Contrapositive Local Class Inference Prediction

by

Omid Kashefi

Bachelor of Science, Iran University of Science and Technology, 2006

Master of Science, Iran University of Science and Technology, 2009

Submitted to the Graduate Faculty of

the School of Computing and Information in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2022

UNIVERSITY OF PITTSBURGH

SCHOOL OF COMPUTING AND INFORMATION

INTELLIGENT SYSTEMS PROGRAM

This dissertation was presented

by

Omid Kashefi

It was defended on

March 29th 2022

and approved by

Prof. Rebecca Hwa, Department of Computer Science

Dr. Adriana Kovashka, Department of Computer Science

Prof. Diane Litman, Department of Computer Science

Dr. Yu-Ru Lin, Department of Informatics and Networked Systems

Dr. Na-Rae Han, Department of Linguistics

Dissertation Advisor: Prof. Rebecca Hwa, Department of Computer Science

Contrapositive Local Class Inference Prediction

Omid Kashefi, PhD

University of Pittsburgh, 2022

Certain types of classification problems may be performed at multiple levels of granularity; for example, we might want to know the sentiment polarity of a document or a sentence, or a phrase. Making more localized prediction (e.g., words or phrases), however, is relatively harder because the role of smaller text units depends on the context in which they are used (e.g., sentences or paragraphs), and training a supervised model to take the context into account, requires labeled training corpora, which is not available in many problem domains. Often, the global prediction at a greater context may be informative for a more localized prediction at a smaller semantic unit. However, directly inferring the most salient local features from the corresponding, easier to learn, global prediction may overlook the semantics of this relationship. This thesis argues that inference along the *contraposition* relationship of the local prediction and the corresponding global prediction makes a more robust and accurate inference scheme, and shows how it can be implemented as a *transfer function* that rewrites a greater context from one class to another.

We study the generalizability of the proposed framework to problem domains with varying data availability profiles and different levels of inference granularity. We demonstrate the robustness of the contrapositive inference to the noisy data and how *data augmentation* can facilitate the generation of weakly-labeled training data for resource-constrained problem domains. We discuss the *transferability* and *adaptability* of the contrapositive relationship to the problem domains with limited amount of training data. In addition, we show the robustness of the contrapositive inference scheme to variability in the size of the local and global contexts: from paragraphs to sentences, and from sentences to words and phrases.

Table of Contents

List of Tables

List of Figures

Preface

I still remember that day when my elementary school teacher asked us what we wanted to be when we grew up. The answers were mostly typical of our age: I want to be a pilot, an astronaut, a footballer, a soldier... 'I want to be a scientist,' I replied. It was some 30 years ago, but I kept chasing that dream, and now, it is hard to believe that I am finally writing the preface to my Ph.D. dissertation. As I look back, all I see are the people who helped me through this journey and I wish to thank them all.

First and foremost, I would like to express my gratitude and respect for my advisor, Prof. Rebecca Hwa. Her support, encouragement, patience, kindness, and invaluable advice during my Ph.D. studies cannot be adequately expressed. Having the privilege of working with such an attentive, wise, and knowledgeable professor will be my lifelong honor.

I would like to sincerely thank my dissertation committee members Prof. Diane Litman, Dr. Yu-Ru Lin, Dr. Adriana Kovashka, and Dr. Na-Rae Han for their time and insightful comments and feedback that greatly helped strengthen this work. Furthermore, I would like to thank my friends, colleagues, and co-authors for helping me through graduate studies and making it more enjoyable. This especially includes: Mohammad Falakmasir, Mahbaneh Eshaghzadeh, Fattaneh Jabbari, Mahdi Pakdaman, Mohammad Hassanzadeh, Zahra Rahimi, Amin Tajgardoon, Changsheng Liu, Mingda Zhang, Tazin Afrin, Christopher Olshefski, Meghan Dale, Muheng Yan, Wen-Ting Chung, Amanda Godley, Nitika Bhaskar, Giridhar Kumaran, Manoochehr Khazaei, Alireza Golestaneh, Saba Dadsetan, Alireza Samadian, Mohammad Hedayati, and Michele Thomas. I am also grateful to all of my teachers and professors along this journey. Among others, I would like to mention Prof. Mohsen Sharifi and Prof. Behrouz Minaei.

Last but not least, I would like to thank my parents Fathollah and Soheila. I am profoundly grateful for their patience, love, and support from day one. I also wish to express my deepest gratitude and love to my best friend, my beautiful wife, Maryam for her unconditional love and support during the hardest of times. Thank you and our four-pawed daughter Catayoon for showing me the beauty of life every day!

1.0 Introduction

1.1 Motivation

In many NLP applications, a piece of text could be analyzed at multiple levels. For example, a product review might be analyzed for whether it is informative overall; its paragraphs might be analyzed for whether they are relevant to a certain aspect of the product; or its words might be analyzed for whether they express intense emotions. These classification tasks of varying scopes of context are often related. For instance, one might posit that a review containing many words carrying extreme emotions might not be very informative. In some cases, the same class prediction task (e.g., sentiment polarity) might be asked at both the global (e.g., sentences or paragraphs) and local levels (e.g., words or phrases).

Classification at the global scope (e.g., sentences or paragraphs) depends on clues gleaned at a more local level (e.g., words or phrases); however, classification at the local scope is often more difficult because the roles of the words and phrases vary depending on the broader context in which they are used. For example, while the word "spade" in the sentence *"No, you can't sit here, go away you spade."* is deemed as "inappropriate", the same word in the inappropriate sentence *"I will cut off your black head with this very spade myself."* is not deemed inappropriate.

Suppose we want to develop a system that automatically determines which words are contributing most to the inappropriateness of the above sentences. A straightforward approach, such as a lexicon lookup that requires domain-specific dictionaries, may not correctly identify the intended usage. For example, assuming "spade" is listed as "inappropriate" in the lexicon, this approach would only correctly identify "spade" as inappropriate in the first example; but in the second example, where the inappropriate word is actually "black", this approach will incorrectly label "spade" as inappropriate instead. Alternatively, a supervised machine learning approach may be better at taking the context into account, it relies on the availability of the training corpus containing class labels for each local scope. For example, in order to identify the inappropriate expressions in a sentence, one should first collect a large set of inappropriate sentences, then manually examine all of their words and label them as inappropriate or not; then, this labeled corpus may be used to train a supervised model to predict whether a word of a sentence is inappropriate or not. This poses a significant bottleneck for new applications for which these resources are not widely available.

The insight that informs this thesis is that a prediction task at a greater-context (which we refer to as a *global prediction*) may be informative for a more localized prediction at a smaller semantic unit. Because the global prediction task is often easier to learn, this thesis argues that the relationship can be exploited to infer a local prediction from the corresponding global prediction task. For example, suppose someone reported a social media comment as "inappropriate" (*global classification*), if we find out which word(s) in the comment contributed the most to the user's decision to report the comment, we have already found the inappropriate words (*local classification*).

Directly inferring the most salient local features from the global prediction, however, may overlook the more subtle semantics of this relationship. For example, irrelevant words or punctuation marks are more influential to the decision of neural text classifiers than verbs or function words (Mudrakarta et al., 2018; Jain and Wallace, 2019); or the presence of snow in an image is the main feature to distinguish huskies from wolves rather than the features related to the animal themselves (Ribeiro et al., 2016).

To address this issue, we propose to perform the inference along the contrapositive relation between the local and global predictions. That is, suppose we know that an instance is globally predicted to belong to some *Class A* (e.g., "inappropriate") and that some local portion l contributed the most to that prediction; if l is replaced so as to *negate* its semantic contribution, then the global label should also change to some *Class B* (e.g. "appropriate"). We propose that, *if, and only if*, the contrapositive constraint is satisfied should the local prediction be inferred from the global problem.

We argue that the contrapositive constraint can improve the robustness of the semantic relationship between the local and its corresponding global problem. For example, suppose that a comma (",") is the item that influences the decision of a neural sentence classifier the most. While direct inference would incorrectly conclude that this comma is therefore the greatest contributor to the global classification, the contrapositive constraint would rule it out because replacing it with another punctuation mark (e.g., a semicolon (";")) would not change the prediction for the sentences. Thus, while logically equivalent, the contrapositive constraint offers a robust alternative for finding the most semantically contributing local item.

Implementing contrapositive inference, however, is not straightforward. A bottom-up approach requires calculating an adversarial semantic alternative for each local feature to assess their contribution, which might be very complicated and resource-intensive. Instead, we argue that contrapositive inference scheme can be modeled after style transfer and controlled text generation methods, and we propose to implement it as a **transfer function** for global class transference. We cast the problem as a rewriting exercise:

> Rewrite the original global instance, which belongs to some *Class A*, so that it becomes more likely to belong to *Class B*; then, those smaller local parts that were changed in the rewriting are likely to be the heavy contributor to *Class A* so we may infer the same class prediction for them.

As an illustrative example, consider the sentiment classification problem. The transfer function might rewrite a *positive* sentence: "the food was great" into a *negative* sentence "The food was awful." While the sentence length is the same, the transfer function chose to replace "great" with "awful," therefore "great" is likely to be a sentiment expressing word.

In theory, the proposed training approach should be applicable to many local prediction tasks because the core of our approach is domain independent. However, to train an appropriate transfer function we need a corpus of training examples for the global prediction, which might not be readily available in many problem domains. Moreover, the relative size of the local and global context and the annotation profiles of the prediction task and domain may pose different challenges and limitations for the inference framework.

Therefore, in this thesis we study the generalizability of our proposed inference framework to problem domains with varying data availability profiles and different levels of inference granularity. We demonstrate the robustness of the contrapositive inference to the noisy data and how *data augmentation* can facilitate the generation of weakly-labeled training data for resource-constrained problem domains. We discuss the *transferability* and *adaptability* of the contrapositive relationship to the problem domains with limited amount of training data. In addition, we show the robustness of our approach to variability in the size of the local and global context: from paragraphs to sentences, and from sentences to word(s) and phrase(s).

1.2 Thesis Statement

Text classification at different scopes (text spans) is fundamental to many NLP problems. In this thesis, I test for the following hypotheses on inferring a category prediction at local scope along its contrapositive relationship with the corresponding global problem:

H1. The **contrapositive** relationship between the local and global problems provides a strong signal for inferring local category predictions.

H2. The contrapositive local inference scheme can be modeled as global scope's **class transfer** exercises: rewrite the original global instance, which belongs to some *Class A* so that it becomes more likely to belong to *Class B*; then, those local textual parts that were changed in the rewriting are likely to be the heavy contributor to *Class A* so we may infer the same class prediction for them.

H3. The contrapositive inference scheme is generalizable to many resource-constrained problem domains with different global **data availability** profiles.

H4. The contrapositive inference scheme is robust to variability in the **relative size** of the local and global contexts.

1.3 Thesis Overview

This thesis argues that the contrapositive constraint between the local and global prediction problems makes their semantic relationship more robust to variability in inference granularity and limitations in the availability of the training data.

This work also discusses how this inference scheme can be implemented as a transfer function model. The experimental results validate our insight that the contrapositive inference scheme outperforms the alternative approaches on different resource-constrained problem domains with variability in the size of local and global contexts, availability of global training data. A brief overview of this thesis is presented below.

1.3.1 Our Contrapositive Local Class Inference Approach

In Chapter 3, we discuss how the relationship between local and global prediction problems can be utilized to create a low-resource prediction framework for local scopes. We argue that this semantic relationship can be made more robust by enforcing the **contrapositive** constraint between the local prediction and its corresponding global prediction. We then explain how this contrapositive inference scheme can be modeled as global context class transference exercise. Thus, we propose a transfer function model as a deep generative approach to implement the contrapositive local inference scheme.

1.3.2 Data Availability Generalization

Developing a transfer function requires a large corpus of training examples for the global prediction. For some scenarios, this data requirement may not be an insurmountable problem. However, there are many cases in NLP where such annotated corpora are not available. In Chapter 4, we discuss how *data augmentation* can facilitate the generation of weakly-labeled training data for resource-constrained problem domains and train a robust transfer function. In addition, for some problem domains with limited data availability, we investigate how *supervised transfer learning* and *unsupervised domain adaptation* may improve the training process of our transfer function.

We study the generalizability of our contrapositive inference scheme by experimenting on three domains with different data availability profiles and settings: *sentiment analysis*, *semantic pleonasm detection*, and *specificity analysis.* These problem domains are described in Section 2.5.

1.3.3 Inference Granularity Generalization

Our contrapositive inference scheme aims to infer a local class label from the corresponding global label. However, different domains and tasks may have varying sizes and annotation profiles for local and corresponding global contexts that may pose different challenges for our proposed approach. In Chapter 5, we investigate the generalizability of the contrapositive inference scheme to different levels of inference granularity: (i) from paragraph to sentence, (ii) from sentence to subsentence, which could be a few words, a phrase, or multiple phrases, and (iii) from sentence to word.

1.4 Thesis Contributions

Identifying the category of a certain part of the text and understanding how it influences the category prediction of the global context is a core problem in many NLP problems. However, the reliance of existing supervised solutions on the availability of large training corpora hinders their applicability to new or resource-constrained problems. This thesis proposes an alternative contrapositive inference framework for making localized category prediction that is only informed by the category prediction of the global problem, so better suited for low-resource classification settings.

- In order to perform the local category prediction inference (Kashefi and Hwa, 2021):
  - We propose to infer the same global category for the most contributing local context(s), if and only if, when the correlation of them satisfies the *contraposition* constraint.
  - We argued that the local contrapositive inference scheme can be modeled as paraphrasing and class transference at global context level.
  - We present a deep generative *transfer function* as the computational model for implementing the contrapositive inference scheme.
- To reconcile the training requirements of the transfer function model:

  - We propose a non-label-reserving *data augmentation scheme* and an *evaluation framework* to measure the suitability of different augmentation heuristics for a classification task. We show that our transfer function can robustly train on an appropriate augmented dataset (Kashefi and Hwa, 2020).
  - We show that the contrapositive semantic relation between a local problem and a corresponding global problem is *transferable* between related domains. Therefore, the training of a transfer function in a resource-constrained problem can be facilitated by utilizing training data from a related resource-rich domain.
  - We demonstrate that *unlabeled* data can also be informative and improve the learning of the transfer function. Therefore, we proposed an unsupervised *domain adaptable* transfer function model approach to enhance the applicability of the contrapositive inference scheme across different domains.
- To expand the generalizability of our contrapositive prediction framework to different levels of inference granularity:
  - We show that regularizing the objective function of the transfer function with our proposed *conciseness loss*, which minimizes the difference between the original and generated global instances, (i) can satisfy the contrapositive constraint with minimal changes, thus improving local prediction accuracy; (ii) allows controlling for the amount of local change during the global rewriting process, thus adapting to the variability in the granularity of the local and global context sizes in different problem domains.
  - We also show that the inclusion of *self-attention* layers during encoding (and decoding) may also provide the *phrasal* representation of the local context, thus extending the application of contrapositive inference scheme to problem domains with the phrase(s) as the contributing local features.
- In addition, we also developed a few benchmark datasets with annotations at local level for evaluating the generalizability of the contrapositive inference scheme to different problem domains. These resources are publicly available to the community and could be used to study a variety of tasks and applications other than those mentioned in this thesis.

- Semantic Pleonasm Corpus (Kashefi et al., 2018) is a collection of three thousand sentences, each featuring a pair of potentially semantically related words. Human annotators determine whether either (or both) of the words is pleonastic (semantically redundant). The corpus offers two improvements over current resources: (i) It filters for grammatical sentences so that the question of redundancy is separated from grammaticality. (ii) It is filtered for a balanced set of positive and negative examples (i.e., not redundant).
- ArgRewrite V2 (Kashefi et al., 2022) is a corpus of annotated argumentative revisions, collected from two cycles of revisions to argumentative essays in response to a prompt. Annotations are provided at different levels of purpose granularity (coarse and fine) and scope (sentential and subsentential), which makes it useful for a wide range of applications.

2.0 Background

In this chapter, we review the literature of research on the concepts that are discussed throughout the dissertation. The first part of the chapter outlines the local classification problems in NLP and the general approaches to solving them, including limitations and requirements. Next, we discuss the low-resource NLP techniques and approaches that may help to overcome these limitations. Lastly, we review research on (controlled) language generation, which shares a similar concept with our generative approach for contrapositive local classification inference.

2.1 Text Classifications in NLP

A local lexical item could be a single word or a sequence of words (phrase) that acts as a unit of meaning. For example, "thesis", "awesome", "accredited investor", "by the way", and "don't count your chickens before they hatch" are all local lexical units to a larger global semantics such as sentences or paragraphs (Sinclair, 1998). Lexical items are important analysis units of the language (Ogden, 1932; Willis, 1990). Linguists believe that language is formed from grammaticalized lexical items and not the lexicalized grammar (Lewis, 1993). This means local context offers far more language generative power than grammatical structure. The human brain stores and processes localized lexical items (chunks) as individual wholes, and the meaning and intent of a global context is further inferred from its localized items (Schmit, 2000). Given the importance of smaller textual units in forming the language, understanding the role of local context and its semantic contribution to the meaning of the global context is fundamental to many NLP tasks and applications such as: part-of-speech (POS) tagging, named entity recognition (NER), spell and grammatical error correction (GEC), sentiment analysis, keyword, and topic extraction, identifying the semantic features of the text (e.g., being concise or verbose, specific or general, argumentative or persuasive), spam detection, automatic essay grading, and many more.

For example, suppose you ask Amazon's Alexa for a quick summary of critic reviews for a movie you plan to watch[1]; Alexa will go through hundreds of comments from different critics, pinpoint the word(s) that is strong indicators of their opinion, and the aggregate the overall consensus of the critics' opinion. Or, suppose a student writer is being asked by the professor to make his argumentative essay more concise and clear; to do that, she needs to read the essay sentence by sentence and find the words and phrases that do not contribute to the overall intent of the essay, or do not strengthen the argument, and perhaps remove them to make the essay more concise. In the above examples, the core problem is identifying certain words and understanding their roles (e.g., positive or negative opinions). However, predicting the category of local items is difficult because the roles of the words and phrases vary depending on the broader context in which they are used. For example, in the sentence *"The story is unpredictable"* the word "unpredictable" is carrying a positive sentiment while the same word in the sentence *"The steering wheel is unpredictable"* is having a negative sentiment.

2.1.1 Lexicon Lookup

A straightforward approach to predict the category of local items is to have a static dictionary of words and their category information. For example, to predict whether a word spelling is *correct* or *incorrect*, a dictionary of languages' lexemes[2] and a set of morphological rules[3] could be used: every word of a sentence would be looked up (pattern matched) through the list of morphological rules to strip to their lexeme, then the lexeme would be looked up in the lexeme of correct languages lexemes, if the word is found it is *correct*, otherwise it is *incorrect* (Mitton, 2010).

Many earlier attempts on local classification tasks in NLP such as, spell and grammatical error correction (Blair, 1960; Damerau, 1964; Mitton, 1987; Mcilroy, 1982), part-of-Speech (or sometimes called grammatical) tagging (Francis and Kucera, 1979; Greene and Rubin,

[1] `https://www.amazon.com/Alexa-Skills-Movie-Info-Reviews/b?ie=UTF8&node=14284848011`

[2] The simple form of a word that may go through inflection (a process of word formation) to being modified and express different grammatical categories such as tense, voice, person, number, gender, etc (Crystal, 2008). For example, "take", "takes", "took", "taken" and "taking" are forms from the lexeme "take."

[3] For example: `present participle/gerund:  <verb lexeme> + -ing`

1971), named entity recognition (Grishma, 1995; Krupka and Hausman, 1998), and sentiment analysis (Stone and Hunt, 1963; Taboada et al., 2011; Moreo et al., 2012) are based on lexicon and knowledge base lookup.

Lexicon lookup approaches require large domain appropriate dictionaries; collecting these dictionaries is a tedious task and may require domain knowledge and expertise; moreover, lexicon lookup approaches are prone to make inaccurate predictions, especially when the local item's category depends on the context. For example, while the word "keep" is categorized as a *verb* in the sentence "keep up the good work!", it is categorized as a *noun* in the sentence "working overtime to earn his keep"; however, a lexicon lookup approach may fail to correctly predict the category of it in one of the example sentences.

2.1.2 Supervised Learning

Another class of approaches towards localized classification is *supervised learning*: inferring a function from already observed items to their category membership, which can be used for mapping new items to a category prediction (Alpaydin, 2020). Supervised learning can be formulated as: given a set of N training examples of the form $D_{train} = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, where x_i is the i^{th} training example and y_i is its category, inferring a function $\mathcal{F} : X \rightarrow Y$, where X is the input space of examples and Y is the output space of class labels. Function $\mathcal{F}$ could be represented using a scoring function $\mathcal{S} : X \times Y \rightarrow \mathbb{R}$ that returns the y value that gives the highest score as:

$$\mathcal{F}(x) = \arg\max_{y} \mathcal{S}(x, y)$$

Thus, for a newly observed local item $x \notin D_{train}$, function $\mathcal{S}$, which is inferred based on the already observed category membership of the training data, will return the most likely category label y that item x may belong to. Functions $\mathcal{F}$ and $\mathcal{S}$ can be any space functions, but majority of supervised learning algorithms are probabilistic models and function $\mathcal{F}$ takes the form of a conditional probability: $\mathcal{F}(x, y) = P(y|x)$, referred to as *discriminative models*, or function $\mathcal{S}$ takes the form of a joint probability model: $\mathcal{S}(x, y) = P(x, y)$, referred to as *generative models*.

A *loss function* $\mathcal{L} : Y \times Y \rightarrow \mathbb{R}$ is then measure how good a function $\mathcal{F}$ can fit the training examples and a supervised learning algorithm tries to minimize the expected loss of function $\mathcal{F}$ as in Equation (2.1).

$$L_{sup} = \min_{\mathcal{F}} \frac{1}{N} \sum_{i} \mathcal{L}(y_i, \mathcal{F}(x_i)) \tag{2.1}$$

Since supervised models have been shown to be capable of making more robust and accurate predictions compared to knowledge-based models, such as lexicon lookup, many NLP approaches shifted their focus on using supervised strategies (Johnson, 2009; Schafer, 2011), including spell and grammatical error correction (Han et al., 2006; Rozovskaya and Roth, 2010; Tetreault et al., 2010; Dahlmeier and Ng, 2011), part-of-Speech tagging (Brants, 2000; Collins, 2002; Toutanova et al., 2003; Choi, 2016; Akbik et al., 2018), named entity recognition (Zhou and Su, 2002; Mccallum and Li, 2003; Ratinov and Roth, 2009; Lample et al., 2016), sentiment analysis (Wilson et al., 2005; Socher et al., 2013; Liu, 2012; Nakov et al., 2019; Xue and Li, 2018) and semantic role labeling (Carreras and Arquez, 2005; Palmer et al., 2010; He et al., 2017; Tan et al., 2018).

Although supervised approaches may better take the context into account, they heavily rely on the availability of the training corpus containing the class labels for each local instance, which is not available in many domains, and creating them requires significant manual effort and expertise. On the other hand, there are many problems with very specific aims within a larger application goal that given their narrow scope and specific application, it's unlikely that a large quantity of annotated corpora would ever be developed for them.

2.1.3 Multi-Instance Learning

Multi-Instance learning (MIL) is a form of supervised learning where (local) instances are not individually labeled, but a (global) label is provided for a bag of instance (Dietterich et al., 1997; Maron and Lozano-Pérez, 1997; Zhou and Zhang, 2002; Zhou, 2004). As a result of allowing to leverage weakly labeled data, MIL is applicable to a wide range of (local) problems from computer vision (Chen et al., 2006; Rahmani and Goldman, 2006; Phan et al., 2015; Cinbis et al., 2017), to document (Bunescu and Mooney, 2007; Zhou et al., 2009) and sound (Briggs et al., 2012) classifications. For example, an object detection model can be trained on images retrieved from the web in response to a query as weak global supervision, instead of locally annotated data; an aspect-based sentiment classifier is trained with an overall rating of a review paragraph instead of costly local annotations for every sentence and aspects.

In the simple binary MIL classification, a bag is classified as negative ($y = -1$) if it contains all negative instances, and as positive ($y = 1$) if it contains at least one positive instance. The MIL problem can be formulated as training a scoring function $\mathcal{S}(C) \in R$ as follows, where C is a set of instances x, and f and g are suitable transformation functions:

$$S(C) = g(\sum_{x \in C} f(x)) \tag{2.2}$$

Depending on the choice of the transformation functions f and g, the MIL approaches fall into two main categories:

- *Instance-based.* The model (f) first classifies each instance separately, then MIL pooling (g) combines the instance labels to assign a label to the bag (Raykar et al., 2008; Cheplygina et al., 2015). Since individual labels are unknown, it is possible that the instance-level classifier makes noisy predictions (Kandemir et al., 2016).
- *Embedding-based.* Instead of classifying the instances individually, the function f maps instances to a low-dimensional embedding. MIL pooling is then used to obtain a bag representation that is independent of the number of instances in the bag. Function g then classifies these bag representations (Pinheiro and Collobert, 2015; Feng and Zhou, 2017; Zhu et al., 2017; Ilse et al., 2018).

The MIL involves two main tasks (i) learn a global function to predict a label for a bag of instances, (ii) predict the local instance(s) that trigger the bag label (Liu et al., 2012; Oquab et al., 2014; Ilse et al., 2018). Our work in this thesis has a similar objective to the second task. However, rather than directly inferring the key local instance from global prediction, which is reported to be noisy (Kandemir et al., 2016; Cheplygina et al., 2015), we propose to infer the most contributing local instance through its *contrapositive* relationship with the global prediction (see Chapter 3).

2.1.4 Annotator's Rationale

Rationales are the clues and reasons that annotators may use when they are deciding what the correct label is on training data (Zaidan et al., 2007). For example, an annotator who is categorizing the following email as *spam* or *not spam* might also highlight some part of it that influenced his annotation decision:

> I'm from the Facebook **security team** and we observed an unusual activity on your account. It looks like your credentials are compromised and someone else is taking over your account. Your account will be **suspended by the end of the day** and **terminated in 3 days**. If you wish to keep your account, please **immediately** respond with your **username and password** to verify that you are the owner of the account so we can provide you with the link to change your password.

Researchers have shown that including even small instances of rationals in the training data may account for the lack of sizable labeled training examples and significantly improve the learning process and classification performance (Zaidan et al., 2007; Zhang et al., 2016b). The straightforward way of using annotator's rationales to improve the learning process has some tie with data augmentation, adding rationales to the training set as *pseudoexamples*. Considering r_{ij} be the rationales for the example (x_i, y_i), the standard supervised learning process in Equation (2.1) would be extended as follows, to exploit the pseudo examples (r_{ij}, y_i), where λ is the rationales weights in the learning process and N_R is the total number of rationales:

$$L_R = \min_{\mathcal{F}}(\frac{1}{N}\sum_i \mathcal{L}(y_i, \mathcal{F}(x_i)) + \lambda \frac{1}{N_R}\sum_{i,j}\mathcal{L}(y_i, \mathcal{F}(r_{ij}))) \tag{2.3}$$

Annotator's rationales are originally introduced to improve the classification. Researchers exploit the rationales to improve the learning process of both traditional classifier models (Zaidan et al., 2007, 2008; Marshall et al., 2016) and neural models (Zhang et al., 2016b; Rabinovich Shachar Mirkin Raj Nath Patel et al., 2016; Strout et al., 2019; Jain and Wallace, 2019).

Our work in this thesis, however, has the opposite aim. Rather than trying to improve the classification of the greater context by relying on rationales, we want to exploit the classification of the greater context (because it is easier to obtain) to help us identify the rationales (Lei et al., 2016; Bao et al., 2018). In Chapter 3, we propose an alternative training framework to infer the location and class label of the local instances (as rationales) from the class label of the greater context.

2.2 Approaches for Low-Resource NLP

Since this thesis's focus is on problem domains with limited resources, this Section reviews the literature of research on the NLP techniques and approaches that may help our approach to better cope with resource scarcity.

2.2.1 Data Augmentation

Data augmentation is a technique for generating additional training data by applying a heuristic transformation to the existing training examples. For example, an existing image could be rescaled or flipped to get more images with the same label to expand the size and diversity of the training dataset and thus train a more reliable and accurate model (Frénay and Verleysen, 2014; Hendrycks et al., 2018; Shorten and Khoshgoftaar, 2019).

Data augmentation could be formulated as Equation (2.4), where h is a heuristic function that transforms the datapoint and label pair of (x, y) to a new augmented sample $(\hat{x}, \hat{y})$.

$$(\hat{x}, \hat{y}) = h(x, y) \tag{2.4}$$

The majority of existing data augmentation approaches are *label-preserving*, which relaxes the Equation (2.4) as $(\hat{x}, y) = h(x, y) = (h(x), y)$; this means, if x belong to some class A, augmented $\hat{x}$ also belong to class A. For example, using a synonym replacement heuristic, a sentence with positive sentiment could be augmented into a new example while preserving the overall positive sentiment:

Original Example: *"they serve high-quality food here!"*

Class Label: *Positive* Sentiment

Augmented Example 1: *"they serve top-notch food here!"*

Augmenting Heuristic: *Synonym Replacement*

Class Label Label: *Positive* Sentiment

Augmented Example 2: *"they serve food here!"*

Augmenting Heuristic: *Random Deletion*

Class Label Label: *Positive* Sentiment

Label-preserving data augmentation requires existing labeled samples for every class that is needed to be augmented. Data augmentation can be *non-label-preserving* as well (Kashefi and Hwa, 2020), where the label itself might also transform using function h_y that expands Equation (2.4) as:

$$(\hat{x}, \hat{y}) = h(x, y) = (h_x(x), h_y(y))$$

This means, while x belongs to class A, augmented $\hat{x}$ might not necessarily belong to class A. For example, by replacing the most positive word(s) of sentences with positive sentiment with an antonym, the sentence's sentiment may become negative. Non-label preserving data augmentation is not bound to the assumption of having labeled samples for the instances of all classes and samples from one class may be enough to generate instances of other classes:

Original Example 1: *"they serve high-quality food here!"*

Class Label: *Positive* Sentiment

Augmented Example 1: *"they serve terrible food here!"*

Augmenting Heuristic: *Antonym Replacement*

Class Label Label: *Negative* Sentiment

Original Example 2: *"they serve high-quality food here!"*
Class Label: *Concise*

Augmented Example 2: *"they serve excellent high-quality food here!"*
Augmenting Heuristic: *Adjectival Synonym Insertion*
Class Label Label: *Verbose*

Data augmentation, can improve the learning process of inferring the function from examples domains to class labels domain by providing more examples, so Equation (2.1), the standard supervised learning process mentioned in Section 2.1.2, could be extended as follows, where λ is the weight of augmentation examples (usually equal to 1), N_{aug} is the total number of augmented examples, and $h_y(y) = y$, when the heuristic is label-preserving, and $h_y(y) : Y \rightarrow Y$, could be any function that transforms label when the heuristic is non-label-preserving:

$$L_{aug} = \min_{\mathcal{F}}(\frac{1}{N}\sum_i \mathcal{L}(y_i, \mathcal{F}(x_i)) + \lambda \frac{1}{N_{aug}}\sum_j \mathcal{L}(h_y(y_j), \mathcal{F}(h(x_j)))) \tag{2.5}$$

Although data augmentation faces some challenges in NLP applications and small random perturbations in the text may lead to a loss of information or the addition of misleading information, data augmentation has been shown to be useful for many NLP applications, with researchers proposing many different approaches for text data augmentation; for example, (Zhang et al., 2015; Wei and Zou, 2019) used thesaurus-based and (Wang and Yang, 2015; Kobayashi, 2018; Jiao et al., 2019) used embedding-based lexical substitution to replace a word with a synonym, (Wei and Zou, 2019; Xie et al., 2019) used random noise injection, including random word insertion, deletion, or sentence shuffling, (Luque, 2019) used instance crossover by combining halves of tweets to generate new samples, (Guo et al., 2019) adapt the mixup approach (Zhang et al., 2018) to text by interpolating the distributed representation of different sentences to generate new (continuous) training samples, (Sennrich et al., 2016b; Fadaee et al., 2017; Xie et al., 2019) used back-translation, and (Hu et al., 2017; Iyyer et al., 2018; Anaby-Tavor et al., 2020; Kumar et al., 2020) used (deep) generative models to augment more training data.

2.2.2 Transfer Learning

Machine learning algorithms are working based on the assumption that training data and the new observations are drawn from the same distribution (Thrun and Pratt, 1998; Pan and Yang, 2010; Zhuang et al., 2020). However, this assumption may not hold for many NLP applications. For example, in a *movie review* sentiment classification task, we may only have sizable training data on *restaurant review* domain, which maybe have a different feature space and distribution than movie reviews, to train a sentiment classifier. Traditionally, when the domain and distribution change, we need to collect and label a large amount of data for the new domain (e.g. movie review) and retrain the model from scratch on the new data. However, it might be expensive or even impossible to collect such labeled data for many NLP problem domains.

Transfer learning is a technique that helps to improve the training process by allowing to transfer the task's semantics learned from a resource-richer domain (e.g. restaurant review) into a resource-poorer domain (e.g. movie review) and save a significant amount of labeling effort (Thrun and Pratt, 1998; Blitzer et al., 2007). The concept of transfer learning is motivated by the fact that people can use previously learned knowledge for solving new problems more efficiently (Pan and Yang, 2010). For example, learning to recognize a *cat* from a *sparrow* may help to recognize a *dog* from an *eagle*, or learning to play *Squash* (a racket and ball sport) may facilitate learning to play Tennis.

In Section 2.1.2, we presented the formal definition of the supervised learning task, which by default assumes source and target examples are belonging to the same feature space or drawn from the same distribution; we expand the definition of the learning process with the notion of *data domain* and *learning task*, following the same notation proposed by (Pan and Yang, 2010), to explore how transfer learning may formally define.

As mentioned in Equation (2.1), a learning task $\mathcal{T} = \{Y, \mathcal{F}(X)\}$ is to infer a predictive function $\mathcal{F}$ that maps the data examples $X = \{x_1, x_2, ..., x_n\}$ into label space Y. A data domain $\mathcal{D} = \{\mathcal{X}, P(X)\}$ is a combination of feature space $\mathcal{X}$, where $X \in \mathcal{X}$, and the marginal probability distribution over data examples $P(X)$. The *source domain* is defined as $\mathcal{D}_S = \{(x_{S_1}, y_{S_1}), (x_{S_2}, y_{S_2}), ..., (x_{S_n}, y_{S_n})\}$, where $x_{S_i} \in \mathcal{X}_S$ is the i^{th} source data example and $y_{S_i} \in Y_S$ is its category label. Similarly, the *target domain* is defined as

$\mathcal{D}_T = \{(x_{T_1}, y_{T_1}), (x_{T_2}, y_{T_2}), ..., (x_{T_n}, y_{T_n})\}$, where $x_{T_i} \in \mathcal{X}_T$ is the i^{th} source example and $y_{T_i} \in Y_T$ is its category label. Given these, transfer learning can be defined as follows:

> *Given the target learning task $\mathcal{T}_T$, improving the target prediction function $\mathcal{F}_T$ in $\mathcal{D}_T$ using the knowledge learned from $\mathcal{T}_S$ and $\mathcal{D}_S$, where $\mathcal{T}_S \neq \mathcal{T}_T$ and $\mathcal{D}_S \neq \mathcal{D}_T$, and in most of the cases $0 \leq n_T << n_S$.*

The condition $\mathcal{T}_S \neq \mathcal{T}_T$ implies that either $Y_S \neq Y_T$, for example, the source task is a binary classification and the target task is multiclass classification, or $g_S \neq g_T$, for example, the distribution of labels in source and target domains are very different. The condition $\mathcal{D}_S \neq \mathcal{D}_T$ implies that either $\mathcal{X}_S \neq \mathcal{X}_T$, for example, source examples are voice messages and the target examples are their textual transcriptions, or $P(X_S) \neq P(X_T)$, for example, the source domain is restaurant reviews and the target domain is movie reviews.

From the relation between the source and target tasks perspective, transfer learning could be categorized into two main family of approaches: *inductive transfer learning*, when $\mathcal{T}_S \neq \mathcal{T}_T$; *transductive transfer learning*, when $\mathcal{T}_S = \mathcal{T}_T$ (Arnold et al., 2007). There is a large body of research in transfer learning studies; transductive approaches are mainly transferring the knowledge of *instances* or *features* between domains; inductive approaches may also aim to transfer the *model parameters* between tasks (Evgeniou and Pontil, 2004; Lawrence and Platt, 2004; Gao et al., 2008; Bonilla et al., 2008), as well as knowledge of instances and features between domains. However, since in this thesis, we are targeting the same classification task in different domains (i.e., $\mathcal{T}_S = \mathcal{T}_T$ and $\mathcal{D}_S \neq \mathcal{D}_T$), we mostly focus on transductive transfer learning, especially from the data availability perspective in source and target domains.

Instance-based transfer learning approaches re-weight the source domain examples to reduce the marginal distribution difference between source and target domain (Huang et al., 2006; Dai et al., 2007; Jiang and Zhai, 2007; Yao and Doretto, 2010; Wan et al., 2011; Sun et al., 2011; Li et al., 2017; Xu et al., 2017); To do so, the loss function of the target learning task, given in Equation (2.1) would be extended as follows , where $\lambda \approx \frac{P(X_T)}{P(X_S)}$ is the weighting parameter for source instances:

$$L_T = \min_{\mathcal{F}} \frac{1}{N_S} \sum_i \lambda\, \mathcal{L}(y_{S_i}, \mathcal{F}(x_{S_i})) \approx \min_{\mathcal{F}} \frac{1}{N_S} \sum_i \frac{P(X_T)}{P(X_S)} \mathcal{L}(y_{S_i}, \mathcal{F}(x_{S_i})) \tag{2.6}$$

Feature-based transfer learning approaches are trying to find a latent space that reduces the difference between source and target domains. The target learning task would be in the form of $\mathcal{T}_t = \{Y_T, \mathcal{F}(\mathcal{X}_{S,T})\}$, where $\mathcal{X}_{S,T}$ is latent feature representation that minimizes the classification error in $\mathcal{D}_T$. Researcher has proposed different approaches for calculating an efficient $\mathcal{X}_{S,T}$, including, feature augmentation (Lee et al., 2007; Li et al., 2014; Ghifary et al., 2015), feature mapping (Zhou and Li, 2005; Wang and Sridhar, 2008; Chen et al., 2011; Beigman Klebanov et al., 2015; Saito et al., 2017; Long et al., 2017), feature selection and construction selection (Jebara, 2004; Ganin et al., 2016; Tzeng et al., 2015, 2017), feature encoding (Glorot et al., 2011; Chen et al., 2015; Vincent et al., 2008), and feature alignment (Argyriou et al., 2008; Gong et al., 2012; Fernando et al., 2013; Tzeng et al., 2014; Sun et al., 2016; Arjovsky et al., 2017; Luo et al., 2017).

2.2.2.1 Domain Adaptation

Domain adaptation is a special case of transfer learning, where there is no labeled data available in the target domain ($\mathcal{D}_T$). It shares the same goal with transfer learning:

> *Improve the target prediction function ($\mathcal{F}_T$) in the target domain ($\mathcal{D}_T$), using the knowledge learned from the source domain ($\mathcal{D}_S$), without supervision at target domain ($\mathcal{D}_T$).*

Similar to supervised transfer learning approaches, domain adaptation approaches can also be categorized as:

- *Instance-Based* domain adaptation, which operates at the instances space through reweighting, diversifying, or adding noise to the source examples to reduce the marginal distribution difference between the source and target domains (Rasmus et al., 2015; Tarvainen and Valpola, 2017; French et al., 2018; Ko et al., 2019).
- *Feature-Based* domain adaptation, which is trying to find a latent space that reduces the difference between source and target domains (Ganchev et al., 2010; Tzeng et al., 2017; Mahadevan et al., 2018; Zellinger et al., 2019; Guo et al., 2020).

2.3 Approaches for Class/Style Transfer

Controlled natural language generation aims to generate realistic sentences while controlling for diffident aspects and attributes of them independent of their content. *Text style transfer* is a special case of Controlled natural language generation, focused on rephrasing a text with desired stylistic attributes while preserving the original intent of the text (Bowman et al., 2016; Hu et al., 2017). For example, while both "absolutely disappointing, he couldn't even trap a bag of cement" and "he cannot control the ball" sentences carry the same meaning, the former has added some *sensational* attribute to the content while the latter is bluntly *informative*.

Our approach, which is based on class transference, shares its core concept with text style transfer. Each class label of greater context could be considered as a stylistic aspect and our goal is to minimally modify the greater context just enough to change its stylistic aspect from one class to another. Therefore, in this section, we review the literature on text style transfer and the computational models that enabled disentangling the style from content and controlling them while regenerating the same content.

2.3.1 Computational Models

Text-based style transfer, in case of availability of parallel data[4], is studied under controlled neural machine translation (NMT) problem (Niu et al., 2017; Sennrich et al., 2016a; Jhamtani et al., 2017). However, the presence of parallel data is limited to a few applications and for the majority of domains, only a set of sentences is available without any correspondence between them. There are some works such as (Iyyer et al., 2018) that used back-translation while enforcing syntactic constraints to generate a pseudo-parallel dataset with sentences with the content but different syntactic structures correspond to each other. Then they used NMT to create a syntactically controlled transfer system. However, such works are a little out of the scope of the style transfer since there is no explicit modulation

[4] A dataset with two (or more) sets of sentences, where a sentence in one set with a certain stylistic attribute (e.g., positive polarity) *correspond* to a sentence in the other set with a similar meaning but different stylistic attribute (e.g., negative polarity).

of stylistic attributes.

What is commonly referred to as text-based style transfer in literature is that categories of style transfer approaches that are trained using *non-parallel data*[5] (Yang et al., 2018; Shen et al., 2017). With exception of a few heuristic attempts with limited applications on style modulation using insertion, deletion, or modification of words and phrases (Reddy and Knight, 2016), the majority of approaches are using *deep generative models*, one way or another.

To briefly formulate the text style transfer problem, let's assume x_1 is a sentence with stylistic attribute y_1 and x_2 is a sentence with stylistic attribute y_2. The goal is to estimate $P(x_1|x_2;y_1,y_2)$ and $P(x_2|x_1;y_1,y_2)$ as the transfer functions between x_1 and x_2. Given the discreet and non-differentiable nature of natural language, unlike vision applications where images are continuous, the transfer functions (i.e., $P(x_1|x_2;y_1,y_2)$) is hard to be learned directly and researchers optimized it through a latent space (Shen et al., 2017; Hu et al., 2018). Thus, the transfer function can be estimated as Equation (2.7), where z is the latent space:

$$P(x_1|x_2;y_1,y_2) = \int_z P(x_1,z|x_2;y_1,y_2)\,dz = \int_z P(z|x_2,y_2)P(x_1|z,y_1)\,dz \tag{2.7}$$

The term $P(z|x_2,y_2)$ can be interpreted as: *to learn a latent representation from a set of sentences with a certain stylistic attribute*, and the term $P(x_1|z,y_1)$ means: *to generate a sentence from the learned latent representation with another stylistic attribute.* Given this formulation, almost all works on text style transfer operate through the two following major tasks:

- *Encode:* learn a latent representation to disentangle the content from the representation of stylistic attributes of the sentence.
- *Decode:* generate realistic sentences with desired stylistic attributes from the learned latent space.

However, different works employ different approaches to *encode* and *decode*. The majority of the works including (Hu et al., 2017; Shen et al., 2017; Yang et al., 2018; Bowman et al.,

[5] A dataset that contains two sets of sentences each set with a different stylistic attribute but *without correspondence* between sentences in two sets.

2016; Mueller et al., 2017) use two main families of the deep generative models to jointly learn the encoding and decoding process:

- *Variational Autoencoders (VAE)* (Kingma and Welling, 2013) and its variations such as Adversarial Autoencoder (Makhzani et al., 2016).
- *Generative Adversarial Network (GAN)* (Goodfellow et al., 2014) and its variations such as InfoGAN (Chen et al., 2016) and SeqGan (Yu et al., 2017).

There are Some other works that learn encoder and decoder through separate processes, such as (Prabhumoye et al., 2018) that uses Seq2Seq (Sutskever et al., 2014) model as the encoder (through back-translation) or (Ficler and Goldberg, 2017) that uses conditioned language model (LM) to learn the latent space, both works use a bi-directional LSTM (Hochreiter and Schmidhuber, 1997) to decode and generate sentences.

Since most of the works on style transfer have some tie to VAE or GAN, and the generative part of our approach is also based on these models, we look deeper into these two families of deep generative models:

Assuming $E : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ be an encoder that learns z and $G : \mathcal{Z} \times \mathcal{Y} \to \mathcal{X}$ be the decoder to generate sentences, the reconstruction loss of an auto-encoder model that can jointly learn encoder and decoder, is as follows, where θ_E and θ_G are the encoder and decoder parameters respectively and the objective of the autoencoder is to minimize the reconstruction loss.

$$\begin{aligned}\mathcal{L}_{rec}(\theta_E, \theta_G) = &\mathbb{E}_{x_1}[-\log p_G(x_1|y_1, E(x_1, y_1))] + \\ &\mathbb{E}_{x_2}[-\log p_G(x_2|y_2, E(x_2, y_2))]\end{aligned} \tag{2.8}$$

However, in order to generate realistic sentences $\hat{x_1}$ and $\hat{x_2}$ that preserve the meaning of x_1 and x_2 respectively, the model has to be constraint, in a way, that the latent space learned from x_1 and x_2 coincides.

The main two families of models that are widely used in text style transfer to provide such constraints to force an autoencoder to generate realistic sentences that belong to the target population of sentences are:

- **VAE** inherits the autoencoder architecture, however, it makes a strong assumption about the prior distribution of the latent space $p(z)$ and sets it to a simple distribution, usually *centered isotropic multivariate Gaussian.* Then it tries to align the posterior distribution of the learned latent space during the encoding phase (i.e., $p(z|x_1, y_1)$ and $p(z|x_2, y_2)$) to the prior probability of the latent space. To do that, it uses KL-divergence (Kullback and Leibler, 1951) to align posteriors to prior as follows :

  $$\mathcal{L}_{KL}(\theta_E) = \mathbb{E}_{x_1}[KLD(z|x_1, y_1)||p(z)] + \mathbb{E}_{x_2}[KLD(z|x_2, y_2)||p(z)]$$

  Then the reconstruction loss of the autoencoder given in Equation (2.8) will be regularized with the KLD loss to form the final objective as in Equation (2.9):

  $$\min_{\theta_E, \theta_G} \mathcal{L}_{rec} + \mathcal{L}_{KL} \tag{2.9}$$

  It is worth mentioning that (Shen et al., 2017) argued that restricting z to a simple distribution, as VAE does, is not a good strategy for non-parallel style transfer since a simple distribution might not be able to recover the effect of the transfer function. However, there are some works such as (Hu et al., 2017) that got impressive results from VAE based models.

- **GAN** employs another technique to force the model to generate realistic samples and instead of aligning the posterior distributions of the latent space with prior, it uses a discriminator to distinguish between distribution of the real data (e.g., $p(x_1)$) and the distribution of the generated data (e.g., $p_G(x_1|E(x_1, y_1), y_2)$). Thus, the discriminator objective is to maximize the difference between two distributions as follows [6]:

  $$\mathcal{L}_{adv_{x_1}}(\theta_E, \theta_G, \theta_D) = \mathbb{E}_{x_1 \sim p(x_1)}[-\log D(x_1)] + \mathbb{E}_{\hat{x_1} \sim p_G(x_1|E(x_1,y_1),y_2)}[-\log(1 - D(\hat{x_1}))]$$
  $$\mathcal{L}_{adv_{x_2}}(\theta_E, \theta_G, \theta_D) = \mathbb{E}_{x_2 \sim p(x_2)}[-\log D(x_2)] + \mathbb{E}_{\hat{x_2} \sim p_G(x_2|E(x_2,y_2),y_1)}[-\log(1 - D(\hat{x_2}))]$$

[6] Given the discreet and non-differentiable nature of natural language, unlike vision applications where images are continuous, the transfer functions cannot be learned and optimized directly, so it has to be learned and optimized through the latent space that needs an auto-encoder so the definition of the GAN I gave here is slightly different from what is originally used in (Goodfellow et al., 2014) and is called adversarial autoencoder (Makhzani et al., 2016)

On the other hand, the model objective is to generate realistic data belonging to the target distribution so that the discriminator cannot distinguish the differences. Thus, the model objective is to minimize the difference so the final objective of the GAN model can be written as:

$$\min_{\theta_E,\theta_G} \max_{\theta_D} \mathcal{L}_{rec} - (\mathcal{L}_{adv_{x_1}} + \mathcal{L}_{adv_{x_2}}) \tag{2.10}$$

2.3.2 Challenges

There are two main challenges in *decoder* part when generating sentences:

- **Discriminator** – how to measure the quality of the generated sentence (i.e., fluency, syntactic/semantic correctness, and having desired stylistic attribute).
- **Continuous approximation** – the discreetness of tokens that are going to be generated, breaks the gradient propagation and makes the optimization of the decoder difficult.

Discriminators are used to guiding the gradient and optimize the decoder to generate sentences more similar to the target distribution. Binary classifiers such as *style classifier* to tell whether a generated sentence contains the desired style is used in some studies including (Hu et al., 2017; Bowman et al., 2016; Prabhumoye et al., 2018). *Real/fake classifier* in used in the line of works that are using adversarial training such as (Shen et al., 2017; Zhang et al., 2016a). Some works including (Yang et al., 2018; Ficler and Goldberg, 2017) used LM as the discriminator.

To provide stronger signals to better optimize the decoder and deal with the discreetness of the language, some works including (Hu et al., 2017; Prabhumoye et al., 2018; Shen et al., 2017) used continuous approximation over *softmax* representation of tokens, some works including (Yu et al., 2017) used REINFORCE algorithm (Sutton et al., 2000) to later finetune the decoder, (Shen et al., 2017) used professor forcing algorithm (Lamb et al., 2016) to match the hidden states of the decoder, while some works including (Yang et al., 2018; Ficler and Goldberg, 2017) used discriminators that can provide a continuous approximation per token, and some other did not tackle the problem and used the discreet signals that lead to generating sentences that seem like a sequence of random words (Bowman et al., 2016).

In addition, the main challenge in *encode* part is to learn a style-independent representation that preserves the meaning so can be used to generate a meaningful sentence. Some works including (Shen et al., 2017; Yang et al., 2018; Ficler and Goldberg, 2017) only relied on the signal from the discriminators (and adversarial training) but some such as (Hu et al., 2017; Bowman et al., 2016; Fu et al., 2018; Mueller et al., 2017) make a strong assumption on the prior distribution of the latent representation to force the content space of the non-parallel sentence to coincide, and (Prabhumoye et al., 2018) used back-translation as encoder, claiming that it preserves the meaning best.

All of the different strategies and considerations to optimize and train *encoder* and *decoder*, in the end, will be realized as adding or removing terms to/from a standard deep generative loss function, and from a higher scope, these works are in some sense all similar.

2.4 Attention Mechanism

In an autoencoder, the encoder is a RNN (e.g., LSTM or GRU), that to process an input sentence $s = \{m_1, m_2, ..., m_n\}$, at each time step t takes two inputs: (i) (vector/soft representation of) a *word* (m_t), and (ii) a *hidden state* from the previous time step (h_{t-1}); then produces the hidden state for the current time step (h_t), and a output vector, which is discarded. After processing all of the sentence words m_n, the last hidden state h_n, which somehow encoded the information of the whole input sequence, would be sent to the decoder as context vector ($\bar{c}$) for producing the output sentence, word by word at each time step (Sutskever et al., 2014), as shown in Figure 1.

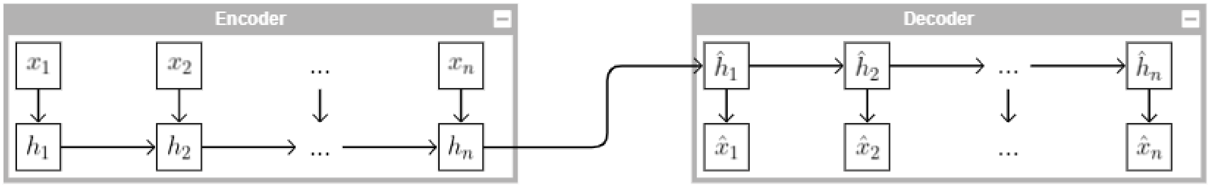

Figure 1: The standard encoder-decoder architecture

However, encoding all the necessary information of the input sentence into a fixed-length context vector makes it difficult to decode long sentences because the individual contribution of each input item to the decoding of each output item would eventually fade away (Cho et al., 2014). The *attention* mechanism proposes a solution to this limitation that instead of encoding the input sequence into a single fixed context vector, encodes it into a context vector pointing to the relevant positions in the input sentence for each output time step (Bahdanau et al., 2015; Luong et al., 2015).

Figure 2 shows the architecture of a standard encoder-decoder model with attention. The encoder first passes all the hidden states ($\{h_1, h_2, ..., h_n\}$), instead of just passing the last one (h_n); these hidden states are each mostly associated with a certain word in the sentence. The attention decoder includes an attention layer (a feed-forward neural network) that is jointly trained with the RNN models. At each time step t, this network learns a score $s_{t,i}$ for each encoder's hidden state (h_i), which corresponds to their relevance to the desired output at that time step. Some popular scoring functions include: additive, (Bahdanau et al., 2015), multiplicative (Luong et al., 2015), and scaled multiplicative (Vaswani et al., 2017) functions.

Next, these scores are softmaxed into attention weight $\alpha_{t,i}$ and multiplied by hidden states h_i to produce the context vector $\overline{c}_t$, as shown in Equation (2.11). The RNN part of the decoder then uses the context vector $\overline{c}_t$ and the generated word from the previous time step $\hat{m}_{t-1}$ to generate the output word $\hat{w}_t$ for time step t. The attention weights are then indicating the relative contribution of the different parts of the sentence (words) to the generation of each output word.

$$\begin{aligned}\alpha_{t,i} &= \frac{\exp(\text{score}(s_{t-1}, h_i))}{\sum_{j=1}^{n} \exp(\text{score}(s_{t-1}, h_j))} \\ \overline{c}_t &= \sum_{i=1}^{n} \alpha_{t,i} h_i\end{aligned} \tag{2.11}$$

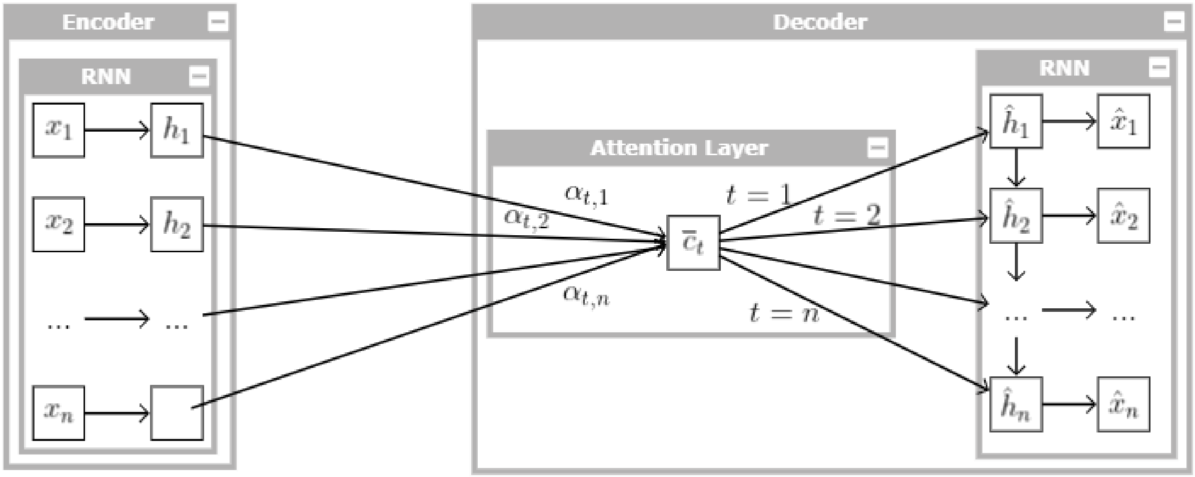

Figure 2: Encoder-decoder architecture with attention

2.4.1 Self-Attention

A conventional encoder-decoder model takes an input sequence (e.g., a sequence of words) and produces a desired set of output (e.g., same words in Persian language or with opposite polarity). Attention is a mechanism that aims to find the related positions in the *input* sequence when generating an *output* word (which input words should contribute to generating an output word) (Bahdanau et al., 2015; Luong et al., 2015). Self-attention (Cheng et al., 2016; Vaswani et al., 2017) fundamentally share the same concept with the attention mechanism, however, it aims to relate each word of a *single* sentence to the different positions of the same sentence (which input words should contribute to represent an input word). Therefore, we might be able to incorporate self-attention weights as a signal to bundle relevant words into a contributing phrase (Q5-1).

Figure 3 shows an example of the correlation between a word and other parts of the sentence, derived from self-attention weights. The current word is highlighted in red and the intensity of the blue shade indicates the self-attention weights for representing the current word. For example, the self-attention weights indicate that the semantic purport of the word "chasing" could be best expressed by paying attention to itself, and just the words "FBI" and "is" and not the rest of the sentence.

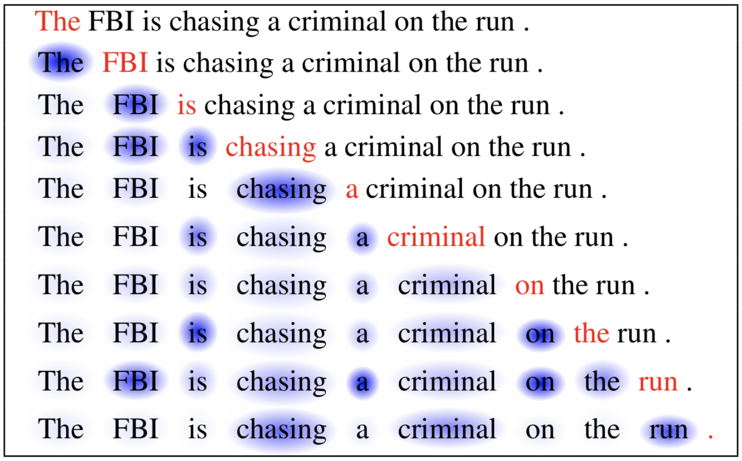
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .

Figure 3: Example of word representation using self-attention (Cheng et al., 2016)

Self-attention involves learning three vectors q, k, and v (sometimes referred to as query, key, and value vectors), for each word of the sentence. Then for word m_i, the score s_i will be calculated by taking the dot product of its query vector q_i with all the key vectors ($\overrightarrow{k}$) and normalized by softmax. Finally, the self-attention vector α_i^s over m_i will be calculated by summing up the dot product of the softmaxed score s_i with all value vectors $\overrightarrow{v}$, indicating how much each other word should be focused to represent m_i, as shown in Equation (2.12).

$$\alpha_i^s = \sum_{j=1}^{n} \frac{\exp(q_i.k_j)}{\sum_{m=1}^{n} \exp(q_i.k_m)} v_j \tag{2.12}$$

2.5 Problem Domains and Resources

In this thesis, we investigate the generalizability of our contrapositive local classification approach to NLP problems with various data availability (see Chapter 4) and annotation (see Chapter 5) profiles by studying three domains with different settings and profiles: *sentiment analysis*, *semantic pleonasm detection*, and *specificity detection*.

2.5.1 Sentiment Analysis

At local level, this task is to identify a word, a phrase, or multiple separated words in a sentence that express strong sentiments. For example, in the sentence “The product itself works perfectly, but their customer service is terrible.”, the words “perfectly” and “useless” are expressing the sentiment of the writer on different aspects of a matter.

Sentiment analysis is a problem domain that has a corresponding sentence-level classification task (does the sentence express *positive*, *negative*, or *neutral* sentiment?), with an easily obtainable large training corpus of labeled sentences. Resources such as Yelp Polarity Dataset (YPD) (Zhang et al., 2015) would be useful for training a transfer function, and Stanford Sentiment Treebank (SST) (Socher et al., 2013), with sentiment annotation at words and phrases level, is making a suitable benchmark for evaluating the sentiment inference at local level.

2.5.2 Semantic Pleonasm Detection

This task aims to find redundant words that are not contributing to the overall meaning of a sentence (Quinn, 1993; Lehmann, 2005). For example, in the sentence: “I received a free *gift*.”, the word “free” may deem semantically redundant at the presence of the word “gift”, which inherently implies being free.

Like sentiment analysis, pleonasm detection has a clear corresponding classification task for the greater context (is the sentence *concise* or *verbose*?), but unlike sentiment analysis, it has a much more limited set of existing resources: NUCLE covers grammatical redundancy (Dahlmeier et al., 2013), and we introduced a small corpus called Semantic Pleonasm Corpus (**SPC**), primarily suitable as a benchmark (Kashefi et al., 2018).

SPC is a collection of three thousand sentences, each sentence features a pair of potentially semantically related words (chosen by a heuristic), and human annotators determined whether either (or both) of the words are semantically redundant.

2.5.3 Specificity Analysis

This task aims to pinpoint the phrases that uniquely relate the sentence to a particular subject (Li and Nenkova, 2015; Lugini and Litman, 2018) and has a wide range of applications in dialogue systems, interactive systems, and educational systems. For example, the phrase "bus accident" in the sentence "10 people killed in bus accident in Pakistan", makes it more specific than the phrase "road accident" in the sentence "10 people killed in road accident in Pakistan."

Similarly, specificity analysis also has a corresponding greater context prediction task (is the sentence conveying a *specific* or *general* piece of information?), however, the existing resources are limited to just a few small corpora (less than 1K sentences) with sentence-level specificity annotation (Louis and Nenkova, 2012; Louis et al., 2013; Tan and Lee, 2014), and the Interpretable Semantic Textual Similarity (iSTS) dataset, which comprises phrase-level specificity annotation, mostly suitable as a benchmark (Agirre et al., 2016).

3.0 The Contrapositive Inference Framework

3.1 Inference Schemes

For many local prediction tasks, there is a corresponding, often easier to learn, global prediction task. Global prediction is relatively easier, in part, because it is attempting to classify a greater context, which is semantically more distinctive than a localized smaller context. Moreover, when a classification task could be performed at multiple levels of granularity (e.g., paragraphs and sentences), there is a much higher chance of having training corpora with label annotation at a larger text span than a smaller text span, which makes the more global version of the task more feasible.

The semantic purport of a larger context is deriving from the co-occurrence of smaller semantic units. This implies that there is a semantic relation between the local and global predictions that could be used to infer the harder-to-learn local prediction from the corresponding easier-to-learn global prediction. It must be noted that prediction inference from the global prediction is only feasible for significantly contributing local features. We define the *direct inference* of local prediction from global prediction as follows:

<u>Direct Inference Scheme:</u>

> ***Global*** → ***Local***: if the global prediction for a larger context be some $Class\ A$, there exists a smaller local portion that significantly influenced the global prediction in the first place so the local prediction for it could infer the same class label as the global prediction.

However, it is reported that the classifiers might sometimes fail to learn the most semantically related features and make predictions based on just salient ones (Ribeiro et al., 2016; Mudrakarta et al., 2018; Jain and Wallace, 2019). We believe adding an extra constraint to the inference, while keeping the relationship between the local and corresponding global prediction intact, can improve the identification of the semantically related local features, therefore, we introduce the *contrapositive inference* of local prediction from the global prediction as follows:

Contrapositive Inference Scheme:

¬Local → ¬Global: the prediction for the local portion of a context could infer the same class label as the global prediction, if, and only if, negating the semantic contribution of that smaller portion, negates the global prediction.

3.2 Adaptation to NLP

In this section, we describe how our contrapositive inference scheme could be applied to the discrete predictions in NLP problems. Assuming $D_{train} = \{(C_1, y_1), (C_2, y_2), ..., (C_n, y_n)\}$ be a collection of N global training examples, where C_i is a global context and y_i is its corresponding class label, the class label of a newly observed instance $C \notin D_{train}$ will be predicted using the function $\mathcal{F} : C \to Y$, which is inferred from the training examples by minimizing the following condition:

$$L_C = \min_{\mathcal{F}} \frac{1}{N} \sum_i \mathcal{L}(y_i, \mathcal{F}(C_i)) \tag{3.1}$$

Now given that C_i, itself, is a collection of smaller local units $C_i = l_{i1}, l_{i2}, ..., l_{in})$, these local segments of a text (l_{ij}) could serve as the features in the training process of the global prediction task, as given in Equation (3.2), where $\mathcal{F}$ is the global prediction function, l_{ij} is the local features within the greater-context C_i, ${l_{ij}}^v$ is some vector representation of the l_{ij} and w_{ij} is its weight, $\mathcal{L}$ is some loss function, and b is the bias value.

$$\begin{aligned} L_C &= \min_{\mathcal{F}} \frac{1}{N} \sum_i \mathcal{L}(y_i, \mathcal{F}(C_i)) \\ &= \min_{\mathcal{F}} \frac{1}{N} \sum_i \mathcal{L}(y_i, \frac{1}{|C_i|} \sum_j w_{ij} {l_{ij}}^v + b) \end{aligned} \tag{3.2}$$

Now, if we can find the local feature l_{ij} that is mainly responsible for making the greater-context C_i belong to the class y_i, we may infer the same class label for that local feature as well. The *direct inference* involves finding the most contributing local features directly from the global prediction function. A straightforward way to implement this scheme is to take the local feature with the *highest weight* as the most contributing one. This approach shares the same concept with the key instance detection in MIL (Liu et al., 2012; Ilse et al., 2018):

$$\frac{\underline{\textit{Direct Inference: MIL(Examine Weights)}}}{\exists\ l_{ij} \mid w_{ij} = \arg\max_{j} \overrightarrow{w_i}} \tag{3.3}$$

Another popular approach to implementing direct inference is to consider the contributing local features as the *rationales* for the global prediction. Rationales are defined as the reason behind the label annotation for the global prediction (Zaidan et al., 2007) and are mostly used to improve the classification of the greater-context (Marshall et al., 2016; Zhang et al., 2016b; Strout et al., 2019; Du et al., 2019). However, some studies have tried to develop systems for automatic extraction of the rationales (Lei et al., 2016; Ehsan et al., 2018) as the smaller text span that could replace the whole global context, while keeping the prediction intact:

$$\frac{\underline{\textit{Direct Inference: Rationale}}}{\exists\ l_{ij} \mid \mathcal{F}(l_{ij}) \approx\ \mathcal{F}(C_i)} \tag{3.4}$$

Alternatively, the contrapositive inference involves finding an adversarial alternative with a negated semantic contribution for local features and reevaluating the global prediction for the resulting greater context as in Equation (3.5), where $\neg l_{ij}$ is the adversarial alternative of the l_{ij}, $C_i^{\neg j}$ is the corresponding greater-context when l_{ij} is replaced with $\neg l_{ij}$. The negated global prediction could be approximated as $\neg y_i \approx 1 - y_i$ for binary classification problems.

$$\frac{\underline{\textit{Contrapositive Inference}}}{\exists\ l_{ij},\ \exists\ \neg l_{ij} \mid \mathcal{F}(C_i{}^{\neg j}) = \neg y_i} \tag{3.5}$$

Implementing contrapositive inference, however, is not as straightforward as direct inference. A bottom-up approach requires calculating an adversarial semantic alternative for each local feature to assess their contribution, which might be very complicated and resource-intensive. Instead of that, we adapt a simpler top-down approach that can be seen as a generalization of machine translation, or as a form of *style transfer* (Bowman et al., 2016; Shen et al., 2017; Yang et al., 2018; Prabhumoye et al., 2018; Zhang et al., 2019).

We aim to develop a *transfer function* (see Section 3.3) that rewrites an arbitrary greater-context (e.g., a sentence) known to be in one class (e.g., *Class A*) into a corresponding text in another class (e.g., *Class B*); the smaller local parts (e.g., words or phrases) that changed during this global rewriting and class transference process are deemed to be the ones that contribute the most to the greater-context's class label, so may infer the same class label as well.

As an illustrative example, consider the sentiment classification problem. The transfer function might rewrite a *positive* sentence: "the food was great" into a *negative* sentence "The food was awful." While the overall sentence length is the same, the learned transfer function chose to replace "great" with "awful," therefore "great" is likely to be a sentiment expressing word. By casting the problem as a style transfer task, we avoid the thorny tasks of quantifying fluency and meaning retention and even if the transfer function does not provide a correct semantic negation, for example, rewrites the "great" as "cold" instead of "awful", we are not concerned with whether "The food was cold" is a meaningful or even fluent sentence; we only care that this new sentence is now classified as *negative*, so it reveals something about the word "great" in the old sentence. This property may also improve the robustness of our approach to some level of noise in the data.

Our proposed contrapositive inference approach is still a complicated method. By relaxing the global prediction negation requirement of the contrapositive inference into maximum prediction deviation, a lighter and easier-to-implement version of it could be seen as a *leave one out (LOO)* baseline. That is, if a local feature be a significant contributor to the global prediction, then removing it from the greater-context should cause a larger prediction deviation from the original prediction as follows, where C_i^{-j} is the corresponding greater-context when l_{ij} is removed:

Semi-Contrapositive Inference: LOO

$$\exists\ l_{ij} \mid \max_{j} \|\mathcal{F}(C_i) - \mathcal{F}(C_i^{-j})\|^2 \tag{3.6}$$

Both direct and contrapositive inference schemes are relaxing the local prediction requirement of having a large training corpus with local annotation into having training corpus with global annotation, which is easier to obtain, thus, they could be beneficial for new or narrowly focused NLP applications for which localized training data is not widely available. However, it must be noted that the localized prediction inference is only applicable to certain classification problems, where the prediction task can be performed in multiple levels of granularity and the local prediction has a corresponding global prediction task in a greater context.

3.3 Transfer Function Model

Since our approach is based on the difference between original and generated greater-contexts, extensive or random modification of the original text might result in meaningless differences, so the key requirements of an ideal transfer function for our approach are:

- **Req. 1.** preserve most of the original content and only make minimal changes.
- **Req. 2.** these minimal changes negate the prediction for the resulting greater-context (makes it belong to the other class).

Any reasonable transfer function that satisfies these requirements may make a suitable model to serve as the core of our proposed approach. As mentioned in Section 2.3.1, we can incorporate a GAN-based approach to train an adversarial transfer function (Chen et al., 2016; Yu et al., 2017; Shen et al., 2017), or a VAE-based (Bowman et al., 2016; Yang et al., 2018; Mueller et al., 2017).

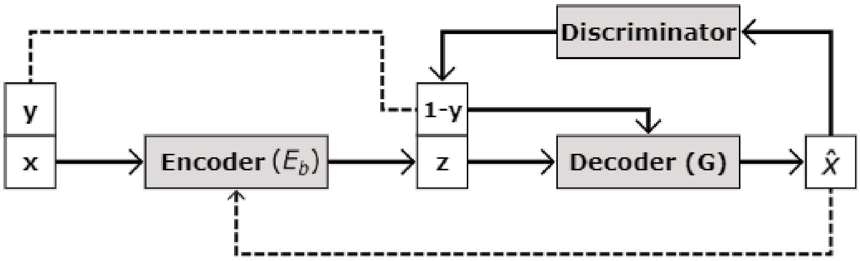

Figure 4: The transfer function model diagram

As an implementations choice, inspired by Hu et al. (2017) and Prabhumoye et al. (2018), we adapt an autoencoder-based architecture as the base of our transfer function model, However, we do not explicitly align the distribution of the original and generated sentence to some prior distribution (Kingma and Welling, 2013), or directly to each other, for example by minimizing the KL-Divergence (Kullback and Leibler, 1951) the instances of *Class A* and instances of *Class B*. We do not however explicitly align the distribution of the original (e.g., *Class A*) and generated (e.g., *Class B*) sentences to some prior distribution (Kingma and Welling, 2013), or directly to each other (for example by minimizing the KL-Divergence between distributions). Instead, we rely on the self-supervised structure of the autoencoder to learn a class-agnostic coinciding latent space for original sentences in one class and their corresponding transferred versions in the other class. As a result, we can develop a transfer function by using only one autoencoder. This is computationally more efficient than alternative models that involve explicit distribution alignment.

Figure 4 shows the overall structure of our transfer function model. Our transfer function model incorporates an encoder, a decoder, and a discriminator for the global prediction. We opt to use a gated recurrent unit (GRU) as the encoder, a bi-directional GRU as the decoder, and a 1D CNN classifier as our discriminator modules, in most of our experiments in Chapter 4. Later in Chapter 5 (Section 5.2.1), we discuss how these RNN units can be replaced by transformers to exploit the self-attention weights to operate on the phrasal representation of the sentences.

The autoencoder is initially trained to learn a latent space (z) from the input greater-context instances (c) by minimizing the reconstruction loss, given in Equation (3.7), where θ_E and θ_G are the encoder and decoder parameters, respectively.

$$\mathcal{L}_{rec}(\theta_E, \theta_G) = \mathbb{E}_{q_E(z|c)}[\log p(c|z, y)] \tag{3.7}$$

Since the latent space z is learning independent of the class labels ($q_E(z|c)$), it could be used to generate instances of both classes. Therefore, the decoder is trained to learn to generate a greater-context $\hat{c}$, given a desired class label ($p(\hat{c}|z, y)$). Now, if we ask the encoder to generate a rewriting alternative for an input greater-context with a flipped class label ($\neg y = 1 - y$), then the generated output would likely be a modified version of the input with some small changes that make it belong to the opposite class (Req. 2). Additionally, in order to learn class transference, the decoder also requires a signal to determine how likely the generated output belongs to the desired class. To provide that signal, we incorporate a class discriminator in our transfer function model, which is optimized as given in Equation (3.8), where θ_D is the discriminator parameter.

$$\mathcal{L}_{\text{disc}}(\theta_D) = \mathbb{E}_C[\log q_D(y|c)] \tag{3.8}$$

As we can see, the discriminator is trained on the input training examples (c) and their associated labels (y), but it is intended to predict a class label for the generated examples ($\hat{c}$), resulting in a noisy global classification. Moreover, local classification depends on classifications at a global level, the discriminators' errors could also propagate to decoder generation, causing an invalid class transference and consequently a wrong local prediction inference. To address this issue, researchers are incorporating different techniques to train a more robust discriminator, including using teacher-forcing (Williams and Zipser, 1989) or professor-forcing (Goyal et al., 2016) algorithms instead of the traditional back-propagation algorithm (Rumelhart et al., 1986) to train a RNN discriminator, using beam search across the predicted probabilities for each word to generate a number of likely candidate output sequences (Bengio et al., 2015), or incorporating bootstrapping (Reed et al., 2015) and regularizing for minimum entropy (Grandvalet and Bengio, 2004).

In this work, however, we use a simple strategy of training the discriminator only with the input sentences (c) and then using it to classify the resulting sentences ($\hat{c}$). Equation (3.9) shows our transfer function's final objective function, where λ_D is the balancing parameter. To avoid the fading gradient phenomenon when training on discrete space, the model feeds with Gumbel-Softmax (Jang et al., 2017) representation over the words to provide a continuous approximation and a stronger signal for optimizing the decoder.

$$\underbrace{\min_{\theta_G} \mathcal{L}_{gen}}_{\text{Generation Loss}} = \underbrace{\mathcal{L}_{rec}}_{\text{Reconstruction Loss}} + \lambda_D \underbrace{\mathcal{L}_{disc}}_{\text{Discriminator Loss}} \tag{3.9}$$

For some initial training runs, we train the discriminator and autoencoder independently ($\lambda_D = 0$); the discriminator will learn to distinguish between classes, and the autoencoder will learn a latent space that could be used to regenerate instances of both classes. After that, for a few more epochs, we jointly train the autoencoder and discriminator ($\lambda_D \neq 0$) to learn the class transference (replace y with $\neg y = 1 - y$). The hyperparameters we used in our experiments are described in Appendix A.1, Appendix A.2, and Appendix A.3.

3.4 Question of Generalizability

In theory, the proposed contrapositive inference scheme should be applicable to many local prediction tasks because the core of our approach is domain-independent. However, to train an appropriate transfer function we need a corpus of training examples for the global prediction, which might not be readily available in many problem domains. Moreover, the relative size of the local and global context and the annotation profiles of the prediction task and domain may pose different challenges and limitations for the inference framework. Therefore, in this thesis we study the generalizability of our proposed inference framework to problem domains with varying data availability profiles and different levels of inference granularity.

3.4.1 Data Availability Perspective

As with many other generative neural models, developing a reasonable transfer function also requires a large training corpus of global examples. For many scenarios, this data requirement may not be an insurmountable problem and such resources already exist. However, there are many cases in NLP where the prediction task does not have available to it a large quantity of annotated corpora, even at global level. Base on the availability of the labeled greater context examples, problems can be classified into one of the four categories:

- **D-P1.** Sizable training corpus with labeled examples for greater context prediction is available;
- **D-P2.** Sizable training corpus with labeled examples for greater context prediction is not available;
- **D-P3.** A small training corpus with labeled examples for greater context prediction is available;
- **D-P4.** Sizable training corpus with labeled examples for greater context prediction is not available, but some unlabeled data is available.

In Chapter 4, we investigate how low-resource NLP techniques (refer to Section 2.2) may facilitate the generalizability of our approach to problems with different data availability profiles, by studying different problem domains with caring data availability profiles: *sentiment analysis* (refer to Section 2.5.1), *semantic pleonasm detection* (refer to Section 2.5.2), and *specificity detection* (refer to Section 2.5.3).

3.4.2 Inference Granularity Perspective

Our contrapositive inference scheme aims to infer a local class label from the corresponding global label. However, the scope of the local and corresponding global contexts may pose different challenges for our proposed approach. For example, it might be practically impossible to associate the class prediction for a document with the presence of a single word; a clear corresponding global prediction task at sentence level is not available for predicting the parts of speech (POS) of words or identifying named entities. In Chapter 5, we investigate the generalizability of the contrapositive inference scheme to different levels of inference granularity:

- **I-P1. Sentence to word inference**, where a *single word* is the major local scope that influenced the class label of the global *sentence*.
- **I-P2. Sentence to subsentence inference**, where the class prediction of the global sentence is influenced by multiple local words and phrases.
- **I-P3. Paragraph to sentence inference**, where a sentence is a rather larger local contributing context to the global class prediction of the paragraph.

4.0 Data Profile Generalizability

In this chapter, we investigate the applicability of our proposed inference scheme to NLP problems with different data availability profiles[1] (refer to Section 3.4.1) to test for the third hypothesis of the thesis (**H3**). However, the first hypothesis of this thesis (**H1**) is that inference by contraposition allows for the identification of semantically relevant local features. Moreover, we argue that this inference scheme can be efficiently implemented as a transfer function (**H2**). To test for these hypotheses, in Section 4.2, I aim to answer the following key question by studying a case in which the availability of a large corpus of global training (D-P1) allows us to train our vanilla transfer function model from scratch:

Q1. *To what extent does the contrapositive inference scheme and the transfer function model outperform alternative methods under similar data requirement?*

In Section 4.3, we investigate the applicability of our approach to the problems for which a training corpus of global prediction is not available (D-P2) and discuss how *data augmentation* can reconcile the training requirement of our transfer function. Thus, our research question would be:

Q2. *How does data augmentation impact the performance of the transfer function?*

In Section 4.4, we explore how *supervised transfer learning* can facilitate the generalizability of our transfer function model to problems with limited availability of training examples (D-P3). The main question we aim to answer in this section is:

Q3 *To what extent does the contrapositive relationship between the local and global prediction tasks transferable across domains?*

In Section 4.5, we argue that utilizing unlabeled data can improve the learning process of the transfer function in a new domains. Therefore, aiming to answer the following research question, we propose a *domain adaptable* transfer function model approach to enhance the

[1] As we proceed through this thesis, the *"data availability"* refers to the extent to which the training data is available for global prediction. We make a firm assumption that: no class label information is available at local scope.

Table 1: Our strategies for local prediction problems with different data availability profiles

Data Availability Profile	**Approach**
D-P1: Available	Our Vanilla Transfer Function
D-P2: Unavailable	+ Data Augmentation
D-P3: Limited	+ Supervised Transfer Learning
D-P4: Unlabeled	+ Unsupervised Domain Adaptation

applicability of the contrapositive inference scheme to problems domains, where only some unlabeled examples are available (D-P4).

Q4. *To what extent does the contrapositive relationship between the local and global prediction tasks adaptable to a new domain, without using any labeled data?*

Table 1, summarizes our approaches for generalizing our model to each data availability profile and our progress status towards them.

4.1 Evaluation Methodology

In order to test for the hypothesis of the thesis (see Section 1.2), I propose several research questions. Aiming to answer these questions, we experiment on a few problem domains (see Section 2.5) with different levels of data availability (see Section 3.4.1 and current Chapter) and inference granularity (see Section 3.4.1 and Chapter 5) settings, as case studies for problems with similar profiles. Through these experiments, we compare the local prediction precision of the different inference schemes and discuss how these results can help us answer our research questions and validate the hypothesis. To ensure fair evaluation, all of these models are tuned on the source and target domains in each experimental setting. The details of our hyperparameter tuning are given in Appendix A.1, Appendix A.2, and Appendix A.3.

- **Direct:MIL.** Inspired by Pappas and Popescu-Belis (2014) and Ilse et al. (2018), this baseline implements an attention-based direct inference scheme in the forms of multi-instance learning (MIL) (refer to Section 2.1.3). For the global prediction, we train a CNN classifier and wrap the weights at each time-step into an attention weight using a multiplicative attention layer (Luong et al., 2015). The local feature with the highest attention weight would then receive the same class prediction as the global class label.
- **Direct:Rationale.** For this baseline we use the model proposed by Lei et al. (2016), as an implementation for the direct inference scheme as the rationales for the global prediction (refer to Section 2.1.4). This model incorporates a RCNN generator and an encoder to predict rationales.
- **Contrapositive:LOO.** This baseline serves as a light version of the contrapositive inference of the local prediction through examining the global prediction for different perturbation of a greater-context by leaving one of its local features out. The local feature that causes the highest deviation in global prediction would infer the original global class label, as given in Equation (3.6). Similar to the *Direct:MIL* baseline, we use a CNN classifier for the global prediction task.
- **Contrapositive:TF.** This model serves as an implementation of our proposed contrapositive inference scheme, as given in Equation (3.5). We use our transfer function model (Section 3.3) for contrapositive local prediction inference through rewriting and class transference of the greater context.

As the evaluation metric in our experiments,we calculate the precision of predicting the correct local context as follows, where $Prediction$ refers to the set of local scopes (e.g., words) that are predicted as the most contributing features by the model, and *Gold* is the set of local scopes with the gold label. This metric rewards the predictions (e.g., words and phrases) within the rationale span and penalizes those outside the rationale span, making it suitable for local prediction problems with varying scopes of rationales, including single-word and subsentential rationales (Zaidan et al., 2007; Lei et al., 2016).

$$Precision = \frac{|Prediction \cap Gold|}{|Prediction|}$$

4.2 Our Vanilla Transfer Function

In this section, we are investigating the answer for Q1 – *to what extent does the contrapositive inference scheme and the transfer function model outperform alternative methods under similar data requirements?*. We set up an experiment to evaluate the prediction inference precision of our model at word-level (local scope) when a large training corpus of labeled sentences (global scope) is available. We opt to study the **sentiment analysis** domain (refer to Section 2.5.1), as case for problem domain with readily available large training corpora of labeled sentences (D-P1). In this domain the different levels of prediction tasks are:

- **Global Prediction Task:** *predict whether a* sentence *has a "positive" or "negative" polarity?*
- **Local Prediction Task:** *which* word(s) *of the sentence expresses a strong sentiment?*

Yelp Polarity Dataset (YPD) (Zhang et al., 2015) is a collection of highly polar restaurant reviews written by Yelp users. Each sentence in this dataset is associated with a polarity label, which makes it suitable for global binary sentiment classification at the sentence level. YPD is large enough to train our transfer function from scratch, so we chose 105K sentences with 30 words or fewer as our training set, which we refer to as YPD-Train. We sampled almost the same amount of *positive* and *negative* examples in YPD-Trains to build a class-balanced training dataset.

As the benchmark dataset to evaluate the local inference performance of the models, we manually annotated some held-out sentences of YPD with word-level sentiment labels. We followed the same annotation scheme proposed by (Socher et al., 2013) and created a unigram pool from all the words of the test set; then randomly picked a word and annotated it as *very positive*, *positive*, *neutral*, *negative*, or *very negative*. Finally, filtering for sentences that contains a *very positive* or *very negative* words, we collected a set of 860 held-out sentences

Table 2: Local sentiment prediction precision of different inference schemes, trained on naturally-occurring labeled examples

Approach	**Local Inference Precision**
Direct:MIL	58.4%
Direct:Rationale	63.3%
Contrapositive:LOO	66.2%
Contrapositive:TF	**77.5**%

with word-level labels as our in-house benchmark dataset, which we refer to as YPD-Test.

Table 2 compares the precision of different inference schemes on predicting the word(s) that express the same sentiment as the sentence in which it is used, based on the gold annotation in the benchmark dataset (YPD-Test). As shown, the contrapositive inference methods, including LOO, are more effective at inferring the sentiment class label of the word(s) than the direct inference alternatives, and our proposed method (*Contrapositive:TF*) substantially outperformed the best direct inference method (*Direct:Rationale*) by more than 14%. This huge performance difference can provide a clear answer to Q1. It suggests that controlling for the contrapositive contribution of local features to the corresponding global prediction can reveal their semantic relationship more robustly, therefore, leading to a more accurate local class prediction inference, as compared to direct inference alternatives.

In our experiment, we observed that direct inference schemes often predict semantically irrelevant words, such as "punctuation marks", as a major local contributor to the global predictions. Contrapositive methods, however, disentangle semantics from surface representations and apply local contrapositive semantic perturbations to the global context. This makes the identification of semantically related local and global contexts more robust to noise. For example, a punctuation mark might influence a class prediction for a sentence, however, replacing it with another punctuation mark will not make the sentence belong to another class, so it is not a *semantically* contributing word.

4.3 Data Augmentation

In this section, we aim to answer the Q2 – *how does data augmentation impact the performance of the transfer function?*, by investigating how data augmentation may reconcile the training requirement of our transfer function model and facilitate its applicability to the problem domains, in which a large training corpus of global prediction is not available (D-P2). For these resource-constrained problems, we propose a *non-label-preserving data augmentation scheme* and based on that, we explore some domain-inspired augmentation heuristics for several problem domains (Section 4.3.1). Since there might be many augmentation heuristic options available, we propose an approach for efficiently assessing the suitability of the heuristic functions for different classification tasks (Section 4.3.2). We then chose the most suitable heuristic(s) and investigate how the resulting augmented datasets may train an appropriate transfer function in those domains (Section 4.3.3).

4.3.1 Augmentation Heuristics

As mentioned in Section 2.2.1, data augmentation has been shown to be an effective way to achieve higher performance in classification tasks by generating more training examples. Textual data augmentation approaches usually assume that a (small) set of labeled data with instances from all of the classes is available and they apply some heuristic(s) to the instances of *Class A* to create more examples for *Class A*. However, these label-preserving augmentation approaches may fail to construct an augmented dataset from scratch.

We propose a non-label-preserving data augmentation scheme that removed the assumption of having labeled samples of all class instances (Kashefi and Hwa, 2020): we begin by identifying a set of sentences that are likely to belong to *Class A*, based on domain knowledge; then, by applying the heuristics, we create the *Class B* examples from the *Class A* instances. There may be multiple heuristics that allow us to create samples of one class from samples of another class, and the choice of heuristic can impact the success of the upstream application. Intuitively, a good heuristic should aim to create near-miss examples (samples of *Class B* "hard to distinguish" from *Class A*).

In this section, we go over some non-label-preserving heuristic options for augmenting training corpora for three problem domains: sentiment analysis, semantic pleonasm detection, and specificity detection.

4.3.1.1 Sentiment Corpus Augmentation

To augment a dataset for the sentiment analysis domain (see Section 2.5.1), we use the Yelp Polarity Dataset (YPD) as our data source to create the augmented dataset. Our overall strategy is to replace the sentiment words of a sentence with their antonyms to create the sentence with an opposite sentiment label. We apply the following heuristics to the positive sentences to create our augmented negative examples, and the other way around for generating the augmented positive examples:

- **ALL.** In this heuristic, we replace all the sentiment words of a sentence with their antonyms. To find the sentiment words, we first collected a vocabulary of positive and negative unigrams by combing the labeled words of *Stanford Sentiment Treebank* (Socher et al., 2013) and the *Opinion Lexicon* (Hu and Liu, 2004). This results in a vocabulary of 3,453 positive and 6,000 negative unigrams.

  Then, for a positive sentence in YPD, we replace every word of it that appeared in the positive portion of the collected vocabulary with one of its randomly chosen antonyms, using WordNet (Miller, 1995), to create the augmented negative sentence. We perform similarly but in the opposite direction to create the augmented positive sentences.
- **ONE.** In this heuristic, instead of replacing all sentiment words with their randomly chosen antonym, we first filtered for antonyms that match the POS and sense of the sentiment word, then we pick the antonym that makes the most fluent augmented sentences, ranked by a language model (LM) trained on YPD. Finally, for every sentence, we only replace one of its sentiment words with its POS, sense, and LM filtered antonym.

  Using this heuristic, for example, a sentence with overall positive polarity may still contain a word that expresses a negative opinion about an aspect, so intuitively, this creates "harder to distinguish" examples compared to the *ALL* heuristic.

Table 3: Examples of data augmentation using the sentiment heuristics

Heuristic	**Sentence**	**Label**
None	super generous portion !	positive
ALL	~~super~~ **+lousy** ~~generous~~ **+meager** portion!	negative
ONE	super ~~generous~~ **+meager** portion !	negative

Overall, we generated 50K positive and 50K negative augmented samples using each heuristic. We removed all of the original YPD sentences so that these datasets contain only augmented samples. We refer to each dataset with the same name as the heuristic function it is augmented with. Table 4 shows examples of sentences augmented using the label-preserving and non-label-preserving sentiment heuristics.

4.3.1.2 Pleonasm Corpus Augmentation

To augment a dataset of concise and verbose sentences for the semantic pleonasm detection domain (see Section 2.5.2), we start by trying to identify an existing real-world data source that has similar characteristics as the target domain. Since the Semantic Pleonasm Corpus (SPC) (Kashefi et al., 2018), which we further will use as an evaluation dataset in this work, is developed over the Yelp Dataset Challenge, we consider using the same data source as well.

One domain-specific feature of Yelp that we exploit is the data category called "tips." Since "tips" are very short sentences, they are likely to be *concise*; we sample for "tips" that contain adjectives because the evaluation corpus mainly focuses on adjectival pleonasms. Now based on domain knowledge, we need to come up with heuristics to create verbose samples based on the collected "concise" sentences. Our overall strategy is to create a verbose sentence by adding a superfluous adjective to the concise sentences using the following heuristics:

- **Duplicate (DUP).** This heuristic is an obvious case for word redundancy by duplicating an adjective word of the sentence right next to itself.
- **Synonym (SYN).** This heuristic inserts a synonym next to an adjective word of the sentence. The conventional way to get synonyms of a word is to use WordNet, however, since these synonyms may express a different quality of the noun clause compared to the original adjective, augmented construction might not be semantically redundant.

  For this reason, we opt to use sense2vec (Trask et al., 2015), a contextual word-embedding fine-tuned on Yelp "tips". Since the adjective synonyms from sense2vec are matching the context and follow the same intent and emotional state of the original adjective, these two adjacent synonyms are likely to make a pleonastic construction.
- **Near-Miss Negative (NMN).** In this heuristic, we try to create **concise** examples that are "hard to distinguish" from the verbose examples. We trained a language model on the Yelp "tips" and used that to predict the most likely words that can occur right after an adjective of the sentence. Let assume for adjective w_{adj} in sentence s, using LM, we retrieved $\{w_{aug1}, w_{aug2}, ..., w_{aug5}\}$ as a sorted list of most likely words that can appear next to w_{adj} given its context s.

  We then filter for w_{aug}s that are adjective themselves and a synonym of w_{adj}, lets assume the filtered list be $\{w_{aug2}, w_{aug5}\}$. Since LM is trained on Yelp, the w_{aug2} is already observed in the Yelp tips after the w_{adj} in some context. Taking into account that Yelp tips are considered concise, the sequence of $... w_{adj}\, w_{aug2} ...$ is also concise. Therefore, we can create concise examples that are containing two adjacent synonyms but are **not** verbose

For each heuristic, we generate only one augmented verbose sample from an original concise sentence. In total, we augmented 100K concise and 100K verbose samples using each heuristic. Since the verbose examples are generated from concise sentences that are included in the augmented corpus, we removed the concise sentences with odd and verbose sentences with even indexes to make sure that non of the concise are verbose sentences in the corpus are corresponding to each other. The final augmented corpus, thus, contains 50K non-parallel samples of each class. We refer to each dataset with the same name as the heuristic function that was used to augment it.

Table 4: Examples of data augmentation using the pleonasm heuristics

Heuristic	**Sentence**	**Label**
None	delicious bread !	concise
DUP	delicious +**delicious** bread !	verbose
SYN	delicious +**tasty** bread !	verbose
NMN	delicious +**redolent** bread !	concise

Table 4 shows examples of sentences augmented using the non-label-preserving pleonasm heuristics. While duplicating the word “delicious“ or adding “tasty” next to it makes the sentence verbose, adding “redolent” does not make it verbose because “redolent” and “delicious“ are describing a different quality of the “bread.”

4.3.1.3 Specificity Analysis

Our benchmark dataset in the specificity detection domain (see Section 2.5.3) is the news headline part of the Interpretable Semantic Textual Similarity (iSTS) (Agirre et al., 2016). Therefore, we are using the sentences of the “all the news[2]” corpus, as the augmentation data source with similar characteristics as the iSTS.

We start by (weakly) labeling the sentences with named-entities as “specific” and sentences with no named-entity as “general.” Now based on domain knowledge, we need to come up with heuristics to create more samples based on the collected weakly-labeled sentences. Our overall strategy is to replace named-entities with their *hyponym* or *hypernyms* to create the sentence with an opposite specificity level. We apply the following heuristics to the sentences to create our augmented examples:

- **ALL.** In this heuristic, we replace all named-entities of a sentence with their *hyponyms* or

[2] `www.kaggle.com/snapcrack/all-the-news`

Table 5: Examples of data augmentation using the specificity heuristics

Heuristic	**Sentence**	**Label**
None	10 *people* killed in *camp*	general
ALL	10 ~~people~~ +**children** killed in +**death** camp	specific
ONE	10 people killed in +**death** camp	specific
None	*Rouhani* wants *nuclear deal* !	specific
ALL	~~Rouhani~~ +**President** wants ~~nuclear~~ deal !	general
ALL	~~Rouhani~~ +**President** wants nuclear deal !	general

hypernym. We use SpaCy[3] (Honnibal and Montani, 2017) to identify the named-entities; then, we use WordNet (Miller, 1995) to retrieve a hypernym for each named-entity and substitute them to augment a *general* sentence from a *specific* sentence;

For weakly-labeled *general* sentences, we lookup every noun of the sentence in the WordNet and replace them with a hyponym, in case they have one, to get an augmented *specific* example.

- **ONE.** This heuristic is similar to the previous one, except, this time we chose the named-entity with the longest hypernymy tree length and substitute that to get an augmented *general* example.

  Similarly, to generate a specific augmented sentence, we substitute the noun with the longest hyponymy tree with a hyponym in a general sentence.

For each heuristic, we generate only one augmented sample from an original sentence. In total, we augmented 110K augmented examples, including about the same portion of general and specific sentences. Table 5 shows examples of sentences augmented using the non-label-preserving pleonasm heuristics. While duplicating the word “delicious“ or adding “tasty” next to it makes the sentence verbose, adding “redolent” does not make it verbose because “redolent” and “delicious“ are describing a different quality of the “bread.”

[3] https://spacy.io/

4.3.2 Quantification of Heuristics Suitability

Given a classification task in general, there may be multiple heuristics and data augmentation approaches that allow us to transform existing samples into new ones, but the choice of heuristic may significantly impact the success of the task. While there is a large body of research on data augmentation for NLP applications (refer to Section 2.2.1), to our best knowledge, there is no guideline for how to choose between different textual augmentation approaches for a task. Therefore, in this section, we aim to answer the following auxiliary question by proposing a low-cost approach to quantify the evaluation of different augmentation heuristics:

AQ1. *Which heuristic and data augmentation approach is more appropriate for a classification task?*

A straightforward approach to assess which heuristic and data augmentation approach is more appropriate for the task is to try every heuristic to generate an augmented dataset, then train a classifier on each and check the final classification performance (Qiu et al., 2020; Wei and Zou, 2019). The training process in this brute-force approach, however, may be time-consuming and resource-intensive, especially in complex training scenarios. Alternatively, we may try to identify qualities that make a heuristic effective. Intuitively, a good heuristic ought to generate augmented samples that are the most similar to the original data distribution. However, this approach may overlook the additional generalization benefit that may come from diverse augmented training examples. Moreover, this approach may not be possible for problem domains with limited resources, where original labeled data is not available for all classes, and one may have to use non-label-preserving heuristics to augment examples for all classes.

From the classification task perspective, however, a good heuristic should aim to generate near-miss examples (samples of class B hard to distinguish from A). We believe, the "hard to distinguish" samples can be quantified by finding a way to compute the difference between the samples of different classes, to sever as a guideline for choosing between different heuristic approaches.

Let us assume samples of class A are drawn from distribution A, which should be different from the distribution B that samples of class B are drawn from. The difference between distribution A and B can be calculated as the KL-divergence (KLD) (Kullback and Leibler, 1951) from B to A as: $D_{KL}(A||B)$. KLD calculates how probability distribution A is different from the reference probability distribution B as the amount of information gained if samples of B are used instead of samples of A. Thus, a lower $D_{KL}(A||B)$ means distribution A is more similar to distribution B, so samples of class A are *harder to distinguish* from samples of class B. Therefore, the extent to which "hard to distinguish" samples can be generated by heuristic h could be quantified as $D_{KL}(A_h||B_h)$, where A_h and B_h indicate the samples of class A and B augmented using heuristic h, and Equation (4.1) could be used to identify which heuristic is generating "harder to distinguish" samples and so more suitable for the classification task.

$$\arg\min_{h} D_{KL}(A_h||B_h) \tag{4.1}$$

Finally, to transform sentences from their discrete word representation into a continuous distribution representation, we utilize a few of the numerous pre-trained embeddings that nowadays are the de facto approach for encoding sentences into vector space (Cho et al., 2014; Le and Mikolov, 2014; Cer et al., 2018; Devlin et al., 2019).

We examine the applicability of our approach by studying *sentiment* and *pleonasm* classification tasks at global and local levels. To validate our approach for quantifying the evaluation of heuristic textual data augmentation methods, we investigate the answer for the key question of this section (AQ1) from the three following dimensions:

- **AQ1-1.** *Can generating "hard to distinguish" examples be an effective way to assess whether a heuristic is generating a suitable augmented training dataset?*
- **AQ1-2.** *To what extent could the notion of "hard to distinguish" examples be quantified by our metric – the difference between the class distribution of the augmented samples?*
- **AQ1-3.** *Is calculating the distribution difference computationally efficient in practice?*

In order to explore the answer for in answering AQ1-1, to measure the accuracy of the classification task in a more general setting (greater context or sentence level), we trained a

BiLSTM (Liu et al., 2016) and a *CNN* (Kim, 2014) classifier on each the augmented dataset. The classification result for each task and augmented dataset is reported in Section 4.3.2.2. The BiLSTM and CNN models are trained on augmented corpora separately for each task; the sentiment classifiers are evaluated on a held-out portion of the YPD, and the pleonasm classifiers are evaluated on SPC. None of the sentences of the held-out YPD and SPC are used during the creation of the augmented datasets.

To answer AQ1-2, we use two pre-trained encoder models: Universal Sentence Encoder (*USE*) (Cer et al., 2018) and Bidirectional Encoder Representations from Transformers (*BERT*) (Devlin et al., 2019), both of which are transformer-based encoders of greater-than-word length text, to transform the sentences into a continuous space so that we can treat them as class distributions and measure their similarity.

4.3.2.1 Classification Accuracy

If a good heuristic is the one that generates "hard to distinguish" examples, the dataset augmented using *ONE* should train a better classifier than *ALL* for the sentiment analysis task, and the pleonasm classifier trained on *NMN* should outperform the classifiers trained on *SYN* and *DUP*.

Table 6 and Table 7 shows the classification accuracy of the neural models trained on different augmented datasets for sentiment and pleonasm prediction tasks, and as we expected, heuristics that intuitively generate "harder to distinguish" examples are more suitable for the prediction task and trained a better classifier on both tasks:

$$\underline{\text{Sentiment Classification Accuracy}}$$

$$ACC(ONE) > ACC(ALL)$$

$$\underline{\text{Verbosity Classification Accuracy}}$$

$$ACC(NMN) > ACC(SYN) > ACC(DUP)$$

Table 6: Sentiment classification accuracy and distribution difference of augmented examples

Augmented Dataset	Model	ACC	KLD	
			USE	BERT
ALL	BiLSTM	.683	26.26	13.97
	CNN	.716		
	AVG	.699		
ONE	BiLSTM	.808	**17.41**	**9.90**
	CNN	.822		
	AVG	**.815**		

Table 7: Pleonasm classification accuracy and distribution difference of augmented examples

Augmented Dataset	Model	ACC	KLD	
			USE	BERT
DUP	BiLSTM	.393	14.86	15.90
	CNN	.442		
	AVG	.417		
SYN	BiLSTM	.526	10.23	12.99
	CNN	.551		
	AVG	.538		
NMN	BiLSTM	.692	**8.91**	**7.77**
	CNN	.738		
	AVG	.715		

These observations, along with the results from *Case Study 4.3.3* for more complex training tasks, suggest that an augmented dataset generated from a heuristic that produces “harder to distinguish” examples for different classes could train a better classifier (AQ1-1).

4.3.2.2 Augmented Distribution Difference

To investigate the extent to which “hard to distinguish” examples might be quantified as a difference between the distribution of the augmented samples of different classes, we first encode the augmented sentences into a continuous high dimensional vector space; then, we computed the difference between the distribution of the augmented samples of different classes as the divergence from a high dimensional representation of one class to another.

For the sentiment analysis task, we computed the difference between augmented positive and negative distribution as follow, where E is either BERT or USE encoders, and *positive* and *negative* indicate augmented positive and negative examples respectively:

$$D_{KL}(E(positive)\|E(negative))$$

The distribution difference for the pleonasm analysis task is calculated as follow, where *concise* and *verbose* indicate augmented concise and verbose examples respectively:

$$D_{KL}(E(concise)\|E(verbose))$$

It must be noted that since there is no correspondence between the augmented examples of different classes, we computed the difference as the average KL-Divergence over mini-batches of the size 64 samples from the shuffled augmented dataset for 10 epochs (the same batch and epoch values used for training BiLSTM and CNN models).

Table 6 shows the distribution difference between augmented *positive* and *negative* samples for the sentiment analysis task. We can observe that augmented dataset with higher classification accuracy has lower divergence between distributions of their positive and negative examples:

Sentiment Classification Accuracy

$$ACC(ONE) >> ACC(ALL)$$

Positive Distribution vs. Negative Distribution:

$$KLD(ONE) < KLD(ALL)$$

Table 7 shows the distribution difference between augmented *concise* and *verbose* samples for the pleonasm prediction task. Here, similar to the sentiment analysis task, we observe the divergence between distributions of augmented concise and verbose examples are following the reverse order of classification accuracy for both BERT and USE representations:

Verbosity Classification Accuracy

$$ACC(NMN) > ACC(SYN) > ACC(DUP)$$

Verbose Distribution vs. Concise Distribution:

$$KLD(NMN) < KLD(SYN) < KLD(DUP)$$

These observations may indicate that the extent to which a heuristic might generate "hard to distinguish" examples could be quantified as the difference (divergence) between the distribution of augmented examples in different classes (AQ1-2).

4.3.2.3 Computational Efficiency

Now that we have investigated the role of “hard to distinguish” examples in the success of training a classifier (AQ1-1) and how to quantify that (AQ1-2), it is time to evaluate the computational efficiency of our purposed approach to see how practical it is compared to training a separate classifier for each augmented dataset and pick the best performing one(s) (AQ1-3). To investigate this, we calculated the time for encoding the augmented examples into continuous space and the time requires for computing the KLD and compared them with the time required for training a classifier on an augmented dataset.

Table 8 shows the average execution time of our approach for evaluating the suitability of different data augmentation heuristics and training neural classifiers on augmented datasets. Reported numbers are averaged over sentiment and pleonasm prediction tasks for all augmented datasets. Encoding is a one-time process for each augmented dataset, and numbers reported under KLD and Classification columns are the overall execution time after 10 epochs of training on an NVIDIA Tesla P100 GPU.

Execution times are showing that our heuristic evaluation approach is about 25 times faster than training a classifier; this may suggest that our approach could be a low-cost alternative solution for assessing the suitability of the heuristic strategies for augmenting training dataset for different classification tasks, especially for complex training scenarios when training many classifiers on different augmented dataset might not be computationally practical (AQ1-3).

We also observed that encoding and divergence calculation times only depend on the number of samples and the classification task and choice of heuristic is not affecting the execution times. Additionally, the training time of both BiLSTM and CNN classifiers also highly depends on the number of training samples, but changing the classification tasks and augmented dataset only slightly change the training time (standard deviation of 9.4s and 6.8s, respectively).

Table 8: Execution time of our heuristic suitability evaluation approach

Our Approach			**Classification**	
USE	**BERT**	**KLD**	**BiLSTM**	**CNN**
33.2s	92.8s	13.4s	2773s	878s
AVG: 63s				
Overall: 76.4s			**AVG:** 1825.5s	

4.3.3 Evaluation

The availability of training corpora is always a bottleneck for the training of complex supervised computational models. Our proposed inference scheme already relaxes the requirement of local prediction tasks from having labeled examples with annotation at the smaller scope to having training examples with global annotation, which is usually easier to obtain. However, in many resource-constrained NLP problem domains, even training corpora for global prediction are not available. Therefore, aiming to provide an answer to Q2, in this experiment we evaluate the extent to which data augmentation can facilitate the training of the local inference approaches by providing noisy training data for global prediction tasks.

Here we study the *pleonasm detection* and *specificity detection* domains (refer to Section 2.5), as two resource-constrained domains. We also make a low-resource case for the *sentiment analysis* domain in order to compare the prediction result with the previous experiment, where a sizable training corpus was available for global prediction (refer to Section 2.

Based on our proposed heuristic evaluation scheme (refer to Section 4.3.2), from a few possible non-label-preserving heuristics for each domain (refer to Section 4.3.1), we choose the best performing heuristics and generate an augmented training corpus for each domain. The size of the training sets are: 105K sentences with either *positive* or *negative* label for sentiment analysis domain (the same size as YPD-Train in Section 4.2), 160K sentences with either *concise* or *verbose* label for pleonasm detection domain, and 110K sentences with either *specific* or *general* for specificity detection domain. Table 9 summarizes the heuristic strategy we used to augment a training dataset for each problem domain.

When evaluating the models for sentiment analysis and pleonasm detection tasks, we use the YPD-Test and SPC, respectively, which are both collected from Yelp. For the specificity detection task, we use the news headline part of the iSTS as the benchmark to evaluate the model. All three datasets have annotation at the word-level, which makes them suitable for local prediction evaluation. Table 10 compares the local prediction inference precision of the different inference schemes across different problem domains, when trained on noisy user-generated corpora, based on the gold annotation in the benchmark datasets.

We can observe that the contrapositive local inference schemes, including *Contrapositive:LOO*, outperform the direct inference schemes in all problem domains and the local prediction precision of our proposed approach (*Contrapositive:TF*) is much higher than all other inference alternatives. As expected, the local prediction inference precision of *Contrapositive:TF* was slightly lower than its corresponding performance in the previous experiment (see Section 4.2), where models were trained on manually-labeled real examples of Yelp (74.2% compared to 77.5%). However, the performance gap between our proposed model and other models is getting larger when trained on noisy data (e.g., 14% difference with *Direct:Rationale* on real data compared to 19% difference with noisy data).

These results may suggest that the contrapositive inference is more noise-tolerant than the alternative methods. In addition, the impact of data augmentation on the performance of our approach is not significant, so it could be beneficial for providing training requirements of our approach and making it applicable to many low-resource NLP problems (Q2).

It is also noteworthy that prior work on semantic pleonasm detection suggests that while a global classifier may be trained to predict whether a sentence is verbose (or concise) with relatively high accuracy, the more local task of finding the pleonastic word(s) is much harder, with the state-of-the-art precision ranging in the 36% (Kashefi et al., 2018). Compared with the results of this experiment, our contrapositive inference scheme has substantially outperformed the previous SOTA results by a huge margin (36% vs. 70.3%). This suggests that our proposed inference scheme could be an efficient alternative solution to many resource-constrained (local) classification problems.

Table 9: Augmentation heuristic strategies for different problem domains

Domain	Source	Heuristic Strategy
Sentiment Analysis	Yelp	• ***Positive*** *Label:* four star and above • *Augmented* ***Negative*** *Heuristic:* substitute a positive word (look up in a polarity vocabulary) with an *antonym* • ***Negative*** *Label:* two star and below • *Augmented* ***Positive*** *Heuristic:* substitute a negative word (look up in a polarity vocabulary) with an *antonym*
Pleonasm Detection	Yelp	• ***Concise*** *Label:* Yelp tips (short sentences) • *Augmented* ***Verbose*** *Heuristic:* add a *synonym* next to an adjective • *Augmented* ***Near-Miss Concise*** *Heuristic:* insert a *non-synonym* word next to an adjective, base on language model prediction
Specificity Detection	News Headline	• ***Specific*** *Label:* contains more than 2 named entities • *Augmented* ***General*** *Heuristic:* substitute a noun with an *hypernym* • ***General*** *Label:* contains no named entities • *Augmented* ***Specific*** *Heuristic:* substitute a noun with an *hyponym*

Table 10: Localized prediction precision of the inference approaches on different problem domains, trained on weakly-labeled augmented eamples

Approaches	**Local Inference Precision**		
	Sentiment Analysis	Pleonasm Detection	Specificity Detection
Direct:MIL	54.3%	24.3%	20.5%
Direct:Rationale	55.6%	30.4%	28.1%
Contrapositive:LOO	59.8%	41.1%	45.5%
Contrapositive:TF	**74.2**%	**70.3**%	**69.5**%

4.3.4 Section Summary

In this section, we presented a non-label-preserving data augmentation scheme that only requires samples from one class to generate instances of other classes. This removes the prior work's assumption of having labeled samples for instances of all classes, therefore, expanding its applicability to a wider range of problem domains.

We found that the quality of augmented examples is a key factor in training an appropriate classifier. As we trained the models on the corpora that are augmented by simpler heuristics that generate obvious examples for different classes, the classification accuracy was dramatically low. However, when the augmented corpus contains "hard to distinguish" examples of different classes, it trained a more robust model.

We argued that the suitability of an augmentation heuristic for a classification task correlates to the extent to which it generates "hard to distinguish" examples. We showed that this quality of the heuristic functions can be quantified as the amount of divergence from one augmented class distribution to another augmented class distribution (AQ1).

Finally, experimental results suggest that considering the contrapositive relation between local and corresponding global tasks makes a more noise-tolerable inference framework and shows that the performance impact of training a transfer function on augmented data is not significant (Q2). Data augmentation may facilitate the training of the transfer function and help our approach be generalizable to various resource-constrained NLP problem domains, as well as the problem domains with an abundance of labeled data.

4.4 Supervised Transfer Learning

In this section we aim to answer the Q3 – *To what extent does the contrapositive relationship between the local and global prediction tasks transferable across domains*, by investigating how supervised transfer learning may facilitate the applicability of our approach to problem domains with limited availability of global training examples (D-P3).

In these problem domains, the lack of training examples will hinder the training of the transfer function models. Therefore, our general approach is to learn the semantics of the task from a related domain with more training examples, and then to use that knowledge to make more robust predictions in the resource-limited target domain.

In this section, we experiment with several transfer learning strategies (see Section 4.4.2) for a local sentiment prediction task in *movie review* domain, as a problem domain with limited resources. We explore how these strategies may help to utilize the knowledge learned from another domain. We experimented with the two following source domains:

- **Related Domain –** we use the *restaurant review* domain, as a related source domain with large enough training data (D-P1) to pre-train the models (see Section 4.4.3.2). More details on *movie review* and *restaurant review* domains and how they compare to each other is given in Section 4.4.1.
- **Augmented Domain –** we augment a training dataset on movie review domain to pre-train the models (see Section 4.4.3.3).

4.4.1 Problem Domains

The Stanford Sentiment Treebank (SST) (Socher et al., 2013) is a dataset with 10K *movie review* sentences with fine-grained sentiment annotation. The relatively small size of SST limits its application in training complex neural network models, such as our transfer function, from scratch. These properties make it a suitable case to study the generalizability of our approach to the resource-limited problem domain. YPD (Zhang et al., 2015), on the other hand, is a sizable resource with sentiment labels for *restaurant review* sentences (see Section 2.5.1 and Section 4.2), which makes it a suitable case as a resource-rich source domain to study the effectiveness of the different transfer learning strategies.

Table 11 shows some statistics of these domains. The sentences from the movie review domain (*SST*) are almost twice as long as the sentences in the restaurant review domain (*YPD*). In addition, although the movie review dataset is 10 times smaller than the restaurant review dataset, its vocabulary (number of distinct words used throughout the dataset), is much richer than the vocabulary of the restaurant review domain. From the 4,665 sentiment words of the movie review domain, which 415 of them are expressing the strong

Table 11: Statistical comparison of movie and restaurant review domains

Domain	**Sentences Number**	**Words / Sentence**	**Vocabulary Size** **(Sentiment : Strong)**	**Domain Specific** **(Sentiment : Strong)**
Movie Review (SST)	9,613	20	14,023 (4,665 : 415)	9,095 (3,566 : 255)
Restaurant Review (YPD)	105,000	10	9,357 (1,507 : 204)	4,429 (408 : 44)

sentiment, 3,566 words are never observed in the restaurant review domain (76%), with 255 of them being words with strong sentiment expression (61%).

Thus, despite the fact that these two domains include sentimental opinions and could potentially be used for the same classification task, they are (statistically, syntactically, and semantically) different.

4.4.2 Transfer Learning Strategies

In Section 2.2.2, we discussed several transfer learning approaches. Here, we experiment with an *instance-based* and a *feature-based* transfer learning approach.

As an instance-based approach, we adapt a simple re-weighting strategy by adding the instances of the target domain (i.e., movie review) to the source training domain with different weights (refer to Section 2.2.2 and Equation (2.6)). We experiment with four different weight setting for target examples: $\lambda = 1$, $\lambda = 2$, $\lambda = 5$, and $\lambda = 10$, by replicating the target domain examples λ times.

As a feature-based approach, we adapt a network-based strategy to align the source features to the target domain by (i) first training the models on the source domain examples, then (ii) fine-tune the trained model on the limited target domain training examples for a few more epochs (in our experiments: 3 epochs). Since our transfer function model contains two parts (autoencoder and discriminator, refer to Section 3.3), which are first trained separately then jointly, we experiment with the following fine-tuning settings:

- **Fine-Tune.** We first train the autoencoder and discriminator on the source domain examples (in our experiments: separately for 10 epochs and then jointly for 2 more epochs), then, we jointly fine-tune them on the target domain examples for a few more epochs (in our experiments: 3 epochs).
- **Fine-Tune$^+$.** We first train the autoencoder and discriminator separately for initial epochs on the source domain examples (in our experiments: 10 epochs), then, we separately fine-tune them on the target domain examples for a few more epochs (in our experiments: 3 epochs), finally, we jointly train them only on the target domain examples (in our experiments: 2 epochs).

4.4.3 Evaluation

In this section, we evaluate how transfer learning can facilitate the training of the inference models on a resource-limited problem domain (sentiment classification in the movie review domain). We first conduct a baseline study to better understand the effect of training data size on the local inference performance of our model. Then we investigate whether the (contrapositive) semantic relationship between the local and global sentiment labels can be transferred across domains (Q3) and whether we can exploit that to improve the training of our model in low-resource settings.

4.4.3.1 Baseline Experiment

The purpose of this experiment is to establish a baseline prediction performance for resource-constrained domains by training the models only on the limited in-domain training examples. This experiment may help us to understand how training data size affects the learning of the global task semantics and its relationship with the contributing local features. Therefore, we split the SST dataset into two parts as follow:

- **SST-Test.** containing 1.1K sentences with one positive (537 sentences) or one negative (574 sentences) words. We only filtered for the sentences with only one sentiment word (very positive, positive, very negative, or negative). We then converted the sentiment

Table 12: Local sentiment prediction precision of the different inference schemes, trained on a small number of movie review sentences

Approach	**Local Inference Precision**
Direct:MIL	27.0%
Direct:Rationale	28.3%
Contrapositive:LOO	29.7%
Contrapositive:TF	**33.2**%

annotations into polarity binary labels by grouping *very positive* and *positive* into positive, and *very negative* and *negative* into negative labels.

- **SST-Train.** the rest of 8.9K sentences from the original SST with just sentence-level polarity annotation.

In this experiment, we use the sentences of the SST-Train to train the inference models and use the SST-Test dataset to evaluate the local prediction performance of our approach on the movie review domain. Table 12 shows the local prediction inference precision of our model on predicting the most polar word of a highly polar movie review sentence. Compared with the results of the same task in Section 4.2, where the training corpus was large enough (YPD-Train), the precision of all models are substantially lower. For example, our model achieved the local precision of 33.2%, while its precision was 77.5% when trained on Yelp.

These results suggest that a relatively small corpus of 10K sentences is not sufficient to train a reasonable inference model. Therefore, in Section 4.4.3.2, we explore how supervised transfer learning may help to improve the training of our approach in the target domain, when there is a sizable training corpus in a related domain. In Section 4.4.3.3, we investigate how data augmentation can couple with transfer learning to improve the performance of our model, when a related domain with sizable training examples is not available.

4.4.3.2 Transfer Learning from a Related Domain

In this section, we evaluate the local inference performance of our approach on detecting the naturally occurring sentiment words in a resource-constrained movie review domain with a limited amount of training examples (D-P3), when there is a related domain with sizable training examples. We use the labeled examples of a resource-rich domain to improve the training of our transfer function in a resource-limited domain. This case study may help us to investigate the transferability of the contrapositive relationship between the local and global sentiment predictions across domains and how transfer learning may help to utilize the training data in a resource-rich domain to improve the training of the transfer function in a resource-constraint domain (Q3).

In this experiment, we start by pre-training the inference models on examples of the restaurant review domain (105K sentences of YPD), as a resource-rich domain with *related* annotation for the sentiment analysis task. Next, we use the small corpus of movie review examples (8.9K sentences of SST-Train) to tune the models using different transfer learning strategies (refer to Section 4.4.2). Finally, we evaluate the word-level sentiment prediction performance of the tuned models on the benchmark corpus (1.1K sentences of SST-Test). Table 13 shows the result of our contrapositive local inference approach in the word-level sentiment detection task on the movie review domain.

Although the restaurant review and movie review domains have similar class information, they are still different in many ways (refer to Section 4.4.1). So as expected, the models trained on the restaurant review domain (YPD-Train) failed to achieve high local inference prediction precision on the movie review domain (SST-Test), without exploiting transfer learning (P: 13.7%). While our model with instance-based re-weighting transfer learning strategies improves the local prediction by 12.9%, compared with our in-domain baseline experiments (refer to Section 4.4.3.1), the best performing re-weighting strategy ($lambda = 5$) shows only around 3% improvement.

The feature-based fine-tuning transfer learning strategies, however, improved the performance of all models by a much greater margin. Using the training examples of the restaurant review domain and transferring their semantics to the target domain using *Fine-Tune*$^+$ strat-

Table 13: Local sentiment classification precision of inference models, trained on restaurant review and tuned on movie review domains

Model	Transfer Learning Strategy	Local Inference Precision
Contrapositive:TF	In-Domain Baseline (§4.4.3.1)	33.2%
	None	13.7%
	Re-Weighting ($\lambda = 5$)	36.6%
	Fine-Tune	47.9%
	Fine-Tune$^+$	**58.3%**
Contrapositive:LOO	Fine-Tune$^+$	45.7%
Direct:MIL	Fine-Tune$^+$	41.6%
Direct:Rationale	Fine-Tune$^+$	45.3%

egy, remarkably improved the local inference prediction of our model on the movie review domain. When compared to the in-domain baseline results, which were trained on the small in-domain corpus (SST-Train), our model yields 25.1% higher word prediction precision (75% improvement), and compared to the settings with no domain adaptation attempt, it yields 44.6% higher local inference precision (3x more accurate). Note that since *Fine-Tune$^+$* strategy works best for all other models as well, we include only the result of this approach for those models in the Table.

Additionally, we observe that by incorporating transfer learning, the performance gap between our model and alternative approaches is getting wider. In the baseline experiment, our model was performing marginally better than other approaches, but with transfer learning, our model (Contrapositive:TF) outperforms the alternatives by a larger margin (30% more accurate). This observation may be due to the fact that our approach is a more complex neural network model with more parameters to learn, which makes it more difficult to train with a smaller training corpus (e.g., in baseline experiments).

4.4.3.3 Transfer Learning from an Augmented Domain

Problem domains with limited training resources (D-P3), can also be approached similarly to the problem domains, to which training data for global prediction is not available (D-P2, refer to Section 4.3). We can *augment* a training dataset for the target domain, train the models and make local predictions, or we can further perform the transfer learning from the augmented domain to the target domain and then do the prediction. In this experiment, we investigate the effectiveness of data augmentation in preparing the training data for transfer learning to enable our approach to make more accurate local predictions in a target domain. Similar to the previous experiment, here we also use SST-Train and SST-Test as the tuning and benchmark datasets to represent the resource-limited *movie review* domain.

In order to augment a sizable training corpus in the movie review domain, since SST-Train is relatively small and, to our best knowledge, there are no other large data sources with sentiment class information, we incorporated two data augmentation strategies at the same time: (i) we use our non-label-preserving heuristics *ONE* (refer to Section 4.3.1.1); (ii) we used the following two simple label-preserving heuristics proposed by Wei and Zou (2019):

- **Synonym Replacement (SR).** Randomly pick a content word from the sentence and replace it with a synonym chosen at random.

- **Random Insertion (RI).** Randomly choose a content word from the sentence and insert one of its synonyms to a random place in the sentence.

We apply the label-preserving heuristics *SR* and *RI* to the sentences of SST-Train to generate more examples with the same polarity label as the original sentence. For each sentence, we repeat each heuristic function until about 20% of its words are changed (α = .2). In contrast, we apply the non-label-preserving heuristic *ONE* to the positive sentences of SST-Train to generate more negative examples, and the other way around for generating more positive examples (similar to the augmentation process in Section 4.3.1.1).

Table 14: Local sentiment classification precision of inference models, trained on the augmented examples and tuned on the real examples of the movie review domain

Model	**Transfer Learning Strategy**	**Local Inference Precision**
Contrapositive:TF	In-Domain Baseline (§4.4.3.1)	33.2%
	None	54.9%
	Re-Weighting ($\lambda = 5$)	55.5%
	Fine-Tune	56.8%
	Fine-Tune$^+$	**64.1%**
Contrapositive:LOO	Fine-Tune$^+$	52.5%
Direct:MIL	Fine-Tune$^+$	51.2%
Direct:Rationale	Fine-Tune$^+$	53.3%

Using these heuristics, for each sentence, we generated up to 10 augmented examples. Overall, we generated 35K positive and 35K negative augmented examples. Since this augmented dataset is generated from the SST-Train examples, it holds the same marginal distribution, so could be considered to belong to the same domain (movie review). We train all models on the *augmented* movie review domain, then the learned task semantics are transferred to the *real* movie review domain (tuned on SST-Train), using different transfer learning strategies (refer to Section 4.4.2), and finally evaluated on *SST-Test* dataset. Table 14 shows the result of predicting the sentiment word in a highly polar movie review sentence using different inference approaches.

The augmented domain is constructed from the target domain examples so they share the same marginal distribution. As a result, even without performing transfer learning, our model is able to achieve a much higher local prediction precision than when it was trained with real examples (P: 54.9% here vs. 13.7% in the previous experiment – Section 4.4.3.2). This observation supports our findings on the success of data augmentation in preparing the training requirement of our transfer function in Section 4.3.3 (Q2).

Even though our model shows promising local prediction precision with only the augmented training data, we see that transfer learning can bring additional improvements. Similar to our observations in Section 4.4.3.2, applying the Fine-Tune^{+} knowledge transfer strategy results in the highest performance gains (from 54.9% to 64.1% for Contrapositive:TF – about 17% improvement), while the best performing re-weighting strategy ($lambda = 2$) yields a negligible improvement. Note that since *Fine-Tune*$^{+}$ transfer learning strategy works best for all other models as well, we include only the result of this approach for those models in the Table.

In addition, fine-tuning our transfer function (Contrapositive:TL), even with a small number of training examples in the movie review domain, increased the precision of predicting the sentiment words from 33.2% to 64.1% (about 93% improvement), as compared with the baseline results (see Table 12).

4.4.4 Section Summary

In this section, we investigated the transferability of the contrapositive semantic relationship between the local and global contexts across (related) domains (Q3). Results from our experiments suggest that some knowledge of this relationship on a certain prediction task could be learned from a domain with more training resources and then used to improve the prediction inference on a domain with limited resources for the same task (D-P3).

In addition, our findings suggest that, when a domain with a sizable training corpus of global prediction is not available, data augmentation can provide an augmented training corpus. This augmented training corpus can also be used as a source domain for further transfer learning attempts.

Finally, we observed that, in general, feature-based transfer learning strategies are more suitable for transferring the class inference semantics of one domain to another, than instance-based strategies.

4.5 Unsupervised Domain Adaptation

In this section, we study the generalizability of our proposed local contrapositive inference approach to resource-constraint problem domains, for which only unlabeled examples are available (D-P4). A lack of training examples in these domains will hinder the training of transfer function models. However, we argue that utilizing unlabeled data may improve the learning process of our model in the new domains. Therefore, the key research question of this section: Q4 – *To what extent does the contrapositive relationship between the local and global prediction tasks adaptable to a new domain, without using any labeled data?.*

In Section 4.5.1, we explore how does our transfer function model perform in a domain that differs from the training domain. This study helps us to better understand how domain mismatch impacts the contrapositive inference of semantics between local and global predictions and what are the issues.

To address these issues, in Section 4.5.2 we propose a method for incorporating an unsupervised domain matching mechanism into our transfer function model so it can make more robust local predictions in new domains. Finally, in Section 4.5.3, we evaluate our proposed domain adaptable transfer function model on local class inference predictions in the movie review domain, without using any labeled training data in the target domain, as a case for problem domains lacking training corpora (D-P4).

4.5.1 Problem Exploration

Let us consider a scenario similar to our experiments in Section 4.4.3.2 and Section 4.4.3.3, where we trained our model on a large set of examples in the *restaurant review* domain, but we want to make the prediction in *movie review* domain. Now, if there is no (small) corpus of training examples in the movie review domain, then we cannot use supervised transfer learning techniques to improve the local prediction in the target domain, nor can we use data augmentation to create a training corpus since there is no (weakly) labeled examples in the target domain from which to augment more examples.

The first row of the Table 13 shows the local prediction precision of our model for movie review domain (SST-Test), when trained on restaurant review domain (YPD-Train), without being exposed to any labeled movie review sentences [4] Our transfer function was only able to predict the sentiment words of movie review sentences with a precision of 13.9%, which indicates that our vanilla transfer function model will fundamentally fail to infer a local prediction when the source and target domains are different.

Now let us take a closer look at the output of our transfer function model when applied to the movie review sentences. The underlined parts are changed during the rewriting and class transfer process, and the super-scripted numbers are indicating the correspondence between the local parts of the input and output sentences.

Example 4.1:

Input Sentence:
"a sloppy^1 *slapstick*^{2} *throwback to long gone bottom of the bill fare like the ghost*^{3} *and mr. chicken ."*

Output Sentence:
"a delicious^1 *pays*^{2} *throwback to long gone bottom of the bill fare like the nugget*^{3} *and mr. chicken ."*

Example 4.2:

Input Sentence:
"in the era of the sopranos^1 *, it feels painfully redundant*^{2} *and inauthentic ."*

Output Sentence:
"in the era of the fast-delivery^1 *, it feels perfectly chatty*^{2} *and inauthentic ."*

Example 4.3:

Input Sentence:
"these characters become wearisome^1 *."*

Output Sentence:
'these characters become surreal^1 *."*

[4] To minimize the impact of out-of-vocabulary (OOV) words on breaking the decoder, we augmented the vocabulary list of the model trained on the restaurant review domain with the words from the movie review domain, however, the model did not expose to any syntactic and semantic attributes of the movie review domain.

Example 4.4:

Input Sentence:
“at a time when commercialism^{1} *has squeezed the life out of whatever idealism*^{2} *american movie making ever had , godfrey reggio ’s career shines like a lonely beacon*^{3} *.”*

Output Sentence:
“at a time when lasagna^{1} *has squeezed the life out of whatever pints*^{2} *,*^{3} *.”*

In *Example 4.1*, in an attempt to rewritten the original *negative* sentence as a *positive* alternative, the transfer function changed the words “sloppy”, “slapstick”, and “ghost”. Choosing these words is justifiable because they all are somehow, regardless of context, expressing negative sentiment. The replacement choices are, however, questionable. Replacing the “sloppy” (negative sentiment) with “delicious” (positive sentiment) might be a valid polarity transference; however, “delicious” is chosen probably because it is just a very likely positive adjective in the source domain (restaurant review), yet it does not match the context. Rewriting “slapstick” to “pays” might be even harder to investigate, the transfer function is expected to struggle to find an appropriate adversarial (opposite sentiment) word for the “slapstick”, but “pay” might be chosen because it matches the source domain context in presence of the word “bill”. Finally, the model might have learned to associate the word “ghost” with the negative sentiment (expected), thus, decide to replace that in an attempt for making the sentence positive; however, the replacement word “nugget” is probably only selected because it yields higher context probability in close proximity of “chicken”.

In *Example 4.2*, the transfer function changed the words “sopranos” (a TV show) to “fast-delivery”. This might be because the word “soprano” is extremely unlikely to appear (perhaps an OOV) in the source domain, so the decoder chooses the “fast delivery” as a likely (and positive) replacement. Choosing the “painfully redundant” as a negative lexical item and rewriting it as “perfectly chatty”, nevertheless, is a justifiable rewriting towards changing the polarity of the sentence.

The transfer function performed much better in *Example 4.3*, probably because the input sentence is relatively short with just a few domain-specific words. The only sentiment (negative) word of the sentence – “wearisome”, is correctly identified. Its adversarial (positive) alternative – “surreal”, however, may not be a suitable option to flip the sentence’s polarity.

The reason behind the first and second changes of the *Example 4.4* is perhaps similar to those behind the first change of the *Example 4.2* – an unlikely word in the source domain ("commercialism" and "idealism") is replaced with an (almost random) frequent word in the target domain ("lasagna" and "pints"). For the third change, the model faced a very unlikely sequence and decided to marginalize the probability of the whole sequence into a very likely word (punctuation mark ",") and make an early termination.

Our observations of applying our transfer function model in a local sentiment classification scenario when the source (restaurant review) and target (movie review) domains are different is summarized as follows:

- In most cases, the sentiment words are identified and rewritten (good prediction recall)
- Words that are not observed or are less likely to appear in the target domain are sometimes rewritten (bad prediction precision; generator issue)
- Our model sometimes fails to correctly predict the polarity of the generated sentences (discriminator issue)

Therefore, it appears that the semantic of the local inference task could be learned from the source domain and be applied to the target domain (rewriting sentiment words). However, the excessive local rewriting may indicate that the discriminator may only be able to classify the extreme global cases (change all sentiment words to flip the polarity). Based on these observations and how our transfer function works, we investigate the following two sub-questions to answer Q4:

Q4-1. *How to improve the extent to which the generator model preserves the content of the sentence of a new domain?*

Q4-2. *How to improve the global prediction in a new domain without supervision?*

To address these questions, we propose new computational models for the generator (Section 4.5.2.1) and discriminator (Section 4.5.2.2) parts of our transfer function model. These models are able to utilize the unlabeled examples to facilitate the adaptability of our approach to new domains.

4.5.2 Domain Adaptable Transfer Function Model

In the previous section, we discussed the issues that prevent our model from being practically applicable to out-of-domain contrapositive inference. In this section, we address these issues by proposing domain-adaptable computation models for the generator and discriminator modules of our transfer function model, which can utilize the unlabeled data in a target domain to make more robust cross-domain local predictions.

Referring back to Section 2.2.2.1, domain adaptation is a special case of transfer learning, where there is no labeled data available in the target domain ($\mathcal{D}_T$), aiming to *improve the target prediction function ($\mathcal{F}_T$), using the knowledge learned from the source domain ($\mathcal{D}_S$), with no supervision in the target domain ($\mathcal{D}_T$).*

Similar to supervised transfer learning approaches, domain adaptation approaches can also be categorized as: *instance-based* approaches, which operate at the instances space, or *feature-based* approaches, which operate at the feature space. In this work, we focus on feature-based domain adaptation approaches and investigate how we can learn and operate on a coinciding latent space for source and target domain.

4.5.2.1 Unsupervised Domain Adaptable Generator

Knowing that the out-of-domain local inference prediction of our model is mainly hampered by failing to generate realistic sentences in a new domain, let us take a look at the generation part of our model (refer to Section 3.3).

Figure 5 illustrates the structure of our autoencoder module, which is responsible for generating sentences. The model is optimized to minimize the reconstruction loss (refer to Equation (3.7)) and the latent space z, thus, interpolated to generate realistic sentences of both $ClassA$ and $ClassB$. Since z is learned using the instance of the source domain (e.g., restaurant review), it may fail to (re)generate instances of another domain (e.g., movie review), as we observed in the previous section. Nevertheless, looking closer, the encoder is trained independent of the upstream global classification task and the class labels of the sentences; the input is only raw sentences, and the output is the regeneration of the input with the aim of making them as similar as possible.

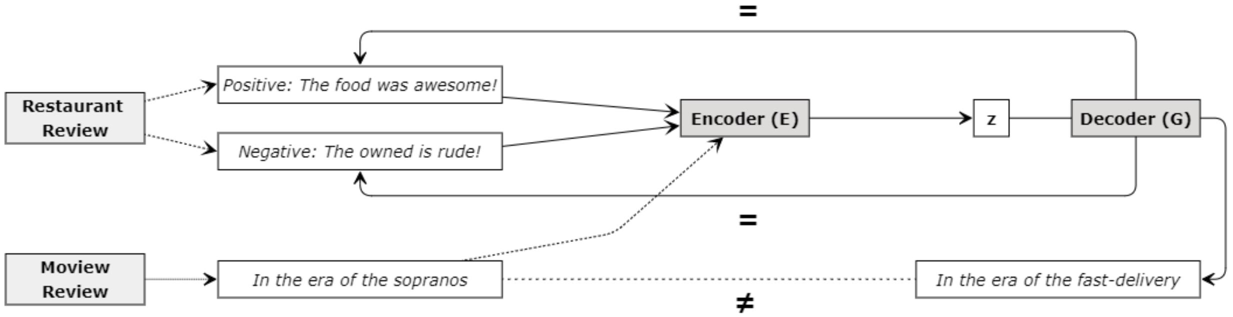
=
Restaurant Review
Positive: The food was awesome!
Negative: The owned is rude!
Encoder (E)
z
Decoder (G)
=
Moview Review
In the era of the sopranos
In the era of the fast-delivery
≠

Figure 5: Cross-domain sentence generation problem

Therefore, we may be able to learn a latent space from the unlabeled data available in the target domain, and align it with the latent space optimized for the source domain so that the model can generate realistic sentences in both domains. Figure 6 shows our proposed hierarchical autoencoder architecture that may improve the generation of sentences in the target domain and facilitate the adaptation of our transfer function model to the new domains, using only unlabeled data.

We can still optimize for a similar reconstruction loss, but during the pre-training process we mask the class labels to the autoencoder, as given in Equation (4.2), where c now represent instances from the source and target domains. Then, as given in Equation (3.7), during the joint-training phase (learning class transference), we start feeding the autoencoder with the class labels (refer to Section 3.3).

$$\mathcal{L}_{rec}(\theta_E, \theta_G) = \mathbb{E}_{q_E(z|c)}[\log p(c|z)] \tag{4.2}$$

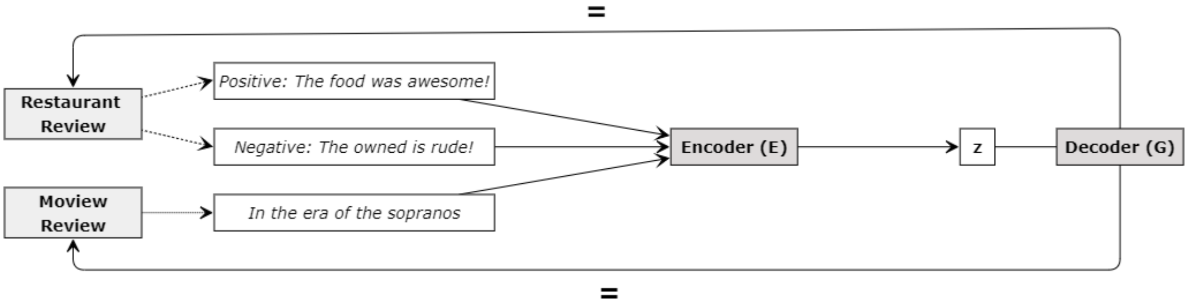
=
Restaurant Review
Positive: The food was awesome!
Negative: The owned is rude!
Moview Review
In the era of the sopranos
Encoder (E)
z
Decoder (G)
=

Figure 6: Our proposed generator structure for a domain adaptable transfer function

4.5.2.2 Unsupervised Domain Adaptable Discriminator

The discriminator (D) module of our transfer function models is responsible for predicting a class label for the sentence ($\hat{c}$) that is generated by the decoder (G). This class prediction would provide feedback to the decoder as to whether the generated sentence has the same class label as the original sentence (c) or if it is transferred into a different class? Figure 7 illustrates how this process works. In our resource-constrained prediction scenario, the discriminator is trained on instances of a source domain (e.g., restaurant review) but attempts to infer a prediction for instances of the target domain (e.g., movie review). This process, however, would not train an appropriate model for the target domain, as we observed in Section 4.5.1.

As a popular solution to this problem, especially when using neural network models, is to match the distributions between the source and target domains (in the kernel-reproducing Hilbert space. Domain adaptation approaches are either directly measure the mean of source and target distribution using different metrics such as maximum mean discrepancy (Gretton et al., 2012), central moment discrepancy (Zellinger et al., 2019), correlation alignment (Sun et al., 2016), cosine similarity (Benaim and Wolf, 2017), association loss (Haeusser et al., 2017), and metric learning (Mahadevan et al., 2018), or they use adversarial strategies to train a classifier that can discriminate between source and target distributions (Ganin et al., 2016; Tzeng et al., 2017; Shen et al., 2018).

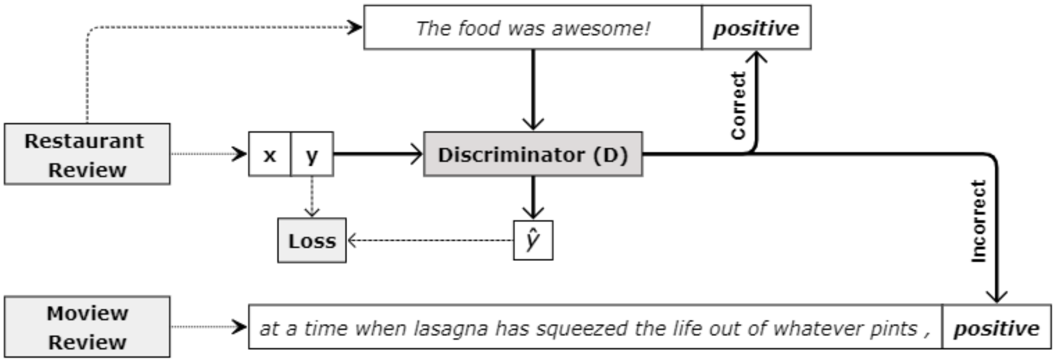

Figure 7: Cross-domain discrimination problem

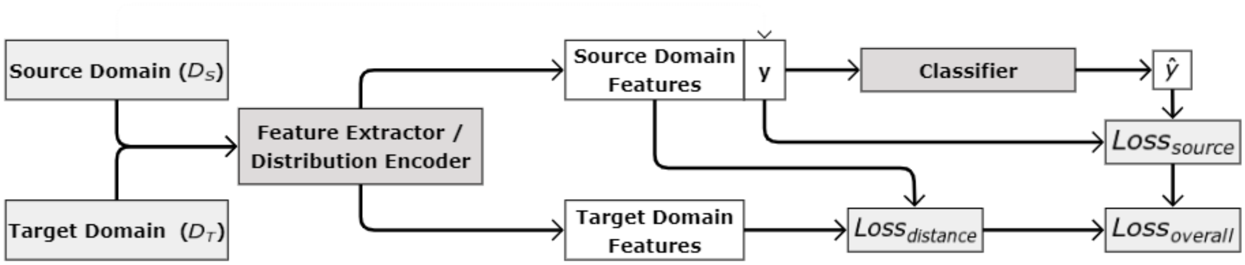

Figure 8: Unsupervised domain adaptation process through distribution matching

Figure 8 shows the overall structure of the unsupervised domain adaptation through distribution matching. In order to learn the semantics of the prediction task, the classifier model is training on the source domain features and their associated label (y), by minimizing the $Loss_{source}$. At the same time, the feature extractor (distribution encoder) model tries to minimize the distance between the extracted source and target feature spaces ($Loss_{distance}$). As a result, the feature extractor model learns to transform the instances of source and target domains into a feature space where they are similar, and the classifier learns to distinguish the instances of different classes in that feature space. Therefore, the model can make robust classifications for instances of the target domain, as well as instances from the source domain.

Following the same principle, to improve the robustness of the global prediction of our model in the target domain, we need to find a latent space in which the instances of the source and target domains are closely sampled. Then, we only need to train our discriminator on this feature space in order to make it adaptable to both domains. Referring back to Figure 6 and Equation (4.2), we are already learning a latent space that can encode instances of both source and target domains. Thus, we can use this feature space to train a domain-adaptable discriminator.

The discriminator's job is to classify the output of the decoder ($\hat{c}$) and provide the gradient guiding signal for training the decoder. Thus, the input of the discriminator is a word sequence that represents $\hat{c}$, which we call *hard-representation*. In order to train our domain-adaptive discriminator, however, we need to first encode the generated sentence $\hat{c}$ into our coinciding latent space z to get a *soft-representation*. Since the encoder is trained on

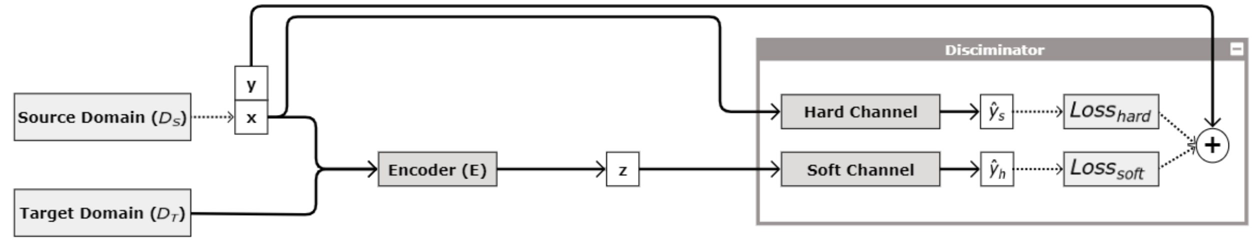

Figure 9: Our domain adaptable discriminator model

the original sentences (c), not the generated ones ($\hat{c}$), this re-encoding process may impose some computational overhead and training instability. To address this issue, we propose a dual-channel discriminator model that is mainly trained on the *hard-representation*, but we jointly guide the training using the *soft-representation*.

Figure 9 shows the structure of our domain-adaptable discriminator model. The *hard channel* of the discriminator is trained on the labeled examples (pair of $< c, y >$) of the source domain by minimizing the $Loss_{hard}$. The *soft channel* of the discriminator is trained on the soft representation of the labeled instances (pair of $<$ Encode$(c) = z, y >$) of the source domain by minimizing the $Loss_{soft}$. The discriminator is then trained by jointly optimizing for both soft and hard channels.

Equation (4.3), thus, supersedes the Equation (3.8) as the discrimination loss in our final objective function, given in Equation (3.9). Note that in Equation (4.3), c is a sentence observation (either original or generated), and λ_S and λ_H are balancing factors

$$\mathcal{L}_{\text{disc}}(\theta_D) = \lambda_H \underbrace{\mathbb{E}_c[\log q_D(\{A|B\}|c)]}_{Loss_{hard}} + \lambda_S \underbrace{\mathbb{E}_c[\log q_D(\{A|B\}|q_E(z|c))]}_{Loss_{soft}} \tag{4.3}$$

During the pre-training phase (refer to Section 3.3), we jointly optimize the $Loss_{soft}$ and $Losshard$ (i.e., $\lambda_S = \lambda_H \neq 0$). During the class transfer training phase, in which we jointly train the generator and discriminator, we will only use the hard channel of the discriminator (i.e., $\lambda_S = 0$) to avoid the re-encoding of the generated sentences ($\hat{c}$) and facilitate the training process.

4.5.3 Experiments

In this section, we investigate whether the contrapositive relationship between the local and global problems can be transferred from one domain to another, without supervision in the target domain (Q4). This helps to evaluate the applicability of our inference scheme to problem domains, for which only unlabeled examples are available (D-P4).

To study the different performance aspects of our domain-adaptable transfer function model on a local prediction task for out-of-domain examples, we experiment with the following scenario: We start by pre-training our model on examples of the restaurant review domain (105K sentences of YPD), as a resource-rich domain with *related* annotation for the sentiment analysis task. Next, we use the 100K unlabeled movie review sentences[5] collected by Pang and Lee (2005) to learn a latent space with matching distribution between restaurant review and movie review domains. Finally, we evaluate the local inference prediction performance of our model on a benchmark dataset on the movie review domain (SST-Test).

In Section 4.5.3.1 we the rewriting and generation quality of our proposed domain-adaptable transfer function model. In Section 4.5.3.2, we evaluate the adaptability of our model to global prediction in a new domain, and in Section 4.5.3.3) we evaluate the unsupervised cross-domain contrapositive local class inference performance of our model.

4.5.3.1 Generation

In natural language generation literature, a popular way of measuring the amount of change made to the sentence is to calculate the *BLEU* score (Papineni et al., 2002) between the *original sentence* (c) and the *generated sentence* ($\hat{c}$), ranging in $[0, 1]$. A value of 0 indicates that generated sentence has no overlap with the original sentence (low generation quality) while a value of 1 means the original and generated sentences are identical (high generation quality). It also would be insightful to know how to interpret the BLEU scores within the range. In machine translation literature (Lavie, 2011), a value less than .1 is considered a useless translation, between .3 and .5 as understandable, and scores above .6 are considered very high-quality generation.

[5] `http://www.cs.cornell.edu/people/pabo/movie-review-data/polarity_html.zip`

Table 15: The BLEU score between the original sentence and the sentence generated by our transfer function on different training scenarios

Model	**In-Domain**	**Out-Of-Domain**	**Domain Adaptation**
	(Yelp → Yelp)	(Yelp → SST)	(Yelp → SST)
Contrapositive:TF	.71	.09	.63

Table 15 shows the BLEU scores of the rewriting process of our model for three different training scenarios: The "In-Domain" setting refers to when our model is trained and tested on the same domain. This setting can be used as a *practical upper-band* and help us to better understand the effectiveness of our domain adaptation efforts on the generation part of our model (refer to Section 4.5.2.1). For this setting, we calculate the BLEU score between the original and generated sentences from our first experiment in Section 2, when the train and test examples were both sampled from Yelp.

The "Out-of-Domain" number indicates how well our (pre-trained) vanilla transfer function performs on rewriting the sentences of a related domain. The low BLEU score of .09 from rewriting the sentences of the movie review domain (SST-Test) using the model trained on the restaurant review domain (Yelp) validates our observations in Section 4.5.1 on the failure of our model on cross-domain rewriting and content preservation.

The number under "Domain Adaptation" shows the generation quality of our domain-adaptable transfer function (refer to Section 4.5.2.1) on rewriting the movie review sentences, when it was only trained on the restaurant review examples. As we can see, the BLUE score has significantly improved from .09 (the out-of-domain baseline) to .63, which suggests that our distribution matching mechanism of feeding examples of both source and target domain to our autoencoder can improve the cross-domain content-preservation ability of our model (Q4-1 and Req. 2).

Furthermore, achieving a comparable score with the in-domain generation score of .71 suggests that our model now can generate more fluent sentences in the movie review domain.

Let us have a look at how our domain adaptable model rewrites some of the examples of the Section 4.5.1:

Example 4.1 (+Domain Adaptation)

Input Sentence:
“a sloppy slapstick^{1} throwback to long gone bottom of the bill fare like the ghost and mr. chicken .”

Output Sentence:
“a big screen^{1} throwback to long gone bottom of the bill fare like the ghost and mr. chicken .”

Example 4.2 (+Domain Adaptation)

Input Sentence:
“in the era of the sopranos , it feels painfully redundant^{2} and inauthentic .”

Output Sentence:
“in the era of the sopranos , it feels perfectly chatty^{2} and inauthentic .”

Example 4.4 (+Domain Adaptation)

Input Sentence:
“at a time when commercialism has squeezed the life out of whatever idealism^{1} american movie making ever had , godfrey reggio ’s^{2} career shines like a lonely beacon^{3} .”

Output Sentence:
“at a time when commercialism has squeezed the life out of whatever real^{1} american movie making ever had , career shines like a lonely bacon^{4} .’ .”

In all of these examples, we can see that our domain adaptable model is more successful in rewriting the movie review sentences. As an instance, in Example 4.1 and Example 4.2, only one local part of each sentence has changed, whereas our vanilla model rewrote at least two parts of the sentences (see Section 4.5.1). Even though some of the word choices in Example 4.4 are incorrect, the rewriting is significantly better than our vanilla model, and most of the sentence remains intact.

It is also worth mentioning that the global class prediction from the discriminator is the main signal for class transference in our model. Despite all of our domain adaptation efforts, our model is not trained with any class information in the target domain, which impacts the rewriting process at the class transference stage.

4.5.3.2 Global Prediction

The discriminator is another component that influences the contrapositive local prediction of our model. It provides feedback to the decoder regarding the likelihood of the generated sentence belonging to another class. The decoder uses this signal to determine which parts of the sentence need to be rewritten for successful class transference. Our discriminator is trained as a traditional supervised classifier on global labeled examples (e.g., sentences). As with other traditional models, our vanilla discriminator is unlikely to perform well for the classification of out-of-domain sentences.

Table 16 shows the global (sentence) classification performance of our model in different training settings. Similar to the previous section, the in-domain performance is collected from our first experiment in Section 2, where the examples used to both train and test the model were from the same domain (restaurant review – Yelp). The precision of 98% indicates a nearly perfect prediction of the polarity of the Yelp review sentences.

Meanwhile, the same model trained on Yelp performs poorly when predicting movie review sentences of the SST-Test (out-of-domain performance of 67.9%). However, incorporating a mechanism to learn a matching distribution for source and target domain in our domain-adaptable model (refer to Section 4.5.2.1) increased the global prediction precision of our model to 77%. This significant performance gain over the out-of-domain baseline suggests that jointly training a classifier on a coinciding feature space and examples of the target domain can improve the domain robustness of the global predictions (Q4-2).

Finally, comparing with the in-domain prediction results shows that the cross-domain prediction is a challenging problem, and even with all of our domain adaptation efforts, we should not expect high classification results in a new domain.

Table 16: The global prediction precision of our approach in different training scenarios

Model	**In-Domain**	**Out-Of-Domain**	**Domain Adaptation**
	(Yelp → Yelp)	(Yelp → SST)	(Yelp → SST)
Contrapositive:TF	98.0%	67.9%	77.1%

4.5.3.3 Local Prediction Inference

Having discussed the intrinsic performance aspects of our proposed domain-adaptable transfer function, in this section, we evaluate the contrapositive local inference performance of our model, following the same training and testing scenario: (i) train the model on YPD, (ii) use the *unlabeled* sentences of IMDB to learn a matching latent space, and (iii) evaluate the model on SST-Test.

Table 17 shows the results of our approach on detecting the words that are strong indicators of sentiment in the movie review sentences. The first line of the table (Transfer Learning Strategy: None) shows the results of our vanilla model, which has not been adapted to the movie review domain using the IMDB sentences. As we already observed in the previous experiment (Section 4.4.3.2), this model fails to make robust predictions on the movie review domain, with a mere local inference prediction precision of 13.7%.

However, our domain-adaptable transfer function with distribution matching mechanism achieves a three-fold increase in local prediction precision (from 13.7% to 45.2%), compared to our vanilla model. This remarkable improvement suggests that the contrapositive relationship between the local and global prediction tasks can be learned from a related domain and then adapted to a new domain, without using global class label information (Q4).

We also observe improvements in global prediction precision (from 67.9% to 77.1%) and sentence generation quality (BLEU from .09 to .63 of our model. However, the discriminator and autoencoder parts of our transfer function are interconnected and jointly trained to perform the class transference. This makes it is difficult to measure the individual contribution of each module to the final local prediction performance.

Table 17: Sentiment classification performance of our domain adaptable model

Model	**Transfer Learning Strategy**	**Local Prediction**	**Global Prediction**	**BLEU**
Contrapositive:TF	None	13.7%	67.9%	.09
	Distribution Matching	**45.2%**	**77.1%**	**.63**

4.5.4 Section Summary

We found that the inability of our model to generate fluent sentences and make robust global predictions in a different domain are the main reasons that prevent it from being effectively applicable to different related domains.

We show that the unlabeled examples of source and target domain can be utilized to learn a latent feature space in which source and target examples are similar to each other. A domain-adaptable transfer function can be learned from the labeled instance of the source domain in this latent space to make robust local inference predictions in the target domain (Q4).

Results from our experiments show that the local-global contrapositive relationship of certain prediction tasks learned from a domain with more training resources is adaptable to a new domain, and thus, our proposed contrapositive inference scheme is applicable to resource-limited problem domains, for which only unlabeled examples are available (D-P4).

5.0 Inference Granularity

The scope of the local context (e.g. a single word or a sentence) and global context (e.g., a sentence or a document) of the prediction task and domain may pose different challenges and limitations for the inference framework. For example, it might be practically impossible to associate the class prediction for a document with the presence of a single word. All of the problems studied in Chapter 4 shared the same inference granularity profile: *inferring a prediction for single word (local scope) from the sentence (global scope) prediction.* Section 5.1 summarizes our sentence to word inference experiments.

In this chapter, I test for the fourth and last hypothesis of the thesis (**H4**) by investigating the generalizability of the contrapositive inference scheme to different levels of inference granularity, where *more than one word* could be the major contributor to the corresponding global problem. In Section 5.2, we study the applicability of our contrapositive approach to problems in which a greater-than-one-word scope of the text is responsible for the class prediction of the global sentence (I-P2). In order to answer the following research question, we investigate how *self-attention* can facilitate getting a phrasal representation of local features and optimizing for *conciseness loss* can allow to control for the scope of the local context:

Q5. *Does the contrapositive inference extend to local predictions of scopes exceeding one word?*

In some problems, the global scope could be larger than a sentence. In Section 5.3, we aim to answer the following question and evaluate our inference approach by studying a paragraph to sentence inference case (I-P3).

Q6. *Does the contrapositive scheme extend to inference from a larger than a sentence global scope?*

5.1 Sentence to Word Inference

In Chapter 4 we experimented with problem domains with varying data availability profiles. In all of these problems, a *single word* was the major contributor to the class prediction of the *sentence*, in which it was used. These experiments demonstrated the applicability of our contrapositive inference scheme and its implementation as a transfer function to problems with sentence-to-word inference granularity.

Table 18 summarizes the inference prediction precision of our model on the sentiment analysis domain. The "in-domain" result refers to cases where a large corpus of global training examples is available in the target domain (D-P1). The "augmented-domain" result refers to the cases when training examples are not available in the target domain so we trained our model on a noisy, weakly-labeled augmented training corpus (D-P2). The "Transfer Learning" refers to the cases where there are only a limited amount of training examples in the target domain, so we incorporate transfer learning (and data augmentation) to improve the training of our model (D-P3). The "Domain Adaptation" refers to the cases in which only a set of unlabeled data is available in the target domain, so we exploit them for adapting the local inference to the instances of the target domain (D-P4). As we can see, our approach has consistently outperformed all other inference alternative approaches (refer to Section 4.1) under similar data requirements, in all of these cases.

Table 18: Word-level precision of our model in problems with varying data availability profiles

In-Domain	**Augmented-Domain**	**Cross-Domain**	
(Section 4.2)	(Section 4.3)	Transfer Learning (+DA)	Domain Adaptation
		(Section 4.4)	(Section 4.5)
77.5%	74.2%	58.3% (64.1%)	45.2%

5.2 Sentence to Subsentence Inference

In Chapter 4 we investigated the generalizability of our single-word local inference framework to different resource-constrained problem domains. However, in many NLP problems, the size of the contributing local scope to the global prediction problem is not limited to a single-word. Let us consider the following examples in different domains:

Example 5.1:

Verbose Sentence: *"it is deja vu, all over again"*

Example 5.2:

Specific Sentence: *"nine children killed by landmine blast in Afghanistan"*
General Sentence: *"nine children killed in Afghanistan"*

Example 5.3:

Positive Sentence 1: *"they have [ridiculously*$^{--}$ *big*$^{+}$*]*$^{++}$ *sandwiches !"*
Positive Sentence 2: *'now with our kids menu, we are open to [all* $^{\times}$ *ages* $^{\times}$*]*$^{+}$*.*

In Example 5.1, the presence of the "deja vu" and "all over again" next to each other makes the context semantically verbose, so either of the two is pleonastic. However, at the word-level, none of the sentence's words are redundant. In Example 5.2, the phrase "landmine blast" expresses a specific piece of information about the cause of an incident, which makes the sentence *specific* rather than *general.* In Example 5.3, the superscripted symbols next to sentiment words indicate their polarity (+: positive, -: negative, ×: neutral). In the first sentence, the very positive phrase "ridiculously big", which is a combination of a negative and a positive word, casts the overall polarity of the sentence (very positive). In the second sentence, the phrase "all ages", which is a combination of two neutral words, makes the sentence express positive sentiment.

As we can see in these examples, the polarity of the sentence is influenced by phrases, and the polarity of the individual words within these phrases differs from the polarity of the sentence itself. Therefore, our model may fail to make appropriate local prediction inferences for such cases.

Referring back to Section 3.1, in our contrapositive learning framework, we infer a class label for a local context l_{ji} from the global context c_j, where $c_j = l_{1j}, l_{2j}, ..., l_{nj}$ (Equation (3.2) and Equation (3.5)). In Chapter 4, our assumption was that the local context was a single word (i.e., $l_{ji} = mji$)[1] and studied the applicability of our framework and transfer function model to various resource-constrained problems. Due to the fact that this assumption does not hold for many NLP problems and that phrasal constructions can also influence global predictions, the l_{ji} could be a sequence of words with an arbitrary length as follows:

$$c_j = \underbrace{m_{1j}, m_{2j}}_{l_{1j}}, \underbrace{m_{3j}}_{l_{3j}}, \underbrace{m_{4j}, m_{5j}, m_{6j}, m_{7j}}_{l_{4j}}, ..., \underbrace{m_{nj}}_{l_{nj}}$$

A simple method to operate on phrasal constructions include extracting sequences of words with fixed lengths (e.g., bigrams or trigrams) as local context; another option could be to extract all potential phrases from the parse tree of the sentence (Klein and Manning, 2003). However, both approaches produce irrelevant and redundant phrases, which poses a significant computational bottleneck for training an appropriate transfer function.

Therefore, finding an efficient set of contributing phrases is one of the key challenges of phrasal local inference. The answer to our high-level research question Q5, thus, could be investigated by attempting to answer the following more technical question:

Q5-1. *How to efficiently bundle single words into contributing phrases?*

In addition, there could be some cases in which the global prediction cannot be determined by a single word or a single phrase, or it is not clear which local feature is the major contributor. Let us consider the following examples:

Example 5.4:

Very Positive Sentence^{++}: *"a fast*$^{+}$ *, funny*$^{+}$ *, [highly enjoyable]*$^{+}$ *movie ."*

Example 5.5:

Negative Sentence^{-}: *"bland*$^{-}$ *murder-on-campus yawner*$^{-}$ *."*

[1] We denote words by m to not to be confused with w, which we used to denote weight

Example 5.4 shows a very positive sentiment. However, it is not clear which of the "fast", "funny", or "highly enjoyable" contributed the most to the polarity of the sentence? In this case, it appears that the mutual contribution of them amplifies and reinforces the polarity of the sentence. In Example 5.5, the words "bland" and "younger" are both contributing to the negative tone of the sentence. However, we can see that the contribution of multiple words are not necessarily strengthen (or weaken) the polarity of the sentence.

As shown in these examples, a global prediction can be influenced by more than one local feature. In such cases, a contrapositive relation between the global context and a single local feature could not be satisfied, so the local prediction inference would be inaccurate. Therefore, to answer Q5 and study the applicability of our contrapositive approach to problems in which global predictions correlate with more than one local feature (I-P2), we must first investigate an answer for the following technical question:

Q5-2. *How to find a set of local features with a combined contrapositive relation with the global prediction?*

5.2.1 Phrasal Transfer Function Model

Referring back to Section 2.4.1, self-attention is a mechanism to (semantically) relate each word of a *single* sentence to the different positions of the same sentence (Cheng et al., 2016; Vaswani et al., 2017). In other word, a self-attention vector α_i^s over the word m_i indicates how much each other word should be focused to represent m_i. For example, a self-attention score of $[.15, .82, .01, .01, ..., .001]$ for word m_{2j} means that m_{1j} also contributes in expressing the semantic purport of m_{2j}, so they may have some semantic dependency and form a phrasal expression l_{1j}, or the self-attention score of $[.001, .001, .01, .06, .10, .15, .66, ..., .001]$ for m_{7j} indicates $\{m_{4j}, m_{5j}, , m_{6j}, m_{7j}\}$ could be bundled as another phrasal item l_{4j}, as illustrated in the following example:

$$c_j = \underbrace{\overbrace{m_{1j}, m_{2j}}^{[.15,.82,.01,.01,...,.001]}}_{l_{1j}} , \underbrace{m_{3j}}_{l_{3j}}, \underbrace{\overbrace{m_{4j}, m_{5j}, m_{6j}, m_{7j}}^{[.001,.001,.01,.06,.10,.15,.66,...,.001]}}_{l_{4j}} , ..., \underbrace{m_{nj}}_{l_{nj}}$$

Therefore, we can incorporate self-attention weights as a signal to bundle relevant words into a contributing phrase (Q5-1). Taking Example 5.1 as a instance, while the individual representation of the words “deja” and “vu” encoded using the RNN encoder of our vanilla transfer function model may be very different, their representation from self-attention network would be more similar, and in the same time, representing their phrasal contribution as “deja vu”. Thus, if our transfer function model can operate on phrasal representations of the input produced by the self-attention layer: (i) it is more likely to be able to capture the semantic relatedness of the “deja vu” and “all over again,” and (ii) since it would be informed by the individual contribution of each word to their phrasal representation, it is more likely to rewrite all of the contributing words (e.g., rewrite both “deja” and “vu”) and make phrasal changes in an attempt to change the class prediction of the sentence.

A phrasal representation of l_{ij} in a global context c_j could be achieved by multiplying the word m_{ij} with its self-attention weight α_{ij}^s. So the global prediction function given in Equation (3.2) could be expanded as given in Equation (5.1).

$$
\begin{aligned}
\alpha_i^s &= \sum_{j=1}^{n} \frac{\exp(q_i.k_j)}{\sum_{m=1}^{n}\exp(q_i.k_m)} v_j && \text{; self-attention weights}\\
l_{ij} &= \alpha_{ij}^s m_{ij} && \text{; phrasal representation of input}\\
\mathcal{F}(c_j) &= \overrightarrow{w_j}.\overrightarrow{l_j} + b_j = \frac{1}{|c_j|}\sum_i w_{ij}\alpha_{ij}^s m_{ij} + b_j && \text{; global prediction function}\\
L_C &= \min_{\mathcal{F}} \frac{1}{|C|}\sum_j \mathcal{L}(y_j, \frac{1}{|c_j|}\sum_i w_{ij}\alpha_{ij}^s m_{ij} + b_j) && \text{; function optimization}
\end{aligned}
\tag{5.1}
$$

In order to integrate this global training framework to our transfer function, we need an encoder to learn a latent representation z from the phrasal representation of the input, and a discriminator that is able to guide the decoder on how to make localized phrasal rewriting to flip the global prediction. Figure 10 illustrate the structure of our transformer transfer function using self-attention layers in autoencoder and discriminator modules. We basically replaced our RNNs (GRU units) with transformers (Vaswani et al., 2017). These transforms include a self-attention layer, which is used to learn the phrasal spans, and a feed-forward layer to prepare the output (Q5-1).

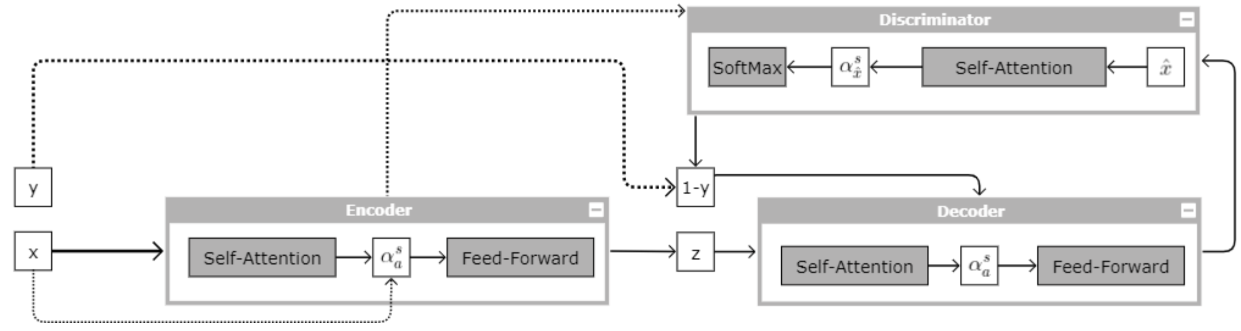

Figure 10: Our transformer-based transfer function model with self-attention layers

The reconstruction loss of our auto-encoder would be modified as Equation (5.2), where α^s indicates the self-attention weights and is calculated through Equation (2.12).

$$\mathcal{L}_{rec}(\theta_E, \theta_G) = \mathbb{E}[-\log p_G(c|q_E(z|\alpha^s m))] \tag{5.2}$$

Equation (5.3) shows the discrimination loss which operates on phrasal local scopes for global prediction. We optimize the model with the same objective function as given in Equation (3.9), following a similar training procedure: (i) separately train the auto-encoder and discriminator for some initial epochs, to learn the content preservation and greater-context prediction, respectively (ii) jointly train the auto-encoder and discriminator to learn the adversarial rewriting and class transference.

$$\mathcal{L}_{\text{disc}}(\theta_D) = \mathbb{E}_X[\log q_D(\{A|B\}|\alpha^s_{\hat{l}} m_{\hat{l}})] \tag{5.3}$$

5.2.1.1 Joint Contrapositive Contribution

One assumption in our contrapositive inference framework (Equation (3.5)) is that there is <u>one</u> local item, which replacing it with an adversarial alternative would flip the class prediction of the global problem. In order to answer Q5-2, therefore, we must first update our inference scheme to enable it to glean the combined contrapositive relationship between <u>multiple</u> local items and the corresponding global context.

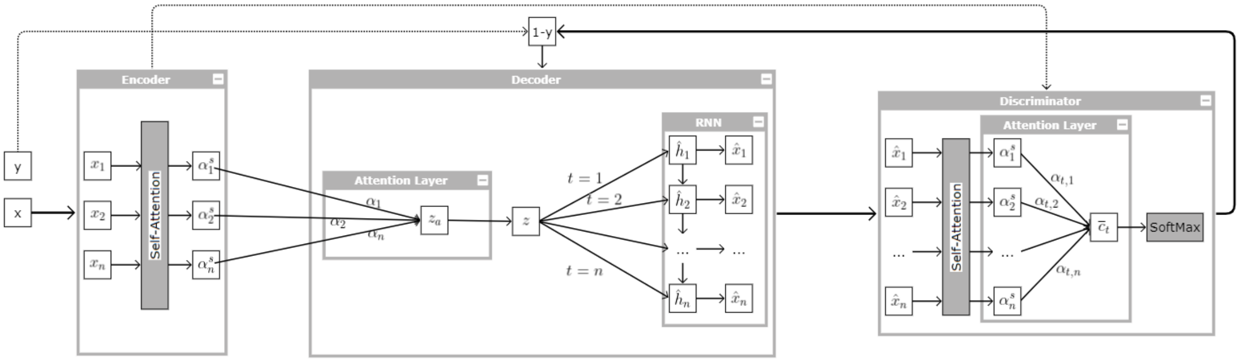

Figure 11: Our self-attending transfer function model

Equation (5.4) shows our contrapositive scheme for inference prediction for multiple local scopes, where $\{l_{ij}\}$ is a set of heavily contributing (phrasal) local items and $\{l^*_{ij}\}$ is the set of corresponding adversarial alternatives:

$$\begin{aligned} \mathcal{F}_X(\{l_{ij}\}) &= \mathcal{F}_C(c_j) \\ \exists \{l^*_{ij}\} \mid \quad \mathcal{F}_C(c_j^{i*}) &= 1 - y_j \end{aligned} \tag{5.4}$$

Referring back to Section 2.12, attention is mechanism to find the related positions in the *input* sequence when generating an *output* word (Bahdanau et al., 2015; Luong et al., 2015) An attention vector α_i indicates the relative contribution of the different parts of the sentence (words) to the generation of each output word. Therefore, we can utilize the attention mechanism to measure the relative (contrapositive) contribution of (phrasal) local items to the global prediction (Q5-2).

Figure 11 shows the structure of our transformer transfer function with attention. The self-attention layer in each encoder provides the phrasal representation over the potential input words. These local phrases then pass through an attention layer incorporated in the decoder, which at each time step, assigns an appropriate attention weight (α_i) to each item, based on its contribution to the generation of each output lexical item. These attention weights are then used to create a latent space z of contributing lexical items. The autoencoder's reconstruction loss of this model is given in Equation 5.5.

$$\begin{aligned}
\alpha_{t,i} &= \frac{\exp(\text{score}(s_{t-1}, h_i))}{\sum_{j=1}^{n} \exp(\text{score}(s_{t-1}, h_j))} && \text{; attention weights} \\
l_i &= \alpha_i^s m_i && \text{; phrasal representation of input} \\
\mathcal{L}_{rec}(\theta_E, \theta_G) &= \mathbb{E}[-\log p_G(c|q_E(z|\alpha_i))] && \text{; reconstruction loss}
\end{aligned} \tag{5.5}$$

We also incorporate an attention layer in the discriminator, so that it can learn the relative contribution of each local part of the sentence to its prediction to guide the generation of the decoder. Equation (5.6) shows the discriminator loss, where α is the attention weight for generated phrasal lexical items $\hat{l}$, which are calculated by applying the self-attention weights α^s to the generated words $\hat{m}$.

$$\begin{aligned}
\hat{l} &= \alpha^s \hat{m} \\
\mathcal{L}_{\text{disc}}(\theta_D) &= \mathbb{E}_X[\log q_D(\{A|B\}|\alpha\hat{l})]
\end{aligned} \tag{5.6}$$

Finally, we train our transfer function model by optimizing the same objective function, given in Equation (3.9). Since the latent spaces now contain the contribution information of all local items, our transformer-based transfer function model can identify the combined contribution of multiple local items to the global prediction, and accordingly determine the appropriate local rewriting (Q5-2).

5.2.1.2 Conciseness Regularization

Different problem domains may have different annotation profiles. The same domain may even be annotated differently by different researchers, so the span (and number) of contributing subsentential parts can vary greatly from one problem and resource to another. Despite the incorporation of transform units (self-attention) to learn to operate on phrasal representation, and attention layers to learn the contribution of different parts of the sentences to the global prediction, we still need a mechanism to somehow control the amount of change during the rewriting, based on the annotation profile of the problem and the associated resources.

In order to address that, we propose the *conciseness loss* ($\mathcal{L}_{con}$) to directly enforce the class transference using minimal changes, as given in Equation (5.7), where c is the original input greater-context and $\hat{c}$ is the transferred version of it to the other class. It should be noted that sometimes the original input and the transferred output may not have the same dimensions. We slice the bigger vector as the dimension of the smaller vector in such cases.

$$\mathcal{L}_{con}(\theta_E, \theta_G) = \mathbb{E}_{q_E(z|c)}[\log p(\hat{c}|z, \neg y)] \tag{5.7}$$

Equation (5.8) shows the transfer function's final objective function, where λ_D and λ_C are the balancing parameter. Regularizing the final objective function of our transfer function with this extra *conciseness loss* provides two major benefits: (i) satisfy the contrapositive constraint with minimal changes (Req. 2), thus, improving the local prediction accuracy; (ii) allows to control for the amount of local change during the global rewriting process, thus, adapting to the variability in the annotation profile and span size of the local and global contexts in different problem domains.

$$\underbrace{\min_{\theta_G} \mathcal{L}_{gen}}_{\text{Generation Loss}} = \underbrace{\mathcal{L}_{rec}}_{\text{Reconstruction Loss}} + \lambda_D \underbrace{\mathcal{L}_{disc}}_{\text{Discriminator Loss}} + \lambda_C \underbrace{\mathcal{L}_{con}}_{\text{Conciseness Loss}} \tag{5.8}$$

Similarly, we train our transfer function model with conciseness loss in two stages: (i) for some initial training runs, we train the discriminator and autoencoder independently ($\lambda_D = 0$); the discriminator will learn to distinguish between classes, and the autoencoder will learn a latent space that could be used to regenerate instances of both classes. At this step, we do not enforce the minimal changes as well ($\lambda_C = 0$). (ii) for a few more epochs, we jointly train the autoencoder and discriminator ($\lambda_D \neq 0$) to learn the class transference (replace y with $\neg y = 1 - y$) while enforcing the minimal required changes ($\lambda_C \neq 0$).

5.2.2 Evaluation

In this section, we study the generalizability of our contrapositive inference scheme to problems for which a larger scope of the text is responsible for the class prediction of the global sentence (I-P2). To provide an answer for the research question Q5, we evaluate our transformer-based transfer function model (Section 5.2.1) with attention (Section 5.2.1.1) and regularization for conciseness (Section 5.2.1.2]) on two sentence to subsentence inference prediction tasks: *rationale prediction* and *multi-word sentiment prediction*.

Rationale Prediction. Referring back to Section 2.1.4, a *rationale* is typically referred to as a subsentential scope of a sentence that influences an annotator's decision on whether to annotate the sentence with a certain class label Zaidan et al. (2007); Bao et al. (2018). In this task, we aim to *predict which part(s) of the sentence is the rationale for the polarity label of it?* As our training set, we use the IMDB dataset (Maas et al., 2011), which is a collection of 50K highly polar movie reviews (multiple sentences), crawled from the IMDB website. We automatically labeled sentences of this dataset with the sentiment polarity, using CoreNLP (Socher et al., 2013; Manning et al., 2014), to extract a balanced portion of 80K positive and negative sentences to form our training corpora with sentence-level labels. Zaidan et al. (2007) annotated 1.8K held-out reviews of the IMDB dataset with subsentential rationale labels, which are the segments of the sentence that are the reason that the sentence has a certain polarity. We use this dataset with subsentential level annotation as the benchmark dataset to evaluate the inference performance of the different methods in our experiment.

Multi-Word Sentiment Prediction. This task and its experimental setting are similar to our experiment in Section 4.2. However, this time, we do not filter for YPD sentences with only one sentiment word. Both our train and test datasets include sentences with multiple sentiment words. To create our benchmark dataset, we extracted a unigram pool from all the words of the test set; then randomly picked a word and annotated it as *very positive*, *positive*, *neutral*, *negative*, or *very negative* to collect a set of 900 held-out sentences with *very positive* or *very negative* words, as our in-house benchmark dataset.

The *Direct:Rationale* approach (Lei et al., 2016) is specifically designed for rationale extraction task so we just tuned the *sparsity* and *coherency* parameters of the model on a

Table 19: Subsentential (and single-word) inference precision of inference approaches

Inference Approach	**Rationale Prediction** (IMDB)	**Multi-Word Sentiment** (Yelp)	**Single-Word Sentiment** (Yelp §4.2)	
Direct:MIL	54.3%	54.2%	58.4%	
Direct:Rationale	62.1%	59.5%	55.6%	
Contrapositive:LOO	59.2%	63.1%	66.2%	
Contrapositive:TF	*65.6%*	*66.6%*	*77.5%*	BLEU: .71
Contrapositive:TF+	**77.6%**	**75.4%**	**81.8%**	BLEU: **.74**

validation set. In the *Direct:MIL* baseline, we picked about 20% (ratio of the length of the rationales to the length of the sentences) of the words of the sentence as local features based on their attention weight rank. In the *Contrapositive:LOO* baseline, we extracted the syntactically plausible phrases no longer than 20% of the sentence length, from its dependency parse tree (using CoreNLP), as potential local features and examined the global prediction by removing them from the sentence. For the multi-word sentiment prediction task, we tuned the balancing parameter of the conciseness loss (λ_C in Equation (5.8)) on a validation set, which controls the amount of change to the original sentence during the rewriting and class transference process. The hyperparameters used in this experiment are described in Appendix A.2.

Table 19 shows the precision of predicting the rationale for the polarity of sentences (IMDB), and the sentiment words of the highly polar sentences (Yelp). The results show that even with an increase in the size of the local span (and consequently an increase in the complexity of semantic relations) the contrapositive inference models outperform alternative methods on both local subsentential inference tasks, which suggest that our contrapositive inference scheme is effectively applicable to inference granularity with varying local span sizes (Q5). Surprisingly, even the *Contrapositive:LOO* baseline, which is a simple model with partial implementation of the contrapositive constraint outperformed the task-specific

Direct:Rationale baseline on the rationale prediction task.

To better evaluate the effectiveness of augmenting our model with attention and custom regularization (Section 5.2.2 and Section 5.2.1.2), we perform an ablation test by including the results from our vanilla transfer function model (*Contrapositive:TF*) and our phrasal transfer function model with self-attention and conciseness regularization (*Contrapositive:TF+*). Despite the small precision gap between our vanilla model and direct inference approaches, the (*Contrapositive:TF+*) model makes a striking improvement on predicting the rationales and sentiment words.

Additionally, we apply our phrasal model to the single-word sentiment prediction task of Section 4.2. As shown in Table 19, our transformer-based model yields 3.3% higher precision on predicting the most polar word in Yelp sentences compared to our vanilla model. Regularizing for rewriting conciseness also improved the generation BLUE score of our model from .71 to .73. These ablation analyses indicate that our transformer-based transfer function model is able to make accurate phrase-level predictions (Q5-1), and regularization for the conciseness of rewriting with multiplicative attention (Luong et al., 2015) facilitates the prediction of jointly-contributing local items.

5.3 Paragraph to Sentence Inference

Some NLP applications may analyze the text beyond the scope of a single sentence. For example, a buyer may write a (few) paragraph of text for a product review. These reviews can then be used by a seller to determine what aspects of the product need to be improved. In this section, we study a case for such applications (I-P3 – paragraph to sentence inference) and evaluate the generalizability of our contrapositive inference scheme to inference prediction problems from larger global text spans (Q6).

It may be easier to infer a local prediction for a larger context, since there is enough material to form a distinctive semantic unit, and identifying the most contributing feature in a smaller set of features (e.g., a sentence in a paragraph) is intuitively easier than in a larger set of features (e.g., a word in a sentence). Therefore, we expect that this experiment will be an easier inference task compared to the inference from sentences to words and phrases. We use a portion of the BeerAdvocate review dataset that is used by Lei et al. (2016), which contains 67K multi-aspect reviews. Each review has a normalized rating for each aspect of a beer, and a written review paragraph. Around 1K held-out reviews of the dataset are annotated by McAuley et al. (2012) with aspect labels for each sentence, which could be used as the benchmark dataset to evaluate the inference performance of the different methods.

We opt to experiment with the "appearance" aspect and we transformed the ratings into binary classes by labeling scores greater than .6 as *positive* sentiment and scores lower than or equal to .5 as *negative* sentiment. Considering the availability of the annotations, the prediction tasks at different levels of granularity could be set up as follows:

- **Global prediction task:** predict whether a review (a paragraph) expresses a positive sentiment towards the "appearance" aspect of a beer or not?
- **Local prediction task:** predict which sentence of the review is expressing the writer's sentiment towards the "appearance" of the beer?

We trained all of the inference approaches on the BeerAdvocate reviews with the label information only at the paragraph level. For generation-based approaches (*Direct:Rationale* and *Contrapositive:TF*), the extracted contributing local feature might be smaller than a sentence (or spread across multiple sentences). For these cases, similar to the evaluation scheme introduced by Lei et al. (2016), we consider the sentence(s) containing the extracted subsentential parts as the predicted sentence.

Table 20 shows the precision of predicting the sentence that talks about the "appearance" aspect, regardless of its polarity. As expected, for this relatively easy inference problem, both direct and contrapositive inference approaches are able to achieve a high localized (sentence) prediction precision. We can see that the *Direct:Rationale* baseline performs slightly better than our vanilla model (*Contrapositive:TF*). This is probably due to the fact that this baseline was developed and tuned specifically for this task and dataset by Lei et al. (2016). Nevertheless, we are able to take back the top spot for local inference using our transformer-based model (*Contrapositive:TF+*). It should be noted that in this experiment we did not regularize our transformer-based model for conciseness loss (i.e., $\lambda_C = 0$ in both

Table 20: Paragraph to sentence inference precision of different inference approaches

Inference Approach	**Inference Precision**
Direct:MIL	79.7%
Direct:Rationale	93.8%
Contrapositive:LOO	82.8%
Contrapositive:TF	92.2%
Contrapositive:TF+	94.3%

pre-training and generation training phases.).

From the computational efficiency perspective, an epoch of training of our model on an NVIDIA Tesla P100 GPU took roughly 45 minutes in all of our sentence to words and phrases inference experiments (refer to Chapter 4, Section 5.2, and Section 5.2). A training epoch of our model for the paragraph to inference task, however, takes roughly 2:30 hours, using the same hardware. Therefore, although inferring a contrapositive relationship between larger local and global context is relatively easy, processing a larger text span requires more computation resources, so applying our contrapositive approach to larger than paragraph context might not be computationally feasible.

5.4 Summary and Discussion

Experimental results demonstrate that the incorporating attention mechanism along with custom rewriting regularization can facilitate the inference prediction for larger local and global contexts (Q5 and Q6), and makes our contrapositive scheme is generalizable to problems with different levels of inference granularity for the size of the local and global contexts.

Similar to the prior work observations (Ribeiro et al., 2016; Ross et al., 2017; Jain and Wallace, 2019), we also found that semantically irrelevant words, including "punctuation marks", are the heavy contributors to the global prediction, suggesting that attention weight might not directly correlate with the semantics of the prediction task.

The *Direct:Rationale* baseline operates through finding a smaller local text span (rationale) that can replace the greater context in the global prediction. Intuitively, the larger rationales, which may carry a distinctive semantic, can successfully train the global prediction, as we observed in our paragraph to sentence inference experiment (Section 5.3). However, rationales with shorter text span (e.g., words or phrases) are likely to provide a noisy and unstable solution for the global problem, as observed in our sentence to word inference experiment (Section 5.1).

The *Contrapositive:LOO* baseline adapts a simple approach to apply a relaxed version of the contrapositive constraint, by removing the local rationale from the global problem's context. A global context, after removing a small text span from it, would still be large and semantically expressive enough for a reliable global prediction, as observed in our sentence to subsentence and word inference experiments (Section 5.1 and Section 5.2).

Since the *Contrapositive:TF* baseline operates by replacing a local rationale of a global context with another (adversarial) rationale, the span size of the greater context remains relatively intact, which makes the prediction inference more robust to variability in the size of the local and global context and their distance, as we observer our experiments with different inference granularity levels (Section 5.1, Section 5.2, and Section 5.3) Moreover, training the model to apply the local contrapositive semantic perturbation to the global context makes it more robust to noise and irrelevant features. For example, a punctuation mark might influences a class prediction for a sentence, however, replacing it with another punctuation mark will not make the sentence belongs to another class, so it is not a *semantically* contributing word.

6.0 Conclusion

This thesis presented a contrapositive scheme for inferring local predictions from class predictions of the corresponding global problem. We argued that the contrapositive inference scheme can be implemented as a transfer function that learns to rewrite a global instance from one class to another (Chapter 3). Experimental results (Section 4.2) support the first hypothesis of the thesis (**H1**) that contrapositive constraint allows for the identification of semantically relevant local features and improves the robustness of the classification training. It also validates the second hypothesis (**H2**) that our proposed transfer function outperforms the alternative inference methods under similar data requirements.

Training such a transfer function requires a large training corpus of global prediction, which may not be available for many resource-constrained problem domains. In Chapter 4, we demonstrated the robustness of our proposed contrapositive local inference approach to resource-constrained problems, when coupled with appropriate *data augmentation* methods (Section 4.3) , as well as its *transferability* (Section 4.4) and *adaptability* (Section 4.5) to problem domains with limited availability of the training corpora. The experimental results (Section 4.3.3, Section 4.4.3, and Section 4.5.3) validated the third hypothesis of the thesis (**H3**) that the our approach is generalizable to problem domains with varying data availability profiles.

The relative size of the local and global context and the annotation profiles of the prediction tasks can pose different challenges for our inference framework. In Chapter 5, we presented a transformer-based transfer function model that facilitates the application of contrapositive constraint to different levels of inference granularity. The results of out experiments (Section 5.1, Section 5.2.2, and Section 5.3) demonstrated the robustness of our proposed approach to variability in the size of the local and global contexts and validated the fourth, and last, hypothesis (**H4**). The following is a summary of our contributions in this thesis:

- We have proposed that the local category inference from the corresponding global problem can be made more robust if it follows the contrapositive constraint.

- We have shown that the contrapositive inference scheme can be modeled after paraphrasing and class transference at global context level and implemented as a deep generative transfer function that learns to transform a context from one class to another.
- We have proposed a *non-label-reserving data augmentation scheme*, which removes the prior work's assumption of having labeled samples for instances of all classes, therefore, being applicable to a wider range of problem domains.
- We have found that the suitability of an augmentation heuristic for a classification (inference) task correlates to the extent to which it generates "hard to distinguish" examples, which can be measured as the amount of divergence from one augmented class distribution to another augmented class distribution.
- We have demonstrated that the contrapositive constraint made the semantic relationship between the local and global problems more robust to noise, so even a weakly-labeled augmented corpus can train an appropriate transfer function.
- We have found that the local-global contrapositive relationship can be learned from a domain with more training resources and then applied to a domain with limited resources.
- We have introduced a domain-adaptable transfer function that can utilize unlabeled examples to learn a similar feature space for different domains to make more robust cross-domain local inference predictions.
- We have presented a transformer-based transfer that integrates a self-attention autoencoder to support phrasal contrapositive inference.
- We have shown that custom regularization for conciseness of rewriting improves the prediction of jointly-contributing local items.

6.1 Limitations and Future Work

While the contrapositive inference scheme provides an efficient solution for resource-constrained local prediction problems, it has some limitations as a full-fledged classifier:

- While our local prediction inference approach does not require labeled examples at local level, developing a transfer function still requires a large corpus of training examples at global level. This may hinder the application of our approach to problems for which such global training corpora do not exist or are hard to obtain.
- A contrapositive inference can be applied only when there is a corresponding global prediction task. For example, our approach is not suitable for predicting the parts of speech (POS) of words or for identifying named entities in a sentence, in which a clear corresponding sentence-level prediction task is not available.
- A Contrapositive inference is not appropriate for cases where there are multiple local features with contrasting class labels within a global context. For example, our approach may fail to detect negative words in a positive sentence, or double negative expressions.
- In problems where the classes are not independent of each other, for example "positive" and 'very positive" that have considerable overlap, meeting the contrapositive constraint can be challenging.
- In our transfer function implementation, we are using a general classifier as the discriminator, which may fail to correctly predict a class label for more subtle instances of a global classification task. For example, while sentence polarity classification may seem straightforward, handling negations can be challenging, and using a single classifier to predict both simple cases (e.g., positive sentences) and subtle cases (e.g., double negatives or negations) may result in noisy class predictions. For such classification tasks, researchers are incorporating custom classifiers to handle more subtle or edge cases, which can make the overall classification more robust (Jia et al., 2009; Wiegand et al., 2010; Barnes et al., 2021).

Therefore, as part of the future work, it would be insightful to investigate how does incorporating task-specific global classifiers in our transfer function affects its contrapositive local prediction inference on the same task.

- Currently, our contrapositive inference scheme is applicable only to "binary" classifications and may not be fully generalizable to "multi-class" classifications. Similar to SVM and other well-known models, which were initially developed as binary classifiers, we might be able to extend the contrapositive constraint to more general classification settings. As part of the feature work, we may be able to train a multi-class contrapositive classifier following an *one-vs-one* training strategy. For example, to classify emotions (Happy, Sad, Angry, and Calm), we can train a transfer function for *Happy-Sad* transference and another for *Angry-Calm* transference. Then, based on the discriminator's global prediction for a given sentence, the appropriate transfer function would trigger to perform the rewriting and infer the local prediction.

Appendix Hyperparameters

A.1 Sentence to Word Inference Experiments

Direct:MIL: attention weight over single words.

Direct:Rationale

- Embedding: GloVe + Yelp
- $\lambda_1 = 1e - 2$
- $\lambda_2 = 2\lambda_1$

Contrapositive:LOO: single words as local text span.

Contrapositive:TF:

	Pre-training	Joint-training
#epochs	10	2
λ_D	.0	1e-1
λ_C	.0	1e-2

A.2 Sentence to Subsentence Inference Experiments

Direct:MIL: pick top 20% words ranked by attention weight.

Direct:Rationale:

- Embedding: GloVe + IMDB
- $\lambda_1 = 1e - 3$
- $\lambda_2 = 2\lambda_1$

Contrapositive:LOO: local context: phrases extracted from the sentence parse tree, where their length is smaller that 20% of the sentence length

Contrapositive:TF:

	Pre-training	Joint-training
#epochs	10	3
λ_D	.0	1e-1
λ_C	.0	1e-2

A.3 Paragraph to Sentence Inference Experiment

Direct:MIL: Attention weight: average over sentence.

Direct:Rationale:

- Embedding: word2vec + BeerAdvocate
- $\lambda_1 = 4e - 4$ (Sparsity)
- $\lambda_2 = 2\lambda_1$ (Coherence)

Contrapositive:LOO: the whole sentence as local context.

Contrapositive:TF:

	Pre-training	Joint-training
#epochs	15	5
λ_D	.0	1e-1
λ_C	.0	0 – we did not enforce the conciseness in this experiment

Bibliography

Agirre, E., Gonzalez-Agirre, A., Lopez-Gazpio, I., Maritxalar, M., Rigau, G., and Uria, L. (2016). SemEval-2016 Task 2: Interpretable Semantic Textual Similarity. In *SemEval*.

Akbik, A., Blythe, D., and Vollgraf, R. (2018). Contextual String Embeddings for Sequence Labeling. In *COLING*.

Alpaydin, E. (2020). *Introduction to Machine Learning*. MIT press, fourth edition.

Anaby-Tavor, A., Carmeli, B., Goldbraich, E., Kantor, A., Kour, G., Shlomov, S., Tepper, N., and Zwerdling, N. (2020). Not Enough Data? Deep Learning to the Rescue! In *AAAI*.

Argyriou, A., Micchelli, C. A., Pontil, M., and Ying, Y. (2008). A Spectral Regularization Framework for Multi-Task Structure Learning. In *NIPS*.

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein GAN. In *ICML*.

Arnold, A., Nallapati, R., and Cohen, W. W. (2007). A comparative study of methods for transductive transfer learning. In *ICDMW*.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural Machine Translation by Jointly Learning to Align and Translate. In *ICLR*.

Bao, Y., Chang, S., Yu, M., and Barzilay, R. (2018). Deriving Machine Attention from Human Rationales. In *EMNLP*.

Barnes, J., Velldal, E., and Ovrelid, L. (2021). Improving sentiment analysis with multi-task learning of negation. *Natural Language Engineering*, 27(2):249–269.

Beigman Klebanov, B., Wee Leong, C., and Flor, M. (2015). Supervised Word-Level Metaphor Detection: Experiments with Concreteness and Reweighting of Examples. In *Workshop on Metaphor in NLP*, pages 11–20.

Benaim, S. and Wolf, L. (2017). One-Sided Unsupervised Domain Mapping. In *NIPS*.

Bengio, S., Vinyals, O., Jaitly, N., and Shazeer, N. (2015). Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks. In *NIPS*, volume 2015-January. Neural information processing systems foundation.

Blair, C. R. (1960). A program for correcting spelling errors. *Information and Control*, 3(1):60–67.

Blitzer, J., Dredze, M., and Pereira, F. (2007). Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In *ACL*. Association for Computational Linguistics.

Bonilla, E. V., Chai, M. A., and Williams, C. K. I. (2008). Multi-task Gaussian Process Prediction. In *NIPS*.

Bowman, S. R., Luke, V., Oriol, V., Dai, A. M., Jozefowicz, R., and Bengio, S. (2016). Generating Sentences from a Continuous Space. In *CoNLL*, pages 10–21.

Brants, T. (2000). TnT: a statistical part-of-speech tagger. In *ANLPC*.

Briggs, F., Fern, X. Z., and Raich, R. (2012). Rank-loss support instance machines for MIML instance annotation. In *KDD*, pages 534–542.

Bunescu, R. C. and Mooney, R. J. (2007). Learning to Extract Relations from the Web using Minimal Supervision. In *ACL*, pages 576–583.

Carreras, X. and Arquez, L. M. (2005). Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling. In *CoNLL*.

Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N., St John, R., Constant, N., Guajardo-Céspedes, M., Yuan, S., Tar, C., Sung, Y.-H., Strope, B., and Kurzweil Google Research Mountain View, R. (2018). Universal Sentence Encoder. *Computing Research Repository*, arXiv:1803.11175.

Chen, M., Weinberger, K. Q., and Blitzer, J. C. (2011). Co-Training for Domain Adaptation. In *NIPS*.

Chen, M., Weinberger, K. Q., Xu, Z., and Sha, F. (2015). Marginalizing Stacked Linear Denoising Autoencoders. *Journal of Machine Learning Research*, 16(116):3849–3875.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. (2016). InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In *NIPS*.

Chen, Y., Bi, J., and Wang, J. Z. (2006). MILES: Multiple-instance learning via embedded instance selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):1931–1947.

Cheng, J., Dong, L., and Lapata, M. (2016). Long Short-Term Memory-Networks for Machine Reading. In *EMNLP*.

Cheplygina, V., Tax, D. M., and Loog, M. (2015). Multiple instance learning with bag dissimilarities. *Pattern Recognition*, 48(1):264–275.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *EMNLP*, pages –.

Choi, J. D. (2016). Dynamic Feature Induction: The Last Gist to the State-of-the-Art. In *NAACL*.

Cinbis, R. G., Verbeek, J., and Schmid, C. (2017). Weakly Supervised Object Localization with Multi-Fold Multiple Instance Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(1):189–203.

Collins, M. (2002). Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In *EMNLP*.

Crystal, D. (2008). *A Dictionary of Linguistics and Phonetics.* Wiley, 6th edition edition.

Dahlmeier, D. and Ng, H. T. (2011). Grammatical Error Correction with Alternating Structure Optimization. In *Proceedings of the 49th Annual Meeting of the ACL-HLT*, pages 915–923.

Dahlmeier, D., Ng, H. T., and Wu, S. M. (2013). Building a Large Annotated Corpus of Learner English: The NUS Corpus of Learner English. In *Workshop on Innovative Use of NLP for Building Educational Applications*, pages 22–31.

Dai, W., Yang, Q., Xue, G. R., and Yu, Y. (2007). Boosting for transfer learning. In *ICML*, volume 227, New York, New York, USA. ACM Press.

Damerau, F. J. (1964). A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 7(3):171–176.

Devlin, J., Chang, M.-W., Lee, K., and Kristina Toutanova (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, pages 4171–4186.

Dietterich, T. G., Lathrop, R. H., and Lozano-Pérez, T. (1997). Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71.

Du, M., Liu, N., Yang, F., and Hu, X. (2019). Learning Credible Deep Neural Networks with Rationale Regularization. In *ICDM*.

Ehsan, U., Harrison, B., Chan, L., and Riedl, M. O. (2018). Rationalization: A Neural Machine Translation Approach to Generating Natural Language Explanations. In *AIES*.

Evgeniou, T. and Pontil, M. (2004). Regularized multi-task learning. In *KDD*, New York, New York, USA. Association for Computing Machinery (ACM).

Fadaee, M., Bisazza, A., and Monz, C. (2017). Data Augmentation for Low-Resource Neural Machine Translation. In *NAACL*.

Feng, J. and Zhou, Z.-H. (2017). Deep MIML Network *. In *AAAI*.

Fernando, B., Habrard, A., Sebban, M., and Tuytelaars, T. (2013). Unsupervised visual domain adaptation using subspace alignment. In *CVPR*, pages 2960–2967. Institute of Electrical and Electronics Engineers Inc.

Ficler, J. and Goldberg, Y. (2017). Controlling Linguistic Style Aspects in Neural Language Generation. In *Workshop on Stylistic Variation*.

Francis, W. N. and Kucera, H. (1979). Brown Corpus Manual. *Letters to the Editor*, 5(2):7.

Frénay, B. and Verleysen, M. (2014). Classification in the Presence of Label Noise: a Survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):845.

French, G., Mackiewicz, M., and Fisher, M. (2018). SELF-ENSEMBLING FOR VISUAL DOMAIN ADAPTATION. In *ICLR*.

Fu, Z., Tan, X., Peng, N., Zhao, D., and Yan, R. (2018). Style Transfer in Text: Exploration and Evaluation. In *AAAI*.

Ganchev, K., Graça, J., Gillenwater, J., and Taskar, B. (2010). Posterior Regularization for Structured Latent Variable Models. *Journal of Machine Learning Research*, 11:2001–2049.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V., Dogan, U., Kloft, M., Orabona, F., Tommasi, T., and Ganin, a. (2016). Domain-Adversarial Training of Neural Networks. *Journal of Machine Learning Research*, 17:1–35.

Gao, J., Fan, W., Jiang, J., and Han, J. (2008). Knowledge transfer via multiple model local structure mapping. In *KDD*, New York, New York, USA. ACM Press.

Ghifary, M., Bastiaan Kleijn, W., and Zhang, M. (2015). Domain adaptive neural networks for object recognition. In *ICML*.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach. In *ICML*.

Gong, B., Shi, Y., Sha, F., and Grauman, K. (2012). Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, pages 2066–2073.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative Adversarial Nets. In *NIPS*.

Goyal, A., Lamb, A., Zhang, Y., Zhang, S., Courville, A., and Bengio, Y. (2016). Professor Forcing: A New Algorithm for Training Recurrent Networks. In *NIPS*.

Grandvalet, Y. and Bengio, Y. (2004). Semi-supervised Learning by Entropy Minimization. In *NIPS*.

Greene, B. B. and Rubin, G. M. (1971). *Automatic grammatical tagging of English*. Department of Linguistics, Brown University.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Smola, A., Schölkopf, B., and Smola GRETTON, A. (2012). A Kernel Two-Sample Test Bernhard Schölkopf. *Journal of Machine Learning Research*, 13(25):723–773.

Grishma, R. (1995). The NYU System for MUC-6 or Where's the Syntax ? In *Message Understanding Conference*.

Guo, H., Mao, Y., and Zhang, R. (2019). Augmenting Data with Mixup for Sentence Classification: An Empirical Study. *Computing Research Repository*.

Guo, H., Pasunuru, R., and Bansal, M. (2020). Multi-Source Domain Adaptation for Text Classification via DistanceNet-Bandits. In *AAAI*.

Haeusser, P., Frerix, T., Mordvintsev, A., and Cremers, D. (2017). Associative Domain Adaptation. In *ICCV*.

Han, N.-R., Chodorow, M., and Leacock, C. (2006). Detecting errors in English article usage by non-native speakers. *Natural Language Engineering*, 12(02):115–129.

He, L., Lee, K., Lewis, M., Zettlemoyer, L., and Allen, P. G. (2017). Deep Semantic Role Labeling: What Works and What's Next. In *ACL*.

Hendrycks, D., Mazeika, M., Wilson, D., and Gimpel, K. (2018). Using Trusted Data to Train Deep Networks on Labels Corrupted by Severe Noise. In *NIPS*.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Honnibal, M. and Montani, I. (2017). SpaCy 2: Natural Language Understanding with Bloom Embeddings, Convolutional Neural Networks and Incremental Parsing. *To appear*.

Hu, M. and Liu, B. (2004). Mining and Summarizing Customer Reviews. In *KDD*.

Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R., and Xing, E. P. (2017). Toward Controlled Generation of Text. In *ICML*.

Hu, Z., Yang, Z., Salakhutdinov, R., and Xing, E. (2018). On Unifying Deep Generative Models. In *ICLR*.

Huang, J., Smola, A. J., Gretton, A., Borgwardt, K. M., and Scholkopf, B. (2006). Correcting sample selection bias by unlabeled data — Proceedings of the 19th International Conference on Neural Information Processing Systems. In *NIPS*.

Ilse, M., Tomczak, J. M., and Welling, M. (2018). Attention-based Deep Multiple Instance Learning. In *ICML*.

Iyyer, M., Wieting, J., Gimpel, K., and Zettlemoyer, L. (2018). Adversarial Example Generation with Syntactically Controlled Paraphrase Networks. In *NAACL*.

Jain, S. and Wallace, B. C. (2019). Attention is not Explanation. In *EMNLP*.

Jang, E., Gu, S., and Poole, B. (2017). Categorical Reparameterization With Gumbel-Softmax. In *ICLR*.

Jebara, T. (2004). Multi-task feature and kernel selection for SVMs. In *ICML*, New York, New York, USA. ACM Press.

Jhamtani, H., Gangal, V., Hovy, E., and Nyberg, E. (2017). Shakespearizing Modern Language Using Copy-Enriched Sequence-to-Sequence Models. In *Workshop on Stylistic Variation*.

Jia, L., Yu, C., and Meng, W. (2009). The effect of negation on sentiment analysis and retrieval effectiveness. In *CIKM*, pages 1827–1830.

Jiang, J. and Zhai, C. (2007). Instance Weighting for Domain Adaptation in NLP. In *ACL*.

Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., and Liu, Q. (2019). TinyBERT: Distilling BERT for Natural Language Understanding. *Computing Research Repository*.

Johnson, M. (2009). How the statistical revolution changes (computational) linguistics. In *EACL*.

Kandemir, M., Haußmann, M., Diego, F., Rajamani, K., Van Der Laak, J., Nl, J. V., and Hamprecht, F. A. (2016). Variational Weakly Supervised Gaussian Processes. In *BMVA*.

Kashefi, O., Afrin, T., Dale, M., Olshefski, C., Godley, A., Litman, D., and Hwa, R. (2022). ArgRewrite V.2: an annotated argumentative revisions corpus. *Language Resources and Evaluation*, pages 1–35.

Kashefi, O. and Hwa, R. (2020). Quantifying the Evaluation of Heuristic Methods for Textual Data Augmentation. In *WNUT-EMNLP*, pages 200–208.

Kashefi, O. and Hwa, R. (2021). Contrapositive Local Class Inference. *EMNLP-WNUT*.

Kashefi, O., Lucas, A. T., and Hwa, R. (2018). Semantic Pleonasm Detection. In *NAACL (Volume 2: Short Papers)*, pages 225–230.

Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. *EMNLP*.

Kingma, D. P. and Welling, M. (2013). Auto-Encoding Variational Bayes. In *ICLR*.

Klein, D. and Manning, C. D. (2003). Accurate unlexicalized parsing. In *ACL*, volume 1.

Ko, W.-J., Durrett, G., and Li, J. J. (2019). Domain Agnostic Real-Valued Specificity Prediction. In *AAAI*.

Kobayashi, S. (2018). Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations. In *NAACL*.

Krupka, G. R. and Hausman, K. (1998). Description of the NetOwl™ Extractor System as Used for MUC-7. In *Message Understanding Conference*.

Kullback, S. and Leibler, R. A. (1951). On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

Kumar, V., Choudhary, A., and Cho, E. (2020). Data Augmentation using Pre-trained Transformer Models. *Computing Research Repository*.

Lamb, A., Goyal, A., Zhang, Y., Zhang, S., Courville, A., and Bengio, Y. (2016). Professor Forcing: A New Algorithm for Training Recurrent Networks. In *NIPS*.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural Architectures for Named Entity Recognition. In *NAACL*.

Lavie, A. (2011). Evaluating the Output of Machine Translation Systems - ACL Anthology. In *MT Summit XIII*.

Lawrence, N. D. and Platt, J. C. (2004). Learning to Learn with the Informative Vector Machine. In *ICML*, New York, New York, USA. ACM Press.

Le, Q. and Mikolov, T. (2014). Distributed Representations of Sentences and Documents. In *ICML*.

Lee, S.-I., Chatalbashev, V., Vickrey, D., and Koller, D. (2007). Learning a Meta-Level Prior for Feature Relevance from Multiple Related Tasks. In *ICML*.

Lehmann, C. (2005). Pleonasm and hypercharacterisation. In *Yearbook of Morphology*, pages 119–154. Springer, Dordrecht.

Lei, T., Barzilay, R., and Jaakkola, T. (2016). Rationalizing Neural Predictions. In *EMNLP*.

Lewis, M. (1993). *The Lexical Approach*, volume 1. Hove, Language Teaching Publications.

Li, J. J. and Nenkova, A. (2015). Fast and Accurate Prediction of Sentence Specificity. In *AAAI*, pages 2281–2287.

Li, N., Hao, H., Gu, Q., Wang, D., and Hu, X. (2017). A transfer learning method for automatic identification of sandstone microscopic images. *Computers and Geosciences*, 103:111–121.

Li, W., Duan, L., Xu, D., and Tsang, I. W. (2014). Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1134–1148.

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–184.

Liu, G., Wu, J., Zhou, Z.-H., Hoi, S. C. H., and Buntine, W. (2012). Key Instance Detection in Multi-Instance Learning. In *JMLR*, volume 25, pages 253–268.

Liu, P., Qiu, X., and Huang, X. (2016). Recurrent Neural Network for Text Classification with Multi-Task Learning. In *IJCAI*.

Long, M., Zhu, H., Wang, J., and Jordan, M. I. (2017). Deep transfer learning with joint adaptation networks. In *ICML*, volume 8862. Springer Verlag.

Louis, A., Dipper, S., Zinsmeister, H., and Webber, B. (2013). A corpus of science journalism for analyzing writing quality. *Dialogue and Discourse*, 4(2):87–117.

Louis, A. and Nenkova, A. (2012). A Corpus of General and Specific Sentences from News. In *LREC*.

Lugini, L. and Litman, D. (2018). Predicting Specificity in Classroom Discussion. In *Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–61.

Luo, Z., Zou, Y., Hoffman, J., and Fei-Fei, L. (2017). Label Efficient Learning of Transferable Representations across Domains and Tasks. In *NIPS*, volume 2017-December. Neural information processing systems foundation.

Luong, M.-T., Pham, H., and Manning, C. D. (2015). Effective Approaches to Attention-based Neural Machine Translation. In *EMNLP*.

Luque, F. M. (2019). Atalaya at TASS 2019: Data Augmentation and Robust Embeddings for Sentiment Analysis. In *CEUR Workshop Proceedings*, volume 2421. CEUR-WS.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning Word Vectors for Sentiment Analysis. In *ACL*.

Mahadevan, S., Mishra, B., and Ghosh, S. (2018). A unified framework for domain adaptation using metric learning on manifolds. In *ECML-PKDD*, volume 11052 LNAI. Springer Verlag.

Makhzani, A., Shlens, J., Jaitly, N., Brain, G., Openai, I. G., and Frey, B. (2016). Adversarial Autoencoders. In *ICLR*.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and Mcclosky, D. (2014). The Stanford CoreNLP Natural Language Processing Toolkit. In *ACL: System Demonstrations*.

Maron, O. and Lozano-Pérez, T. (1997). A Framework for Multiple-Instance Learning. In *NIPS*.

Marshall, I. J., Kuiper, J., and Wallace, B. C. (2016). RobotReviewer: evaluation of a system for automatically assessing bias in clinical trials. *Journal of the American Medical Informatics Association*, 23(1):193–201.

McAuley, J., Leskovec, J., and Jurafsky, D. (2012). Learning attitudes and attributes from multi-aspect reviews. In *ICDM*.

Mccallum, A. and Li, W. (2003). Early Results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-Enhanced Lexicons. In *NAACL*.

Mcilroy, M. D. (1982). Development of a Spelling List. *IEEE Transactions on Communications*, 30(1):91–99.

Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

Mitton, R. (1987). Spelling checkers, spelling correctors and the misspellings of poor spellers. *Information Processing and Management*, 23(5):495–505.

Mitton, R. (2010). Fifty years of spellchecking. *Writing Systems Research*, 2(1):1–7.

Moreo, A., Romero, M., Castro, J. L., and Zurita, J. M. (2012). Lexicon-based Comments-oriented News Sentiment Analyzer system. *Expert Systems with Applications*, 39(10):9166–9180.

Mudrakarta, P. K., Taly, A., Sundararajan, M., and Dhamdhere, K. (2018). Did the model understand the question? In *ACL*.

Mueller, J., Gifford, D., and Jaakkola, T. (2017). Sequence to Better Sequence: Continuous Revision of Combinatorial Structures. In *ICML*.

Nakov, P., Ritter, A., Rosenthal, S., Sebastiani, F., and Stoyanov, V. (2019). SemEval-2016 Task 4: Sentiment Analysis in Twitter. *SemEval*.

Niu, X., Martindale, M., and Carpuat, M. (2017). A Study of Style in Machine Translation: Controlling the Formality of Machine Translation Output. In *EMNLP*, pages 2814–2819.

Ogden, C. (1932). *Basic English: A General Introduction with Rules and Grammar*. K. Paul, Trench, Trubner & Company, Limited.

Oquab, M., Bottou, L., Laptev, I., and Sivic, J. (2014). Weakly supervised object recognition with convolutional neural networks. In *NIPS*.

Palmer, M., Gildea, D., and Xue, N. (2010). Semantic role labeling. *Synthesis Lectures on Human Language Technologies*, 3(1):1–103.

Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

Pang, B. and Lee, L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *ACL*.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: a method for automatic evaluation of machine translation. In *ACL*, Morristown, NJ, USA.

Pappas, N. and Popescu-Belis, A. (2014). Explaining the Stars: Weighted Multiple-Instance Learning for Aspect-Based Sentiment Analysis. In *EMNLP*, pages 455–466.

Phan, S., Le, D.-D., and Satoh, i. (2015). Multimedia Event Detection Using Event-Driven Multiple Instance Learning. In *ACM international conference on Multimedia*, New York, NY, USA. ACM.

Pinheiro, P. O. and Collobert, R. (2015). From Image-level to Pixel-level Labeling with Convolutional Networks. In *CVPR*.

Prabhumoye, S., Tsvetkov, Y., Salakhutdinov, R., and Black, A. W. (2018). Style Transfer Through Back-Translation. In *ACL*.

Qiu, S., Xu, B., Zhang, J., Wang, Y., Shen, X., de Melo, G., Long, C., and Li, X. (2020). EasyAug: An Automatic Textual Data Augmentation Platform for Classification Tasks. In *The Web Conference*, New York, NY, USA. ACM.

Quinn, A. (1993). *Figures of speech : 60 ways to turn a phrase*. Psychology Press.

Rabinovich Shachar Mirkin Raj Nath Patel, E., Carmel, M., Gulmohar Cross Road No, H., and Specia Shuly Wintner, L. (2016). Personalized Machine Translation: Preserving Original Author Traits. In *ACL*, volume 1, pages 1074–1084.

Rahmani, R. and Goldman, S. A. (2006). MISSL: Multiple-Instance Semi-Supervised Learning. In *ICML*.

Rasmus, A., Valpola, H., Honkala, M., Berglund, M., and Raiko, T. (2015). Semi-Supervised Learning with Ladder Networks. In *NIPS*, volume 28.

Ratinov, L. and Roth, D. (2009). Design Challenges and Misconceptions in Named Entity Recognition * † ‡. In *CoNLL*.

Raykar, V. C., Krishnapuram, B., Bi, J., Dundar, M., and Rao, R. B. (2008). Bayesian multiple instance learning. In *ICML*, pages 808–815. Association for Computing Machinery (ACM).

Reddy, S. and Knight, K. (2016). Obfuscating Gender in Social Media Writing. In *EMNLP Workshop on Natural Language Processing and Computational Social Science*, pages 17–26.

Reed, S. E., Lee, H., Anguelov, D., Szegedy, C., Erhan, D., and Rabinovich, A. (2015). TRAINING DEEP NEURAL NETWORKS ON NOISY LABELS WITH BOOTSTRAPPING. In *ICLR*.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "Why should i trust you?" Explaining the predictions of any classifier. In *KDD*.

Ross, A. S., Hughes, M. C., and Doshi-Velez, F. (2017). Right for the right reasons: Training differentiable models by constraining their explanations. In *IJCAI*.

Rozovskaya, A. and Roth, D. (2010). Generating Confusion Sets for Context-Sensitive Error Correction. In *Proceedings of the EMNLP*, pages 961–970.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088):533–536.

Saito, K., Ushiku, Y., and Harada, T. (2017). Asymmetric Tri-training for Unsupervised Domain Adaptation. In *ICML*.

Schafer, R. E. (2011). *Statistical Models and Methods for Lifetime Data*. John Wiley & Sons.

Schmit, N. (2000). Lexical chunks. *ELT Journal*, 54(4):400–401.

Sennrich, R., Haddow, B., and Birch, A. (2016a). Controlling Politeness in Neural Machine Translation via Side Constraints. In *NAACL*, pages 35–40.

Sennrich, R., Haddow, B., and Birch, A. (2016b). Improving Neural Machine Translation Models with Monolingual Data. In *ACL*.

Shen, J., Qu, Y., Zhang, W., and Yu, Y. (2018). Wasserstein Distance Guided Representation Learning for Domain Adaptation. In *AAAI*.

Shen, T., Lei, T., Barzilay, R., Jaakkola, T., and Csail, M. (2017). Style Transfer from Non-Parallel Text by Cross-Alignment. In *NIPS*.

Shorten, C. and Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1):1–48.

Sinclair, J. M. (1998). The Lexical Item. In Weigand, E., editor, *Contrastive Lexical Semantics*, chapter 1, page 24. John Benjamins.

Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. (2013). Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *EMNLP*.

Stone, P. J. and Hunt, E. B. (1963). A computer approach to content analysis: Studies using the general inquirer system. In *AFIPS Conference Proceedings*, pages 241–256, New York, New York, USA. Association for Computing Machinery, Inc.

Strout, J., Zhang, Y., and Mooney, R. J. (2019). Do Human Rationales Improve Machine Explanations? In *BlackboxNLP Workshop at ACL*.

Sun, B., Feng, J., and Saenko, K. (2016). Return of Frustratingly Easy Domain Adaptation. In *AAAI*. AAAI press.

Sun, Q., Chattopadhyay, R., Panchanathan, S., and Ye, J. (2011). A Two-Stage Weighting Framework for Multi-Source Domain Adaptation. In *NIPS*.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. In *NIPS*.

Sutton, R. S., Mcallester, D., Singh, S., and Mansour, Y. (2000). Policy Gradient Methods for Reinforcement Learning with Function Approximation. In *NIPS*.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., and Stede, M. (2011). Lexicon-basedmethods for sentiment analysis. *Computational Linguistics*, 37(2):267–307.

Tan, C. and Lee, L. (2014). A Corpus of Sentence-level Revisions in Academic Writing: A Step towards Understanding Statement Strength in Communication. In *ACL (Volume 2: Short Papers)*, pages 403–408.

Tan, Z., Wang, M., Xie, J., Chen, Y., and Shi, X. (2018). Deep Semantic Role Labeling with Self-Attention. In *AAAI*.

Tarvainen, A. and Valpola, H. (2017). Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NIPS*.

Tetreault, J., Foster, J., and Chodorow, M. (2010). Using Parse Features for Preposition Selection and Error Detection. In *Proceedings of the ACL Short Papers*, pages 353–358.

Thrun, S. and Pratt, L. (1998). *Learning to Learn*. Kluwer Academic Publishers, USA.

Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (2003). Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In *NAACL*.

Trask, A., Michalak, P., and Liu, J. (2015). sense2vec - A Fast and Accurate Method for Word Sense Disambiguation In Neural Word Embeddings. *Computing Research Repository*, arXiv:1511.06388.

Tzeng, E., Hoffman, J., Darrell, T., and Saenko, K. (2015). Simultaneous deep transfer across domains and tasks. In *ICCV*.

Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T. (2017). Adversarial discriminative domain adaptation. In *CVPR*, volume 2017-January. Institute of Electrical and Electronics Engineers Inc.

Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., and Darrell, T. (2014). Deep Domain Confusion: Maximizing for Domain Invariance. *Computing Research Repository*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Lukasz, K., and Polosukhin, I. (2017). Attention Is All You Need. In *NIPS*.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P. A. (2008). Extracting and composing robust features with denoising autoencoders. In *ICML*, New York, New York, USA. Association for Computing Machinery (ACM).

Wan, C., Pan, R., and Li, J. (2011). Bi-weighting domain adaptation for cross-language text classification — Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume Volume Two. In *IJCAI*.

Wang, C. and Sridhar, M. (2008). Manifold Alignment using Procrustes Analysis. In *ICML*, volume 64.

Wang, W. Y. and Yang, D. (2015). That's So Annoying!!!: A Lexical and Frame-Semantic Embedding Based Data Augmentation Approach to Automatic Categorization of Annoying Behaviors using #petpeeve Tweets. In *EMNLP*.

Wei, J. and Zou, K. (2019). EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *EMNLP*, pages 6382–6388.

Wiegand, M., Balahur, A., Roth, B., Klakow, D., and Montoyo, A. (2010). A Survey on the Role of Negation in Sentiment Analysis. In *Workshop on negation and speculation in natural language processing*, pages 60–68.

Williams, R. J. and Zipser, D. (1989). A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Computation*, 1(2):270–280.

Willis, D. (1990). *The Lexical Syllabus: A New Approach to Language Teaching.* Harper Collins Publishers.

Wilson, T., Wiebe, J., and Hoffmann, P. (2005). Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In *EMNLP*.

Xie, Q., Dai, Z., Hovy, E., Luong, M.-T., and Le, Q. V. (2019). Unsupervised Data Augmentation for Consistency Training. *Computing Research Repository.*

Xu, Y., Pan, S. J., Xiong, H., Wu, Q., Luo, R., Min, H., and Song, H. (2017). A Unified Framework for Metric Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 29(6):1158–1171.

Xue, W. and Li, T. (2018). Aspect Based Sentiment Analysis with Gated Convolutional Networks. In *ACL*. Association for Computational Linguistics.

Yang, Z., Hu, Z., Dyer, C., Xing, E. P., and Berg-Kirkpatrick, T. (2018). Unsupervised Text Style Transfer using Language Models as Discriminators. In *NIPS*.

Yao, Y. and Doretto, G. (2010). Boosting for transfer learning with multiple sources. In *CVPR*, pages 1855–1862.

Yu, L., Zhang, W., Wang, J., and Yu, Y. (2017). SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. In *AAAI*.

Zaidan, O. F., Eisner, J., and Piatko, C. D. (2007). Using "Annotator Rationales" to Improve Machine Learning for Text Categorization. In *NAACL*.

Zaidan, O. F., Eisner, J., and Piatko, C. D. (2008). Machine Learning with Annotator Rationales to Reduce Annotation Cost. In *NIPS Workshop on Cost Sensitive Learning*.

Zellinger, W., Moser, B. A., Grubinger, T., Lughofer, E., Natschläger, T., and Saminger-Platz, S. (2019). Robust unsupervised domain adaptation for neural networks via moment alignment. *Information Sciences*, 483:174–191.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2018). mixup: Beyond Empirical Risk Minimization. In *ICLR*.

Zhang, X., Zhao, J., and Lecun, Y. (2015). Character-level Convolutional Networks for Text Classification. In *NIPS*.

Zhang, Y., Gan, Z., and Carin, L. (2016a). Generating text via adversarial training. In *NIPS workshop on Adversarial Training*.

Zhang, Y., Marshall, I., and Wallace, B. C. (2016b). Rationale-Augmented Convolutional Neural Networks for Text Classification. In *EMNLP*.

Zhang, Z., Ren, S., Liu, S., Wang, J., Chen, P., Li, M., Zhou, M., and Chen, E. (2019). Style Transfer as Unsupervised Machine Translation. In *AAAI*.

Zhou, G. and Su, J. (2002). Named Entity Recognition using an HMM-based Chunk Tagger. In *ACL*.

Zhou, Z.-H. (2004). Multi-Instance Learning: A Survey. Technical report, Department of Computer Science & Technology, Nanjing University.

Zhou, Z. H. and Li, M. (2005). Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering*, 17(11):1529–1541.

Zhou, Z.-H., Sun, Y.-Y., and Li, Y.-F. (2009). Multi-Instance Learning by Treating Instances As Non-I.I.D. Samples. In *ICML*.

Zhou, Z.-H. and Zhang, M.-L. (2002). Neural Networks for Multi-Instance Learning. Technical report, AI Lab, Computer Science & Technology Department Nanjing University.

Zhu, W., Lou, Q., Vang, Y. S., and Xie, X. (2017). Deep Multi-instance Networks with Sparse Label Assignment for Whole Mammogram Classification. In *MICCAI*, volume 10435 LNCS, pages 603–611. Springer, Cham.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., and He, Q. (2020). A Comprehensive Survey on Transfer Learning. *Proceedings of the IEEE*.