*, because they indicate opposing sentiment in isolation, while the phrase itself is focused on *waste time*, which is very difficult for most NLP systems to understand. This indicates that representations of natural language may be doing more than simply adding or transforming the word embeddings, and instead non-compositionally transforming the logic structure of the sentence.

Also, attacking words or phrases which are ambiguous between positive and negative sentiment is also a potentially effective approach. For example, *rock* is used predominantly in positive-sentiment contexts, in reference to jewels or strong/reliable people, meaning that systems are likely to learn that it has exclusively positive sentiment due to bias in the training set. However, when the negative substitution phrase *on the rocks* (meaning "in trouble") is used, the builders' systems might still predict the idiom as having positive sentiment.

Based on these observations, we can conclude that state-of-the-art statistical models have only minimal "understanding" of natural language. The examples we showed above are relatively simple, but in real cases, they can be more complex. And we are not even considering the ambiguity of language or tone of the language in different contexts. To summarize, our approach provides a method to study the robustness of modern NLP systems over a sentiment analysis task. Our results demonstrate that NLP systems are still far from turning the corner to real language understanding.

4 Conclusions

In this paper, we have described our builder and breaker systems, in the context of the BIBI the Language Edition shared task. We built sentiment analysis systems using text-based convolutional neural networks, trained on either phrase- and sentence-level data. Also, we used reinforcement learning and substitution methods to generate adversarial test examples automatically. We performed error analysis to better understand the robustness of statistical NLP models.

Acknowledgements

We would like to thank the three anonymous reviewers for their helpful feedback and suggestions, and to Meng Fang for assisting with the implementation of the RL system.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*.

Steven Bird. 2006. NLTK: The natural language toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*. pages 69–72.

Zsolt Bitvai and Trevor Cohn. 2015. Non-linear text regression with a deep convolutional neural network. 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 180–185.

William H Cunningham, S Thomas McCormick, and Maurice Queyranne. 1996. *Integer Programming and Combinatorial Optimization*. Springer.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *Proceedings of the International Conference on Learning Representations*.

Bo Han and Timothy Baldwin. 2011. Lexical normalisation of short text messages: Makn sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. pages 368–378.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. pages 655–665.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. pages 1746–1751.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*.

Yitong Li, Trevor Cohn, and Timothy Baldwin. 2017. Robust training under linguistic adversity. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. pages 21–27.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*. pages 3111–3119.

George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. 1990. Introduction to WordNet: an on-line lexical database. *International Journal of Lexicography* 3(4):235–244.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. 2013. Playing atari with deep reinforcement learning. *CoRR* abs/1312.5602.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pages 142–148.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*. pages 115–124.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. pages 1631–1642.

Richard S Sutton and Andrew G Barto. 1998. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, USA.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *Proceedings of the International Conference on Learning Representations*.

Christopher JCH Watkins and Peter Dayan. 1992. Q-learning. *Machine learning* 8(3-4):279–292.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*. pages 649–657.

3.4 NAACL 2018: Multi-domain Learning with Domain Supervision

Li, Yitong, Timothy Baldwin and Trevor Cohn (2018) What's in a Domain? Learning Domain-Robust Text Representations Using Adversarial Training, In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (NAACL HLT 2018, Short Papers)*, New Orleans, USA, pp. 474–479.

3.4.1 Research Contributions

This work and the following section (Section 3.5) focus on multi-domain learning, where a model is trained over a corpus comprised of text from multiple domains. In this first paper, I assume that model has access to the domain labels during the training phase, although domain is unknown at test time.

Inspired by "frustratingly easy domain adaptation" architectures (Daumé III, 2007) (see Section 2.3.2.1) and domain-adversarial nets (Ganin et al., 2016) (Section 2.3.1.2), two learning approaches are proposed to tackle the multi-domain problem:

(1) a domain-conditional model which learns domain-invariant features with adversarial learning, where the hidden features generalise well with respect to the target variables but are poor predictors of domain;

(2) a domain-generative model, which additionally learns all domain-specific features through one channel which has good predictions of both label and domain.

To evaluate the robustness of our approaches, two classification tasks are considered – language identification and multi-domain product review. Again, both in-domain and out-of-domain experiments are performed. To investigate how robust the model is towards domain variation, I focus on out-of-domain evaluation, where models have no preliminary domain knowledge of the test instances.

What's in a Domain? Learning Domain-Robust Text Representations using Adversarial Training

Yitong Li and **Timothy Baldwin** and **Trevor Cohn**
School of Computing and Information Systems
The University of Melbourne, Australia
yitongl4@student.unimelb.edu.au, {tbaldwin,tcohn}@unimelb.edu.au

Abstract

Most real world language problems require learning from heterogenous corpora, raising the problem of learning robust models which generalise well to both similar (*in domain*) and dissimilar (*out of domain*) instances to those seen in training. This requires learning an underlying task, while not learning irrelevant signals and biases specific to individual domains. We propose a novel method to optimise both in- and out-of-domain accuracy based on joint learning of a structured neural model with domain-specific and domain-general components, coupled with adversarial training for domain. Evaluating on multi-domain language identification and multi-domain sentiment analysis, we show substantial improvements over standard domain adaptation techniques, and domain-adversarial training.

1 Introduction

Heterogeneity is pervasive in NLP, arising from corpora being constructed from different sources, featuring different topics, register, writing style, etc. An important, yet elusive, goal is to produce NLP tools that are capable of handling all types of texts, such that we can have, e.g., text classifiers that work well on texts from newswire to wikis to micro-blogs. A key roadblock is application to new domains, unseen in training. Accordingly, training needs to be robust to domain variation, such that domain-general concepts are learned in preference to domain-specific phenomena, which will not transfer well to out-of-domain evaluation. To illustrate, Bitvai and Cohn (2015) report learning formatting quirks of specific reviewers in a review text regression task, which are unlikely to prove useful on other texts.

This classic problem in NLP has been tackled under the guise of "domain adaptation", also known as unsupervised transfer learning, using feature-based methods to support knowledge transfer over multiple domains (Blitzer et al., 2007; Daumé III, 2007; Joshi et al., 2012; Williams, 2013; Kim et al., 2016). More recently, Ganin and Lempitsky (2015) proposed a method to encourage domain-general text representations, which transfer better to new domains.

Inspired by the above methods, in this paper we propose a novel technique for multitask learning of domain-general representations.[1] Specifically, we propose deep learning architectures for multi-domain learning, featuring a *shared* representation, and domain *private* representation. Our approach generalises the feature augmentation method of Daumé III (2007) to convolutional neural networks, as part of a larger deep learning architecture. Additionally, we use adversarial training such that the *shared* representation is explicitly discouraged from learning domain identifying information (Ganin and Lempitsky, 2015). We present two architectures which differ in whether domain is conditioned on or generated, and in terms of parameter sharing in forming private representations.

We primarily evaluate on the task of language identification ("LangID": Cavnar and Trenkle (1994)), using the corpora of Lui and Baldwin (2012), which combine large training sets over a diverse range of text domains. Domain adaptation is an important problem for this task (Lui and Baldwin, 2014; Jurgens et al., 2017), where text resources are collected from numerous sources, and exhibit a wide variety of language use. We show that while domain adversarial training overall improves over baselines, gains are modest. The same applies to twin shared/private architectures, but when the two methods are combined, we observe substantial improvements. Overall, our

[1] Code, data and evaluation scripts available at https://github.com/lrank/Domain_Robust_Text_Representation.git

Proceedings of NAACL-HLT 2018, pages 474–479
New Orleans, Louisiana, June 1 - 6, 2018.

methods outperform the state-of-the-art (Lui and Baldwin, 2012) in terms of out-of-domain accuracy. As a secondary evaluation, we use the Multi-Domain Sentiment Dataset (Blitzer et al., 2007), where we once again observe a clear advantage for our approaches, illustrating the potential of our technique more broadly in NLP.

2 Multi-domain Learning

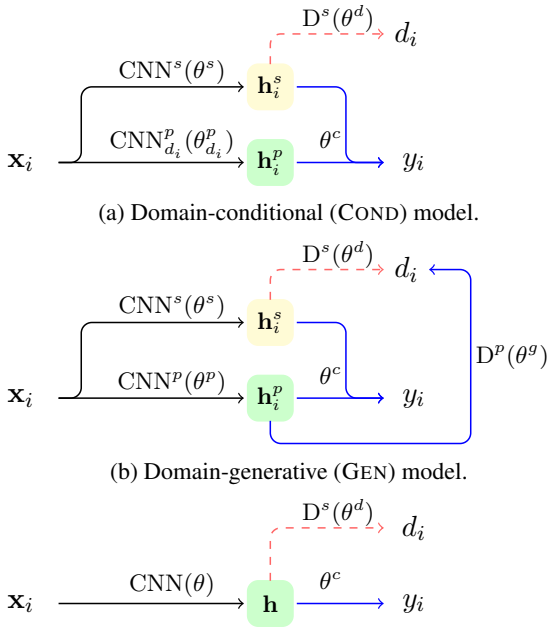
A primary consideration when formulating models of multi-domain data is how best to use the domain. Basic methods might learn several separate models, or simply ignore the domain and learn a single model. Neither method is ideal: the former fails to share statistics between the models to capture the general concept, while the latter discards information that can aid classification, e.g., domain-specific vocabulary or class skew.

To address these issues, we propose two architectures as illustrated in Figure 1 (a and b), parameterised as a convolutional network (CNN) over the input instance, chosen based on the success of CNNs for text categorisation problems (Kim, 2014); note, however, that our method is general and can be applied with other network types. Both representations are based on the idea of twin representations of each instance,[2] denoted *shared* and *private* representations, which are trained to capture domain-general versus domain-specific concepts, respectively. This is achieved using various loss functions, most notably an adversarial loss to discourage learning of domain-specific concepts in the shared representations. The two architectures differ in whether the domain is provided as an input (COND) or an output (GEN). Below, we elaborate on the details of the two models.

2.1 Domain-Conditional Model (COND)

The first model, illustrated in Figure 1a, includes a collection of domain-specific CNNs, and for each training instance $\mathbf{x}$, the domain-specific $\text{CNN}^p_{d_i}$ is used to compute its private representation $\mathbf{h}^p$. In this manner, the model *conditions* on the domain identifier. The COND model also computes a shared representation, $\mathbf{h}^s$, directly from $\mathbf{x}$, using a shared CNN^s, and the two representations are concatenated together to form input to linear softmax classification function c for predicting class label y. Thus far, the approach resembles Daumé III (2007), a method for multitask learning based on feature augmentation in a linear model, which works by replicating the input features to create both general shared features, and domain-specific features. Note that the approaches differ in that our method uses deep learning to form the two representations, in place of feature replication.

[2]This differs from standard architectures, e.g., 'baseline' in Figure 1c, which uses a single representation.



(a) Domain-conditional (COND) model.

(b) Domain-generative (GEN) model.

(c) Baseline CNN model with adversarial loss.

Figure 1: Proposed model architectures, showing a single training instance $(\mathbf{x}_i, y_i)$ with domain d_i, and baseline model with domain adversarial loss. CNN denotes a convolutional network, D indicates a discriminator (d for domain adversarial and g for domain generation), and red dashed and blue lines denote adversarial and standard loss, resp.

Adversarial Supervision A key challenge for the COND model is that the 'shared' representation can be contaminated by domain-specific concepts. To address this, we borrow ideas from adversarial learning (Goodfellow et al., 2014; Ganin et al., 2016). The central idea is to learn a good general representation (suitable for the shared component) to maximize end task performance, yet obscure the domain information, as modelled by a discriminator, D^s. This reduces the domain-specific information in the shared representation, however note that important domain-specific components can still be captured in the private representation.

Overall, this results in the training objective:

$$\mathcal{L}^{\text{COND}} = \min_{\theta^c, \theta^s, \{\theta^p\}} \max_{\theta^d} \mathcal{X}(\mathbf{y}|\mathbf{H}^s, \mathbf{H}^p, \mathbf{d}; \theta^c) - \lambda_d \mathcal{X}(\mathbf{d}|\mathbf{H}^s; \theta^d) \quad (1)$$

where $\mathcal{X}$ denotes the cross-entropy classification loss, $\mathbf{H}^s = \{\mathbf{h}_i^s(\mathbf{x}_i)\}_{i=1}^n$ are the shared representations for the training set of n instances, and likewise $\mathbf{H}^p = \{\mathbf{h}_i^p(\mathbf{x}_i, d_i)\}_{i=1}^n$ are the private representations, which are both functions of θ^s and $\{\theta^p_{\cdot}\}$, respectively. Note the negative sign of the adversarial loss (referred to as d), and the maximisation with respect to the discriminator parameters θ^d. This has the effect of learning a maximally accurate discriminator *wrt* θ^d, while making it maximally inaccurate *wrt* representation $\mathbf{H}^s$, and is implemented using a gradient reversal step during backpropagation (Ganin et al., 2016).

Minimum Entropy Inference As COND conditions on the domain, this imposes the requirement that the domain of the test data is known (and covered in training), which is incompatible with our goal of unsupervised adaptation. To deal with this situation, we consider each domain in the test set as belonging to one of the training domains, and then select the domain with the minimum entropy classification distribution. This is based on an assumption that a closely matching domain should be able to make confident predictions.[3]

2.2 Domain-Generative Model (GEN)

The second model is based on *generation* of, rather than *conditioning* on, the domain, which allows the model to learn domain signals that transfer across some, but not all, domains. Most components are common with the COND model as described in §2.1, including the use of private and shared representations, their use in the classification output, and the adversarial loss based on discriminating the domain from the shared representation. There are two key differences: (1) the private representation, $\mathbf{h}^p$, is computed using a single CNN^p, rather than several domain-specific CNNs, which confers benefits of domain-generalisation, a more compact model, and simpler test inference;[4] and (2) the private representation is used to positively predict the domain, which further encourages the split between domain general and domain-specific aspects of the representation.

[3]The minimum entropy method is quite effective, trailing oracle selection by only 0.8% accuracy.

[4]The domain need not be known for test examples, so the model can be used directly.

GEN has the following training objective,

$$\begin{aligned}\mathcal{L}^{\text{GEN}} = &\min_{\theta^c,\theta^s,\theta^p,\theta^g} \max_{\theta^d} \mathcal{X}(\mathbf{y}|\mathbf{H}^s, \mathbf{H}^p; \theta^c) \\ &- \lambda_d \mathcal{X}(\mathbf{d}|\mathbf{H}^s; \theta^d) + \lambda_g \mathcal{X}(\mathbf{d}|\mathbf{H}^p; \theta^g)\end{aligned} \tag{2}$$

where notation follows that used in §2.1, with the exception of $\mathbf{H}^p = \{\mathbf{h}_i^p(\mathbf{x}_i)\}_{i=1}^n$ that is redefined, with $\mathbf{h}_i^p(\mathbf{x}_i)$ a function of θ^p, and the addition of the last term to capture the generation loss g. The same gradient reversal method from §2.1 is used during training for the adversarial component.

3 Experiments

3.1 Language Identification

To evaluate our approach, we first consider the language identification task.

Data We follow the settings of Lui and Baldwin (2012), involving 5 training sets from 5 different domains with 97 languages in total: Debian, JRC-Acquis, Wikipedia, ClueWeb and RCV2, derived from Lui and Baldwin (2011).[5] We evaluate accuracy on seven holdout benchmarks: EuroGov, TCL, Wikipedia2[6] (all from Baldwin and Lui (2010)), EMEA (Tiedemann, 2009), EuroPARL (Koehn, 2005), T-BE (Tromp and Pechenizkiy, 2011), and T-SC (Carter et al., 2013).

Documents are tokenized as a byte sequence (consistent with Lui and Baldwin (2012)), and truncated or padded to a length of 1k bytes.[7]

Hyper-parameters We perform a grid search for the hyper-parameters, and selected the following settings to optimise accuracy over heldout data from each of the training domains. All byte tokens are mapped to byte embeddings, which are random initialized with size 300. We use the filter sizes of $2, 4, 8, 16$ and 32, with 128 filters for each, to capture n-gram features of different lengths. A dropout rate of 0.5 was applied to all the representation layers. We set the factors λ_d and λ_g to 10^{-3}. All the models are optimized using the Adam Optimizer (Kingma and Ba, 2015) with a learning rate of 10^{-4}.

[5]As ClueWeb in Lui and Baldwin (2012) is not publicly accessible, we used a slightly different set of languages but comparable number of documents for training.

[6]Note that the two Wikipedia datasets have no overlap.

[7]We also tried different document length limits, such as 10k, but observed no substantial change in performance.

Models	EuroGov	TCL	Wikipedia2	EMEA	EuroPARL	T-BE	T-SC	ALL_{out}
baseline CNN	**99.9**	91.7	88.9	93.1	98.2	85.2	92.2	92.7
$+d$	**99.9**	92.4	88.4	90.2	98.2	87.7	93.1	92.8
$+g$	**99.9**	92.0	88.7	91.6	98.4	86.8	92.8	92.9
COND	**99.9**	91.3	88.2	92.0	98.7	91.5	94.5	93.7
$+d$	**99.9**	**93.5**	90.1	91.3	98.7	92.6	**97.9**	**94.9**
GEN	**99.9**	92.3	88.0	93.3	98.6	87.1	93.8	93.3
$+d+g$	**99.9**	93.1	88.7	92.5	99.1	91.2	96.1	94.4
LANGID.PY	98.7	90.4	**91.3**	**93.4**	**99.2**	**94.1**	88.6	93.6
CLD2	99.0	85.0	85.3	90.7	98.5	85.0	93.4	91.0

Table 1: Accuracy [%] of the different models over the seven heldout datasets, and the macro-averaged accuracy out-of-domain over the 7 test domains ("ALL_{out}"). The best result for each dataset is indicated in **bold**. Key: $+d$ = domain adversarial, $+g$ = domain generation component.

3.1.1 Results and Analysis

Baseline and comparisons For comparison, we implement a CNN baseline[8] which is trained using all the data without domain knowledge (i.e. the simple union of the different training datasets). We also employ adversarial learning (d) and generation (g) of domain to the baseline model to better understand the utility of these methods. Note that the baseline +d is a multi-domain variant of Ganin and Lempitsky (2015), albeit trained without any text in the testing domains. For our models, we report results of configurations both with and without the d and g components. We also report the results for two state-of-the-art off-the-shelf LangID tools: (1) LANGID.PY[9] (Lui and Baldwin, 2012); and (2) Google's CLD2.[10]

Out-of-domain Results Our primary concern in terms of evaluating the ability of the different models to generalise, is out-of-domain performance. Table 1 provides a breakdown of out-of-domain results over the 7 holdout domains. The accuracy varies greatly between test domains, depending on the mix of languages, length of test documents, etc. Both our models, COND and GEN, achieve competitive performance, and are further improved by d and g.

For the baseline, applying either d or g results in mild improvements over the baseline, which is surprising as the two forms of supervision work in opposite directions. Overall the small change in performance means neither method appears to be a viable technique for domain adaptation.

Overall, the raw COND and GEN perform better than the baseline. Specifically, for COND, we observed performance gains on EuroPARL, T-BE and T-SC. These three datasets are notable in containing shorter documents, which benefit the most from shared learning. However, as discussed earlier, multi-domain data can introduce noise to the shared representation, causing the performance to drop over TCL, Wikipedia2 and EMEA. This observation demonstrates the necessity of applying adversarial learning to COND. On the other hand, it is a different story for GEN: vanilla GEN achieves accuracy gains relative to the baseline over 5 domains, but is slightly below COND for 4 domains, a result of parameter-sharing over the private representation.

In terms of the adversarial learning, we see that by adding an adversarial component ($+d$ or $+d+g$), COND and GEN realises substantial improvements out of domain, with the exception of EMEA. As we motivated, the domain adversarial part d can obscure the domain-specific information in the shared representation, which helps COND have better generalisation to other domains. Additionally, applying g to GEN helps the private representation to generalize better. These results demonstrate that both d and g are necessary components of multi-domain models. EMEA is noteworthy in that its pattern of results is overall different to the other domains, in that applying d hurts performance. For this domain, the baseline CNN performs very well, and GEN does much better than COND. We believe the reason is that, as a medical domain, EMEA is very much an outlier and does not align to any single training domain. Also, there is a lot of borrowing of terms such as drug and disease names verbatim between lan-

[8]The baseline here used a double capacity hidden representation, in order to better match the increased expressivity of the shared/private models.

[9]https://github.com/saffsd/langid.py

[10]https://github.com/CLD2Owners/cld2

Models	Deb	JRCA	Wiki	ClWb	RCV2	ALL_{in}
baseline CNN	96.6	99.8	97.8	90.7	97.9	96.6
baseline $+d$	96.5	99.8	97.8	90.7	97.0	96.4
COND	97.0	99.9	97.8	90.9	98.0	96.7
COND $+d$	97.0	**99.9**	**98.4**	90.8	98.1	96.8
GEN $+d+g$	**97.8**	**99.9**	98.0	91.1	97.9	96.9
LANGID.PY	97.4	99.8	97.6	**91.3**	**99.3**	**97.1**
CLD2	92.2	99.8	92.3	92.3	89.8	93.3

Table 2: Accuracy [%] of different models over five in-domain datasets using cross-validation evaluation and macro-averaged accuracy ("ALL_{in}").

guages, further complicating the task.

Overall, our best models (COND $+d$ and GEN $+d+g$) outperform both LANGID.PY and CLD2 in terms of average out-of-domain accuracy.

In-domain Results Table 2 reports the in-domain performance over the 5 training domains, using 5-fold cross validation, as well as the macro-averaged accuracy. Our proposed methods (COND $+d$ and COND $+d+g$) consistently achieve better performance than the baseline. Both COND and GEN achieve competitive performance with the state-of-the-art LANGID.PY in the in-domain scenario. Although LANGID.PY performs slightly better on average accuracy, our best model outperforms LANGID.PY for three of the five datasets.

3.2 Product Reviews

To evaluate the generalization of our methods to other tasks, we experiment with the Multi-Domain Sentiment Dataset (Blitzer et al., 2007).[11] We select the 20 domains with the most review instances, and discard the remaining 5 domains.

For model parameterization, we adopt the same basic hyper-parameter settings and training process as for LangID in §3.1, but change the filter sizes to 3, 4 and 5, use word-based tokenisation, and truncate sentences to 256 tokens, for better compatible with shorter documents.

We perform a out-of-domain evaluation over four target domains, "book" (B), "dvd" (D), "electronics" (E) and "kitchen & housewares" (K), as used in Blitzer et al. (2007). Our experimental setup differs from theirs, in that they train on a single domain and then evaluate on another, while we train over 16 domains, then evaluate on the four test domains.

[11] From https://www.cs.jhu.edu/~mdredze/datasets/sentiment/, using the positive and negative files from unprocessed, up to 2,000 instances per domain. For the four test domains we automatically aligned the reviews in the processed and unprocessed, such that we can compare results directly against prior work.

Models	B	D	E	K
baseline CNN	79.6	81.2	86.3	87.2
baseline $+d$	78.7	81.6	86.6	87.1
COND	79.2	81.8	85.8	87.2
COND $+d$	79.8	82.3	86.8	87.4
GEN $+d+g$	**80.2**	**82.4**	**87.3**	**87.8**
SCL-MI ♣	76.0	78.5	77.9	85.9
DANN ♢	72.3	78.4	84.3	85.4
IN DOMAIN ♣	82.4	80.4	84.4	87.7

Table 3: Accuracy [%] of different models over 4 domains (B, D, E and K) under out-of-domain evaluations on Multi Domain Sentiment Dataset. Key: ♣ from Blitzer et al. (2007); ♢ from Ganin and Lempitsky (2015).

Table 3 presents the results. Overall, our proposed methods consistently outperform the baselines, with the GEN $+d+g$ approach a consistent winner over all other techniques. Note also the lacklustre performance when the baseline is trained with the adversarial loss, mirroring our findings for language identification in §3.1. For comparison, we also report the best results of SCL-MI and DANN, in both cases using an oracle selection of source domain. Our method consistently outperform these approaches, despite having no test oracle, although note that we use more diverse data sources for training.

4 Conclusions

We have proposed a novel deep learning method for multi-domain learning, based on joint learning of domain-specific and domain-general components, using either domain conditioning or domain generation. Based on our evaluation over multi-domain language identification and multi-domain sentiment analysis, we show our models to substantially outperform a baseline deep learning method, and set a new benchmark for state-of-the-art cross-domain LangID. Our approach has potential to benefit other NLP applications involving multi-domain data.

Acknowledgments

We thank the anonymous reviewers for their helpful feedback and suggestions, and the National Computational Infrastructure Australia for computation resources. This work was supported by the Australian Research Council (FT130101105).

References

Timothy Baldwin and Marco Lui. 2010. Language identification: The long and the short of the matter. In *Proceedings of Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics*. pages 229–237.

Zsolt Bitvai and Trevor Cohn. 2015. Non-linear text regression with a deep convolutional neural network. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing Short Papers*.

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

Simon Carter, Wouter Weerkamp, and Manos Tsagkias. 2013. Microblog language identification: Overcoming the limitations of short, unedited and idiomatic text. *Language Resources and Evaluation* 47(1):195–215.

William B Cavnar and John M Trenkle. 1994. N-gram-based text categorization. In *Proceedings of the Third Symposium on Document Analysis and Information Retrieval*.

Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*. pages 256–263.

Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning 2015*. pages 1180–1189.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of Machine Learning Research* 17:59:1–59:35.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*. pages 2672–2680.

Mahesh Joshi, Mark Dredze, William W. Cohen, and Carolyn Penstein Rosé. 2012. Multi-domain learning: When do domains matter? In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. pages 1302–1312.

David Jurgens, Yulia Tsvetkov, and Dan Jurafsky. 2017. Incorporating dialectal variability for socially equitable language identification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. pages 51–57.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. pages 1746–1751.

Young-Bum Kim, Karl Stratos, and Ruhi Sarikaya. 2016. Frustratingly easy neural domain adaptation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. pages 387–396.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT Summit 2005*. pages 79–86.

Marco Lui and Timothy Baldwin. 2011. Cross-domain feature selection for language identification. In *Fifth International Joint Conference on Natural Language Processing*. pages 553–561.

Marco Lui and Timothy Baldwin. 2012. langid.py: An off-the-shelf language identification tool. In *Proceedings of ACL 2012 System Demonstrations*. pages 25–30.

Marco Lui and Timothy Baldwin. 2014. Accurate language identification of Twitter messages. In *Proceedings of the 5th workshop on language analysis for social media*. pages 17–25.

Jörg Tiedemann. 2009. News from OPUS – a collection of multilingual parallel corpora with tools and interfaces. In *Recent Advances in Natural Language Processing*. volume 5, pages 237–248.

Erik Tromp and Mykola Pechenizkiy. 2011. Graph-based n-gram language identification on short texts. In *Proceedings of the 20th Machine Learning Conference of Belgium and The Netherlands*. pages 27–34.

Jason Williams. 2013. Multi-domain learning and generalization in dialog state tracking. In *Proceedings of the SIGDIAL 2013 Conference*. pages 433–441.

3.5 ACL 2019: Multi-domain Learning using Latent Domain

Li, Yitong, Timothy Baldwin and Trevor Cohn (2019) Semi-supervised Stochastic Multi-Domain Learning using Variational Inference, In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, Florence, Italy, pp. 1923–1934.

3.5.1 Research Contributions

While still focusing on multi-domain learning, this work makes an assumption that not all training instances are accompanied with domain information, which means a model does not have domain labels for some instances during training. This task is thus a semi-supervised learning problem with respect to domain, which I refer to as *domain semi-supervised learning*. The worst case can be no domain knowledge of the training data at all, referred to as *domain unsupervised learning*.

To tackle this problem, I propose a stochastic domain learning technique, treating domain as a latent variable, which performs like a gating function over each domain channel. To parameterise the latent domain, I use either discrete or continuous representations, to prevent model complexity from expanding linearly with the number of domains. In this paper, variational inference is adopted to tackle the intractable problem of inferring continuous with stochastic variables.

To evaluate the proposed approaches, the tasks and dataset from Section 3.4 are reused. Once again, this paper focuses on out-of-domain evaluation to demonstrate the robustness and advantages of the proposed models.

This section and the last section (Section 3.4) tackle the second questions of robust multi-domain learning. For limitations, I discuss them from several aspects in Section 4.3.1, including better adversarial training technique, as well as future opportunities.

Semi-supervised Stochastic Multi-Domain Learning using Variational Inference

Yitong Li **Timothy Baldwin** **Trevor Cohn**
School of Computing and Information Systems
The University of Melbourne, Australia
yitongl4@student.unimelb.edu.au
{tbaldwin,tcohn}@unimelb.edu.au

Abstract

Supervised models of NLP rely on large collections of text which closely resemble the intended testing setting. Unfortunately matching text is often not available in sufficient quantity, and moreover, within any domain of text, data is often highly heterogenous. In this paper we propose a method to distill the important domain signal as part of a multi-domain learning system, using a latent variable model in which parts of a neural model are stochastically gated based on the inferred domain. We compare the use of discrete versus continuous latent variables, operating in a domain-supervised or a domain semi-supervised setting, where the domain is known only for a subset of training inputs. We show that our model leads to substantial performance improvements over competitive benchmark domain adaptation methods, including methods using adversarial learning.

1 Introduction

Text corpora are often collated from several different sources, such as news, literature, microblogs, and web crawls, raising the problem of learning NLP systems from heterogenous data, and how well such models transfer to testing settings. Learning from these corpora requires models which can generalise to different domains, a problem known as transfer learning or domain adaptation (Blitzer et al., 2007; Daumé III, 2007; Joshi et al., 2012; Kim et al., 2016). In most state-of-the-art frameworks, the model has full knowledge of the domain of instances in the training data, and the domain is treated as a discrete indicator variable. However, in reality, data is often messy, with domain labels not always available, or providing limited information about the style and genre of text. For example, web-crawled corpora are comprised of all manner of text, such as news, marketing, blogs, novels, and recipes, however the type of each document is typically not explicitly specified. Moreover, even corpora that are labelled with a specific domain might themselves be instances of a much more specific area, e.g., "news" articles will cover politics, sports, travel, opinion, etc. Modelling these types of data accurately requires knowledge of the specific domain of each instance, as well as the domain of each test instance, which is particularly problematic for test data from previously unseen domains.

A simple strategy for domain learning is to jointly learn over all the data with a single model, where the model is not conditioned on domain, and directly maximises $p(y|\mathbf{x})$, where $\mathbf{x}$ is the text input, and y the output (e.g. classification label). Improvements reported in multi-domain learning (Daumé III, 2007; Kim et al., 2016) have often focused on learning twin representations (*shared* and *private* representations) for each instance. The private representation is modelled by introducing a domain-specific channel conditional on the domain, and the shared one is learned through domain-general channels. To learn more robust domain-general and domain-specific channels, adversarial supervision can be applied in the form of either domain-conditional or domain-generative methods (Liu et al., 2016; Li et al., 2018a).

Inspired by these works, we develop a method for the setting where the domain is unobserved or partially observed, which we refer to as unsupervised and semi-supervised, respectively, with respect to domain. This has the added benefit of affording robustness where the test data is drawn from an unseen domain, through modelling each test instance as a mixture of domains. In this paper, we propose methods which use latent variables to characterise the domain, by modelling the discriminative learning problem $p(y|\mathbf{x}) = \int_z p(z|\mathbf{x})p(y|\mathbf{x}, z)$, where z encodes the domain, which must be marginalised

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1923–1934
Florence, Italy, July 28 - August 2, 2019.

out when the domain is unobserved. We propose a sequence of models of increasing complexity in the modelling of the treatment of z, ranging from a discrete mixture model, to a continuous vector-valued latent variable (analogous to a topic model; Blei et al. (2003)), modelled using Beta or Dirichlet distributions. We show how these models can be trained efficiently, using either direct gradient-based methods or variational inference (Kingma et al., 2014), for the respective model types. The variational method can be applied to domain and/or label semi-supervised settings, where not all components of the training data are fully observed.

We evaluate our approach using sentiment analysis over multi-domain product review data and 7 language identification benchmarks from different domains, showing that in out-of-domain evaluation, our methods substantially improve over benchmark methods, including adversarially-trained domain adaptation (Li et al., 2018a). We show that including additional domain unlabelled data gives a substantial boost to performance, resulting in transfer models that often outperform domain-trained models, to the best of our knowledge, setting a new state of the art for the dataset.

2 Stochastic Domain Adaptation

In this section, we describe our proposed approaches to Stochastic Domain Adaptation (SDA), which use latent variables to represent an implicit 'domain'. This is formulated as a joint model of output classification label, y and latent domain z, which are both conditional on $\mathbf{x}$,

$$p(y, z|\mathbf{x}) = p_\phi(\mathbf{z}|\mathbf{x})p_\theta(y|\mathbf{x}, \mathbf{z})\,.$$

The two components are the prior, $p_\phi(\mathbf{z}|\mathbf{x})$, and classifier likelihood, $p_\theta(y|\mathbf{x}, \mathbf{z})$, which are parameterised by ϕ and θ, respectively. We propose several different choices of prior, based on the nature of $\mathbf{z}$, that is, whether it is: (i) a discrete value ("DSDA", see Section 2.2); or (ii) a continuous vector, in which case we experiment with different distributions to model $p(\mathbf{z}|\mathbf{x})$ ("CSDA", see Section 2.3).

2.1 Stochastic Channel Gating

For all of our models the likelihood, $p_\theta(y|\mathbf{x}, \mathbf{z})$, is formulated as a multi-channel neural model, where $\mathbf{z}$ is used as a gate to select which channels should be used in representing the input. The model comprises k channels, with each channel computing an independent hidden representation,

$$\mathbf{h}_i = \text{CNN}_i(\mathbf{x}; \theta)|_{i=1}^k$$

using a convolutional neural network.[1] The value of z is then used to select the channel, by computing $\mathbf{h} = \sum_{i=1}^k z_k \mathbf{h}_i$, where we assume $\mathbf{z} \in \mathbb{R}^k$ is a continuous vector. For the discrete setting, we represent integer z by its 1-hot encoding $\mathbf{z}$, in which case $\mathbf{h} = \mathbf{h}_z$. The final step of the likelihood passes $\mathbf{h}$ through a MLP with a single hidden layer, followed by a softmax, which is used to predict class label y.

2.2 Discrete Domain Identifiers

We now turn to the central part of our method, the prior component. The simplest approach, DSDA (see Figure 1a), uses a discrete latent variable, i.e., $z \in [1, k]$ is an integer-valued random variable, and consequently the model can be considered as a form of mixture model. This prior predicts z given input $\mathbf{x}$, which is modelled using a neural network with a softmax output. Given z, the process of generating y is as described above in Section 2.1. The discrete model can be trained for the maximum likelihood estimate using the objective,

$$\log p(y|\mathbf{x}) = \log \sum_{z=1}^{k} p_\phi(z|\mathbf{x})p_\theta(y|\mathbf{x}, z), \quad (1)$$

which can be computed tractably,[2] and scales linearly in k.

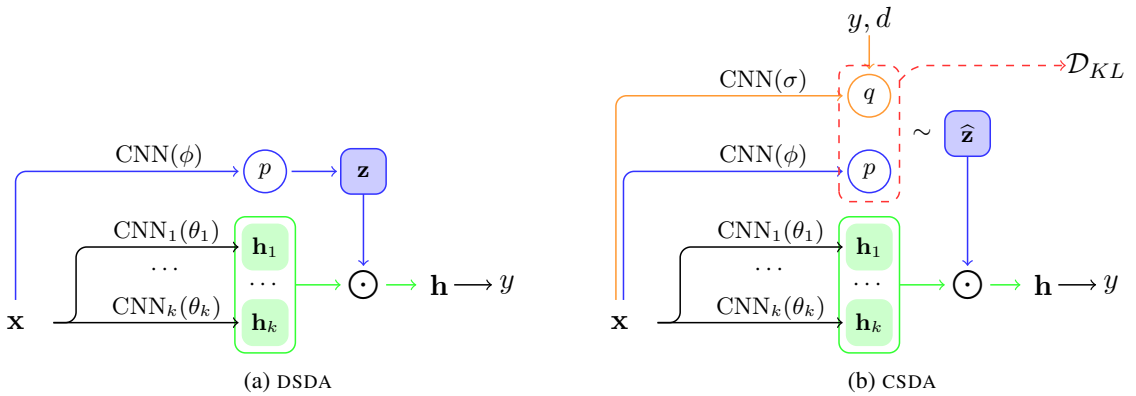
DSDA can be applied with supervised or semi-supervised domains, by maximising the likelihood $p(z = d|\mathbf{x})$ when the ground truth domain d is observed. We refer to this setting as "DSDA +sup." or "DSDA +semisup", respectively, noting that in this setting we assume the number of channels, k, is equal to the known inventory of domains, D.

2.3 Continuous Domain Identifiers

For the DSDA model to work well requires sufficiently large k, such that all the different types of data can be clearly separated into individual mixture components. When there is not a clear delineation between domains, the inferred domain posterior is likely to be uncertain, and the approach

[1] Our approach is general, and could be easily combined with other methods besides CNNs.

[2] This arises from the finite summation in (1), which requires each of the k components to be computed separately, and their results summed. This procedure permits standard gradient back-propagation.



(a) DSDA (b) CSDA

Figure 1: Model architectures for latent variable models, DSDA and CSDA, which differ in the treatment of the latent variable, which is discrete ($d \in [1, k]$), or a continuous vector ($\hat{\mathbf{z}} \in \mathbb{R}^k$). The lower green model components show k independent convolutional network components, and the blue and yellow component the prior, p, and the variational approximation, q, respectfully. The latent variable is used to gate the k hidden representations (shown as $\odot$), which are then used in a linear function to predict a classification label, y. During training CSDA draws samples ($\sim$) from q, while during inference, samples are drawn from p.

reduces to an ensemble technique. Thus, we introduce the second modelling approach as Continuous domain identifiers (CSDA), inspired by the way in which LDA models the documents as mixtures of several topics (Blei et al., 2003).

A more statistically efficient method would be to use binary functions as domain specifiers, i.e., $\mathbf{z} \in \{0, 1\}^k$, effectively allowing for exponentially many domain combinations (2^k). Each element of the domain z_i acts as a gate, or equivalently, attention, governing whether hidden state $\mathbf{h}_i$ is incorporated into the predictive model. In this way, individual components of the model can specialise to a very specific topic such as politics or sport, and yet domains are still able to combine both to produce specialised representations, such as the politics of sport. The use of a latent bit-vector renders inference intractable, due to the marginalisation over exponentially many states. For this reason, we instead make a continuous relaxation, such that $\mathbf{z} \in \mathbb{R}^k$ with each scalar z_i being drawn from a probability distribution parameterised as a function of the input $\mathbf{x}$. These functions can learn to relate aspects of $\mathbf{x}$ with certain domain indexes, e.g., the use of specific words like *baseball* and *innings* relate to a domain corresponding to "sport", thereby allowing the text domains to be learned automatically.

Several possible distributions can be used to model $\mathbf{z} \in \mathbb{R}^k$. Here we consider the following distributions:

Beta which bounds all elements to the range $[0, 1]$, such that $\mathbf{z}$ lies in a hyper-cube;

Dirichlet which also bounds all elements, as for Beta, however $\mathbf{z}$ are also constrained to lie in the probability simplex.

In both cases,[3] each dimension of $\mathbf{z}$ is controlled by different distribution parameters, themselves formulated as different non-linear functions of $\mathbf{x}$. We expect the Dirichlet model to perform the best, based on their widespread use in topic models, and their desirable property of generating a normalised vector, resembling common attention mechanisms (Bahdanau et al., 2015).

Depending on the choice of distribution, the prior is modelled as

$$p(\mathbf{z}|\mathbf{x}) = \text{Beta}\left(\boldsymbol{\alpha}^B, \boldsymbol{\beta}^B\right) \quad (2a)$$

$$\text{or} \quad p(\mathbf{z}|\mathbf{x}) = \text{Dirichlet}\left(\alpha_0 \boldsymbol{\alpha}^D\right), \quad (2b)$$

where the prior parameters are parameterised as neural networks of the input. For the Beta prior,

$$\boldsymbol{\alpha}^B = \text{elu}(\text{f}_{\alpha,B}(\mathbf{x})) + 1 \quad (3a)$$

$$\boldsymbol{\beta}^B = \text{elu}(\text{f}_{\beta,B}(\mathbf{x})) + 1, \quad (3b)$$

where $\text{elu}(\cdot) + 1$ is an element-wise activation function which returns a positive value (Clevert et al., 2016), and $\text{f}_\omega(\cdot)$ is a nonlinear function with parameters ω—here we use a CNN. The Dirichlet prior uses a different parameterisation,

$$\alpha_0 = \exp(\text{f}_{D,0}(\mathbf{x})) \quad (4a)$$

$$\boldsymbol{\alpha}^D = \text{sigmoid}(\text{f}_D(\mathbf{x})), \quad (4b)$$

[3] We also compared Gamma distributions, but they underperformed Beta and Dirichlet models.

where α_0 is a positive-valued overall concentration parameter, used to scale all components in (2b), thus capturing overall sparsity, while $\boldsymbol{\alpha}^D$ models the affinity to each channel.

2.4 Variational Inference

Using continuous latent variables, as described in Section 2.3, gives rise to intractable inference; for this reason we develop a variational inference method based on the variational auto-encoder (Kingma and Welling, 2014). Fitting the model involves maximising the evidence lower bound (ELBO),

$$\begin{aligned}\log p_{\phi,\theta}(y|\mathbf{x}) &= \log \int_{\mathbf{z}} p_\phi(\mathbf{z}|\mathbf{x}) p_\theta(y|\mathbf{z},\mathbf{x}) \\ &\geq \mathop{\mathbb{E}}_{q_\sigma}\left[\log p_\theta(y|\mathbf{z},\mathbf{x})\right] \\ &\quad - \lambda \mathcal{D}_{\mathrm{KL}}\big(q_\sigma(\mathbf{z}|\mathbf{x},y,d)||p_\phi(\mathbf{z}|\mathbf{x})\big)\,,\end{aligned} \tag{5}$$

where q_σ is the variational distribution, parameterised by σ, chosen to match the family of the prior (Beta or Dirichlet) and λ is a hyper-parameter controlling the weight of the KL term. The ELBO in (5) is maximised with respect to σ, ϕ and θ, using stochastic gradient ascent, where the expectation term is approximated using a single sample, $\hat{\mathbf{z}} \sim q_\sigma$, which is used to compute the likelihood directly. Although it is not normally possible to backpropagate gradients through a sample, which is required to learn the variational parameters σ, this problem is usually sidestepped using a reparameterisation trick (Kingma and Welling, 2014). However this method only works for a limited range of distributions, most notably the Gaussian distribution, and for this reason we use the implicit reparameterisation gradient method (Figurnov et al., 2018), which allows for inference with a variety of continuous distributions, including Beta and Dirichlet. We give more details of the implicit reparameterisation method in Appendix A.2.

The variational distribution q, is defined in an analagous way to the prior, p, see (2–4b), i.e., using a neural network parameterisation for the distribution parameters. The key difference is that q conditions not only on $\mathbf{x}$ but also on the target label y and domain d. This is done by embedding both y and d, which are concatenated with a CNN encoding of $\mathbf{x}$, and then transformed into the distribution parameters. Semi-supervised learning with respect to the domain can easily be facilitated by setting d to the domain identifier when it is observed, otherwise using a sentinel value $d = \texttt{UNK}$, for domain-unsupervised instances. The same trick is used for y, to allow for vanilla semi-supervised learning (with respect to target label). The use of y and d allows the inference network to learn to encode these two key variables into z, to encourage the latent variable, and thus model channels, to be informative of both the target label and the domain. This, in concert with the KL term in (5), ensures that the prior, p, must also learn to discriminate for domain and label, based solely on the input text, $\mathbf{x}$.

For inference at test time, we assume that only $\mathbf{x}$ is available as input, and accordingly the inference network cannot be used. Instead we generate a sample from the prior $\hat{\mathbf{z}} \sim p(\mathbf{z}|\mathbf{x})$, which is then used to compute the maximum likelihood label, $\hat{y} = \arg\max_y p(y|\mathbf{x},\hat{\mathbf{z}})$. We also experimented with Monte Carlo methods for test inference, in order to reduce sampling variance, using: (a) prior mean $\bar{\mathbf{z}} = \mu$; (b) Monte Carlo averaging $\bar{y} = \frac{1}{m}\sum_i p(y|\mathbf{x},\hat{\mathbf{z}}_i)$ using $m = 100$ samples from the prior; and (c) importance sampling (Glynn and Iglehart, 1989) to estimate $p(y|\mathbf{x})$ based on sampling from the inference network, q.[4] None of the Monte Carlo methods showed a significant difference in predictive performance versus the single sample technique, although they did show a very tiny reduction in variance over 10 runs. This is despite their being orders of magnitude slower, and therefore we use a single sample for test inference hereafter.

3 Experiments

3.1 Multi-domain Sentiment Analysis

To evaluate the proposed models, we first experiment with a multi-domain sentiment analysis dataset, focusing on out-of-domain evaluation where the test domain is unknown.

We derive our dataset from Multi-Domain Sentiment Dataset v2.0 (Blitzer et al., 2007).[5] The task is to predict a binary sentiment label, i.e., positive vs. negative. The unprocessed dataset has more than 20 domains. For our purposes, we filter out domains with fewer than 1k labelled instances

[4] Importance sampling estimates $p(y|\mathbf{x}) = \mathbb{E}_q[p(y,\mathbf{z}|\mathbf{x})/q(\mathbf{z}|\mathbf{x},y,d)]$ for each setting of y using $m = 100$ samples from q, and then finds the maximising y. This is tractable in our settings as y is a discrete variable, e.g., a binary sentiment, or multiclass language label.

[5] From https://www.cs.jhu.edu/~mdredze/datasets/sentiment/.

Domain	$\mathcal{F}(\mathbf{x}, y, d)$	$\mathcal{Y}(\mathbf{x}, y, ?)$
apparel	1,000	1,000
baby	950	950
camera & photo	1000	999
health & personal care	1,000	1,000
magazines	985	985
music	1,000	1,000
sports & outdoors	1,000	1,000
toys & games	1,000	1,000
video	1,000	1,000

Table 1: Numbers of instances (reviews) for each training domain in our dataset, under the two categories $\mathcal{F}$ (domain and label known) and $\mathcal{Y}$ (label known; domain unknown), in which "?" represents the "UNK" token, meaning the given attribute is unobserved.

or fewer than 2k unlabelled instances, resulting in 13 domains in total.

To simulate the semi-supervised domain situation, we remove the domain attributions for one half of the labelled data, denoting them as domain-unlabelled data $\mathcal{Y}(\mathbf{x}, y, ?)$. The other half are sentiment- and domain-labelled data $\mathcal{F}(\mathbf{x}, y, d)$. We present a breakdown of the dataset in Table 1.[6]

For evaluation, we hold out four domains—namely books ("B"), dvds ("D"), electronics ("E"), and kitchen & housewares ("K")—for comparability with previous work (Blitzer et al., 2007). Each domain has 1k test instances, and we split this data into *dev* and *test* with ratio 4:6. The *dev* dataset is used for hyper-parameter tuning and early stopping,[7] and we report accuracy results on *test*.

3.1.1 Baselines and Comparisons

For comparison, we use 3 baselines. The first is a single channel CNN ("S-CNN"), which jointly over all data instances in a single model, without domain-specific parameters. The second baseline is a multi channel CNN ("M-CNN"), which expands the capacity of the S-CNN model (606k parameters) to match CSDA and DSDA (roughly 7.5m-8.3m parameters). Our third baseline is a multi-domain learning approach using adversarial learning for domain generation ("GEN"), the best-performing model of Li et al. (2018a) and state-of-the-art for unsupervised multi-domain adaptation over a comparable dataset.[8] We report results for their best performing GEN +d+g model.

3.1.2 Training Strategy

For the hyper-parameter setups, we provide the details in Appendix A.1. In terms of training, we simulate two scenarios using two experimental configurations, as discussed above: (a) domain supervision; and (2) domain semi-supervision. For domain supervised training, only $\mathcal{F}$ is used, which covers only 9 of the domains, and the test domain data is entirely unseen. For domain semi-supervised training, we use combinations of $\mathcal{F}$ and $\mathcal{Y}$, noting that both sub-corpora do not include data from the target domains, and none of which is explicitly labelled with sentiment, y, and domain, d. These simulate the setting where we have heterogenous data which includes a lot of relevant data, however its metadata is inconsistent, and thus cannot be easily modelled.

For λ in (5), according to the derivation of the ELBO it should be the case that $\lambda = 1$, however other settings are often justified in practice (Alemi et al., 2018). Accordingly, we tried both annealing and fixed schedules, but found no consistent differences in end performance. We performed a grid search for the fixed value, $\lambda = 10^a$, $a \in \{-3, -2, -1, 0, 1\}$, and selected $\lambda = 10^{-1}$, based on development performance. We provide further analysis in the form of a sensitivity plot in Section 3.2. The latent domain size k for DSDA is set to the true number of training domains $k = D = 9$. Note that, even for DSDA, we could use $k \neq D$, which we explore in the $\mathcal{F} + \mathcal{Y}$ supervision setting in Section 3.1.3. For CSDA we present the main results with $k = 13$, set to match the total number of domains in training and testing.

3.1.3 Results

Table 2 reports the performance of different models under two training configurations: (1) with $\mathcal{F} + \mathcal{Y}$ (domain semi-supervised learning); and (2) with $\mathcal{F}$ only (domain supervised learning). In each case, we report the standard deviation based on 10 runs with different random seeds.

Overall, domain B and D are more difficult than E and K, consistent with previous work. Comparing the two configurations, we see that when we use domain semi-supervised training (with the addition of $\mathcal{Y}$), all models perform bet-

[6] The dataset, along with the source code, can be found at https://github.com/lrank/Code_VariationalInference-Multidomain

[7] This confers light supervision in the target domain. However we would expect similar results were we to use disjoint held out domains for development wrt testing.

[8] The dataset used in Li et al. (2018a) differs slightly in that it is also based off Multi-Domain Sentiment Dataset v2.0, but uses slightly more training domains and a slightly different composition of training data. We retrain the model of the authors over our dataset, using their implementation.

Data			B	D	E	K	Average
$\mathcal{F}+\mathcal{Y}$	S-CNN		78.9 ± 1.3	80.9 ± 1.5	82.4 ± 0.8	84.1 ± 1.8	81.6 ± 0.9
	M-CNN		79.0 ± 1.5	82.5 ± 1.3	84.1 ± 0.8	85.9 ± 0.8	82.9 ± 0.9
	GEN		78.4 ± 0.9	81.2 ± 1.0	83.9 ± 1.7	**87.5** ± 1.2	82.8 ± 1.1
	DSDA		76.8 ± 1.4	79.6 ± 1.7	83.1 ± 1.5	85.8 ± 2.0	81.3 ± 1.0
		+ semi-sup.	77.1 ± 1.6	79.9 ± 1.0	83.1 ± 1.7	85.4 ± 1.3	81.4 ± 0.6
	CSDA	w. Beta	78.4 ± 0.8	**84.4** ± 0.7	82.9 ± 1.1	87.2 ± 1.3	83.2 ± 0.9
		w. Dirichlet	**80.0** ± 1.4	84.3 ± 1.4	**86.2** ± 1.5	87.0 ± 0.3	**84.4** ± 0.9
$\mathcal{F}$ only	S-CNN		76.0 ± 1.8	77.0 ± 1.0	81.5 ± 1.3	82.8 ± 1.6	79.3 ± 0.7
	M-CNN		76.7 ± 1.8	79.2 ± 0.4	82.0 ± 1.2	83.1 ± 1.8	79.8 ± 1.3
	GEN		76.7 ± 2.0	79.1 ± 1.3	82.1 ± 1.6	84.0 ± 1.1	80.5 ± 0.7
	DSDA		74.3 ± 1.4	75.8 ± 2.2	80.5 ± 1.3	82.8 ± 1.4	78.4 ± 0.9
		+ unsup.	74.1 ± 2.0	75.6 ± 2.3	80.8 ± 1.3	83.0 ± 1.7	78.4 ± 0.6
	CSDA	w. Beta	**78.0** ± 1.9	80.5 ± 1.1	83.7 ± 1.3	85.7 ± 1.3	82.0 ± 1.1
		w. Dirichlet	77.9 ± 1.6	**80.6** ± 0.9	**84.4** ± 1.1	**86.5** ± 0.9	**82.3** ± 0.6
	IN DOMAIN ♣		80.4	82.4	84.4	87.7	83.7

Table 2: Accuracy [%] and standard deviation of different models under two data configurations: (1) using both $\mathcal{F}$ and $\mathcal{Y}$ (domain semi-supervised learning); and (2) using $\mathcal{F}$ only (domain supervised learning). In each case, we evaluate over the four held-out test domains (B, D, E and K), and also report the accuracy. Best results are indicated in **bold** in each configuration. Key: ♣ from Blitzer et al. (2007).

ter, demonstrating the utility of domain semi-supervised learning when annotated data is limited.

Comparing our discrete and continuous approaches (DSDA and DSDA, resp.), we see that CSDA consistently performs the best, outperforming the baselines by a substantial margin. In contrast DSDA is disappointing, underperforming the baselines, and moreover, shows no change in performance between domain supervision versus the semi-supervised or unsupervised settings. Among the CSDA based methods, all the distributions perform well, but the Dirichlet distribution performs the best overall, which we attribute to better modelling of the sparsity of domains, thus reducing the influence of uncertain and mixed domains. The best results are for domain semi-supervised learning ($\mathcal{F}+\mathcal{Y}$), which brings an increase in accuracy of about 2% over domain supervised learning ($\mathcal{F}$) consistently across the different types of model.

3.2 Analysis and Discussion

To better understand what the model learns, we focus on the CSDA model, using the Dirichlet distribution.

First, we consider the model capacity, in terms of the latent domain size, k. Figure 2 shows the impact of varying k. Note that the true number of domains is $D = 13$, comprising 9 training and 4 test domains. Setting k to roughly this value appears to be justified, in that the mean accuracy

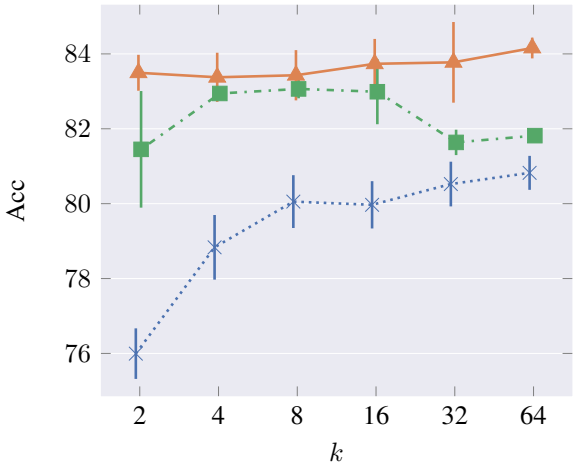


Figure 2: Performance with standard error (|) as latent domain size k is increased in $\log 2$ space with DSDA (×) and with three CSDA methods using Beta (■) and Dirichlet (▲) averaged accuracy, over $\mathcal{F}+\mathcal{Y}$.

increases with k, and plateaus around $k = 16$. Interestingly, when $k \geq 32$, the performance of CSDA with Beta drops, while performance for Dirichlet remains high—indeed Dirichlet is consistently superior even at the extreme value of $k = 2$, although it does show improvement as k increases. Also observe that DSDA requires a large latent state inventory, supporting our argument for the efficiency of continuous cf. discrete latent variables.

Next, we consider the impact of using different combinations of $\mathcal{F}$ and $\mathcal{Y}$. Table 3 shows the performance of difference configurations. Overall, $\mathcal{F}+\mathcal{Y}$ gives excellent performance. Interestingly,

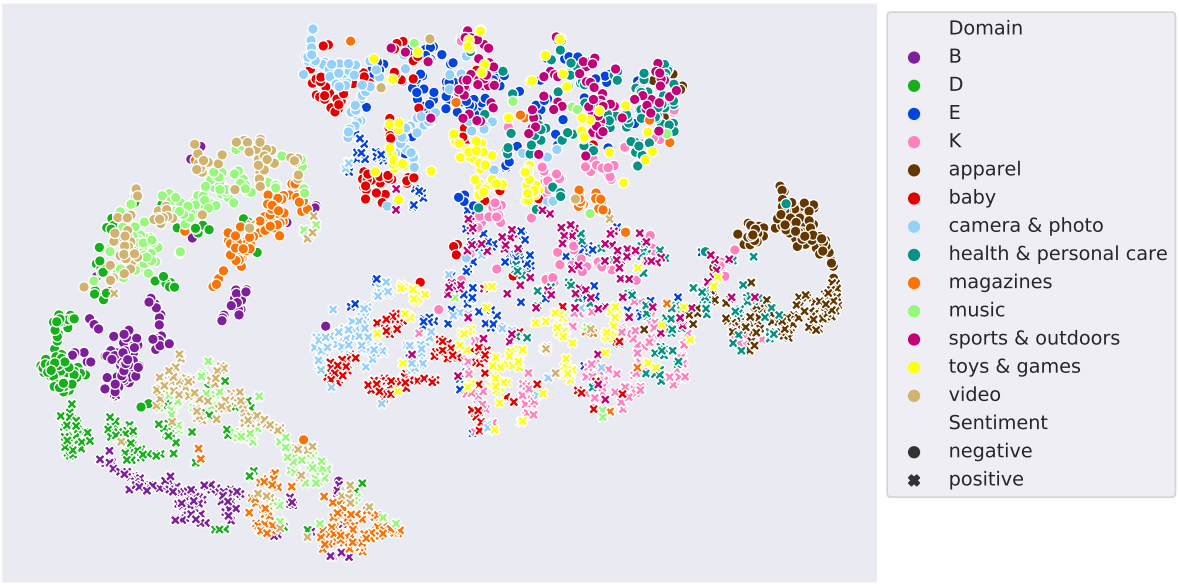


Figure 3: t-SNE of hidden representations in CSDA over all 13 domains, comprising 4 held-out testing domains (B, D, E and K), and the remaindering 9 domains are used only for training. Each point is a document, and the symbol indicates its gold sentiment label, using a filled circle for negative instances and cross for positive.

CSDA	B	D	E	K	Average
$\mathcal{F}$	77.9	80.6	84.4	86.5	82.3
$\mathcal{F}+\mathcal{Y}$	80.0	84.3	86.2	87.0	84.4
$\mathcal{Y}$	77.6	81.5	83.7	85.2	82.0

Table 3: Accuracy [%] of CSDA w. Dirichlet trained with different configurations of $\mathcal{F}$ and $\mathcal{Y}$.

$\mathcal{Y}$ on its own is only a little worse than only $\mathcal{F}$, showing that target labels y are more important for learning than the domain d. The $\mathcal{Y}$ configuration fully domain unsupervised training still results in decent performance, boding well for application to very messy and heterogenous datasets with no domain metadata.

Finally, we consider what is being learned by the model, in terms of how it learns to use the k dimensional latent variables for different types of data. We visualise the learned representations, showing points for each domain plotted in a 2d t-SNE plot (Maaten and Hinton, 2008) in Figure 3. Notice that each domain is split into two clusters, representing positive (×) and negative (•) instances within that domain. Among the test domains, B (books) and D (dvds) are clustered close together but are still clearly separated, which is encouraging given the close relation between these two media. The other two, E (electronics) and K (kitchen & housewares) are mixed together and intermingled with other domains. Overall across all domains, the APPAREL cluster is quite distinct,

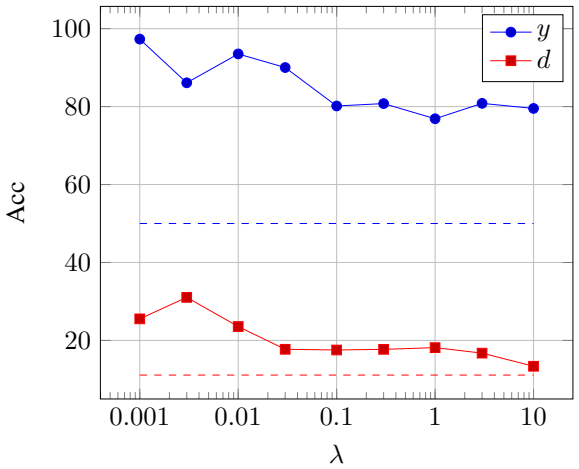


Figure 4: Diagostic classifier accuracy [%] over $\mathbf{z}$ to predict the sentiment label y and domain label d, with respect to different λ, shown on a log scale. Dashed horizontal lines show chance accuracy for both outputs.

while VIDEO and MUSIC are highly associated with D, and part of the cluster for MAGAZINES is close to B; all of these make sense intuitively, given similarities between the respective products. E is related to CAMERA and GAMES, while K is most closely connected to HEALTH and SPORTS.

To obtain a better understanding of what is being encoded in the latent variable, and how this is effected by the setting of λ, we learn simple diagnostic classifiers to predict sentiment label y and domain label d, given only $\mathbf{z}$ as input. To do so, we first train our model over the training set, and

Data			EUROGOV	TCL	WIKIPEDIA	EMEA	EUROPARL	TBE	TSC	Average
$\mathcal{Y}$	S-CNN		98.8	92.3	85.9	98.5	92.3	79.3	91.7	91.3
	M-CNN		**98.9**	**93.6**	86.2	99.2	96.0	88.3	91.7	93.4
	DSDA		98.3	91.9	86.3	97.8	95.2	86.0	79.0	90.6
	CSDA	w. Beta	98.7	93.0	**89.0**	**99.3**	96.8	93.1	**95.2**	**95.0**
		w. Dirichlet	**98.9**	93.0	**89.0**	99.2	**96.7**	**93.2**	94.5	94.9
$\mathcal{F}$	DSDA		98.0	91.8	85.7	97.7	95.3	85.4	78.1	90.3
	CSDA	w. Beta	**99.3**	**93.7**	89.1	99.2	**96.9**	**93.6**	93.9	95.1
		w. Dirichlet	99.0	**93.7**	**89.3**	**99.3**	**96.9**	93.3	**96.1**	**95.4**
	GEN		99.9	93.1	88.7	92.5	97.1	91.2	96.1	94.1
	LANGID.PY		98.7	90.4	91.3	93.4	97.4	94.1	92.7	94.0

Table 4: Accuracy [%] over 7 LangID benchmarks, as well as the averaged score, for different models under two data configurations: (1) using domain unsupervised learning ($\mathcal{Y}$); and (2) using domain supervised learning ($\mathcal{F}$). The best results are indicated in **bold** in each configuration. Note that the training data for GEN and LANGID.PY is slightly different from that used in the original papers.

record samples of $\mathbf{z}$ from the inference network. We then partition the training set, using 70% to learn linear logistic regression classifiers to predict y and d, and use the remaining 30% for evaluation. Figure 4 shows the prediction accuracy, based on averaging over three runs, each with different $\mathbf{z}$ samples. Clearly very small $\lambda \leq 10^{-2}$, leads to almost perfect sentiment label accuracy which is evidence of overfitting by using the latent variable to encode the response variable. For $\lambda \geq 10^{-1}$ the sentiment accuracy is still above chance, as expected, but is more stable. For the domain label d, the predictive accuracy is also above chance, albeit to a lesser extent, and shows a similar downward trend. At the setting $\lambda = 0.1$, used in the earlier experiments, this shows that the latent variable encodes captures substantial sentiment, and some domain knowledge, as observed in Figure 3.

In terms of the time required for training, a single epoch of training took about 25min for the CSDA method, using the default settings, and a similar time for DSDA and M-CNN. The runtime increases sub-linearly with increasing latent size k.

3.3 Language Identification

To further demonstrate our approaches, we then evaluate our models with the second task, language identification (LangID: Jauhiainen et al. (2018)).

For data processing, we use 5 training sets from 5 different domains with 97 language, following the setup of Lui and Baldwin (2011). We evaluate accuracy over 7 holdout benchmarks: EUROGOV, TCL, WIKIPEDIA from Baldwin and Lui (2010), EMEA (Tiedemann, 2009), EUROPARL (Koehn, 2005), TBE (Tromp and Pechenizkiy, 2011) and TSC (Carter et al., 2013). Differently from sentiment tasks, here, we evaluate our methods using the full dataset, but with two configurations: (1) domain unsupervised, where all instance have only labels but no domain (denoted $\mathcal{Y}$); and (2) domain supervised learning, where all instances have labels and domain ($\mathcal{F}$).

3.3.1 Results

Table 4 shows the performance of different models over 7 holdout benchmarks and the averaged scores. We also report the results of GEN, the best model from Li et al. (2018a), and one state-of-the-art off-the-shelf LangID tool: LANGID.PY (Lui and Baldwin, 2012). Note that, both S-CNN and M-CNN are domain unsupervised methods. In terms of results, overall, both of our CSDA models consistently outperform all other baseline models. Comparing the different CSDA variants, Beta vs. Dirichlet, both perform closely across the LangID tasks. Furthermore, CSDA out-performs the state-of-the-art in terms of average scores. Interestingly the two training configurations show that domain knowledge $\mathcal{F}$ provides a small performance boost for CSDA, but not does help for DSDA. Above all, the LangID results confirm the effectiveness of our proposed approaches.

4 Related Work

Domain adaptation ("DA") typically involves one or more training domains and a single target domain. Among DA approaches, single-domain adaptation is the most common scenario, where a model is trained over one domain and then transferred to a single target domain using prior

knowledge of the target domain (Blitzer et al., 2007; Glorot et al., 2011). Adversarial learning methods have been proposed for learning robust domain-independent representations, which can capture domain knowledge through semi-supervised learning (Ganin et al., 2016).

Multi-domain adaptation uses training data from more than one training domain. Approaches include feature augmentation methods (Daumé III, 2007), and analagous neural models (Joshi et al., 2012; Kim et al., 2016), as well as attention-based and hierarchical methods (Li et al., 2018b). These works assume the 'oracle' source domain is known when transferring, however we do not require an oracle in this paper. Adversarial training methods have been employed to learn robust domain-generalised representations (Liu et al., 2016). Li et al. (2018a) considered the case of the model having no access to the target domain, and using adversarial learning to generate domain-generation representations by cross-comparison between source domains.

The other important component of this work is Variational Inference ("VI"), a method from machine learning that approximates probability densities through optimisation (Blei et al., 2017; Kucukelbir et al., 2017). The idea of a variational auto-encoder has been applied to language generation (Bowman et al., 2016; Kim et al., 2018; Miao et al., 2017; Zhou and Neubig, 2017; Zhang et al., 2016) and machine translation (Shah and Barber, 2018; Eikema and Aziz, 2018), but not in the context of semi-supervised domain adaptation.

5 Conclusion

In this paper, we have proposed two models—DSDA and CSDA—for multi-domain learning, which use a graphical model with a latent variable to represent the domain. We propose models with a discrete latent variable, and a continuous vector-valued latent variable, which we model with Beta or Dirichlet priors. For training, we adopt a variational inference technique based on the variational autoencoder. In empirical evaluation over a multi-domain sentiment dataset and seven language identification benchmarks, our models outperform strong baselines, across varying data conditions, including a setting where no target domain data is provided. Our proposed models have broad utility across NLP applications on heterogenous corpora.

Acknowledgements

This work was supported by an Amazon Research Award. We thank the anonymous reviewers for their helpful feedback and suggestions.

References

Alexander Alemi, Ben Poole, Ian Fischer, Joshua Dillon, Rif A Saurous, and Kevin Murphy. 2018. Fixing a broken elbo. In *International Conference on Machine Learning*, pages 159–168.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*.

Timothy Baldwin and Marco Lui. 2010. Language identification: The long and the short of the matter. In *Proceedings of Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics*, pages 229–237.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. 2017. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022.

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

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21.

Simon Carter, Wouter Weerkamp, and Manos Tsagkias. 2013. Microblog language identification: Overcoming the limitations of short, unedited and idiomatic text. *Language Resources and Evaluation*, 47(1):195–215.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2016. Fast and accurate deep network learning by exponential linear units (ELUs). In *Proceedings of the International Conference on Learning Representations*.

Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263.

Bryan Eikema and Wilker Aziz. 2018. Auto-encoding variational neural machine translation. *arXiv preprint arXiv:1807.10564*.

Mikhail Figurnov, Shakir Mohamed, and Andriy Mnih. 2018. Implicit reparameterization gradients. In *Advances in Neural Information Processing Systems 31*, pages 439–450.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17:59:1–59:35.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning*, pages 513–520.

Peter W Glynn and Donald L Iglehart. 1989. Importance sampling for stochastic simulations. *Management Science*, 35(11):1367–1392.

Tommi Jauhiainen, Marco Lui, Marcos Zampieri, Timothy Baldwin, and Krister Lindén. 2018. Automatic language identification in texts: A survey. *CoRR*, abs/1804.08186.

Mahesh Joshi, Mark Dredze, William W. Cohen, and Carolyn Penstein Rosé. 2012. Multi-domain learning: When do domains matter? In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1302–1312.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751.

Yoon Kim, Sam Wiseman, Andrew Miller, David Sontag, and Alexander Rush. 2018. Semi-amortized variational autoencoders. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2678–2687.

Young-Bum Kim, Karl Stratos, and Ruhi Sarikaya. 2016. Frustratingly easy neural domain adaptation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 387–396.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*.

Diederik P. Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. 2014. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014*, pages 3581–3589.

Diederik P Kingma and Max Welling. 2014. Auto-encoding variational Bayes. In *Proceedings of the International Conference on Learning Representations*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT Summit 2005*, pages 79–86.

Alp Kucukelbir, Dustin Tran, Rajesh Ranganath, Andrew Gelman, and David M Blei. 2017. Automatic differentiation variational inference. *Journal of Machine Learning Research*, 18(1):430–474.

Yitong Li, Timothy Baldwin, and Trevor Cohn. 2018a. What's in a domain? learning domain-robust text representations using adversarial training. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 474–479.

Zheng Li, Ying Wei, Yu Zhang, and Qiang Yang. 2018b. Hierarchical attention transfer network for cross-domain sentiment classification. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Deep multi-task learning with shared memory for text classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 118–127.

Marco Lui and Timothy Baldwin. 2011. Cross-domain feature selection for language identification. In *Fifth International Joint Conference on Natural Language Processing*, pages 553–561.

Marco Lui and Timothy Baldwin. 2012. langid.py: An off-the-shelf language identification tool. In *Proceedings of ACL 2012 System Demonstrations*, pages 25–30.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605.

Yishu Miao, Edward Grefenstette, and Phil Blunsom. 2017. Discovering discrete latent topics with neural variational inference. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2410–2419.

Harshil Shah and David Barber. 2018. Generative neural machine translation. In *Advances in Neural Information Processing Systems*, pages 1346–1355.

Jörg Tiedemann. 2009. News from OPUS – a collection of multilingual parallel corpora with tools and interfaces. In *Recent Advances in Natural Language Processing*, volume 5, pages 237–248.

Erik Tromp and Mykola Pechenizkiy. 2011. Graph-based n-gram language identification on short texts.

In *Proceedings of the 20th Machine Learning Conference of Belgium and The Netherlands*, pages 27–34.

Biao Zhang, Deyi Xiong, Jinsong Su, Hong Duan, and Min Zhang. 2016. Variational neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 521–530.

Chunting Zhou and Graham Neubig. 2017. Multi-space variational encoder-decoders for semi-supervised labeled sequence transduction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 310–320.

A Appendices

A.1 Base Model Architecture

For the sentiment task, all the hidden representations are learned by convolutional neural networks (CNN), following Kim (2014). All documents are lower-cased and truncated to maximum 256 tokens, and then each word is mapped into a 300 dimensional vector representation using randomly-initialised word embeddings. In each CNN channel, filter windows are set to $\{3, 4, 5\}$, with 128 filters for each. Then, ReLU and pooling are applied after the filtering, generating 384-d ($128 * 3$) hidden representations. Dropout is applied to the hidden $\mathbf{h}$, at a rate of 0.5. For simplicity, we use the same CNN architecture to encode the functions f used in the prior q and in the inference networks p, in each case with different parameters. Specifically, in prior q, the embedding sizes of domain and label are set to 16 and 4, respectively. $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ share the same CNN but with different output projections. After gating using $\mathbf{z}$, the final hidden goes through a one-hidden MLPwith hidden size 300. We use the Adam optimiser (Kingma and Ba, 2015) throughout, with the learning rate set to 10^{-4} and a batch size of 32, optimising the loss functions (1) or (5), for DSDA and CSDA, respectively.

For the language identification task, all documents are tokenized as a byte sequence, truncated or padded to a length of 1k bytes. We use the same CNN architecture and hyper-parameter configurations as for the sentiment task.

A.2 Implicit Reparameterisation Gradient

In this section, we outline the implicit reparameterisation gradient method of Figurnov et al. (2018). First, we review some background on variational inference. We start by defining a differentiable and invertible standardization function as

$$S_\sigma(\mathbf{z}) = \epsilon \sim q(\epsilon)\,, \tag{6a}$$

which describes a mapping between points drawn from a specific distribution function and a standard distribution, q. For example, for a Gaussian distribution $z \sim \mathcal{N}(\mu, \psi)$, we can define $S_{\mu,\psi}(z) = (z - \mu)/\psi \sim \mathcal{N}(0, 1)$ to map to the standard Normal. We aim to compute the gradient of the expectation of a objective function $f(\mathbf{z})$,

$$\nabla_\sigma \mathop{\mathbb{E}}_{q_\sigma(\mathbf{z})}[f(\mathbf{z})] = \mathop{\mathbb{E}}_{q(\epsilon)}[\nabla_\sigma f(S^{-1}(\epsilon))]\,, \tag{6b}$$

where in ELBO (5) in our case, $f(\mathbf{z}) = p_\theta(y|\mathbf{z}, \mathbf{x})$ is the likelihood function.

The implicit reparameterisation gradient technique is a way of computing the reparameterisation without the need for inversion of the standardization function. This works by applying $\nabla_\sigma S^{-1}(\epsilon) = \nabla_\sigma \mathbf{z}$,

$$\nabla_\sigma \mathop{\mathbb{E}}_{q_\sigma(\mathbf{z})}[f(\mathbf{z})] = \mathop{\mathbb{E}}_{q_\sigma(\mathbf{z})}[\nabla_\mathbf{z} f(\mathbf{z}) \nabla_\sigma \mathbf{z}]\,. \tag{6c}$$

However, we still need to calculate $\nabla_\sigma \mathbf{z}$. The key insight here is that we can compute $\nabla_\sigma \mathbf{z}$ by *implicit differentiation*. We apply the total gradient $\nabla^{\text{TD}}_\sigma$ over (6a),

$$\nabla^{\text{TD}}_\sigma S_\sigma(\mathbf{z}) = \nabla^{\text{TD}}_\sigma \epsilon\,. \tag{6d}$$

From the definition of a standardization function, the noise ϵ is independent of σ, and we apply the multi-variable chain rule over left side of (6d),

$$\frac{\partial S_\sigma(\mathbf{z})}{\partial \mathbf{z}} \nabla_\sigma \mathbf{z} + \frac{\partial S_\sigma(\mathbf{z})}{\partial \sigma} = \mathbf{0}\,. \tag{6e}$$

Therefore, the key of the implicit gradient calculation in this process can be summarised as

$$\nabla_\sigma \mathbf{z} = -(\nabla_\mathbf{z} S_\sigma(\mathbf{z}))^{-1} \nabla_\sigma S_\sigma(\mathbf{z})\,. \tag{6f}$$

This expression allows for computation of (6c), which can be applied to a range of distribution families. We refer the reader to Figurnov et al. (2018) for further details.

3.6 ACL 2018: Towards Model Fairness and Privacy

Li, Yitong, Timothy Baldwin and Trevor Cohn (2018) Towards Robust and Privacy-preserving Text Representations, In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018, Volume 2: Short Papers)*, Melbourne, Australia, pp. 25–30.

3.6.1 Research Contributions

In this paper, I focus on the third research question of improving model fairness, and how this can help preserve privacy.

First, to investigate model fairness, a POS tagging task is considered, with two demographic variables (gender and age). The test data is partitioned according to age or gender. To demonstrate model fairness, the model should be unbiased and performance equally over the two groups of test data. To demonstrate model robustness, models are further tested over an out-of-domain POS tagging dataset, containing African-American vernacular English (AAVE), to investigate whether the demographic-adversarial model is also fair over unseen demographic variables.

Second, to investigate the question of user privacy, the Trustpilot sentiment dataset is used, with the addition of location variables for users. To assess the privacy-preserving abilities of a model, I adopt an inference attack over the learned hidden representation to detect the three demographic variables. The experiments show that conventional training tends to expose user information.

To achieve better model fairness and privacy-preservation, I reuse the technique from Section 3.4 to adversarially remove demographic information. The results show the proposed approaches do not sacrifice performance but learn an unbiased representation towards the two attributes, achieving better model fairness. Moreover, we observe improvements over the out-of-domain AAVE corpus. The experiments demonstrate that the proposed model does not reduce overall performance, while reducing the vulnerability of the models to inference attacks.

This paper address the third research question to model fairness and privacy-preservation. And I return to discuss the limitations in Section 4.3.5.

Towards Robust and Privacy-preserving Text Representations

Yitong Li **Timothy Baldwin** **Trevor Cohn**
School of Computing and Information Systems
The University of Melbourne, Australia
yitongl4@student.unimelb.edu.au
{tbaldwin,tcohn}@unimelb.edu.au

Abstract

Written text often provides sufficient clues to identify the author, their gender, age, and other important attributes. Consequently, the authorship of training and evaluation corpora can have unforeseen impacts, including differing model performance for different user groups, as well as privacy implications. In this paper, we propose an approach to explicitly obscure important author characteristics at training time, such that representations learned are invariant to these attributes. Evaluating on two tasks, we show that this leads to increased privacy in the learned representations, as well as more robust models to varying evaluation conditions, including out-of-domain corpora.

1 Introduction

Language is highly diverse, and differs according to author, their background, and personal attributes such as gender, age, education and nationality. This variation can have a substantial effect on NLP models learned from text (Hovy et al., 2015), leading to significant variation in inferences across different types of corpora, such as the author's native language, gender and age. Training corpora are never truly representative, and therefore models fit to these datasets are biased in the sense that they are much more effective for texts from certain groups of user, e.g., middle-aged white men, and considerably poorer for other parts of the population (Hovy, 2015). Moreover, models fit to language corpora often fixate on author attributes which correlate with the target variable, e.g., gender correlating with class skews (Zhao et al., 2017), or translation choices (Rabinovich et al., 2017). This signal, however, is rarely fundamental to the task of modelling language, and is better considered as a confounding influence. These auxiliary learning signals can mean the models do not adequately capture the core linguistic problem. In such situations, removing these confounds should give better generalisation, especially for out-of-domain evaluation, a similar motivation to research in domain adaptation based on selection biases over text domains (Blitzer et al., 2007; Daumé III, 2007).

Another related problem is privacy: texts convey information about their author, often inadvertently, and many individuals may wish to keep this information private. Consider the case of the *AOL search data leak*, in which AOL released detailed search logs of many of their users in August 2006 (Pass et al., 2006). Although they de-identified users in the data, the log itself contained sufficient personally identifiable information that allowed many of these individuals to be identified (Jones et al., 2007). Other sources of user text, such as emails, SMS messages and social media posts, would likely pose similar privacy issues. This raises the question of how the corpora, or models built thereupon, can be distributed without exposing this sensitive data. This is the problem of *differential privacy*, which is more typically applied to structured data, and often involves data masking, addition or noise, or other forms of corruption, such that formal bounds can be stated in terms of the likelihood of reconstructing the protected components of the dataset (Dwork, 2008). This often comes at the cost of an accuracy reduction for models trained on the corrupted data (Shokri and Shmatikov, 2015; Abadi et al., 2016).

Another related setting is where latent representations of the data are shared, rather than the text itself, which might occur when sending data from a phone to the cloud for processing, or trusting a third party with sensitive emails for NLP

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 25–30
Melbourne, Australia, July 15 - 20, 2018.

processing, such as grammar correction or translation. The transfered representations may still contain sensitive information, however, especially if an adversary has preliminary knowledge of the training model, in which case they can readily reverse engineer the input, for example, by a GAN attack algorithm (Hitaj et al., 2017). This is true even when differential privacy mechanisms have been applied.

Inspired by the above works, and recent successes of adversarial learning (Goodfellow et al., 2014; Ganin et al., 2016), we propose a novel approach for privacy-preserving learning of unbiased representations.[1] Specially, we employ Ganin et al.'s approach to training deep models with adversarial learning, to explicitly obscure individuals' private information. Thereby the learned (hidden) representations of the data can be transferred without compromising the authors' privacy, while still supporting high-quality NLP inference. We evaluate on the tasks of POS-tagging and sentiment analysis, protecting several demographic attributes — gender, age, and location — and show empirically that doing so does not hurt accuracy, but instead can lead to substantial gains, most notably in out-of-domain evaluation. Compared to differential privacy, we report gains rather than loss in performance, but note that we provide only empirical improvements in privacy, without any formal guarantees.

2 Methodology

We consider a standard supervised learning situation, in which inputs $\mathbf{x}$ are used to compute a representation $\mathbf{h}$, which then forms the parameterisation of a generalised linear model, used to predict the target $\mathbf{y}$. Training proceeds by minimising a differentiable loss, e.g., cross entropy, between predictions and the ground truth, in order to learn an estimate of the model parameters, denoted θ_M.

Overfitting is a common problem, particular in deep learning models with large numbers of parameters, whereby $\mathbf{h}$ learns to capture specifics of the training instances which do not generalise to unseen data. Some types of overfitting are insidious, and cannot be adequately addressed with standard techniques like dropout or regularisation. Consider, for example, the authorship of each sentence in the training set in a sentiment prediction task. Knowing the author, and their general disposition, will likely provide strong clues about their sentiment wrt any sentence. Moreover, given the ease of authorship attribution, a powerful learning model might learn to detect the author from their text, and use this to predict the sentiment, rather than basing the decision on the semantics of each sentence. This might be the most efficient use of model capacity if there are many sentences by this individual in the training dataset, yet will lead to poor generalisation to test data authored by unseen individuals.

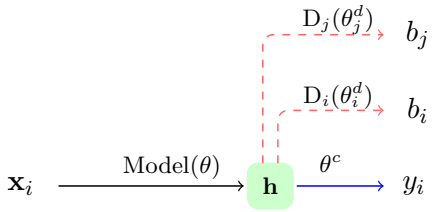


Figure 1: Proposed model architectures, showing a single training instance $(\mathbf{x}_i, y_i)$ with two protected attributes, b_i and b_j. D indicates a discriminator, and the red dashed and blue lines denote adversarial and standard loss, respectively.

Moreover, this raises privacy issues when $\mathbf{h}$ is known by an attacker or malicious adversary. Traditional privacy-preserving methods, such as added noise or masking, applied to the representation will often incur a cost in terms of a reduction in task performance. Differential privacy methods are unable to protect the user privacy of $\mathbf{h}$ under adversarial attacks, as described in Section 1.

Therefore, we consider how to learn an unbiased representations of the data with respect to specific attributes which we expect to behave as confounds in a generalisation setting. To do so, we take inspiration from adversarial learning (Goodfellow et al., 2014; Ganin et al., 2016). The architecture is illustrated in Figure 1.

2.1 Adversarial Learning

A key idea of adversarial learning, following Ganin et al. (2016), is to learn a discriminator model D jointly with learning the standard supervised model. Using gender as an example, a discriminator will attempt to predict the gender, b, of each instance from $\mathbf{h}$, such that training involves joint learning of both the model parameters, θ_M, and the discriminator parameters θ_D. However,

[1] Implementation available at https://github.com/lrank/Robust_and_Privacy_preserving_Text_Representations.

the aim of learning for these components are in opposition – we seek a $\mathbf{h}$ which leads to a good predictor of the target $\mathbf{y}$, while being a poor representation for prediction of gender. This leads to the objective (illustrated for a single training instance),

$$\hat{\theta} = \min_{\theta_M} \max_{\theta_{\mathrm{D}}} \mathcal{X}(\hat{\mathbf{y}}(\mathbf{x}; \theta_M), \mathbf{y}) - \lambda \cdot \mathcal{X}(\hat{b}(\mathbf{x}; \theta_{\mathrm{D}}), b)\,, \quad (1)$$

where $\mathcal{X}$ denotes the cross entropy function. The negative sign of the second term, referred to as the adversarial loss, can be implemented by a gradient reversal layer during backpropagation (Ganin et al., 2016). To elaborate, training is based on standard gradient backpropagation for learning the main task, but for the auxiliary task, we start with standard loss backpropagation, however gradients are reversed in sign when they reach $\mathbf{h}$. Consequently the linear output components are trained to be good predictors, but $\mathbf{h}$ is trained to be maximally good for the main task and maximally poor for the auxiliary task.

Furthermore, Equation 1 can be expanded to scenarios with several (N) protected attributes,

$$\hat{\theta} = \min_{\theta_M} \max_{\{\theta_{\mathrm{D}^i}\}_{i=1}^{N}} \mathcal{X}(\hat{\mathbf{y}}(\mathbf{x}; \theta_M), \mathbf{y}) - \sum_{i=1}^{N} \left(\lambda_i \cdot \mathcal{X}(\hat{b}(\mathbf{x}; \theta_{\mathrm{D}^i}), b_i)\right)\,. \quad (2)$$

3 Experiments

In this section, we report experimental results for our methods with two very different language tasks.

3.1 POS-tagging

This first task is part-of-speech (POS) tagging, framed as a sequence tagging problem. Recent demographic studies have found that the author's gender, age and race can influence tagger performance (Hovy and Søgaard, 2015; Jørgensen et al., 2016). Therefore, we use the POS tagging to demonstrate that our model is capable of eliminating this type of bias, thereby leading to more robust models of the problem.

Model Our model is a bi-directional LSTM for POS tag prediction (Hochreiter and Schmidhuber, 1997), formulated as:

$$\begin{aligned}\mathbf{h}_i &= \mathrm{LSTM}(\mathbf{x}_i, \mathbf{h}_{i-1}; \theta_h)\\ \mathbf{h}'_i &= \mathrm{LSTM}(\mathbf{x}_i, \mathbf{h}'_{i+1}; \theta'_h)\\ \mathbf{y}_i &\sim \mathrm{Categorical}(\phi([\mathbf{h}_i; \mathbf{h}'_i]); \theta_o)\,,\end{aligned}$$

for input sequence $\mathbf{x}_i|_{i=1}^{n}$ with terminal hidden states $\mathbf{h}_0$ and $\mathbf{h}'_{n+1}$ set to zero, where ϕ is a linear transformation, and $[\cdot;\cdot]$ denotes vector concatenation.

For the adversarial learning, we use the training objective from Equation 2 to protect gender and age, both of which are treated as binary values. The adversarial component is parameterised by 1-hidden feedforward nets, applied to the final hidden representation $[\mathbf{h}_n; \mathbf{h}'_0]$. For hyperparameters, we fix the size of the word embeddings and $\mathbf{h}$ to 300, and set all λ values to 10^{-3}. A dropout rate of 0.5 is applied to all hidden layers during training.

Data We use the TrustPilot English POS tagged dataset (Hovy and Søgaard, 2015), which consists of 600 sentences, each labelled with both the sex and age of the author, and manually POS tagged based on the Google Universal POS tagset (Petrov et al., 2012). For the purposes of this paper, we follow Hovy and Søgaard's setup, categorising SEX into female (F) and male (M), and AGE into over-45 (O45) and under-35 (U35). We train the taggers both with and without the adversarial loss, denoted ADV and BASELINE, respectively.

For evaluation, we perform a 10-fold cross validation, with a train:dev:test split using ratios of 8:1:1. We also follow the evaluation method in Hovy and Søgaard (2015), by reporting the tagging accuracy for sentences over different slices of the data based on SEX and AGE, and the absolute difference between the two settings.

Considering the tiny quantity of text in the TrustPilot corpus, we use the Web English Treebank (WebEng: Bies et al. (2012)), as a means of pre-training the tagging model. WebEng was chosen to be as similar as possible in domain to the TrustPilot data, in that the corpus includes unedited user generated internet content.

As a second evaluation set, we use a corpus of African-American Vernacular English (AAVE) from Jørgensen et al. (2016), which is used purely for held-out evaluation. AAVE consists of three very heterogeneous domains: LYRICS, SUBTITLES and TWEETS. Considering the substantial

	SEX			AGE		
	F	M	Δ	O45	U35	Δ
BASELINE	90.9	91.1	0.2	91.4	89.9	1.5
ADV	**92.2**	**92.1**	**0.1**	**92.3**	**92.0**	**0.3**

Table 1: POS prediction accuracy [%] using the Trustpilot test set, stratified by SEX and AGE (higher is better), and the absolute difference (Δ) within each bias group (smaller is better). The best result is indicated in **bold**.

difference between this corpus and WebEng or TrustPilot, and the lack of any domain adaptation, we expect a substantial drop in performance when transferring models, but also expect a larger impact from bias removal using ADV training.

Results and analysis Table 1 shows the results for the TrustPilot dataset. Observe that the disparity for the BASELINE tagger accuracy (the Δ column), for AGE is larger than for SEX, consistent with the results of Hovy and Søgaard (2015). Our ADV method leads to a sizeable reduction in the difference in accuracy across both SEX and AGE, showing our model is capturing the bias signal less and more robust to the tagging task. Moreover, our method leads to a substantial improvement in accuracy across all the test cases. We speculate that this is a consequence of the regularising effect of the adversarial loss, leading to a better characterisation of the tagging problem.

Table 2 shows the results for the AAVE held-out domain. Note that we do not have annotations for SEX or AGE, and thus we only report the overall accuracy on this dataset. Note that ADV also significantly outperforms the BASELINE across the three heldout domains.

Combined, these results demonstrate that our model can learn relatively gender and age *de*-biased representations, while simultaneously improving the predictive performance, both for in-domain and out-of-domain evaluation scenarios.

3.2 Sentiment Analysis

The second task we use is sentiment analysis, which also has broad applications to the online community, as well as privacy implications for the authors whose text is used to train our models. Many user attributes have been shown to be easily detectable from online review data, as used extensively in sentiment analysis results (Hovy et al., 2015; Potthast et al., 2017). In this paper, we focus on three demographic variables of gender, age, and location.

	LYRICS	SUBTITLES	TWEETS	Average
BASELINE	73.7	81.4	59.9	71.7
ADV	**80.5**	**85.8**	**65.4**	**77.0**

Table 2: POS predictive accuracy [%] over the AAVE dataset, stratified over the three domains, alongside the macro-average accuracy. The best result is indicated in **bold**.

Model Sentiment is framed as a 5-class text classification problem, which we model using Kim (2014)'s convolutional neural net (CNN) architecture, in which the hidden representation is generated by a series of convolutional filters followed a maxpooling step, simply denote as $\mathbf{h} = \text{CNN}(\mathbf{x}; \theta_M)$. We follow the hyper-parameter settings of Kim (2014), and initialise the model with word2vec embeddings (Mikolov et al., 2013). We set the λ values to 10^{-3} and apply a dropout rate of 0.5 to $\mathbf{h}$.

As the discriminator, we also use a feed-forward model with one hidden layer, to predict the private attribute(s). We compare models trained with zero, one, or all three private attributes, denoted BASELINE, ADV-*, and ADV-all, respectively.

Data We again use the TrustPilot dataset derived from Hovy et al. (2015), however now we consider the RATING score as the target variable, not POS-tag. Each review is associated with three further attributes: gender (SEX), age (AGE), and location (LOC). To ensure that LOC cannot be trivially predicted based on the script, we discard all non-English reviews based on LANGID.PY (Lui and Baldwin, 2012), by retaining only reviews classified as English with a confidence greater than 0.9. We then subsample 10k reviews for each location to balance the five location classes (US, UK, Germany, Denmark, and France), which were highly skewed in the original dataset. We use the same binary representation of SEX and AGE as the POS task, following the setup in Hovy et al. (2015).

To evaluate the different models, we perform 10-fold cross validation and report test performance in terms of the F_1 score for the RATING task, and the accuracy of each discriminator. Note that the discriminator can be applied to test data, where it plays the role of an adversarial attacker, by trying to determine the private attributes of

	F_1		Discrim. [%]		
	dev	test	AGE	SEX	LOC
Majority class			57.8	62.3	20.0
BASELINE	41.9	40.1	65.3	66.9	53.4
ADV-AGE	**42.7**	40.1	**61.1**	65.6	41.0
ADV-SEX	42.4	39.9	61.8	62.9	42.7
ADV-LOC	42.0	**40.2**	62.2	66.8	**22.1**
ADV-all	42.0	**40.2**	61.8	**62.5**	28.1

Table 3: Sentiment F_1-score [%] over the RATING task, and accuracy [%] of all the discriminator across three private attributes. The best score is indicated in **bold**. The majority class with respect to each private attribute is also reported.

users based on their hidden representation. That is, lower discriminator performance indicates that the representation conveys better privacy for individuals, and vice versa.

Results Table 3 shows the results of the different models. Note that all the privacy attributes can be easily detected in BASELINE, with results that are substantially higher than the majority class, although AGE and SEX are less well captured than LOC. The ADV trained models all maintain the task performance of the BASELINE method, however they clearly have a substantial effect on the discrimination accuracy. The privacy of SEX and LOC is substantially improved, leading to discriminators with performance close to that of the majority class (conveys little information). AGE proves harder, although our technique leads to privacy improvements. Note that AGE appears to be related to the other private attributes, in that privacy is improved when optimising an adversarial loss for the other attributes (SEX and LOC).

Overall, these results show that our approach learns hidden representations that hide much of the personal information of users, without affecting the sentiment task performance. This is a surprising finding, which augurs well for the use of deep learning as a privacy preserving mechanism when handling text corpora.

4 Conclusion

We proposed a novel method for removing model biases by explicitly protecting private author attributes as part of model training, which we formulate as deep learning with adversarial learning. We evaluate our methods with POS tagging and sentiment classification, demonstrating our method results in increased privacy, while also maintaining, or even improving, task performance, through increased model robustness.

Acknowledgements

We thank Benjamin Rubinstein and the anonymous reviewers for their helpful feedback and suggestions, and the National Computational Infrastructure Australia for computation resources. We also thank Dirk Hovy for providing the Trustpilot dataset. This work was supported by the Australian Research Council (FT130101105).

References

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318.

Ann Bies, Justin Mott, Colin Warner, and Seth Kulick. 2012. English Web Treebank. *Linguistic Data Consortium, Philadelphia, USA*.

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

Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263.

Cynthia Dwork. 2008. Differential privacy: A survey of results. In *International Conference on Theory and Applications of Models of Computation*, pages 1–19. Springer.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17:59:1–59:35.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, pages 2672–2680.

Briland Hitaj, Giuseppe Ateniese, and Fernando Pérez-Cruz. 2017. Deep models under the gan: information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC Conference*

on Computer and Communications Security, pages 603–618. ACM.

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

Dirk Hovy. 2015. Demographic factors improve classification performance. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 752–762.

Dirk Hovy, Anders Johannsen, and Anders Søgaard. 2015. User review sites as a resource for large-scale sociolinguistic studies. In *Proceedings of the 24th International Conference on World Wide Web*, pages 452–461.

Dirk Hovy and Anders Søgaard. 2015. Tagging performance correlates with author age. 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)*, volume 2, pages 483–488.

Rosie Jones, Ravi Kumar, Bo Pang, and Andrew Tomkins. 2007. I know what you did last summer: query logs and user privacy. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management (CIKM 2007)*, pages 909–914.

Anna Jørgensen, Dirk Hovy, and Anders Søgaard. 2016. Learning a POS tagger for AAVE-like language. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1115–1120.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751.

Marco Lui and Timothy Baldwin. 2012. langid.py: An off-the-shelf language identification tool. In *Proceedings of ACL 2012 System Demonstrations*, pages 25–30.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119.

Greg Pass, Abdur Chowdhury, and Cayley Torgeson. 2006. A picture of search. In *The First International Conference on Scalable Information Systems*, volume 152, page 1.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation*.

Martin Potthast, Francisco Rangel, Michael Tschuggnall, Efstathios Stamatatos, Paolo Rosso, and Benno Stein. 2017. Overview of PAN17: Author identification, author profiling, and author obfuscation. In *8th International Conference of the CLEF on Experimental IR Meets Multilinguality, Multimodality, and Visualization*.

Ella Rabinovich, Raj Nath Patel, Shachar Mirkin, Lucia Specia, and Shuly Wintner. 2017. Personalized machine translation: Preserving original author traits. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1074–1084.

Reza Shokri and Vitaly Shmatikov. 2015. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1310–1321.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2017. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2979–2989.

Chapter 4

Discussion and Conclusions

In this chapter, I compare and contrast the approaches proposed in Chapter 3 for robust modelling. These approaches share commonalities in terms of motivation, methodology, and evaluation, which I outline first. I then discuss the similarities between privacy and demographic bias. Next, I summarise the subsequent work to our publications in Chapter 3. I discuss future directions given the contributions of this thesis. Some are based on the limitations of the current work, while others are based on related work in other machine learning areas. I finish with an overall summary of the thesis.

4.1 Commonalities Between Different Forms of Robustness

I look into the commonalities in the different forms of robustness according to the three research questions: robustness to language variation, robustness to domain variation, and robustness to demographic bias related to fairness and privacy-preservation. In this section, I discuss the relationships between these aspects, and review the commonalities between proposed methods.

4.1.1 Motivation

The central goal of this thesis, robustness, was motivated from a broad range of perspectives. I first review these motivations, then discuss how they interrelate.

A general motivation for achieving robustness in machine learning is that models need to generalise well over both seen data distributions and unseen closely related distributions. This relates to the assumption that the observations are independent and identically distributed, which generally does not hold in real world applications, including most NLP applications. That is when an NLP model deals with unseen data instances, for example as associated with language variation, domain variation, or the demographic of the author, the model should be robust under all these scenarios.

All three factors are capable of changing or shifting the data distribution. Language variations (Sections 3.1 to 3.3), such as dialect and slang can cause shifts in the data distribution of text. Particularly for word-level language applications, out-of-vocabulary issues caused by lexical variations can also be viewed as one kind of distribution change. Syntactic variations also shift the distribution of the data, by changing grammar usage and sentence structures. These language phenomena are also types of distribution shifts from a machine learning point of view. From a different perspective, domain changes (Sections 3.4 and 3.5) also cause distribution changes, as domain is a machine learning concept which governs the data distribution as a latent variable (Pan and Yang, 2009). From a socio-linguistic perspective, demographic variables (Section 3.6) like author's gender and age also influence data distributions. For example, people of different ages have different habits of language usage, and thus their language follows different distributions (Hovy and Søgaard, 2015).

Now, let us look into relationships in each pairing of the three factors of robustness.

Some language variations can be viewed as a kind of domain change. For example, applying models trained over formal language to informal or spoken language can be considered as a form of domain transfer.

One of the motivations of this thesis is dealing with demographic variables, specifically gender, age and location of the author, to build robust and unbiased NLP models. One can view author demographic as a form of domain. For example, one can partition the dataset into different groups by author age, and thus different groups can be recognised as different domains. However, the concept of bias can be a broader concept of domain. For example,

each data instance can have multiple demographic variables, such as age and gender, but it can only belong to a domain or a sub-domain.

Multi-domain learning, a central problem in this thesis, relates to learning an unbiased and fair model. One motivation in multi-domain modelling is to learn a domain-invariant representation. One can see that in the context of fairness in machine learning, learning a representation that is unbiased towards demographic variables follows the same motivation of domain-invariant learning. In fact, the same technique, adversarial training, was applied to tackle this problem.

Considering bias from a socio-linguistic point of view, demographic bias in text data can be reflected as a language variation. As we mentioned, demographic variables also govern the data distribution. For example, people of different ages and education levels tend to use language with different vocabulary, grammar, and semantics, and these variations can be detected by machine learning approaches (Martinc et al., 2017) and then used to shortcut learning, failing to learn the core linguist process. Therefore, these biases are one cause of language variation. Learning robust models to reduce these biases can hopefully lead to text robustness.

4.1.2 Privacy and Demographic Bias

The third research question focused on the relationship between privacy and demographic variables.

Section 3.6 considered privacy under inference attacks on demographic variables. Inference attacks aim to predict demographic variables from the shared representations learned by the model, to expose the privacy of the users. The motivation for building a privacy-preserving model was that the representations should encode less information about these variables, thus reducing the effectiveness of such attack.

From the perspective of model fairness, the model aims to predict equally accurately over different demographic groups. However, machine learning methods tend to disadvantage minority groups and only fit the majority groups, as they are a larger component of the training objective. Sometimes, only fitting the majority achieves better aggregated performance over

the test data, but this could potentially lead to overfitting and a lack of generalisation (see Section 3.6). Therefore, the motivation here is to build models that are capable of learning the primary target variable and ignoring demographic information.

Finally, both perspectives relate to the same motivation of learning unbiased representations. Bias of demographic of authors is the private attributes under inference attacks. And, in this paper, model fairness is demonstrated to improve the privacy-preservation of the model, by learning representations that are less biased with respect to demographic variables.

4.1.3 Methodology

In terms of methodology, all three research questions are framed as machine learning problems, and the proposed methods all fall under the scope of machine learning algorithms. In detail, two categories of task are conducted, text classification, and sequence classification. From the machine learning point of view, these tasks can be formalised as optimisation problems by finding the best parameters to maximise the conditional probability $\theta^* = \arg\max_\theta p(y|\mathbf{x};\theta)$ given the observations $\mathbf{x}$. In terms of modelling approaches, deep learning models, to be specific CNNs and RNNs, were chosen to tackle this optimisation problem, and solved by stochastic gradient descent or close variants. However, the different robustness emphasise different motivations, and thus different machine learning approaches have been applied to tackle different questions. Specific to this thesis, regularisation methods and data augmentation methods are used to improve robustness to language variation, and adversarial learning and variational inference are adopted to achieve robustness for multi-domain learning.

Adversarial learning is pervasive in this thesis. First, language variations are generated to corrupt models, such as lexical noise (Section 3.1) and adversarial generation (Section 3.3), and then models are trained by augmenting with the generated variations to improve robustness (Section 3.2). That is, difficult language variations and robustness to language variation are adversaries of one another. The idea of adversarial training is applied to multi-domain learning (Section 3.4), and further extended to tackle demographic bias and privacy-preservation in Section 3.6.

4.1.4 Evaluation Setup

To demonstrate the robustness of the proposed approaches, a similar experimental setup was considered throughout the thesis. A robust model must generalise to in-domain instances, but also to out-of-domain data. Accordingly, both in-domain and out-of-domain evaluations are conducted for the different models and datasets. That is the second research question of robustness to domain shifts, which is explored throughout the thesis.

The out-of-domain setting we consider is where no knowledge of the target data is assumed. Following this principle, all the proposed approaches and baseline models are evaluated in in-domain and out-of-domain settings. Even for fairness, out-of-domain evaluation was used to demonstrate that the unbiased model improves robustness to text corpora from unseen demographics.

In terms of datasets, I mostly focused on text classification with the one exception of using a sequence prediction POS-tagging task. The movie review and product review datasets used in Sections 3.1 to 3.3 were extended to multi-domain versions for multi-domain learning (Sections 3.4 and 3.5). The Trustpilot dataset was used for both fairness and privacy evaluation (Section 3.6), while the fairness evaluation was tested over the tagging task and the privacy evaluation was conducted in the context of a classification task.

4.2 Subsequent Work

In this section, I discuss subsequent work that has appears in the literature, building upon or related to each paper as presented in Chapter 3.

4.2.1 Linguistic Variation and Adversarial Generation

Following the idea of applying random dropout noise to evaluate the robustness (Section 3.1), Heigold et al. (2018) investigated the model robustness against lexical variation, such as scrambled words and random noise, in morphological tagging and machine translation. They empirically evaluated the robustness of different models on character-based word

embeddings and they showed that state-of-the-art NLP systems are sensitive to slightly perturbed words. Based on the proposed robust regularisation in Section 3.1, Newman-Griffis and Fosler-Lussier (2017) also considered learning second-order vector representations of words, induced from nearest neighborhood topological features, and they analysed the effects of using second order representations is able to handle highly heterogeneous data better than first-order representations.

Our ideas of augmentation approaches (Section 3.2) have been extended to a range of NLP tasks, such as conventional AI (Du and Black, 2018), sentiment classification (Chen and Ji, 2019), and neural machine translation (Anastasopoulos et al., 2019). The adversarial generation (Section 3.3) is also adopted to other language tasks, such as relation extraction (Wu et al., 2017), vision question answering (Shah et al., 2019), and reading comprehension (Jia and Liang, 2017).

4.2.2 Multi-domain Learning

Our approaches for multi-domain learning have also been extended. For example, Jauhiainen et al. (2019) studied a task of language and dialect identification via language model adaptation. They applied a close idea to Section 3.4, with a but different approach of applying a language model methods to do language identification. Inspired by our ideas of selecting domain channels (Section 3.5), Dai et al. (2019a) considered to select pretraining data that is close to target task data for Named Entity Recognition (NER) task.

4.2.3 Fairness and Privacy-preservation

Our last contribution focuses on fairness and privacy-preservation (Section 3.6). I have reviewed some key literature in Section 2.4, and here I look at papers inspired by our ideas of adversarial training.

In terms of fairness, Schumann et al. (2019) studied the machine learning fairness in the scenarios of transfer learning. Kaneko and Bollegala (2019) proposed gender-fair learning for pre-trained word embeddings.

Subsequent to our work of privacy-preservation, Elazar and Goldberg (2018) also proposed a similar method of adversarial learning to remove the demographic attributes from text data. Different from our paper, they claimed that training another classifier to do external verification for discrimination accuracies is crucial. Barrett et al. (2019) revisited Elazar and Goldberg (2018)'s method. And they constructed a new dataset, combining data from multiple domains, and they showed that Elazar and Goldberg (2018)'s method does not generalize well to out-of-domain instances. More broadly, de-identification approaches using our adversarial idea have been applied on different language tasks, such as syntactic parsing (Hu et al., 2019) and legal text classification (Garat and Wonsever, 2019), as well as to many NLP models, such as the transformer (Lang et al., 2019).

4.3 Limitations and Future Work

In this section, I address the limitations of our work and potential extensions. Some of my earlier work was based on assumptions which were reasonable at the time of writing. I re-evaluate the work and outline how these limitations could be addressed.

4.3.1 Character-level and Subword-level Methods

The first limitation is that I only considered word-level approaches and followed the assumption that lexical variation causes an out-of-vocabulary problem for word-level models. With recent advantages of deep learning, the out-of-vocabulary problem can be mitigated to some degree using character-level and subword-level methods (Devlin et al., 2019; Kim et al., 2016a). Therefore, one future direction is to evaluate the robustness of these models. Accordingly, our proposed augmentation methods can still be applied.

Recently, pre-trained methods (see Section 2.1.2) have shown great success for NLP applications, involving better model architectures like the transformer (Vaswani et al., 2017). BERT embedding (Devlin et al., 2019), which is an application of transformers to language modelling and sentence representation, also works on subword level learning. Studying the robustness to character-edits, such as typos, punctuation changes, and changes to capitalisa-

tion, of BERT embeddings and applying transformer architecture for text learning should be considered in future work. This can help better understand the transformers and build more robust NLP applications in the future.

4.3.2 Other Language Tasks

A second limitation is the experimental evaluation was largely using classification. To be specific, two document classification tasks – sentiment analysis and language identification – was used, and in one case, sequence classification POS tagging.

In terms of future work, most of the proposed approaches in this thesis should be general. For example, our approaches can be applied to syntactic and semantic tasks, such as structured learning, machine translation, and natural language understanding. These tasks can be conducted in two different group, depending on the difficulty of these new tasks.

The first group is to directly apply the proposed approaches to other text classification tasks and sequence tagging tasks, and Section 4.2 has mentioned some work towards these tasks. For example, the data augmentation method proposed in Section 3.2 can be instantly applied to other tasks when annotated data is insufficient, to improve the generalisation of the model. Also, the multi-domain learning methods can be directly applied to multi-domain POS tagging, multi-domain NER and multi-domain parsing, where the data is heterogeneous. This can be modelled using a structured model, and the multi-domain approaches (Sections 3.4 and 3.5) can easily be transferred to sequence learning architectures. These approaches can be expected to help model automatically learn better domain knowledge.

However, for the other group, they might need tailering of our methods to suit new tasks. For instance, language understanding tasks could require more complex models, for example, understanding tree structures and the underlying logic of language, which cannot be resolved by a simple model. Therefore, for these new tasks, a better understanding of the task and design of new methods is required, such as using transformers to model language understanding tasks (Vaswani et al., 2017).

4.3.3 Multi-lingual Scenarios

The third limitation is that, in this thesis, the experiments are limited to English, with the exception of language identification, which considers 97 languages. Generally, the proposed approaches should generalise to other languages, as long as there is enough data. Therefore, one direction for future work is to test the robustness of the proposed methods using datasets in languages other than English. For example, the augmentation approach to language variation (Section 3.2) should also work pretty well to French sentiment analysis, as French and English are close to obtain good language resources for generation. When applying with other languages, such as Chinese sentiment analysis, I believe the proposed augmentation approach should still work relatively well, as long as the generated instances have good quality. Method and baselines can also be applied to the new languages.

Next, I consider extending the mono-lingual setup to cross-lingual scenarios. All of the proposed approaches are evaluated with out-of-domain data. For future work, cross-lingual variations could be considered as an additional form of out-of-domain evaluation by viewing languages as "domains". In other words, the out-of-domain evaluation could be extended to cross-lingual evaluation using a transfer learning setup, where the proposed domain approaches would require some form of bridge, such as cross-lingual embeddings (CONNEAU and Lample, 2019; Upadhyay et al., 2016). In fact, I have some preliminary results applying the adversarial training approaches to cross-lingual NER, where I train an NER system over one language and evaluated over a different language in an unsupervised fashion. The adversarial idea was applied to discriminate and align the source and target language representations. The model performance improves compared with a simple LSTM baseline, but for some language pairs, it does not work as well as state-of-the-art transfer methods. I suspect part of this is due to the cross-lingual embeddings from Luong et al. (2015) not being good enough for some language pairs, and better cross-lingual embeddings based on BERT (CONNEAU and Lample, 2019) should improve the performance.

Besides the cross-language adaptation tasks, another future direction is to adopt multi-domain adaptation to multi-lingual adaptation. This setup considers the case of training over a corpus made up of multiple languages, and aiming to transfer knowledge from multiple

candidates to the target languages. Some tasks like multi-lingual POS tagging and multi-lingual NER can be considered. One idea is to extend the proposed multi-domain learning approach to the setup of Rahimi et al. (2019), in multi-lingual NER. The multi-domain approach can automatically learn attention weights over the source languages, according to the given target language. This can help the model estimate the true distribution of the target, which could better transfer linguistic knowledge. Based on this idea, similar tasks like multi-lingual POS tagging (Lin et al., 2018; Plank et al., 2016) and multi-lingual parsing (McDonald et al., 2013; Nivre et al., 2016) could also be considered.

Additionally, another future research opportunity is to link the multi-domain and multi-lingual problems. Imagine a case of a corpus in multiple languages, with each language focusing on different domains. For example, the English reviews on *cars* and *ships*, Chinese reviews on *books* and *dvds*, and Spanish reviews on *movies*. Is it possible to learn a model that can transfer the knowledge from Chinese *dvds* to Spanish *movies*? To tackle this problem, based on the variational approach in Section 3.5, one idea would be introducing another latent variable l to model language and learn a joint model $p(y, z, l|\mathbf{x})$.

Above all, I consider to extend our approaches to the multi-lingual scenarios with different setups, and come up with potential solutions. This will be a future direction that I will focus on.

4.3.4 Better Adversarial Training Techniques

Another limitation relates to adversarial training. In the thesis, I applied Ganin et al. (2016)'s architecture and derived the training algorithm from Goodfellow et al. (2014). However, lots of research has shown that adversarial nets are hard to train and propose robust training algorithms (Arjovsky et al., 2017; Berthelot et al., 2017; Chrysos et al., 2019; Gao et al., 2019). During my experiments, I also found that the model training could be unstable, as the generator and discriminator converge very slowly. Therefore, a better training algorithm or better loss than cross-entropy could be applied for more robust training.

One particular question here relates to Equation 2.25 in Section 2.3.1.2, the min-max objective:

$$\mathscr{L}^{(D)} = \min_{\theta^{(G)}} \max_{\theta^{(D)}} \mathscr{H}\left(\mathbf{y}|\,\mathrm{G}(\mathbf{x})\right) - \lambda \mathscr{H}\left(d|\,\mathrm{G}(\mathbf{x})\right) . \tag{4.1}$$

where the first term $\mathscr{H}\left(\mathbf{y}|\,\mathrm{G}(\mathbf{x})\right)$ is the cross-entropy loss of the task and the second term $\mathscr{H}\left(d|\,\mathrm{G}(\mathbf{x})\right)$ is the cross-entropy loss of the domain, measured by the discriminator D, leveraged by hyper-parameter λ. In this function, the discriminator tries to make the model un-learn the domain label. However, manually setting λ to a large value will make it dominate the objective, making G completely uninformative, while setting it to a small value will make the discriminator useless. To improve the stability of learning, other distance measurements for the discriminator could be introduced, such as Wasserstein loss (Arjovsky et al., 2017).

There are also several variations of adversarial nets from the computer vision, which could have potential utility for our applications. For examples, the Conditional GAN (Mirza and Osindero, 2014) can make generation conditional on given variables, such as domain or demographic variable in our case. In details, both generator G and discriminator D are conditioned on some auxiliary information $\mathbf{y}$, and the objective is:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}\left[\log D(\mathbf{x}|\mathbf{y})\right] + \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}\left[\log(1 - D(G(\mathbf{z}|\mathbf{y})))\right] , \tag{4.2}$$

This could be useful for personalised generation tasks (Michel and Neubig, 2018; Rabinovich et al., 2017), which provide a different perspective on how to deal with and model demographic variables. Also, CycleGAN (Zhu et al., 2017) can learn style transfer using unpaired unsupervised training data, where the style could also be generalised to the task of domain adaptation. In our case, the idea of cycleGAN can be useful in automatically aligning representations of different domains in an unsupervised scenarios (Chen et al., 2019).

Until recently, adversarial learning has had limited use in language applications. This may be due to the discreteness and the sparsity of text, which gives no gradients during back-propagation. This can be tackled by reinforcement learning (Sutton et al., 1999), but

problems still remain, such as training efficiency and generation quantity. Solving these problems is an important open question in NLP.

4.3.5 Formal Privacy

A limitation concerns privacy. In this thesis, I only measure privacy empirically, in the form of the effectiveness of an inference attack, limiting the scope of this work. Formal privacy, such as Differential Privacy (DP) (Dwork, 2008), can provide a privacy guarantee, as discussed in Section 2.4.2.1. Therefore, in future work, DP methods can be considered to introduce model robustness and help ensure machine learning privacy.

However, there are certain issues in applying DP to NLP models. One disadvantage of applying DP methods is that there is a trade-off between the privacy budget and model performance. Simply using DP methods, such as applying Laplace or dropout noise (Holohan et al., 2018; Kairouz et al., 2014), during inference will diminish model accuracy, as illustrated in Section 3.1. Therefore, combining formal privacy and robustness approaches for inference is one immediate area of future work. I have preliminary results to tackle the trade-off during inference. Taking inspiration from our approaches to lexical noise (Section 3.1) and adversarial defence to fast gradient attacks (Goodfellow et al., 2015), I can train a model to be more stable to DP noise. Generally, the algorithm reinforces the learned hidden representation $\mathbf{h}$ with fast gradient noise to reduce the influence of DP noise. In detail, assuming a hidden presentation $\mathbf{h}$ requires protection, Laplacian noise is applied before being shared: $\mathbf{h}' = \mathbf{h} + \text{Lap}(\epsilon)$. I applied the noise training using the fast gradient noise $\widehat{\mathbf{g}}$ (Goodfellow et al., 2015) and retrain the output model with noise-augmented representation $\mathbf{h}^* = \mathbf{h}' + \widehat{\mathbf{g}}$. Preliminary experiments show positive results, that this noise training approach can improve the robustness in inference, however considering the time limitations of the thesis, they are not presented here.

Nowadays, machine learning methods, such as inference attacks, are considered to attack formal privacy mechanisms, where these conventional privacy-preserving methods requires high privacy budget. Therefore, another interesting and related question is to investigate the relationship between formal privacy and inference attacks in NLP. For example, how much

information can inference attacks extract from DP-protected data instances. This has not been well studied, and I believe this question is a promising research direction for privacy and robustness of NLP models.

4.3.6 Adversarial Attacks and Defences

Another limitation relates to adversarial attacks. In the section on robustness to language variation (Section 3.3), I proposed a method to generate adversarial text examples. These adversarial examples can be considered as an attack, as known as adversarial attacks, which expose the brittleness of machine learning models (Yuan et al., 2019). Note that this is a different attack from inference attacks above. However, in this thesis, I did not take adversarial evaluation into consideration.

There are some reasons for not utilising adversarial evaluation in this thesis. First is the overall quality of the adversarial opponents – attackers. In the research community, many automatic attack methods for generating adversarial text have been studied (Alzantot et al., 2018; Subramanian et al., 2017). However, the overall quality of the adversarial text is not good. As mentioned in Section 4.3.4, adversarial learning has not shown superior effectiveness in language applications, as generating adversarial text examples is much more difficult than it is for images. Second, human experts can generate high-quality adversarial text examples but this is time-consuming. In Section 3.3, I also experimented with augmentation via adversarial text generated by linguistics, however, it is hard to generalise to other tasks and datasets. Due to these reasons, I did not consider adversarial evaluation as a criterion of robustness.

Adversarial attacks and defences are important questions for NLP to study the robustness of machine learning models, as a robust machine learning model should be able to defend against adversarial text. Therefore, this should be considered in the future work.

4.3.7 Relationship with Disentangled Representation Learning

In Sections 3.4 and 3.5, I introduced multi-channel methods where individual channels could capture information for each domain. This is related with the work of disentanglement representation from computer vision (Chen et al., 2018; Higgins et al., 2018), where each neuron of the learned representation ideally captures an independent feature, such as the color, shape, or texture of an object. By learning disentangled representations, the model is more interpretable and more transparent for decision making, which can also improve model fairness (Creager et al., 2019; Locatello et al., 2019).

In fact, the proposed domain-conditional and domain-generative methods (Section 3.4) are an efficient way of learning disentangled representations for each domain. With adversarial learning, domain-specific representations are captured by each domain channel, where each domain representation is independent with respect to domain. Approaches proposed in Section 3.4 take the form of supervised disentangled representation learning, which requires annotation of the domain. It is also possible to perform unsupervised disentangled representation learning (Chen et al., 2016; Hsu et al., 2017). In Section 3.5, the setup was close to unsupervised learning, where the domain label is not provided. To make learning tractable, I proposed to use variational inference which is close to disentanglement learning methods such as Categorical VAE (CatVAE) (Pineau and Lelarge, 2018), which uses multimodal distributions for the prior and the inference network, and maximises the mutual information between the categories. The motivation is to represent each major categories of the data within independent subspace of the latent space.

Disentangled representation learning is one future direction to look at, borrowing ideas from disentangled representation (Higgins et al., 2018) to analyse the proposed approaches (Section 3.5). Further, approaches can be borrowed for learning domain-disentangled representation in multi-domain learning, which can help minimise the number of channels, effectively reducing the model complexity. Most NLP datasets are heterogeneous and this would benefit the robustness for a range of language applications.

4.4 Conclusion

In this thesis, I aimed to learn robust representations for text applications. By reviewing the literature, robustness of NLP applications has been considered as an essential problem when dealing with real-world corpora. And with machine learning algorithms involving more in our world, fairness and privacy-preservation are needed. Therefore, I considered robustness with respect to three different aspects, namely robustness to language variation, robustness to domain variation, and robustness to demographic of authors in terms of model fairness and privacy-preservation.

To tackle each research question, I broke them up into sub-questions and proposed a range of approaches. For robustness to language variation, I first studied how lexical variation influence model robustness, and then proposed a regularisation method to deal with it. Moreover, I proposed a data augmentation approach by generating language variation from both linguistic and machine learning perspectives to improve both model generalisation and robustness. For domain variation, I focused on multi-domain learning and investigated the semi-supervised learning with respect to domain label. Two state-of-the-art approaches, adversarial training and variational inference, were used to learn more robust multi-domain representations. For robustness to demographic variables, both model fairness and privacy perspectives were considered, tackled via adversarial learning to reduce bias in representation learning.

For evaluation, a range of tasks and datasets were considered, with different configurations for different tasks. Throughout this thesis, in-domain and out-of-domain evaluations were used to demonstrate the generalisation and robustness of the proposed approaches. Domain supervised and semi-supervised learning configurations were simulated to exemplify the proposed multi-domain approaches.

Based on empirical experiments, the proposed approaches were found to outperform a range of baselines, demonstrating the advantages and robustness of our models. I hope these results and discussions contribute towards and inspire future work in this space.

References

Martin Abadi, Andy Chu, Ian Goodfellow, Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318.

Rakesh Agrawal and Ramakrishnan Srikant. 2000. Privacy-preserving data mining. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pages 439–450.

Laura Aina, Kristina Gulordava, and Gemma Boleda. 2019. Putting words in context: LSTM language models and lexical ambiguity. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3342–3348.

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896.

Antonios Anastasopoulos, Alison Lui, Toan Nguyen, and David Chiang. 2019. Neural machine translation of text from non-native speakers. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long and Short Papers)*, pages 3070–3080.

Antreas Antoniou, Amos Storkey, and Harrison Edwards. 2017. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 214–223.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*.

Timothy Baldwin, Marie Catherine de Marneffe, Bo Han, Young-Bum Kim, Alan Ritter, and Wei Xu. 2015. Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 126–135.

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 597–604.

Solon Barocas, Moritz Hardt, and Arvind Narayanan. 2019. *Fairness and Machine Learning*. fairmlbook.org.

Maria Barrett, Yova Kementchedjhieva, Yanai Elazar, Desmond Elliott, and Anders Søgaard. 2019. Adversarial removal of demographic attributes revisited. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 6331–6336.

Jerome R Bellegarda. 2004. Statistical language model adaptation: review and perspectives. *Speech Communication*, 42(1):93–108.

Emily M. Bender, Dan Flickinger, and Stephan Oepen. 2002. The grammar matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In *Proceedings of the 2002 Workshop on Grammar Engineering and Evaluation*, pages 1–7.

Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1415–1425.

David Berthelot, Thomas Schumm, and Luke Metz. 2017. BEGAN: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. 2017. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, Bollywood, boomboxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447.

Danushka Bollegala, David Weir, and John Carroll. 2012. Cross-domain sentiment classification using a sentiment sensitive thesaurus. *IEEE transactions on knowledge and data engineering*, 25(8):1719–1731.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. In *Advances in Neural Information Processing Systems 29*, pages 4349–4357.

Francis Bond, Eric Nichols, Darren Scott Appling, and Michael Paul. 2008. Improving statistical machine translation by paraphrasing the training data. In *Proceedings of the 2008 International Workshop on Spoken Language Translation*.

Shikha Bordia and Samuel Bowman. 2019. Identifying and reducing gender bias in word-level language models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 7–15.

Léon Bottou. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of 19th International Conference on Computational Statistics*, pages 177–186.

Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21.

Rich Caruana, Steve Lawrence, and Lee Giles. 2001. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *Advances in Neural Information Processing Systems 13*, pages 402–408.

Nicolo Cesa-Bianchi, Claudio Gentile, and Luca Zaniboni. 2006. Worst-case analysis of selective sampling for linear classification. *Journal of Machine Learning Research*, 7(Jul):1205–1230.

Mahawaga Arachchige Pathum Chamikara, Peter Bertok, Ibrahim Khalil, Dongxi Liu, Seyit Camtepe, and Mohammed Atiquzzaman. 2019. Local differential privacy for deep learning. *IEEE Internet of Things Journal*.

David Chen and William Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 190–200.

Hanjie Chen and Yangfeng Ji. 2019. Improving the interpretability of neural sentiment classifiers via data augmentation. *arXiv preprint arXiv:1909.04225*.

Jiawei Chen, Janusz Konrad, and Prakash Ishwar. 2019. A cyclically-trained adversarial network for invariant representation learning. *arXiv preprint arXiv:1906.09313*.

Tian Qi Chen, Xuechen Li, Roger Grosse, and David Duvenaud. 2018. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems 31*, pages 2610–2620.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems 29*, pages 2172–2180.

Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 551–561.

Travers Ching, Daniel Himmelstein, Brett Beaulieu-Jones, Alexandr Kalinin, Brian Do, Gregory Way, Enrico Ferrero, Paul-Michael Agapow, Michael Zietz, Michael Hoffman, Wei Xie, Gail Rosen, Benjamin Lengerich, Johnny Israeli, Jack Lanchantin, Stephen Woloszynek, Anne Carpenter, Avanti Shrikumar, Jinbo Xu, Evan Cofer, Christopher Lavender, Srinivas Turaga, Amr Alexandari, Zhiyong Lu, David Harris, Dave DeCaprio, Yanjun Qi, Anshul Kundaje, Yifan Peng, Laura Wiley, Marwin Segler, Simina Boca, Joshua Swamidass, Austin Huang, Anthony Gitter, and Casey Greene. 2018. Opportunities and obstacles for deep learning in biology and medicine. *Journal of The Royal Society Interface*, 15(141):20170387.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of the 8th Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111.

Alexandra Chouldechova. 2017. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big Data*, 5(2):153–163.

Alexandra Chouldechova and Fernando Diaz, editors. 2019. *Proceedings of the Conference on Fairness, Accountability, and Transparency*. ACM.

Grigorios Chrysos, Jean Kossaifi, and Stefanos Zafeiriou. 2019. Roc-GAN: Robust conditional GAN. In *International Conference on Learning Representations*.

Chenhui Chu and Rui Wang. 2018. A survey of domain adaptation for neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1304–1319.

James Clarke and Mirella Lapata. 2008. Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research*, 31:399–429.

Maximin Coavoux, Shashi Narayan, and Shay Cohen. 2018. Privacy-preserving neural representations of text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1–10.

Trevor Cohn and Mirella Lapata. 2008. Sentence compression beyond word deletion. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 137–144.

Trevor Cohn and Mirella Lapata. 2009. Sentence compression as tree transduction. *Journal of Artificial Intelligence Research*, 34:637–674.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 160–167.

Alexis CONNEAU and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems 32*, pages 7057–7067.

Ann Copestake and Dan Flickinger. 2000. An open source grammar development environment and broad-coverage English grammar using HPSG. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation*.

Elliot Creager, David Madras, Jörn-Henrik Jacobsen, Marissa A Weis, Kevin Swersky, Toniann Pitassi, and Richard Zemel. 2019. Flexibly fair representation learning by disentanglement. In *Proceedings of the 36th International Conference on Machine Learning*, pages 1436–1445.

David Alan Cruse. 1986. *Lexical semantics*. Cambridge University Press.

Ido Dagan, Oren Glickman, Alfio Gliozzo, Efrat Marmorshtein, and Carlo Strapparava. 2006. Direct word sense matching for lexical substitution. In *Proceedings of the 21st*

International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, pages 449–456.

Xiang Dai, Sarvnaz Karimi, Ben Hachey, and Cecile Paris. 2019a. Using similarity measures to select pretraining data for NER. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long and Short Papers)*, pages 1460–1470.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019b. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988.

Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263.

Yann Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 933–941.

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

Arthur Dempster, Nan Laird, and Donald Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society (Series B: Methodological)*, 39(1):1–22.

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

Julia Dressel and Hany Farid. 2018. The accuracy, fairness, and limits of predicting recidivism. *Science Advances*, 4(1).

Wenchao Du and Alan Black. 2018. Data augmentation for neural online chats response selection. In *Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI*, pages 52–58.

Cynthia Dwork. 2008. Differential privacy: A survey of results. In *International Conference on Theory and Applications of Models of Computation*, pages 1–19.

Cynthia Dwork. 2011. Differential privacy. *Encyclopedia of Cryptography and Security*, pages 338–340.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36.

Yanai Elazar and Yoav Goldberg. 2018. Adversarial removal of demographic attributes from text data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 11–21.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 567–573.

Natasha Fernandes, Mark Dras, and Annabelle McIver. 2019. Generalised differential privacy for text document processing. In *Principles of Security and Trust*, pages 123–148.

Katja Filippova, Enrique Alfonseca, Carlos Colmenares, Lukasz Kaiser, and Oriol Vinyals. 2015. Sentence compression by deletion with LSTMs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 360–368.

Anthony Flores, Kristin Bechtel, and Christopher Lowenkamp. 2016. False positives, false negatives, and false analyses: A rejoinder to "Machine bias: There's software used across

the country to predict future criminals. And it's biased against blacks.". *Federal Probation*, 80:38–46.

Nissim Francez. 1986. *Fairness*. Springer-Verlag.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in Neural Information Processing Systems 29*, pages 1019–1027.

Sébastien Gambs, Ahmed Gmati, and Michel Hurfin. 2012. Reconstruction attack through classifier analysis. In *Annual Conference on Data and Applications Security and Privacy*, pages 274–281.

Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by back-propagation. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pages 1180–1189.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17:59:1–59:35.

Chao Gao, jiyi Liu, Yuan Yao, and Weizhi Zhu. 2019. Robust estimation via generative adversarial networks. In *International Conference on Learning Representations*.

Diego Garat and Dina Wonsever. 2019. Towards de-identification of legal texts. *arXiv preprint arXiv:1910.03739*.

Andrea Gesmundo and Tanja Samardžić. 2012. Lemmatisation as a tagging task. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 368–372.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning*, pages 513–520.

Hila Gonen and Yoav Goldberg. 2019. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long and Short Papers)*, pages 609–614.

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

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, pages 2672–2680.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *Proceedings of the 3rd International Conference on Learning Representations*.

Robert Granville. 1984. Controlling lexical substitution in computer text generation. In *Proceedings of the 10th International Conference on Computational Linguistics and 22nd Annual Meeting on Association for Computational Linguistics*, pages 381–384.

Ankush Gupta, Arvind Agarwal, Prawaan Singh, and Piyush Rai. 2018. A deep generative framework for paraphrase generation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5149–5156.

Jihun Hamm. 2017. Minimax filter: Learning to preserve privacy from inference attacks. *The Journal of Machine Learning Research*, 18(1):4704–4734.

Bo Han and Timothy Baldwin. 2011. Lexical normalisation of short text messages: Makn sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 368–378.

Moritz Hardt, Eric Price, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems 29*, pages 3315–3323.

Zellig Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

Martin Haspelmath. 2009. Lexical borrowing: Concepts and issues. In *Loanwords In the World's Languages: A Comparative Handbook*, pages 35–54. De Gruyter Martin Berlin.

Samer Hassan, Andras Csomai, Carmen Banea, Ravi Sinha, and Rada Mihalcea. 2007. UNT: SubFinder: Combining knowledge sources for automatic lexical substitution. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 410–413.

Wei He, Shiqi Zhao, Haifeng Wang, and Ting Liu. 2011. Enriching SMT training data via paraphrasing. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 803–810.

Georg Heigold, Stalin Varanasi, Günter Neumann, and Josef van Genabith. 2018. How robust are character-based word embeddings in tagging and MT against wrod scramlbing or randdm nouse? In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pages 68–80.

Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. 2018. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*.

Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. 2017. Deep models under the GAN: Information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 603–618.

Sepp Hochreiter. 1998. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Naoise Holohan, Spiros Antonatos, Stefano Braghin, and Pól Mac Aonghusa. 2018. The bounded laplace mechanism in differential privacy. *arXiv preprint arXiv:1808.10410*.

Dirk Hovy. 2015. Demographic factors improve classification performance. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 752–762.

Dirk Hovy, Anders Johannsen, and Anders Søgaard. 2015. User review sites as a resource for large-scale sociolinguistic studies. In *Proceedings of the 24th International Conference on World Wide Web*, pages 452–461.

Dirk Hovy and Anders Søgaard. 2015. Tagging performance correlates with author age. 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)*, volume 2, pages 483–488.

Jeremy Howard and Sebastian Ruder. 2018. 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.

Wei-Ning Hsu, Yu Zhang, and James Glass. 2017. Unsupervised learning of disentangled and interpretable representations from sequential data. In *Advances in Neural Information Processing Systems 30*, pages 1878–1889.

Zhifeng Hu, Serhii Havrylov, Ivan Titov, and Shay B Cohen. 2019. Obfuscation for privacy-preserving syntactic parsing. *arXiv preprint arXiv:1904.09585*.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Tommi Jauhiainen, Krister Lindén, and Heidi Jauhiainen. 2019. Language model adaptation for language and dialect identification of text. *Natural Language Engineering*, 25(5):561–583.

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

Thorsten Joachims. 1999. Transductive inference for text classification using support vector machines. In *Proceedings of 16th International Conference on Machine Learning*, pages 200–209.

Bart Jongejan and Hercules Dalianis. 2009. Automatic training of lemmatization rules that handle morphological changes in pre-, in- and suffixes alike. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 145–153.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*.

Peter Kairouz, Sewoong Oh, and Pramod Viswanath. 2014. Extremal mechanisms for local differential privacy. In *Advances in Neural Information Processing Systems 27*, pages 2879–2887.

Masahiro Kaneko and Danushka Bollegala. 2019. Gender-preserving debiasing for pre-trained word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1641–1650.

Emiko Kashima and Yoshihisa Kashima. 1998. Culture and language: The case of cultural dimensionsand personal pronoun use. *Journal of Cross-Cultural Psychology*, 29(3):461–486.

David Kauchak and Regina Barzilay. 2006. Paraphrasing for automatic evaluation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 455–462.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander Rush. 2016a. Character-aware neural language models. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*.

Young-Bum Kim, Karl Stratos, and Ruhi Sarikaya. 2016b. Frustratingly easy neural domain adaptation. In *Proceedings of the 26th International Conference on Computational Linguistics*, pages 387–396.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*.

Diederik Kingma and Max Welling. 2014. Auto-encoding variational Bayes. In *Proceedings of the International Conference on Learning Representations*.

Kevin Knight and Daniel Marcu. 2000. Statistics-based summarization - step one: Sentence compression. In *Proceedings of the 17th National Conference on Artificial Intelligence and 12th Conference on Innovative Applications of Artificial Intelligence*, pages 703–710.

Kevin Knight and Daniel Marcu. 2002. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artificial Intelligence*, 139(1):91–107.

Wessel Kraaij and Renée Pohlmann. 1994. Porter's stemming algorithm for Dutch. *Informatiewetenschap*, pages 167–180.

Gerhard Kremer, Katrin Erk, Sebastian Padó, and Stefan Thater. 2014. What substitutes tell us - analysis of an "all-words" lexical substitution corpus. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 540–549.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105.

John Krumm. 2007. Inference attacks on location tracks. In *International Conference on Pervasive Computing*, pages 127–143.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 2267–2273.

Jiaqi Lang, Linjing Li, Weiyun Chen, and Daniel Zeng. 2019. Privacy protection in transformer-based neural network. In *2019 IEEE International Conference on Intelligence and Security Informatics*, pages 182–184.

Samuel Leeman-Munk, James Lester, and James Cox. 2015. NCSU_SAS_SAM: Deep encoding and reconstruction for normalization of noisy text. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 154–161.

Zheng Li, Yun Zhang, Ying Wei, Yuxiang Wu, and Qiang Yang. 2017. End-to-end adversarial memory network for cross-domain sentiment classification. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 2237–2243.

Zichao Li, Xin Jiang, Lifeng Shang, and Hang Li. 2018. Paraphrase generation with deep reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3865–3878.

Zichao Li, Xin Jiang, Lifeng Shang, and Qun Liu. 2019. Decomposable neural paraphrase generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3403–3414.

Xi Victoria Lin, Richard Socher, and Caiming Xiong. 2018. Multi-hop knowledge graph reasoning with reward shaping. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3243–3253.

Wang Ling, Guang Xiang, Chris Dyer, Alan Black, and Isabel Trancoso. 2013. Microblogs as parallel corpora. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 176–186.

Cheng-Yuan Liou, Wei-Chen Cheng, Jiun-Wei Liou, and Daw-Ran Liou. 2014. Autoencoder for words. *Neurocomputing*, 139:84–96.

Quan Liu, Hui Jiang, Si Wei, Zhen-Hua Ling, and Yu Hu. 2015a. Learning semantic word embeddings based on ordinal knowledge constraints. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1501–1511.

Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye-Yi Wang. 2015b. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 912–921.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Francesco Locatello, Gabriele Abbati, Tom Rainforth, Stefan Bauer, Bernhard Schölkopf, and Olivier Bachem. 2019. On the fairness of disentangled representations. In *Advances in Neural Information Processing Systems 32*, pages 14584–14597.

Minh-Thang Luong, Hieu Pham, and Christopher Manning. 2015. Bilingual word representations with monolingual quality in mind. In *NAACL Workshop on Vector Space Modeling for NLP*.

Nitin Madnani and Bonnie Dorr. 2010. Generating phrasal and sentential paraphrases: A survey of data-driven methods. *Computational Linguistics*, 36(3):341–387.

Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. 2017. Paraphrasing revisited with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 881–893.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. 2009. Domain adaptation with multiple sources. In *Advances in Neural Information Processing Systems 21*, pages 1041–1048.

Matej Martinc, Iza Skrjanec, Katja Zupan, and Senja Pollak. 2017. PAN 2017: Author profiling - gender and language variety prediction. In *Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum*.

Diana McCarthy and Roberto Navigli. 2007. SemEval-2007 task 10: English lexical substitution task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations*, pages 48–53.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–97.

Oren Melamud, Omer Levy, and Ido Dagan. 2015. A simple word embedding model for lexical substitution. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 1–7.

Paul Michel and Graham Neubig. 2018. Extreme adaptation for personalized neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 312–318.

Rada Mihalcea, Ravi Sinha, and Diana McCarthy. 2010. SemEval-2010 task 2: Cross-lingual lexical substitution. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 9–14.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119.

George Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Ahmadreza Mosallanezhad, Ghazaleh Beigi, and Huan Liu. 2019. Deep reinforcement learning-based text anonymization against private-attribute inference. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 2360–2369.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 142–148.

Anoop Namboodiri and Anil Jain. 2007. Document structure and layout analysis. In *Digital Document Processing*, pages 29–48.

Denis Newman-Griffis and Eric Fosler-Lussier. 2017. Second-order word embeddings from nearest neighbor topological features. *arXiv preprint arXiv:1705.08488*.

Andrew Ng. 2004. Feature selection, L1 vs. L2 regularization, and rotational invariance. In *Proceedings of the 21st International Conference on Machine Learning*, pages 78–85.

Eric Nichols, Francis Bond, Scott Appling, and Yuji Matsumoto. 2010. Paraphrasing training data for statistical machine translation. *Information and Media Technologies*, 5(2):950–971.

Malvina Nissim, Rik van Noord, and Rob van der Goot. 2019. Fair is better than sensational: Man is to doctor as woman is to doctor. *arXiv preprint arXiv:1905.09866*.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, pages 1659–1666.

Masataka Ono, Makoto Miwa, and Yutaka Sasaki. 2015. Word embedding-based antonym detection using thesauri and distributional information. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 984–989.

Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. 2010. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th International Conference on World Wide Web*, pages 751–760.

Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135.

Christos Papadimitriou. 1981. On the complexity of integer programming. *Journal of the Association for Computing Machinery*, 28(4):765–768.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2012. Understanding the exploding gradient problem. *arXiv preprint arXiv:1211.5063*, 2.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. 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.

Carol Pfaff. 1979. Constraints on language mixing: intrasentential code-switching and borrowing in Spanish/English. *Language*, pages 291–318.

Edouard Pineau and Marc Lelarge. 2018. InfocatVAE: Representation learning with categorical variational autoencoders. *arXiv preprint arXiv:1806.08240*.

Jakub Piskorski, Marcin Sydow, and Anna Kupść. 2007. Lemmatization of polish person names. In *Proceedings of the Workshop on Balto-Slavonic Natural Language Processing*, pages 27–34.

Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 412–418.

Joël Plisson, Nada Lavrac, Dunja Mladenic, et al. 2004. A rule based approach to word lemmatization. *Proceedings of the Seventh International Multiconference Information Society*, pages 83–86.

Martin Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.

Chris Quirk, Chris Brockett, and William Dolan. 2004. Monolingual machine translation for paraphrase generation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 142–149.

Ella Rabinovich, Raj Nath Patel, Shachar Mirkin, Lucia Specia, and Shuly Wintner. 2017. Personalized machine translation: Preserving original author traits. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1074–1084.

Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Massively multilingual transfer for NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 151–164.

Piyush Rai, Avishek Saha, Hal Daumé, and Suresh Venkatasubramanian. 2010. Domain adaptation meets active learning. In *Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing*, pages 27–32.

John Rawls. 2001. *Justice as fairness: A restatement*. Harvard University Press.

Avik Ray, Yilin Shen, and Hongxia Jin. 2018. Robust spoken language understanding via paraphrasing. In *Proceedings of Interspeech 2018*, pages 3454–3458.

Philip Resnik, Olivia Buzek, Chang Hu, Yakov Kronrod, Alex Quinn, and Benjamin B. Bederson. 2010. Improving translation via targeted paraphrasing. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 127–137.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging NLP models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865.

Sebastian Ruder and Barbara Plank. 2018. 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.

Rachel Rudinger, Chandler May, and Benjamin Van Durme. 2017. Social bias in elicited natural language inferences. In *Proceedings of the 1st ACL Workshop on Ethics in Natural Language Processing*, pages 74–79.

Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. 2018. Gender bias in coreference resolution. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2, Short Papers)*, pages 8–14.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial Winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. 2016. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems 29*, pages 2234–2242.

Wael Salloum and Nizar Habash. 2011. Dialectal to standard Arabic paraphrasing to improve Arabic-English statistical machine translation. In *Proceedings of the 1st Workshop on Algorithms and Resources for Modelling of Dialects and Language Varieties*, pages 10–21.

David Sankoff. 1988. Sociolinguistics and syntactic variation. *Linguistics: the Cambridge Survey*, 4:140–161.

Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah Smith. 2019. The risk of racial bias in hate speech detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1668–1678.

Candice Schumann, Xuezhi Wang, Alex Beutel, Jilin Chen, Hai Qian, and Ed Chi. 2019. Transfer of machine learning fairness across domains. *arXiv preprint arXiv:1906.09688*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. 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.

Meet Shah, Xinlei Chen, Marcus Rohrbach, and Devi Parikh. 2019. Cycle-consistency for robust visual question answering. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 6649–6658.

Reza Shokri and Vitaly Shmatikov. 2015. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1310–1321.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy*, pages 3–18.

Congzheng Song, Thomas Ristenpart, and Vitaly Shmatikov. 2017. Machine learning models that remember too much. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 587–601.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Gabriel Stanovsky, Noah Smith, and Luke Zettlemoyer. 2019. Evaluating gender bias in machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1679–1684.

Sandeep Subramanian, Sai Rajeswar, Francis Dutil, Chris Pal, and Aaron Courville. 2017. Adversarial generation of natural language. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 241–251.

Tony Sun, Andrew Gaut, Shirlyn Tang, Yuxin Huang, Mai ElSherief, Jieyu Zhao, Diba Mirza, Elizabeth Belding, Kai-Wei Chang, and William Yang Wang. 2019. Mitigating gender bias in natural language processing: Literature review. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1630–1640.

Richard Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems 12*, pages 1057–1063.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *Proceedings of the International Conference on Learning Representations*.

Songbo Tan, Xueqi Cheng, Yuefen Wang, and Hongbo Xu. 2009. Adapting naive bayes to domain adaptation for sentiment analysis. In *Advances in Information Retrieval*, pages 337–349.

Martin Tanner and Wing Hung Wong. 1987. The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association*, 82(398):528–540.

Anna Tordai and Maarten De Rijke. 2005. Four stemmers and a funeral: Stemming in hungarian at clef 2005. In *Workshop of the Cross-Language Evaluation Forum for European Languages*, pages 179–186.

Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Guillaume Lample, and Chris Dyer. 2015. Evaluation of word vector representations by subspace alignment. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2049–2054.

Alan Turing. 1950. Computing machinery and intelligence. *Mind*, 59(236):433–460.

Jenine Turner and Eugene Charniak. 2005. Supervised and unsupervised learning for sentence compression. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 290–297.

Shyam Upadhyay, Manaal Faruqui, Chris Dyer, and Dan Roth. 2016. Cross-lingual models of word embeddings: An empirical comparison. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1661–1670.

Teun Adrianus Van Dijk. 1977. *Text and context: Explorations in the semantics and pragmatics of discourse*. Addison-Wesley Longman Ltd.

Tim Van Erven, Wojciech Kotłowski, and Manfred Warmuth. 2014. Follow the leader with dropout perturbations. In *Conference on Learning Theory*, pages 949–974.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103.

Stefan Wager, Sida Wang, and Percy Liang. 2013. Dropout training as adaptive regularization. In *Advances in Neural Information Processing Systems 26*, pages 351–359.

Karl Weiss, Taghi Khoshgoftaar, and DingDing Wang. 2016. A survey of transfer learning. *Journal of Big Data*, 3(1):9.

Fangzhao Wu and Yongfeng Huang. 2016. 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.

Lijun Wu, Yingce Xia, Fei Tian, Li Zhao, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2018. Adversarial neural machine translation. In *Asian Conference on Machine Learning*, pages 534–549.

Yi Wu, David Bamman, and Stuart Russell. 2017. Adversarial training for relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1778–1783.

Sander Wubben, Antal van den Bosch, and Emiel Krahmer. 2010. Paraphrase generation as monolingual translation: Data and evaluation. In *Proceedings of the 6th International Natural Language Generation Conference*.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc Le. 2019. Unsupervised data augmentation. *arXiv preprint arXiv:1904.12848*.

Caiming Xiong, Victor Zhong, and Richard Socher. 2017. Dynamic coattention networks for question answering. In *Proceedings of the 5th International Conference on Learning Representations*.

Huan Xu and Shie Mannor. 2012. Robustness and generalization. *Machine Learning*, 86(3):391–423.

Yi Yang and Jacob Eisenstein. 2015. Unsupervised multi-domain adaptation with feature embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 672–682.

Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. 2018. Improving neural machine translation with conditional sequence generative adversarial nets. 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 1346–1355.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32*, pages 5754–5764.

Wen-tau Yih, Xiaodong He, and Christopher Meek. 2014. Semantic parsing for single-relation question answering. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 643–648.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pages 2852–2858.

Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. 2019. Adversarial examples: Attacks and defenses for deep learning. *IEEE Transactions on Neural Networks and Learning Systems*, 30(9):2805–2824.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28*, pages 649–657.

Han Zhao, Jianfeng Chi, Yuan Tian, and Geoffrey J Gordon. 2019a. Adversarial task-specific privacy preservation under attribute attack. *arXiv preprint arXiv:1906.07902*.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Ryan Cotterell, Vicente Ordonez, and Kai-Wei Chang. 2019b. Gender bias in contextualized word embeddings. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long and Short Papers)*, pages 629–634.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2017. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2979–2989.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018a. Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 15–20.

Jieyu Zhao, Yichao Zhou, Zeyu Li, Wei Wang, and Kai-Wei Chang. 2018b. Learning gender-neutral word embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4847–4853.

Shiqi Zhao, Xiang Lan, Ting Liu, and Sheng Li. 2009. Application-driven statistical paraphrase generation. In *Proceedings of the Joint Conference of the 47th Annual Meeting of*

the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, pages 834–842.

Shiqi Zhao, Cheng Niu, Ming Zhou, Ting Liu, and Sheng Li. 2008a. Combining multiple resources to improve SMT-based paraphrasing model. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1021–1029.

Shiqi Zhao, Haifeng Wang, Ting Liu, and Sheng Li. 2008b. Pivot approach for extracting paraphrase patterns from bilingual corpora. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 780–788.

Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2018c. Generating natural adversarial examples. In *International Conference on Learning Representations*.

Wangchunshu Zhou, Tao Ge, Ke Xu, Furu Wei, and Ming Zhou. 2019. BERT-based lexical substitution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3368–3373.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *2017 IEEE International Conference on Computer Vision*, pages 2242–2251.

Yftah Ziser and Roi Reichart. 2017. Neural structural correspondence learning for domain adaptation. In *Proceedings of the 21st Conference on Computational Natural Language Learning*, pages 400–410.