UNIVERSITY OF CALIFORNIA
SANTA CRUZ

SEVERAL STUDIES OF WEAKLY SUPERVISED LEARNING IN TEXT CLASSIFICATION

A dissertation submitted in partial satisfaction of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

by

Tianyi Luo

June 2022

The Dissertation of Tianyi Luo
is approved:

Professor Yang Liu, Chair

Professor Xin Eric Wang

Professor Dong Wang

Peter Biehl
Vice Provost and Dean of Graduate Studies

Contents

List of Figures

List of Tables

Abstract

Several Studies of Weakly Supervised Learning

in Text Classification

by

Tianyi Luo

Text classification is one of the most fundamental tasks in Natural Language Processing. How to effectually utilize the unlabeled dataset in text classification and apply weakly supervised learning methods to further improve the performance based on the existing labeled dataset, especially for supervision-starved tasks (hard to obtain high-quality labeled data), is challenging. In this PhD thesis, we show several studies of weakly supervised learning methods in text classification.

We first focus on improving the accuracy and interpretability in text classification tasks using weakly supervised learning methods with the help of unlabeled dataset. More specifically, we proposed several new methods to further improve the accuracy and interpretability on both of two main research directions in weakly supervised learning methods: learning with noisy labels and semi-supervised learning. For learning with noisy labels, we proposed two weakly supervised learning aided methods on the special supervision-starved text classification task: Research Replication Prediction. For semi-supervised learning, we presented a new weakly interpretable model to improve the interpretability on the long text classification tasks. We also proposed a new ensemble method to assign better pseudo or noisy labels to the samples in the unlabeled dataset for semi-supervise learning methods.

Furthermore, we conducted the research on fairness on weakly supervised learning. More specifically, we reveal the disparate impacts in different sub-populations (e.g., race and gender) when applying the semi-supervised learning methods. Finally, we also contribute a weakly supervised learning benchmark (Research Replication Prediction) to the community.

Acknowledgments

The most special thanks to my wife. You are the most beautiful sunshine in my life. Language is too limited to express my deepest love for you. I also want to express my greatest gratitude to my parents for their unconditional love and support throughout my life.

I would like to sincerely thank my advisor Dr. Yang Liu. This doctoral dissertation is impossible without your guidance and support. Many thanks to my other committee members Drs. Xin Eric Wang, Dong Wang, and Yi Zhang for their mentorship and valuable advice on my research and this dissertation. Additional special thanks to Dr. Dong Wang. With your direction and encouragement at Tsinghua University, I started enjoying doing research and was able to embark on my journey as a PhD student in the United States. I am also grateful for all my lab group members, collaborators, and friends in graduate school.

Chapter 1

Introduction

Text classification, also known as text categorization, is one of the most fundamental tasks in Natural Language Processing (NLP), which aims to assign one or more classes or categories to a textual unit e.g., sentences, paragraphs, and documents [37, 166, 69, 30, 165, 126, 82]. The text classification technologies are widely applied in the real world such as sentiment analysis [10], news categorization [129], spam filtering [36], and question answering [179].

Over the last decade, machine learning (ML) has made unprecedented progress in the text classification tasks [107]. These successes have been largely obtained by training the model with strong supervisions in supervised learning. However, supervised learning is an arduous process, requiring collecting massive amounts of data, cleaning it up, manually labelling it, training and perfecting a model purpose-built for the text classification tasks, and then using it to predict labels for unknown data. Collecting a large size of strong-supervised labels in the supervised learning is too expensive and time-consuming especially for some supervision-starved tasks, e.g. medical data [145] and research replication prediction [101] mentioned above. However, it is usually much easier for us to get a large size of unlabeled dataset. These unlabelled examples, though possibly noisy, but are informative and useful information [159]. Therefore, how to effectually utilize the unlabeled dataset in text classification to further improve the performance based on the existing labeled dataset becomes an interesting and important problem. For solving the problem, the weakly supervised learning models are proposed to deal with the data containing few labeled examples and a large number of unlabelled examples [54, 86, 106, 178, 8, 108, 109, 162, 24].

In this PhD thesis, we show several studies of weakly supervised learning methods in text classification. In the remaining sections in this chapter, we first describe the problems and motivations. Then we provide our corresponding results and contributions.

1.1 Problems and Motivations

The problems we want to resolve in this thesis are how to make better use of the unlabeled dataset to further improve the model performance in different aspects such as accuracy and interpretability for text classification tasks. In addition, when having machine learning models with better performance, the potential unfairness issue among different sub-populations such as race and gender arises and need to be explored when deploying them in the real world. Furthermore, the researches mentioned above need to be evaluated on the suitable datasets and building new standard datasets will facilitate the researchers to speed up the development of new weakly supervised learning methods.

1.1.1 Noisy Labels in Weakly Supervised Learning for Text Classification

While ML models are increasingly applied in text classification tasks and deployed in the real world, high-quality training data are often limited in amount [1, 160], especially for the supervision-starved tasks [101, 145]. Therefore, it is imperative to propose weakly supervised learning method to make the most use of the unlabeled dataset. In weakly supervised learning, we usually assign the pseudo labels to unlabeled dataset for conducting additional training based on the labeled dataset. However, the generated pseudo label are often noisy and inaccurate. How to utilize the unlabeled dataset with noisy labels is challenging.

1.1.2 Interpretability in Weakly Supervised Learning for Text Classification

In recent years, deep learning neural network models draw a lot of attentions and are widely applied in text classification. However, these neural network models are often black-box and lacks of model interpretability [16]. Building an interpretable neural text classifier for text classification is necessary and it will make the deployed model in the real-world more reliable and trustworthy. Some studies on building the interpretable models have been conducted for short text classification tasks and few are for long text [23]. Furthermore, the prior works on model interpretation mainly focused on improving the model interpretability at the word/phrase level, which are insufficient especially for long documents. In addition, the existing methods cannot utilize a large size of unlabeled dataset to further improve the model interpretability. How to address these limitations are challenging.

1.1.3 Quality of Noisy Labels in Weakly Supervised Learning for Text Classification

Assigning the high-quality pseudo or noisy labels to the unlabeled dataset is crucial for weakly supervised learning in the text classification tasks. Ensemble methods are usually applied to conduct the better aggregating in the step of generating pseudo or noisy labels to the unlabeled dataset [9]. Nonetheless, the existing aggregating rule would fail when the majority answer of all the constituent algorithms is more likely to be wrong. It is challenging on whether we can propose a new ensemble method which can reveal the correct minority label when the majority answer is wrong.

1.1.4 Fairness in Weakly Supervised Learning for Text Classification

Weakly supervised learning have been successfully applied in text classification tasks, where the high-quality supervised data is severely limited. Although the average accuracy for the whole population of data is improved, it is unclear on the improvement for different sub-populations. It may raise the fairness concerns when the sub-populations are defined by the demographic groups such as race and gender [76]. Verifying whether there exists fairness issue for different sub-populations and how to qualitatively show the fairness issue in a suitable way for this setting is challenging.

1.1.5 Datasets in Weakly Supervised Learning for Text Classification

Many standard text classification datasets have been built [20, 103, 87, 174, 122, 121, 170] and they facilitate the development of the corresponding ML approaches. However, few standard datasets are built specially for developing weakly supervised learning methods.

1.2 Results and Contributions

In this thesis, we proposed some new weakly supervised learning methods to further improve the accuracy and interpretability on both of two main research directions in weakly supervised learning methods: learning with noisy labels and semi-supervised learning. We also reveal the disparate impacts in different sub-populations (e.g., race and gender) when deploying semi-supervised learning methods. In addition, we also contributes a weakly supervised learning benchmark (Research Replication Prediction) to the community. Our main results are described in Chapter 3 (published in EMNLP 2020 Findings), Chapter 4 (published in ACL 2022 Findings), Chapter 5 (Minor revision in Machine Learning Journal), Chapter 6 (published in ICLR 2022), and Chapter 7 (will be submitted to Nature Scientific Data Journal).

1.2.1 Research Replication Prediction Using Weakly Supervised Learning

Research Replication Prediction (RRP), which aims to predict if a published research result can be replicated, is a supervision-starved task. Carrying out direct replication of published research to obtain the high-quality labels incurs a high cost. We propose two weakly supervised learning approaches (Variational Inference based and Peer Loss based) to further improve the accuracy on this supervision-starved RRP task using both labeled and unlabeled datasets. This paper [101] was published in EMNLP 2020 Findings.

1.2.2 Interpretable Research Replication Prediction via Variational Contextual Consistency Sentence Masking

For long text classification tasks such as Research Replication Prediction (RRP) and European Convention of Human Rights (ECHR), we built an interpretable neural model which can provide sentence-level explanations and apply weakly supervised approach to further leverage the large corpus of unlabeled datasets to boost the interpretability in addition to improving prediction performance as existing works have done. More specifically, we propose the Variational Contextual Consistency Sentence Masking (**VCCSM**) method to automatically extract key sentences based on the context in the classifier, using both labeled and unlabeled datasets. This paper [102] was published in ACL 2022 Findings.

1.2.3 Machine Truth Serum: a Surprisingly Popular Approach to Improving Ensemble Methods in Classification

To further improve the quality of noisy or pseudo labels of unlabeled dataset generated by ensemble methods in the weakly supervised learning, we present two machine learning aided methods which can reveal the truth when the minority instead of majority has the true answer on both settings of supervised and semi-supervised classification tasks. We name our proposed method the Machine Truth Serum (MTS). Our experiments on a set of classification tasks (image, text, etc.) show that the classification performance can be further improved by applying MTS in the ensemble final predictions step (supervised) and in the ensemble data augmentations step (semi-supervised). This paper obtained minor revision in Machine Learning Journal.

1.2.4 The Rich Get Richer: Disparate Impact of Semi-Supervised Learning

With weakly supervised learning methods widely applied in various applications, fairness issue among different sub-populations such as gender and race arises. We reveal the disparate impacts of deploying semi-supervised leanring (SSL): the sub-population who has a higher baseline accuracy without using SSL (the “rich" one) tends to benefit more from SSL; while the sub-population who suffers from a low baseline accuracy (the “poor" one) might even observe a performance drop after adding the SSL module. We hope our paper will alarm the potential pitfall of using SSL and encourage a multifaceted evaluation of future SSL algorithms. This paper [184] was published in ICLR 2022.

1.2.5 A New Weakly Supervised Learning Dataset — Research Replication Prediction

To help the community develop better weakly supervised learning ML models, we present a new weakly supervised dataset — Research Replication Prediction (RRP). In this RRP dataset, we collected two types of data with different costs: one with direct verification (expensive with smaller size) and one using crowdsourcing (larger scale but potentially noisy). In total, our dataset contains 399 directly replicated samples and 2,682 crowdsourced samples. We benchmark the performances of several representative weakly supervised baseline methods. We report several commonly used metrics (accuracy, precision, recall, and F1) to evaluate the models.

Chapter 2

Preliminaries

In this chapter we present definitions and notations for supervised learning, semi-supervised learning and learning with noisy labels methods for text classification tasks.

2.1 Supervised Text Classification Tasks

Consider a K-class classification task given N_L labeled training examples denoted by $D_L := \{(x_l, y_l)\}_{l=1}^{N_L}$ and N_T labeled testing examples denoted by $D_T := \{(x_t, y_t)\}_{t=1}^{N_T}$, $x_l \ or \ x_t \in X$ is an input feature of a text unit, $y_l \ or \ y_t \in \{0, 1, ..., K-1\}$ represents its corresponding clean class label. The clean data distribution with full supervision is denoted by $\mathcal{D}$. Examples $(x_l, y_l) \ or \ (x_t, y_t)$ are drawn according to random variables $(X, Y) \sim \mathcal{D}$. The classification task aims to learn a classifier f that maps X to Y accurately denoted by $f : X \rightarrow Y$. We define the K-classification cross entropy loss as $\ell(f(x_l), y_l) := -\ln(f_{x_l}[y_l]), y_l \in \{0, 1, .., K-1\}$, where $f_{x_l}[y_l]$ denotes the y_l-th component of $f(x_l)$, for the supervised text classifier on each data point (x_l, y_l) in the training dataset. Therefore, the empirical risk of the supervised text classifier using clean class labels is as follows:

$$L_1(f, D_L) = \frac{1}{N_L} \sum_{l=1}^{N_L} \ell(f(x_l), y_l).$$

2.2 Semi-Supervised Text Classification Tasks

In the semi-supervised text classification tasks, there is also one additional unlabeled dataset $\mathcal{D}_U := \{(x_u, \cdot)\}_{u=1}^{N_U}$, where the labels are missing or unobservable. Many methods are proposed to generate the high-quality pseudo labels of unsupervised dataset [9, 162, 8, 141, 164] and we can have a new $\mathcal{D}_U := \{(x_u, y_u)\}_{u=1}^{N_U}$. Compared with supervised classification tasks, the information of unsupervised should be leveraged to improve the performance. The empirical risk of the semi-supervised text classifier for $f(\cdot)$ using pseudo labels is as follows:

$$L_2(f, D_L, D_U) = \frac{1}{N_L} \sum_{i=1}^{N_L} \ell(f(x_l), y_l) + \frac{1}{N_U} \sum_{u=1}^{N_U} \ell(f(x_u), y_u).$$

2.3 Learning with Noisy Labels Text Classification Tasks

Semi-supervised learning and learning with noisy labels methods are two main research lines which can improve the performance with the help of unlabeled dataset. Although both semi-supervised and learning with noisy labels methods try to assign high-quality pseudo or noisy labels to the unlabeled dataset, learning with noisy labels utilizes the non-standard loss functions including the information of error rates instead of directly using standard loss functions such as Cross Entropy [38]. In this subsection, we introduce two commonly used learning with noisy label methods.

Loss correction A main research line of learning with noisy labels are loss correction methods require estimating the error rates. A representative method is variational inference (VI) aided weakly supervised method [101]. In VI, several basic text classifiers (only for estimating the error rates) are first trained on the labeled dataset and then the pseudo or noisy labels of unlabeled dataset are obtained applying the majority rule based on the predictions of the trained basic classifiers. Then pseudo or noisy labels of unlabeled dataset as well as the predictions of basic classifiers are used to estimate the error rates using the variational inference methods proposed by Liu et al. [89]. In the final step, the noisy training is conducted on the unlabeled dataset with the proxy loss function [112] as shown below (only show binary classification case for simplicity):

$$\ell_{noise\_correct} = \sum_{u=1}^{N_U} \frac{(1-\rho_{1-y_u})\ell(y_u^p, y_u) - \rho_{y_u}\ell(y_u^p, 1-y_u)}{1-\rho_1-\rho_0}$$

where $\ell(y_u^p, y_u)$ is a standard cross entropy loss function where y_u^p is the u-th training sample's prediction in the unlabeled dataset and y_u is its corresponding pseudo or noisy label. N_U is the number of unlabeled training dataset and $\rho_0 := \Pr(y_u^p \neq y_u | y_u = 0)$, $\rho_1 := \Pr(y_u^p \neq y_u | y_u = 1)$ are two classes' error rates estimated using variational inference method.

Peer loss Instead of estimating the noise rates in VI (may introduce the extra errors), Liu and Guo [97] provided an alternative, peer loss, to deal with noisy labels without requiring an additional estimation step for the noise rates. To apply peer loss, we first construct peer samples for each sample in the unlabeled training dataset. More specifically, for the u-th training sample (x_u, y_u) in the unlabeled dataset, we randomly choose two other samples $(x_{u_1}, y_{u_1}), (x_{u_2}, y_{u_2})$ such that $u_1 \neq u_2$ and $u_1, u_2 \neq u$. Then we can construct the peer sample $(x_{u_1}, y_{u_1}), (x_{u_2}, y_{u_2})$ for (x_u, y_u). Then we can calculate peer loss function as shown below:

$$\ell_{noise\_peer} = \sum_{u=1}^{N_U} \ell(y_u^p, y_u) - \alpha \cdot \ell(y_{u_1}^p, y_{u_2})$$

where $\ell(y_u^p, y_u)$ is a standard cross entropy loss function. y_u^p is the u-th sample's prediction and y_u is the corresponding noisy label. α is an important hyperparameter that need to be tuned with in the peer loss function. N_U is the number of unlabeled training dataset.

Based on two learning with noisy labels methods mentioned above, the empirical risk of the learning with noisy labels text classifier for $f(\cdot)$ using pseudo labels is as follows:

$$L_3(f, D_L, D_U) = \frac{1}{N_L} \sum_{l=1}^{N_L} \ell(f(x_l), y_l) + \frac{1}{N_U} \sum_{u=1}^{N_U} \ell_{noise}(f(x_u), y_u).$$

Chapter 3

Research Replication Prediction Using Weakly Supervised Learning

3.1 Introduction

This chapter focuses on proposing the new weakly supervised learning methods to improve the accuracy for a typical supervision-starved text classification task: Research Replication Prediction (RRP) in the research line of learning with noisy labels.

Non-reproducible scientific results will mislead the progress of science and undermine the trustworthiness of the research community. Therefore, it is important to know whether a published research result can be reproduced or not. In recent years, researchers have conducted several direct replication projects for hundreds of classic and contemporary published findings in the social sciences studies [14, 15, 44, 80, 33]. However, such direct replication is very time-consuming and expensive [47]. Therefore, machine learning (ML), a much cheaper and more efficient alternative is used to conduct the replication prediction. After being modeled as a ML prediction problem, it becomes a very typical supervision-starved task due to the high cost of obtaining labeled dataset mentioned above.

Existing ML works for this RRP task only utilized a small amount of expensive labeled dataset to train the model, where the use of more sophisticated but more accurate deep learning techniques is limited [43, 167, 3]. Even though we only have a small size of labeled dataset, large amounts of unlabeled research articles are available. These unlabeled examples, although possibly noisy, can provide useful information. Therefore, we aim to propose the new methods to leverage the large size of unlabeled dataset to further improve the performance.

To make use of the unlabeled dataset, we explore the possibility of using the weakly supervised learning methods . More specifically, we focus on utlizing the techniques from the research line of learning with noisy labels [89, 112, 130, 149, 97]. The high level idea is to first train several weak classifiers based on the small size of labeled data. Then these weak classifiers can help us assign the noisy label to the large size of unlabeled data. Finally, the tools from learning with noisy labels can be utilized to further improve the performance with additional training on these noisy or weakly supervised samples.

In this chapter, we proposed two weakly supervised learning approaches based on text information of research papers to further improve the prediction accuracy of research replication using both labeled and unlabeled datasets. The first method is Variational Inference (VI) aided Weakly Supervised Learning. In VI, the efficient variational inference method [89] are firstly used to estimate the error rates of weakly supervised samples. Then the loss correction can be conducted to further improve the performance with the estimated error rates. The second method

— Peer Loss (PL) aided Weakly Supervised Learning applied the peer loss function [97] which can directly be trained on the peer samples without requiring the knowledge of error rates.

In addition, the labeled and unlabeled datasets for the RRP task are constructed the by ourselves. The labeled dataset containing 399 research articles are collected by summarizing eight research replication projects. As for the unlabeled dataset, we implemented a python crawler to obtain the pdf files of 2,170 research papers from the websites of corresponding journals.

3.2 Learning with Noisy Labels based Weakly Supervised Learning

In this section we present two learning with noisy labels based weakly supervised methods.

The first method relies on proxy loss function [112] to correct the error in the noisy labels and variational inference approaches [89] to estimate the error rates. These two techniques jointly provide us a bias-corrected training process to improve the model's robustness against errors in the noisy labels. The first method is named as *Variational Inference aided Weakly Supervised Learning*.

The second method is built on the peer loss approach [97]. The peer loss approach, where estimating the error rates are not required, is particularly suitable for our RRP task when the errors in the noisy labels are unclear. We name this solution as *Peer Loss aided Weakly Supervised Learning*.

3.2.1 Variational Inference aided Weakly Supervised Learning

Algorithm 1 Variational Inference aided Weakly Supervised Learning

```
Require:
    Input:
    $\mathcal{D}_L = \{(x_1, y_1), ..., (x_{N_L}, y_{N_L})\}$: labeled data
    $\mathcal{D}_U = \{x_1, ..., x_{N_U}\}$: unlabeled data
    $\mathcal{D}_T = \{(x_1, y_1), ..., (x_{N_T}, y_{N_T})\}$: test data
    $\mathcal{F} = \{f_1, ..., f_J\}$: classifiers
Ensure:
 1: Train $J$ classifiers ($\mathcal{F}$) on the labeled training data $\mathcal{D}_L$.
 2: for $j = 1$ to $J$ do
 3:     for $u = 1$ to $N_U$ do
 4:         Compute $y_u^j$ using $j$-th basic classifier.
 5:     end for
 6: end for
 7: Aggregate above labels into $\{y_u\}_{u=1}^{N_U}$ and estimate the error rates according to mean field
    method described in [89].
 8: Train the LSTM model using the proxy loss function mentioned in Section 5.1 with the
    estimated error rates in line 7 as the inputs on the weakly supervised dataset. Also train the
    LSTM model using the standard cross entropy on the labeled datset.
 9: for $t = 1$ to $N_T$ do
10:     Output prediction.
11: end for
```

We first use five basic classifiers (LR, RF, SVM, MLP, and LSTM) trained on the small size of labeled dataset to generate the noisy labels for each sample in the unlabeled dataset. Then these noisy labels will be aggregated and the error rates will be estimated utilizing a variational inference procedure [89], which we reproduce below:

μ_i is denoted as the probability of different class labels for the u-th training sample in the unlabeled dataset and ω_j represents the weight or ability of the j-th classifier, α and β are the hyperparameters, $\delta_{uj} = \mathbb{1}[y_u^j = y_u]$ where y_u^j is the noisy label of the j-th basic classifiers and y_u is the aggregated noisy labels by applying the majority voting rule on the y_u^j for each unlabeled data sample. We first estimated μ_i and ω_j using the Expectation-Maximization (EM) algorithms. Then the EM predictions y_u^{em} are obtained based on the above estimated μ_u and ω_k. Finally, we estimate the error rates $\sigma_0 := \mathbb{P}(y_u = 1|y_u^{em} = 0)$ and $\sigma_1 := \mathbb{P}(y_u = 0|y_u^{em} = 1)$ by using y_u^{em} as the proxy for the ground truth label. The steps of estimating the error rates are summarized in Algorithm 2. More details of EM estimation are described in [89].

Algorithm 2 Aggregation and Error Rates

```
1: Update $\mu_u$ :
       $$\mu_u(z_u) = \prod_{j \in K} \omega_j^{\delta_{uj}} (1 - \omega_j)^{1-\delta_{uj}}$$
2: Update $\omega_j$ : $\omega_j = \frac{\sum_{u \in N_U} \mu_u(y_u^j) + \alpha}{N_U + \alpha + \beta}$
3: VI Predictions : $y_u^{em} = \text{argmax}_z \, \mu_u(z_u)$
4: Error rates :
       $$\sigma_0 = \frac{|u : y_u^{em} = 0, y_u = 1|}{|u : y_u^{em} = 0|}$$
       $$\sigma_1 = \frac{|i : y_u^{em} = 1, y_u = 0|}{|u : y_u^{em} = 1|}$$
```

In the final step, an LSTM neural network model with proxy loss function as described in [112] is used to conduct the training. The definition of proxy loss function is as follows:

$$\sum_{u=1}^{N_U} \frac{(1 - \rho_{1-y_u})\ell(y_u^p, y_u) - \rho_{y_u}\ell(y_u^p, 1 - y_u)}{1 - \rho_1 - \rho_0}$$

where y_u^p is the u-th sample's prediction of final LSTM model and y_u is the corresponding noisy label. ℓ is the standard cross entropy loss function.

The whole procedure of Variational Inference based Weakly Supervised Learning method is summarized in Algorithm 1.

3.2.2 Peer Loss aided Weakly Supervised Learning

As described in last subsection, Variational Inference aided Weakly Supervised Learning method needs to estimate the error rates, where the additional estimating step may introduce extra errors. Liu and Guo [97] proposed a new learning with noisy label method without requiring estimating the error rates. Therefore, we proposed Peer Loss aided Weakly Supervised Learning method.

Similar to Variational Inference aided Weakly Supervised Learning method, five basic classifiers (LR, RF, SVM, MLP, and LSTM) trained on the small size of labeled dataset are utilized to generate the noisy labels for each training sample in the unlabeled dataset $\{y_u\}_{u=1}^{N_U}$ via a majority voting rule.

For each training sample (x_u, y_u) in the unlabeled dataset, we randomly draw another two samples

$$\textbf{Peer Samples: } (x_{u_1}, y_{u_1}), (x_{u_2}, y_{u_2})$$

such that $u_1 \neq u_2$ and $u_1, u_2 \neq u$. $(x_{u_1}, y_{u_1}), (x_{u_2}, y_{u_2})$ are the u-th data's peer samples. Then we calculate peer loss function as shown in [97]. The definition of total peer loss function is given as follows:

$$\sum_{u=1}^{N_U} \ell(y_u^p, y_u) - \alpha \cdot \ell(y_{u_1}^p, y_{u_2})$$

where y_u^p is the u-th sample's prediction of final LSTM model and y_u is the corresponding noisy label. ℓ is a standard cross entropy loss function and α is a hyperparameter that we need to tune with.

Finally, an LSTM neural network model with the above peer loss function are trained. The whole procedure of Peer Loss aided Weakly Supervised Learning method is further illustrated in Algorithm 3.

Algorithm 3 Peer Loss aided Weakly Supervised Learning

```
Require:
    Input:
    $\mathcal{D}_L = \{(x_1, y_1), ..., (x_{N_L}, y_{N_L})\}$: labeled data
    $\mathcal{D}_U = \{x_1, , ..., x_{N_U}\}$: unlabeled data
    $\mathcal{T} = \{(x_1, y_1), ..., (x_{N_T}, y_{N_T})\}$: test data
    $\mathcal{F} = \{f_1, ..., f_J\}$: classifiers
Ensure:
    Train $J$ classifiers ($\mathcal{F}$) on the labeled training data $\mathcal{D}_L$.
    for $j = 1$ to $J$ do
        for $u = 1$ to $N_U$ do
            Compute $y_u^j$ using $j$-th basic classifier.
        end for
    end for
    Compute $\{y_u\}_{u=1}^{N_U}$ using majority rule.
    for $u = 1$ to $N_U$ do
        Construct $\{(x_u, y_u), (x_{u_1}, y_{u_2})\}$.
    end for
    Create noisy training dataset: $\mathcal{D}^{noise} = \{(x_u, y_u), (x_{u_1}, y_{u_2})\}_{u=1}^{N_U}$.
    Train the LSTM model using peer loss function as shown in Section 5.2 on $\mathcal{D}^{noise}$ on the
    unlabeled dataset. Also train the LSTM model using the standard cross entropy on the
    labeled datset.
    for $t = 1$ to $N_T$ do
        Output prediction.
    end for
```

3.3 Datasets

Annotated Data we obtained 399 annotated articles from eight research replication projects which are the Registered Replication Report (RRR) [135], Many Labs 1 [78], Many Labs 2 [80], Many Labs 3 [44], Social Sciences Replication Project (SSRP) [15], PsychFileDrawer [115], Experimental Economics Replication Project [14], and Reproducibility Project: Psychology (RPP) [32].

There are different standards to claim that one research paper is replicable. To include as many annotated samples as possible, we adopt the comonly used definition — “statistically significant (p-value $<= 0.05$) effect in the same direction as in the original study." [3]

In the labeled dataset, label ‘1’ is used to denote that the research paper can be reproduced. Otherwise, the label ‘0’ is used to represent it. In the 399 annotated samples, there are 201 samples with label ‘1’ (replicable) and 198 samples are ‘0’ (non-replicable). From the class distribution, we observe that the labeled dataset is class-balanced.

Unsupervised Data Along with collecting the labeled dataset, we write a python crawler to obtain an unlabeled dataset. Since the research papers in the labeled dataset are mainly from American Economic Review and Psychological Science and the remaining papers are mainly in the economic and psychology fields, we applied the python crawler to obtain 981 unlabeled research papers (PDF files) from the website of American Economic Review (Jan 2011 - Dec 2014) and 1,189 unlabeled research papers (PDF files) from the website of Psychological Science (Jan 2006 - Dec 2012).

Datasets	Number of documents	Average length	Maximum length	Minimum length
Train	300	8948	68998	1446
Test	99	8343	33354	3599
Unlabeled	2170	6647	28994	1260

Table 3.1: Number, average length, maximum length, and minimum length of documents in different datasets

Overall Dataset We list the average length, minimum length, and maximum length information of different datasets in Table 3.1. The length means the number of words.

3.4 Experiments

3.4.1 Experimental Setup

In total, we have 399 labeled and 2,170 unlabeled samples. 300 (150:1;150:0) samples are randomly selected from labeled dataset and all the samples from unlabeled dataset construct our training dataset. The remaining 99 (51:1;48:0) samples in the labeled dataset are considered as the testing dataset.

Considering this special task, both text and statistics features of research papers are utilized. As for the statistics features, p-value, effect size, sample size are considered. As for the text features, tf-idf and BERT word embeddings are used for bag-of-words and sequential models respectively. BERT models can help obtain better context-aware features. More specifically, we used fine-tuned BERT model based on our own corpus as the pretrained model. In addition, we set the maximum length of our model to 10,000 since the average length of all the documents are 10,000.

Because the text features and statistics features represent different types of information, we trained models using these two types of features separately. We also try to combine the predictions of these two types of models. In the combination, the model trained on the statistics features are fixed to SVM because it obtains the best performance.

Model	Train Setting	Test Accuracy (Text)	Test Accuracy (Text + Statistics)
LR	300 (L)	57.58% (57/99)	58.59% (58/99)
RF	300 (L)	51.52% (51/99)	52.53% (52/99)
SVM	300 (L)	58.59% (58/99)	60.61% (60/99)
MLP	300 (L)	59.60% (59/99)	60.61% (60/99)
DIVIDEMIX	300 (L) + 2,170 (U)	62.63% (62/99)	63.64% (63/99)
BERT	300 (L)	64.65% (64/99)	65.66% (65/99)
BERT	300 (L) + 2,170 (U)	65.66% (65/99)	67.68% (67/99)
MixText	300 (L) + 2,170 (U)	64.65% (64/99)	65.66% (65/99)
VI	300 (L) + 2,170 (U)	**68.69% (68/99)**	**69.70% (69/99)**
PL	300 (L) + 2,170 (U)	**71.72% (71/99)**	**75.76% (75/99)**

Table 3.2: Comparison on Train Setting, Test Accuracy (Text), and Test Accuracy (Text + Statistics) between different nine models. VI is our variational inference based weakly supervised learning method, and PL is our peer loss based weakly supervised learning approach. 300 (L) means that 300 labeled samples are used to train. 300 (L) + 2,170 (U) means that 300 labeled and 2,170 unlabeled samples are used to train.

3.4.2 Results

The experimental results of text only and text + statistics are listed in Table 7.9. As shown in Table 7.9, we can observe that the combination model (text + statistics) performs better than the ones trained only on the text features, which indicates that the statistics feature are complementary to text feature. As for the models only trained on the statistical features, we report that LR, RF, and SVM (non-deep learning models) are only able to obtain 54.55%, 50.51%, and 56.57% accuracy on the testing dataset respectively. The experimental results of models trained on the statistics features only confirms that the models train on the text features are better.

In Table 7.9, we compare nine different approaches LR, RF, SVM, MLP, LSTM, DIVIDEMIX [88], MixText[24], VI (our variational inference based weakly supervised learning method), and PL (our peer loss based weakly supervised learning method). From Table 7.9, we can observe that our two proposed weakly supervised learning methods obtain the best performance and it shows the effectiveness of our proposed methods. Furthermore, among our two proposed approaches, PL perform better than VI. It suggests that PL works better in handling the noise and likely the extra error are introduced to VI when estimating the error rates.

In addition, we also trained the BERT on both supervised and weakly supervised settings without using noise-resistant loss functions. We observe that they get the same performance. It suggest that the performance cannot be improve if the noise-resistant loss functions are not applied to correct the biases in the noisy labels.

3.4.3 Case Study

In this subsection, we showed two cases with the same text but obtain different results predicted by Logistic Regression and Peer Loss aided Weakly Supervised Learning methods. The showing text is a paragraph selected from a research paper “Avoiding overhead aversion in charity” in Behavioral Economics. The ground truth label of this paper is replicable. Showing the case study aim to provide an intuitive view about how different classifiers work and identify replicable related words.

In Table 3.3, we highlight the words with larger weights using Logistic Regressions classifier. Since we used tf-idf features in the Logistic Regression classifier, each word has its unique weight. In Table 3.4, we highlight the words with larger weights using Peer Loss aided Weakly Supervised Learning classifier. Because PL used a neural network model, each word (node) in the input layer has multiple links to hidden states and we calculate a summation of all the weights of the corresponding links for each word. Comparing Table 3.3 and 3.4, PL provide

Donors **tend** to **avoid** charities that dedicate a high **percentage** of expenses to administrative and fundraising costs, limiting the ability of nonprofits to be effective. We propose a solution to this problem: Use donations from **major** philanthropists to cover overhead expenses and offer **potential** donors an overhead-free donation opportunity. A laboratory experiment testing this solution confirms that donations **decrease** when overhead increases, but only when donors **pay** for overhead themselves. In a field **experiment** with 40,000 potential donors, we compared the overhead-free solution with other **common** uses of initial donations. Consistent with **prior** research, informing donors that seed money has already been raised **increases** donations, as does a $1:$1 matching campaign. Our main result, however, clearly **shows** that informing **potential** donors that overhead **costs** 3 are covered by an initial donation significantly **increases** the donation **rate** by 80% (or 94%) and **total** donations by 75% (or 89%) compared with the seed (or matching) approach.

Table 3.3: Red color highlights words having positive weights and the absolute value is larger than 0.1. Blue color highlights words having negative weights and the absolute value is larger than 0.1. Classification result of Logistic Regression for this paper is Non-replicable (Wrong)

Donors tend to avoid **charities** that dedicate a **high** percentage of **expenses** to administrative and **fundraising** costs, **limiting** the ability of nonprofits to be effective. We **propose** a solution to this problem: Use **donations** from **major** philanthropists to **cover** overhead expenses and **offer** **potential donors** an overhead-free **donation** opportunity. A laboratory experiment testing this **solution confirms** that **donations decrease** when **overhead** increases, but only when **donors** **pay** for **overhead** themselves. In a field experiment with 40,000 potential donors, we compared the overhead-free solution with other **common** uses of initial donations. **Consistent** with **prior** research, **informing donors** that **seed money** has already been raised increases donations, as does a $1:$1 **matching** campaign. Our main result, however, clearly shows that **informing potential** donors that **overhead costs**3 are covered by an initial donation **significantly** increases the donation rate by 80% (or 94%) and **total** donations by 75% (or 89%) compared with the **seed** (or matching) approach.

Table 3.4: Red color highlights words having positive weights and the absolute value is larger than 0.15. Blue colors highlight words having negative weights and the absolute value is larger than 0.15. Classification result of Peer Loss for this paper is Replicable (Correct)

a correct prediction result because it is able to capture more relevant keywords such as charity, donors, overhead, significantly, etc.

3.5 Related Work

Replication crisis has spurred several large-scale direct replication projects which are conducted by the professional individuals or teams in the social science [14, 15, 44, 79, 80, 33].

However, such direct replication is expensive and time-consuming [47]. Therefore, machine learning serves as a much more efficient method to conduct the replication prediction task. Altmejd et al. [3] applied machine learning methods on the data from four large-scale replication projects in experimental psychology and economics. But they trained only on the small size of labeled dataset.

Weakly supervised learning methods have been proposed to leverage both labeled and unlabeled datasets [178, 114, 109]. We focus on the research line of learning with noisy labels in the weakly supervised learning methods [17, 13, 131, 130, 149]. Particularly relevant to us, [112] proposed a proxy loss function which can provide an unbiased estimation of the loss on the clean dataset using only noisy labels. Liu and Guo [97] introduced a new family of loss functions, peer loss functions which can conduct the empirical risk minimization without requiring estimating the error rates of noisy labels.

Chapter 4

Interpretable Research Replication Prediction via Variational Contextual Consistency Sentence Masking

4.1 Introduction

In this chapter, we focus on proposing the new weakly supervised learning methods to improve the model interpretability with the help of unlabeled dataset for text classification tasks. We started with the same typical supervision-starved text classification task: Research Replication Prediction (RRP) and then generally extended our method to other long text classification tasks.

It is important to know whether a published research result can be replicated or not. In recent years, several direct replication projects on social science have been conducted [14, 15, 44, 80, 33]. However, such direct replication is very time-consuming and expensive. Therefore, a much more efficient and cheaper alternative—ML method is applied to predict

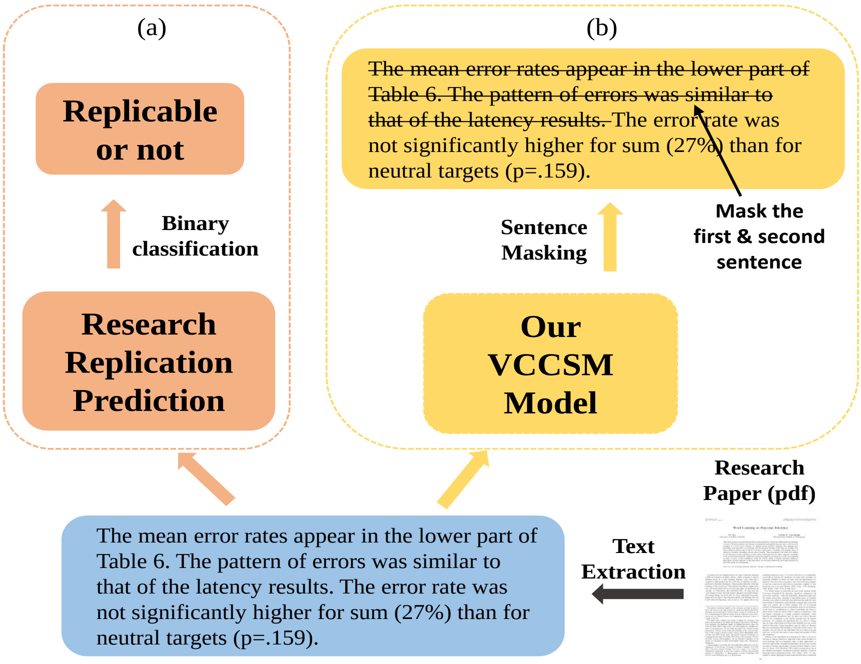

(a)
Replicable or not
Binary classification
Research Replication Prediction
(b)
~~The mean error rates appear in the lower part of Table 6. The pattern of errors was similar to that of the latency results.~~ The error rate was not significantly higher for sum (27%) than for neutral targets (p=.159).
Sentence Masking
Mask the first & second sentence
Our VCCSM Model
Research Paper (pdf)
Text Extraction
The mean error rates appear in the lower part of Table 6. The pattern of errors was similar to that of the latency results. The error rate was not significantly higher for sum (27%) than for neutral targets (p=.159).

Figure 4.1: (a) Given the text information of a research paper, Research Replication Prediction (RRP) task predicts whether the paper can be reproduced or not. (b) Having the same input as (a), our VCCSM model can keep the important sentences (through masking unimportant ones) which are related to reproducibility.

research replication [43, 167, 3, 101]. In this chapter, we model the task of predicting research replication as a binary text classification problem — Research Replication Prediction (RRP) task which is shown in Figure 4.1(a). Nonetheless, applying the recent deep neural network models on the RRP task faces two challenges. The first challenge is the existing neural network models applied on RRP task lack of interpretability. The results of RRP may not be widely accepted as reliable and trustworthy if the understandable explanations are provided for the predictions. The second challenge is the small size of labeled dataset in RRP due to its high cost of direct replications. Although we have proposed some new weakly supervised learning methods to further improve the accuracy with the help of unlabeled dataset, how to make use of the unlabeled dataset to further improve the model interpretabity is challenging.

As for the first challenge, the existing interpretable machine learning methods mostly focus on improving the interpretability only at the word/phrase level which may work well for short documents (the average length of words is less than 500) [63, 140, 138, 132, 57, 23, 22]. However, the average length of words for the research papers in RRP are about 10,000 which are lengthy. As for the second challenge, the existing weakly supervised approaches mainly focused on improving the accuracy but not the interpretability [9, 162, 24]. We aim to explore a new weakly interpretable neural text classifier for predicting research replication which can utilize the large size of unlabeled dataset to improve the model interpretability.

For tackling the first challenge, we built an interpretable neural network model which can automatically select key sentences based on the contexts instead of words/phrases by adding a *variational sentence masking* layer (information bottleneck framework [148, 2] can be used) on the input layer. We considered these selected key sentences after masking as our interpretations for each research paper. To tackle the second challenge, we proposed a new weakly supervised method to leverage the unlabeled dataset to improve the model interpretability. More specifically, we proposed a consistency training method through replacing the noise-added input of unlabeled dataset by masked sentences. For each research paper, we make the first prediction using the key sentences after masking and then get the second prediction using all the sentences without masking. Then the consistency check is conducted on these two predictions by minimizing the difference between them. Therefore, a large size of unlabeled dataset is utilized to further improve the model interpretability.

In sum, we proposed a variational contextual consistency sentence masking (VCCSM) method as shown in Figure 4.1(b) which can extract the key sentences based on their contexts and leverage a large size of unlabeled dataset to further improve the model interpretability by using a consistency checking mechanism. In addition, our proposed VCCSM method can also be generally applied on other long text classification tasks.

4.2 Variational Contextual Consistency Sentence Masking

4.2.1 Model Overview

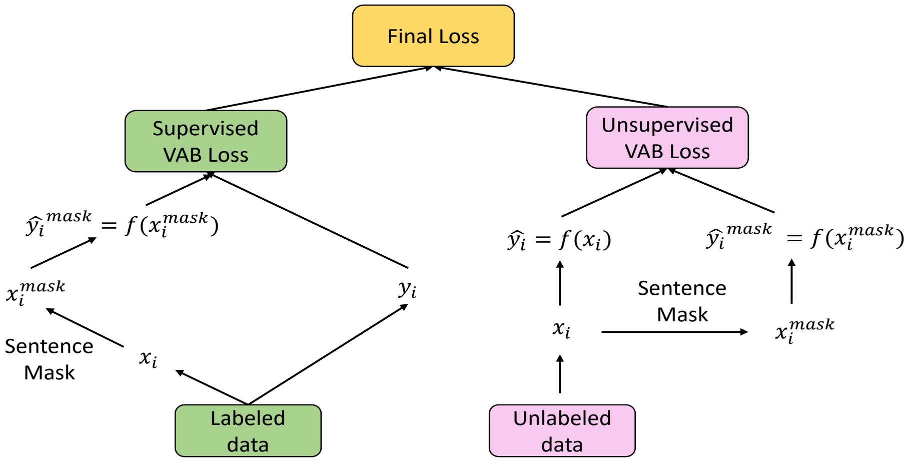


Figure 4.2: The architecture of variational contextual consistency sentence masking (VCCSM).

There are two key modules (variational contextual sentence masking and consistency training) in our proposed model. Variational contextual sentence masking is applied on both labeled and unlabeled datasets. Consistency training is only utilized on the unlabeled dataset.

In the training on the labeled dataset, variational contextual sentence masking module are used to extract the key sentences by using contextual masking implemented by a LSTM model. Then the supervised variational bottleneck loss is calculated between the predictions on the key sentences after masking and the ground truth label. The model architecture on how to train the labeled dataset is shown in the left part of Figure 4.2.

In the training on the unlabeled dataset, different from the prior works, the consistency training are conducted to improve the model interpretability along with the accuracy. More specifically, we replace the traditional noise injection methods (e.g., additive Gaussian noise, dropout noise, and adversarial noise [127, 109, 30]) by our sentence masking method in our consistency training. We first make the first prediction using the key sentences after masking and the second prediction based on all the sentences without masking. The unsupervised variational bottleneck loss is calculated between these two predictions and the goal is to minimize the difference. The model architecture on how to train the unlabeled dataset is shown in the right part of Figure 4.2.

4.2.2 Variational Contextual Sentence Masking

Inspired by Chen and Ji [23], we add one mask layer M after the sentence embedding layer to help the model extract the key sentences. The sentence masking layer is denoted by $M = [M_1, M_2, ...M_j..., M_S]$ and S is the maximum number of sentences in a research paper in the RRP dataset. The embedding of each sentence is concatenated by word embeddings included in this sentence.

In our model, each $M_j \in \{0, 1\}$ is a binary random variable which decides whether mask the j-th sentence or not. For each sentence, whether mask it or not should based on both itself and its context (sentences around it). Therefore, an LSTM model is used to predict M_j given the whole document and the current j-th sentence as the input, where $M_j = \text{LSTM}(x, x_j)$,

$j = 1, 2, ..., S$. The contextual sentence mask layer M_j together with sentence embeddings construct the real input of our neural network text classifier, which is denoted by:

$$Z = X^{\text{mask}} = M \bigodot X, \tag{4.1}$$

where $\bigodot$ is an element-wise multiplication, X represents the original input of all the samples, X^{mask} denotes the real input of neural text classifier after masking X. We aims to optimize M so that our VCCSM model can extract the key sentences for each research paper in the RRP dataset.

The information bottleneck theory is used to learn the input X's encoding Z with maximal information on predicting the target Y while keeps the least redundant information of input X [148, 2]. As proven effective in identifying important features [23], we applied the information bottleneck framework in our model and want to make $Z = X^{\text{mask}}$ maximally expressive on predicting the target Y while being maximally compressive on the original input X. Therefore, according to the standard information bottleneck theory [148], our objective function is denoted as follows:

$$\max_{Z} I(Z; Y) - \beta \cdot I(Z; X), \tag{4.2}$$

where the definitions of X and $Z = X^{\text{mask}}$ are given in Equation 4.1. Y is the target, $I(\cdot; \cdot)$ denotes the mutual information, and $\beta \in \mathbb{R}_+$ is a coefficient that balances the two terms in the information bottleneck objective function.

Nonetheless, directly computing the Equation 4.2 is usually computationally challenging. Therefore, we used the variational inference method to construct a lower bound of Equation 4.2. Having this lower bound, the reparameterization trick [77] can be applied to conduct the optimizing utilizing stochastic gradient descent. In this chapter, we just listed the lower bound in the Equation 4.3 and the derivation details is shown in Appendix A.1.

We assume that the true joint distribution is $P(X, Y, Z)$ and X, Y, Z are random variables having the following conditional dependency property: $Y \leftrightarrow X \leftrightarrow Z$. x, y, z are instances of random variables X, Y, Z respectively. The lower bound of Equation 4.2 is listed as follows:

$$\begin{aligned}&\sum_{x,y,z} P_X(x)P_{Y|X}(y|x)P_{Z|X}(z|x)\log Q_{Y|X}(y|z)\\&-\beta\sum_{z,x} P_X(x)P_{Z|X}(z|x)\log\frac{P_{Z|X}(z|x)}{Q_Z(z)}\end{aligned}\tag{4.3}$$

To compute Equation 4.3, we use the empirical data distribution including two Delta functions to approximate the $P_{X,Y}(x, y)$. Therefore we have the loss function of variational information bottleneck (VAB) as follows:

$$\begin{aligned}\ell_{vib} = -(&\mathbb{E}_{P_{X,Y}(x,y)}[\mathbb{E}_{P_{Z|X}(z|x)}[\log(Q_{Y|Z}(y|z)]\\&-\beta\cdot\text{KL}[P_{Z|X}(z|x)||Q_Z(z)]])\end{aligned}\tag{4.4}$$

4.2.3 Consistency Training based on Variational Contextual Sentence Masking

In this chapter, we utilized a particular consistency training to leverage the unlabeled dataset to further improve the interpretability. More specifically, we replace the traditional noise inject method in the regular consistency training by our Contextual Sentence Masking module to generate the masked input x^{mask} given each input x in the unlabeled dataset which can be written as follows: $x^{\text{mask}} = M \cdot x$. We also used the information bottleneck framework in the consistency training. The only difference comparing with the supervised training is that we replace the ground truth label by the prediction $\hat{y}_u$ give the all the sentences as then input without masking.

4.2.4 Variational Information Bottleneck (VAB) Loss Function

As shown in Figure 4.2, our VAB loss function has two key parts: a supervised VAB loss ℓ_{su} and an unsupervised VAB loss ℓ_{un}. The same model is optimized in both losses.

Supervised VAB Loss Since we have ground truth labels in the labeled dataset, the supervised VAB loss ℓ_{su} is the same as the VAB loss ℓ_{vlb} in Equation 4.4 and it is denoted as follows:

$$\begin{aligned} \ell_{su} = -(\mathbb{E}_{P_{X,Y}(x,y)}[\mathbb{E}_{P_{Z|X}(z|x)}[\log(Q_{Y|Z}(y|z)] \\ - \beta \cdot \text{KL}[P_{Z|X}(z|x)||Q_Z(z)]]]) \end{aligned} \tag{4.5}$$

where $P_{X,Y}(x, y)$ refers to empirical distribution of complete observations.

Unsupervised VAB Loss As for the unsupervised VAB loss, the only difference comparing with the supervised one is to replace the ground truth label y by the prediction $\hat{y} = f(x)$ given the all the sentences in the research paper x (without masking) as the input and and it is denoted as follows:

$$\ell_{un} = -(\mathbb{E}_{P_X(x)}[\mathbb{E}_{P_{Z|X}(z|x)}[\log(Q_{Y|Z}(\hat{y}|z)] \\ - \beta \cdot \mathrm{KL}[P_{Z|X}(z|x)||Q_Z(z)]]) \tag{4.6}$$

where $P_X(x)$ refers to empirical distribution of incomplete observations.

Total Loss In summary, our full training objective ℓ can be written as follows:

$$\ell = \ell_{su} + \alpha \cdot \ell_{un} \tag{4.7}$$

where $\alpha > 0$ is a balancing hyper parameter about these two items of losses. Our goal is to minimize the full training objective ℓ.

4.3 Experiments

Our proposed VCCSM method is evaluated with two typical neural network models commonly used in text classification, LSTM [66] and BERT [39]. In the experiments, we mainly show the results on RRP datasets. We also show the results on another ECHR Dataset since our proposed method can be generally extended to other text classification tasks.

4.3.1 Experimental Setup

4.3.1.1 Datasets

RRP Dataset Luo et al. [101] proposed the RRP dataset and the details are described in last chapter. In summary, RRP dataset contains 399 labeled and 2,170 unlabeled research articles in social science fields. As for the training/testing splitting, we follow the same setting as in last chapter. 300 (150:1;150:0) samples are randomly selected from labeled dataset and all the samples from unlabeled dataset construct our training dataset. The remaining 99 (51:1;48:0) samples in the labeled dataset are considered as the testing dataset.

ECHR Dataset European Convention of Human Rights (ECHR) [18] is a publicly available English legal judgment prediction dataset containing 11,478 cases. In each case, there are a list of paragraphs describing the facts. The task is to predict whether one given case is judged as violated or not based on the text description. Training, development, testing datasets contains 7,100, 1,380 and 2,998 cases. The average number of words for training, development, and testing datasets are 2,421, 1,931, and 2,588, respectively.

4.3.1.2 Implementation Details

The LSTM model we used has a bidirectional hidden layer, and it's initialized with 300-dimensional google's pre-trained word embeddings. We fix the embedding layer and update other parameters in LSTM to achieve the best performance. As for BERT model, a published BERT pre-trained model ("bert-base-uncased"[1]) is utilized as the embedding layer of LSTM

[1] https://huggingface.co/bert-base-uncased

model. We first use our corpus to pre-train the BERT model and then fine-tune it in the VCCSM classifier's training. In each epoch, the model is first trained on labeled data, followed by unlabeled data. The hidden state of the [CLS] token of the last layer is considered as the sentence representation.

Because the average length (words) of all the documents in the labeled and unlabeled datasets is about 10,000, we set the the maximum length of words in our paper to 10,000. Since VCCSM method is sentence masking and we need to split the text of research paper into sentences. We use period, question mark, and semicolon to conduct the splitting. After some statistical analysis, the average length (words) of each sentence is around 25. For a fair comparison with word masking method, we set the maximum length of sentences in each document to 400. It means that we set the maximum length of words in each document to 10,000 in all models.

4.3.1.3 Interpretability Metrics

AOPC The first interpretability metric we used is area over the perturbation curve (AOPC) [128, 113] which is obtained by computing the average change of prediction probability by deleting top n important words and it can evaluate the model interpretablity on faithness. Since our proposed VCCSM is sentence masking method, we calculate the average change of prediction probability by deleting top n key sentences in the explanations of the papers. Therefore, AOPC used in our paper is defined as follows:

$$\text{AOPC}(f) = \frac{1}{T+1} \sum_{i=1}^{T} \left(f(x_i) - f(x_i \backslash \{s_1, ..., s_n\}) \right),$$

where $f(x_i \backslash \{s_1, ..., s_n\})$ is the probability for the predicted class on the i_{th} document in RRP when the top n sentences on importance are removed. Higher AOPC score is better.

Post-hoc Accuracy The second interpretability metric utilized in this paper is post-hoc accuracy metric [25] which is computed by counting how many testing examples' predictions are changed by utilizing only extracted top n words to classify. For our VCCSM models, we used top n key sentences. The formula to calculate the post-hoc accuracy in our paper is as follows:

$$\text{ACC}_{post}(f, n) = \frac{1}{T} \sum_{i=1}^{T} \mathbb{1}[f(\{s_1, ..., s_n\}) = f(x_i)],$$

where T is the number of examples in the testing dataset, $\{s_1, ..., s_n\}$ are the top n sentences on importance in the i_{th} document. Higher post-hoc accuracy is better.

4.3.2 Experimental Results

We tested our proposed models on two text classification datasets (RRP along with ECHR), and the details about prediction accuracy and interpretability are described in this section.

	RRP			ECHR		
Methods	Acc	AOPC	Post-hoc	Acc	AOPC	Post-hoc
LSTM Word Masking [23]	60.61%	11.16%	50.51%	84.86%	10.32%	65.84%
BERT's Attention Weights (words)	64.65%	11.70%	60.61%	84.26%	15.06%	73.75%
BERT Word Masking [23]	65.66%	12.05%	61.62%	85.06%	16.30%	76.38%
SOTA Extractive Summarization [35]	65.66%	12.86%	57.58%	85.39%	19.57%	75.52%
BERT's Attention Weights (sentences)	65.66%	13.62%	62.63%	85.39%	22.61%	81.49%
LSTM Sentence Masking + Contextual + Consistency	65.66%	22.19%	63.64%	86.06%	30.53%	84.22%
BERT Sentence Masking + Contextual + Consistency	**68.69%**	**24.02%**	**65.66%**	**87.66%**	**32.78%**	**86.59%**

Table 4.1: Comparison between VCCSM and other methods on testing accuracy, area over the perturbation curve (AOPC), and post-hoc accuracy on RRP and ECHR datasets.

4.3.2.1 Quantitative Evaluation

We evaluate the interpretability of VCCSM model against other types of models via the AOPC [128, 113] and post-hoc accuracy [25] metrics. We also listed the performance with varying number of the unlabeled data in Appendix A.2 and it shows that the performance become higher with more unlabeled data.

Table 4.1 shows the results of VCCSM (LSTM & BERT) and other interpretable models on the RPP and ECHR datasets with top 500 words (word based methods) or 20 sentences (sentence based methods). Simialr results are obtained with varying number of sentences. For BERT's attention weights model, we extracted the words' attention weights of all heads in the last layer and average them. As for BERT's attention weights (sentences), we average the words' averaged weights in each sentence as its sentence representation. Extractive summarization models can also extract the key sentences for each document. In this section, we used the recent extractive summarization method [35] as the baseline. We conduct the training on arXiv + PubMed [31] and our labeled + unlabeled datasets (the abstract are the summary). Training on arXiv + PubMed aims to generalize the model and make the model extract a more comprehensive of information instead of only abstract in the research paper. We can observe that our proposed models perform better than other methods in both interpretability and prediction performance on both RRP and ECHR datasets.

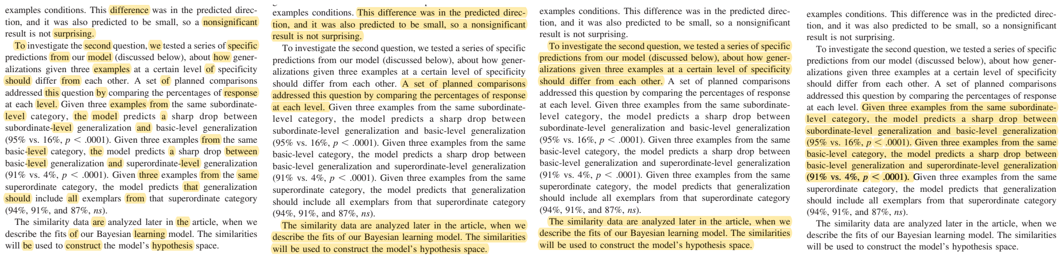
Model	Methods	Accuracy	AOPC	Post-hoc
LSTM	Proposed LSTM VCCSM	65.66%	22.19%	63.64%
	w/o consistency training	62.63%	14.29%	60.61%
	w/o contextual masking	63.64%	19.10%	62.63%
BERT	Proposed BERT VCCSM	68.69%	24.02%	65.66%
	w/o consistency training	65.66%	16.38%	62.63%
	w/o contextual masking	66.67%	21.16%	64.65%

Table 4.2: Ablation study of proposed VCCSM (LSTM & BERT Sentence Masking + Contextual + Consistency) on testing accuracy, area over the perturbation curve (AOPC), and post-hoc accuracy on RRP dataset.

Ablation Study In order to validate different modules in our proposed VCCSM method, we conduct the ablation study on the RRP dataset as shown in Table 4.2. We observe the drop after removing contextual masking or consistency training (on the unlabeled data) which shows that each component benefit to the model. It is noting that we observe a larger drop on both accuracy and two interpreatability metrics without the consistency training on the unlabeled data which demonstrates that consistency training contributes more to the model.

4.3.2.2 Qualitative Evaluation

In this section, we conduct the qualitative evaluations and compare the explanations of different models intuitively by highlighting the words or sentences. Specifically, we draw on the Open Science pratices (e.g., mentioning how to access the data) as indicators of high reproducibility, because these practices are proposed as solutions to the reproducibility crisis in the science community [136, 46, 12, 41, 104]. Some of those indicators which are easier



examples conditions. This difference was in the predicted direction, and it was also predicted to be small, so a nonsignificant result is not surprising.

To investigate the second question, we tested a series of specific predictions from our model (discussed below), about how generalizations given three examples at a certain level of specificity should differ from each other. A set of planned comparisons addressed this question by comparing the percentages of response at each level. Given three examples from the same subordinate-level category, the model predicts a sharp drop between subordinate-level generalization and basic-level generalization (95% vs. 16%, $p < .0001$). Given three examples from the same basic-level category, the model predicts a sharp drop between basic-level generalization and superordinate-level generalization (91% vs. 4%, $p < .0001$). Given three examples from the same superordinate category, the model predicts that generalization should include all exemplars from that superordinate category (94%, 91%, and 87%, *ns*).

The similarity data are analyzed later in the article, when we describe the fits of our Bayesian learning model. The similarities will be used to construct the model's hypothesis space.

BERT Word Masking

examples conditions. This difference was in the predicted direction, and it was also predicted to be small, so a nonsignificant result is not surprising.

To investigate the second question, we tested a series of specific predictions from our model (discussed below), about how generalizations given three examples at a certain level of specificity should differ from each other. A set of planned comparisons addressed this question by comparing the percentages of response at each level. Given three examples from the same subordinate-level category, the model predicts a sharp drop between subordinate-level generalization and basic-level generalization (95% vs. 16%, $p < .0001$). Given three examples from the same basic-level category, the model predicts a sharp drop between basic-level generalization and superordinate-level generalization (91% vs. 4%, $p < .0001$). Given three examples from the same superordinate category, the model predicts that generalization should include all exemplars from that superordinate category (94%, 91%, and 87%, *ns*).

The similarity data are analyzed later in the article, when we describe the fits of our Bayesian learning model. The similarities will be used to construct the model's hypothesis space.

Attention Weights (sentences)

examples conditions. This difference was in the predicted direction, and it was also predicted to be small, so a nonsignificant result is not surprising.

To investigate the second question, we tested a series of specific predictions from our model (discussed below), about how generalizations given three examples at a certain level of specificity should differ from each other. A set of planned comparisons addressed this question by comparing the percentages of response at each level. Given three examples from the same subordinate-level category, the model predicts a sharp drop between subordinate-level generalization and basic-level generalization (95% vs. 16%, $p < .0001$). Given three examples from the same basic-level category, the model predicts a sharp drop between basic-level generalization and superordinate-level generalization (91% vs. 4%, $p < .0001$). Given three examples from the same superordinate category, the model predicts that generalization should include all exemplars from that superordinate category (94%, 91%, and 87%, *ns*).

The similarity data are analyzed later in the article, when we describe the fits of our Bayesian learning model. The similarities will be used to construct the model's hypothesis space.

Extractive Summarization

examples conditions. This difference was in the predicted direction, and it was also predicted to be small, so a nonsignificant result is not surprising.

To investigate the second question, we tested a series of specific predictions from our model (discussed below), about how generalizations given three examples at a certain level of specificity should differ from each other. A set of planned comparisons addressed this question by comparing the percentages of response at each level. Given three examples from the same subordinate-level category, the model predicts a sharp drop between subordinate-level generalization and basic-level generalization (95% vs. 16%, $p < .0001$). Given three examples from the same basic-level category, the model predicts a sharp drop between basic-level generalization and superordinate-level generalization (91% vs. 4%, $p < .0001$). Given three examples from the same superordinate category, the model predicts that generalization should include all exemplars from that superordinate category (94%, 91%, and 87%, *ns*).

The similarity data are analyzed later in the article, when we describe the fits of our Bayesian learning model. The similarities will be used to construct the model's hypothesis space.

BERT VCCSM

Figure 4.3: Highlighted explanations (words or sentences) of BERT word masking, attention weights (sentences), SOTA extractive summarization, and BERT VCCSM methods for a paragraph in one replicable research paper "Word Learning as Bayesian Inference" in Psychological Review.

to check are listed as below: (1) Publish materials, data, and code; (2) Preregister studies and submit the reports; (3) Conduct the replications by themselves; (4) Collaborate with others; (5) P-value[2] is close to 0.5.

We conduct the case studies on the testing dataset and find that our proposed methods can highlight more sentences which are related to the indicators mentioned above. A case study is shown in Figure 4.3. More specifically, Figure 4.3 shows highlighted explanations (words or sentences) of BERT word masking, attention weights (sentences), SOTA extractive summarization, and BERT VCCSM methods for a paragraph in one replicable research paper "Word Learning as Bayesian Inference" [163] in Psychological Review. In this case study, we extracted top 200 sentences or 5,000 words (only for BERT word masking method) but only show one paragraph highlighted results. Although all the methods provide the correct prediction, our VCCSM highlights the sentences which are related to the indicators described above. It is noting that the highlight words of BERT word masking is not so readable for the long research

[2] Probability of obtaining test results at least as extreme as the results actually observed, under the assumption that the null hypothesis is correct.

paper. Attention weights (sentences) and SOTA extractive summarization methods can provide informational sentences but the highlighted sentence are not related to the indicators described above. BERT VCSSM can highlight p-value sentences which are related to the indicators mentioned above.

4.3.2.3 Discussion on Plausibility of Predicting Research Replicability using Text

By looking into RRP's labeled dataset and conducting the cases studies carefully such as in Figure 4.3, we discuss on whether classifying results in a research paper as replicable using text is actually sufficient to replicate the results, which is the central premise this paper is based on. Non-replicability of scientific studies largely results from unscientific, unethical research practices (e.g., p-hacking, selective reporting, data manipulation). Such practices can be manifested in the texts of research papers such as the reports of p-values, experimental procedures, etc. Generally speaking, the more problematic practices a research paper involves, the less likely its findings are valid, and the less likely it will be reproduced. Hence, by modeling the replicability of research paper with regard to its textual components that are potentially linked with the problematic practices, we can classify whether a research paper can be replicated and identify the focal sentences relevant to the prediction.

4.4 Related Work

Blackbox Research Replication Prediction Research Replication Prediction, knowing whether a published research result is replicable or not, is important. Recently, several large scale of direct replication projects have been conducted in social science studies to alleviate the replication crisis. But the cost of direct replication is too high to have a large size of annotated dataset. Therefore, an alternative ML method that is much cheaper and more efficient than direction replication is utilized in RRP. Luo et al. [101] proposed a neural text classifier to achieve the best performance on RRP. But their model is a blackbox and cannot provide faithful explanations about why a research paper is predicted as replicable or non-replicable.

Interpreting Neural Networks Various approaches have been proposed to interpret neural network models from the post-hoc manner, such as gradient-based [137, 63, 143], attention-based [132], decomposition-based [111, 138], example-based methods [81, 57], and word masking [23]. However, these interpretation methods have their own limitations, including only work with specific neural network model, render doubts on faithfulness, and need additional work to provide the explanations based on trained models. In this paper, we focus on model-agnostic explanation methods. More specifically, we follow the research of masking methods which can improve both the prediction performance and interpretability by adopting information bottleneck framework [148, 2] to identify important sentences.

Improving interpretability via word masking Chen and Ji [23] proposed a word masking method which can automatically select important words in the training process and build interpretable neural text classifiers by formulating their problem in the framework of information bottleneck. The proposed solution mainly deals with the short text and the average length (words) in all the seven datasets they used are less than 300. Four of them are less than 25. In constrast, the average length (words) of research papers in our RRP task is about 10,000 which is much longer than the ones used in [23]. Therefore, we view word masking as insufficient for our task.

On the other hand, Chen and Ji [23] learn independently on whether each word is masked or not. Different from prior work, we utilized the context information (whether other sentences in the same paper are masked or not) of each sentence by applying LSTM models to decide whether to mask this sentence or not. We hypothesize that context masking is better than independent masking, especially for long documents such as the research papers in RRP.

Consistency Training on Unlabeled Dataset The annotated data in RRP is collected using direction replication and its size is small. Therefore, weakly supervised learning methods need to be used to improve the model performance in RRP with the help of the unlabeled dataset. The existing weakly supervised methods applied in RRP focus mainly on improving the prediction performance, but less so about the model interpretability.

Consistency training can improve the robustness of models by regularizing model predictions to be invariant to small noise applied to input examples [127, 30]. Xie et al. [162] proposed to substitute the traditional noise injection methods in the consistency training with high quality data augmentations so that a new consistency training based weakly supervised method is proposed and the performance is improved with the help of unlabeled dataset. But they focused only on improving the prediction performance.

In this paper, we conduct the consistency training on the unlabeled dataset to improve both prediction performance and interpretability by substituting the traditional noise injection methods with sentence masking methods. More specifically, we first mask the unimportant sentences and keep the critical sentences. Then we make the predictions on the kept key sentences the same as the ones based on all the sentences in the research paper without masking. Finally, we conducted the consistency check by minimizing the difference between them.

Chapter 5

Machine Truth Serum: a Surprisingly Popular Approach to Improving Ensemble Methods in Classification

5.1 Introduction

In this chapter, we focus on improving the quality of noisy labels by proposing a new ensemble method to reveal the correct minority answer when the majority answer is wrong in the weakly supervise learning methods.

Wisdom of the crowd shows the power of aggregating opinion from a diverse of groups. Even though this idea was proposed to aggregate the human opinions, it has been successfully applied in the Machine Learning (ML) and the most typical one is Ensemble method. Ensemble methods are widely applied in various settings such as supervised (SL) and semi-supervised (SSL) learning by combining several different types of ML models [42]. More specifically, ensemble methods can be utlized to enhance the final predictions in SL and improve the quality of pseudo or noisy labels for the data augmentations methods in SSL (e.g., MixMatch [9]). Many ensemble methods are proposed to solve various types of problems [65, 124, 175, 89, 176, 177]. However, all the methods mentioned above are based the same assumption that the majority is likely to be correct. While enjoying the assumption that the majority answer is tending to be correct, it is questionable where the special knowledge is required for obtaining the ground truth answer, especially when the knowledge is not widely shared and owned by a few experts [26, 134, 119]. Similar to the setting of aggregating the human knowledge, the same challenge is faced in the ensemble methods. For example, when we ensemble several deep learning models [53], one of them is a state-of-the-art (SOTA) and gets the best performance. For some samples, the prediction result of this SOTA model may be the correct minority. In this situation, applying the majoirty voting rule can lead to the wrong answer.

Our goal is to explore whether there is a better ensemble method which overcome the shortcoming of the majority voting rule, where the minority correct answer can also be revealed. Inspired by the *Bayesian Truth Serum* (BTS) [118, 119] which are proposed to solve the problem in the setting of aggregating the human opinions, we transferred the idea to ensemble methods in the ML field. The core idea in BTS is simple and elegant: the "surprisingly" popular one (having a higher posterior than prior) is the correct answer instead of the one obtained by applying the majority voting rule (only having a higher posterior). In BTS, the prior is constructed by eliciting a peer prediction information which is "how many other people would agree with you" for each agent. However, in ensemble methods, we cannot ask the classier the subjective question "how many other classifiers would agree with you". Therefore, if we want to transferred the idea in

BTS into ensemble methods, the new methods are needed to proposed.

In this chapter, we aims to extended the idea in BTS to further improve the performance in the ensemble methods in the context of SL and SSL text classification. Again, the challenge is that we cannot elicit a belief from a classifier "how many other classifiers would agree with themselves" which make computing the prior difficult. Therefore, we proposed two ML aided algorithms to mimic the procedure of reporting the peer prediction information for each classier, which are jointly named *Machine Truth Serum* (MTS). In Heuristic Machine Truth Serum (HMTS), for each classifier (an agent), a regressor model is trained to prediction the peer prediction information utilizing a processed training dataset. Having the predictions from these regressors, we can computer the prior and directly apply the idea in BTS to decide minority answer is "surprising popular" by comparing the prior and posterior. In Discriminative Machine Truth Serum (DMTS), we directly train a model to predict whether the minority answer is correct or not. We applied our proposed HMTS and DMTS on SL and SSL text classification tasks. For SL, MTS methods are used to enhance the ensemble methods in the final predictions step. As for SSL, MTS methods are utilized to improve the quality of pseudo or noisy labels in the data augmentation steps. The theoretical analysis for the correctness of our proposed MTS approaches is also provided and they are very practical to implement and run based on the computational complexity analysis.

5.2 Preliminary

In this chapter, we consider supervised and semi-supervised classification problems. Nonetheless, for simplicity of demonstration, our main presentation focuses on binary classification. A multi-class extension of our method is presented in Section 5.3.3.

5.2.1 Supervised Classification Tasks

Suppose that we have a training dataset $\mathcal{D}_L := \{(x_l, y_l)\}_{l=1}^{N_L}$ and a test dataset $\mathcal{T} := \{(x_t, y_t)\}_{t=1}^{N_T}$, where $x_l \; or \; x_t \in X \subseteq \mathbb{R}^d$ is a d-dimensional feature vector and $y_l \; or \; y_t$ is its true class label. We have J baseline classifiers $\mathcal{F} := \{f_1, f_2, ..., f_J : X \to \{0, 1\}\}$ that map each feature vector to a binary classification outcome. Ensemble method such as boosting algorithms can combine $\{f_1, f_2, ..., f_J\}$ to get better prediction results than each single one. For instance, Random Forest first applies the bootstrap aggregating to train multiple different decision trees to correct overfitting problems of decision trees. After training, the majority rule will be applied to generate the prediction result. We define the binary cross-entropy (BCE) loss of supervised classification as $\ell(f_j(x_l), y_l) := -[y_l \cdot \ln(f_j(x_l)) + (1 - y_l) \cdot \ln(1 - f_j(x_l))]$ for the j-th classifier on each data point (x_l, y_l) in the training dataset. Therefore, the empirical risk of the supervised classifier for $f_j, j = 1, ..., J$ using true labels is as follows:

$$L_1(f_j, D_L) = \frac{1}{N_L} \sum_{l=1}^{N_L} \ell(f_j(x_l), y_l).$$

The above dependence on the majority voting rule is ubiquitous in ensemble methods. The key assumption of using the majority rule is that the majority is more likely to be correct than random guessing. Denoting as $\mathsf{Maj}(\{f_1(x), f_2(x), ..., f_J(x)\})$ the majority answer from the J classifiers, formally, most, if not all, methods require that

$$P(\mathsf{Maj}(\{f_1(x), f_2(x), ..., f_J(x)\}) \neq y) < 0.5$$

Our goal is still to construct a single aggregator $\mathcal{A}_{D_L}(\{f_1, f_2, ..., f_J\})$ that takes the classifiers' predictions on each supervised data point as inputs and generates an accurate aggregated prediction. But we aim to provide instruction to cases where it is possible that

$$P(\mathsf{Maj}(\{f_1(x), f_2(x), ..., f_J(x)\}) \neq y) > 0.5$$

The challenge is to detect when the minority population has the true answer.

5.2.2 Semi-supervised Classification Tasks

In the semi-supervised classification tasks, there is also an unlabeled dataset $\mathcal{D}_U := \{(x_u, \cdot)\}_{u=1}^{N_U}$, where the labels are missing or unobservable. Many methods are proposed to generate the high-quality pseudo labels of unsupervised dataset [9, 162, 8, 141, 164] and we can have a new $\mathcal{D}_U := \{(x_u, y_u)\}_{u=1}^{N_U}$. Let $N := N_L + N_U$. We unify the whole data including both labeled and unlabeled as $\mathcal{D} := \{(x_n, y_n)\}_{n=1}^{N}$. $\{y_n\}_{n=1}^{N_L}$ are the true labels of supervised dataset and $\{y_n\}_{n=N_L+1}^{N}$ are the pseudo labels of unsupervised dataset. Compared with supervised classification tasks, the information of unsupervised should be leveraged to improve the performance. Recent SSL methods usually apply the consistency regularization methods to make use of unsupervised data, where the output of original inputs and their data augmented ones should be consistent [86, 123, 146, 109, 71, 9, 141, 24]. In this paper, we consider MixMatch [9] and MixText [24] as our ensemble baseline methods because they generated the high-quanlity explicit pseudo labels for unsupervised data using ensemble methods.

For each unlabeled data $x_u, u = 1, ..., N_U$, the pseudo label can be generated by ensemble the model predictions of its data augmentations. We set the number of data augmentations for each unlabeled data to M. The data augmentation is denoted by $x_{u,m} := f_{\text{augment}}(x_u), m = 1, ..., M; u = 1, ..., N_U$. The pseudo label y_u can be generated based on M model predictions of data augmentations as $y_u = f_{\text{sharpen}}\left(\frac{1}{M}\sum_{m=1}^{M}\bar{f}_m(x_{u,m})\right), m = 1, ..., M; u = 1, ..., N_U$, where $\{\bar{f}_1, \bar{f}_2, ..., \bar{f}_M\}$ are extra M classifiers which are only utilized to generate better pseudo labels of unsupervised data and ensemble methods are limited to applying on this pseudo labeling process (not used in final classification prediction). We denoted the classifier conducting the final classification prediction as $f(\cdot)$. The function $f_{\text{sharpen}}(\cdot)$ can reduce the entropy of pseudo labels, e.g., setting to one-hot encoding based on the probabilities of different class labels [141]. The empirical risk of the semi-supervised classifier for $f(\cdot)$ using pseudo labels is as follows:

$$L_2(f, D_L, D_U) = \frac{1}{N_L}\sum_{l=1}^{N_L}\ell(f(x_l), y_l) + \frac{1}{N_U}\sum_{u=1}^{N_U}\ell(f(x_u), y_u).$$

Similar to 5.2.1, our goal is to construct a single aggregator $\mathcal{A}_{D_L,D_U}(\{\bar{f}_1, \bar{f}_2, ..., \bar{f}_M\})$ that takes the model predictions of data augmentations on each unsupervised data point as inputs and generates a high-quality pseudo label even the majority of model predictions is wrong. The challenge is still to detect when the minority population has the true answer.

5.2.3 Bayesian Truth Serum

[118] considers the following human judgement elicitation problem: There are a set of agents denoted by $\{a_j\}_{j=1}^J$. The designer aims to collect subjective judgement from each agent about an unknown event $y \in \{0,1\}$ and aggregate accordingly. Each of the agent j needs to report his own predicted label $l_j \in \{0,1\}$ for y, and the percentage of other agents he believes will agree with him $p_j \in [0,1]$. We will also call this second belief information as the *peer prediction information*. Denote the j's local belief of $l_r, r \neq j$ as $l^b_{j,r}, r \neq j$. p_j is defined as follows:

$$p_j = \mathbb{E}_{l^b_{j,r}, r\neq j}\left[\frac{\sum_{r\neq j} \mathbb{1}(l^b_{j,r} = l_j)}{J-1}\right]$$

In above the expectation is w.r.t. $l^b_{j,r}, r \neq j$ - this definition rigorously sets up the formulation, since in BTS, each agent only observes his/her private signals but not others.

We, as the designer, obtain the prediction labels $\{l_j\}_{j=1}^J$ and the percentage information $\{p_j\}_{j=1}^J$ from all the agents. The posterior for each label is defined as the actual percentage of this label which can be easily calculated utilizing the prediction results: (for label 1)

$$\textsf{Posterior}(1) = \frac{\sum_j \mathbb{1}(l_j = 1)}{J} \tag{5.1}$$

In [118, 119], Prelec et al. promote the idea of using the average predicted percentage of the responding label as the approximation of the priors: (for label 1).

$$\textsf{Prior}(1) = \frac{\sum_{j=1}^J p_j^{\mathbb{1}(l_j=1)} \cdot (1-p_j)^{1-\mathbb{1}(l_j=1)}}{J} \tag{5.2}$$

If $\mathsf{Posterior}(1) > \mathsf{Prior}(1)$, label 1 will be taken as the surprisingly more popular answer, which should be considered as the true answer $\hat{y}$, even though it might be in minority's hands. The same rule is applied to label 0. Formally, if we denote $\hat{y}$ as the aggregated answer:

$$\hat{y} = \begin{cases} 1 & \text{if } \mathsf{Prior}(1) < \mathsf{Posterior}(1); \\ 0 & \text{if } \mathsf{Prior}(1) > \mathsf{Posterior}(1). \end{cases} \tag{5.3}$$

The rest of the paper will focus on generalizing the above idea to aggregate classifiers' predictions.

5.3 Machine Truth Serum

In this section, we introduce Machine Truth Serum (MTS). We aim to build a more robust ensemble method which can recover the true answer (in minority's hands) if the majority's answer is wrong. Suppose we have access to a set of basic classifiers. We'd like to build a BTS-ish ensemble method to further improve the model's robustness. The challenge is to compute the priors from the classifiers - machine-trained classifiers do not encode beliefs as human agents do, so we cannot elicit the peer prediction information from them directly. We propose two machine learning aided approaches to perform the generation of this peer prediction information. We first introduce two MTS approaches for binary classification in supervised learning. Then we extend these approaches to multiclass classification case in supervised learning. After describing our proposed methods in supervised learning, we show the MTS methods for binary classification in SSL. Finally, the theoretical analysis of our MTS methods are provided.

5.3.1 Heuristic Machine Truth Serum

We first introduce Heuristic Machine Truth Serum (HMTS). The high-level idea is to train a regression model for each classifier to predict the percent of the agreement from other classifiers on the prediction of each particular data point. After getting the predicted labels and the predicted peer prediction information of the classifiers, we can again approximate the priors using the predicted peer prediction information for each classifier, compute the average and compare it to posterior. In this part, HMTS for binary classification in supervised learning is introduced firstly and its multiclass extension is stated in Section 4.3.

Algorithm 4 Heuristic Machine Truth Serum (Binary classification)

```
Require:
    Input:
    $\mathcal{D}_L = \{(x_1, y_1), ..., (x_{N_L}, y_{N_L})\}$: training data
    $\mathcal{T} = \{(x_1, y_1), ..., (x_{N_T}, y_{N_T})\}$: testing data
    $\mathcal{F} = \{f_j, ..., f_J\}$: classifiers
Ensure:
    Train $J$ classifiers ($\mathcal{F}$) on the training data
    for $j = 1$ to $J$ do
        for $l = 1$ to $N_L$ do
            Compute $\bar{y}_l^j$ according to Eqn.(5.4)
        end for
    end for
    Train machine belief regressors $p_j^-, p_j^+$ on training dataset $\mathcal{D}_j^H := \{(x_l, \bar{y}_l^j)\}_{l=1}^{N_L}$.
    for $t = 1$ to $N_T$ do
        Get $\mathsf{Prior}(x_t, k = 1)$ and $\mathsf{Posterior}(x_t, k = 1)$ according to Eqn.(5.5) and Eqn.(5.7).
        if $\mathsf{Prior}(x_t, k = 1) < \mathsf{Posterior}(x_t, k = 1)$ then
            Output “surprising” answer 1 as the final prediction.
            if $\mathsf{Prior}(x_t, k = 1) > \mathsf{Posterior}(x_t, k = 1)$ then
                Output “surprising” answer 0 as the final prediction.
            end if
        end if
    end for
```

Given the training data $D = \{(x_l, y_l)\}_{l=1}^{N_L}$ and multiple classifiers $\{f_j\}_{j=1}^{J}$, we first try to compute the j-th classifier's "belief" of the fraction of other classifiers that would "agree" with it. Denote this number as $\bar{y}_l^j$ for each training example (x_l, y_l). $\bar{y}_l^j$ can be computed as follows:

$$\bar{y}_l^j = \frac{\sum_{c \neq k} \mathbb{1}\left(f_c(x_l) = f_j(x_l)\right)}{J-1} \tag{5.4}$$

By above, we have pre-processed the training data to obtain $D_{H|j} := \{(x_l, \bar{y}_l^j)\}_{l=1}^{N_L},\ j = 1, ..., J$, which can serve as the training data to predict the peer prediction information of classifier j (again to recall, peer prediction information is the fraction of other classifiers that classifier j believes would agree with it). We then train peer prediction regression models $\{\bar{p}_j\}_{j=1}^{J}$ on $D_{H|j} := \{(x_l, \bar{y}_l^j)\}_{l=1}^{N_L},\ j = 1, ..., J$ respectively to map x_l to $\bar{y}_l^j$. We consider different class labels[1] and will first train two regression models: p_j^- and p_j^+ are two belief regression models of classifier j and trained on the examples whose predicted labels are 0s ($D_{H|j}^- := \{(x_l, \bar{y}_l^j) : f_j(x_l) = 0\}_{l=1}^{N_L}$) and 1s ($D_{H|j}^+ := \{(x_l, \bar{y}_l^j) : f_j(x_l) = 1\}_{l=1}^{N_L}$) respectively.

Then we compute the following prior of label 1 for each $(x_t, y_t) \in \mathcal{T}$ in the testing dataset:

$$\bar{p}_j(x_t) = \begin{cases} p_j^+(x_t) & \text{if } f_j(x_t) = 1; \\ 1 - p_j^-(x_t) & \text{if } f_j(x_t) = 0. \end{cases} \tag{5.5}$$

After obtaining these peer prediction regression results $\bar{p}_j(x_t)$ for all test data points,

[1] In BTS, an agent predicts how many other agents agree with it depending on its own prediction.

the prior and posterior of $(x_t, y_t) \in \mathcal{T}$ in the test dataset are then calculated by,

$$\begin{aligned} P(x_t, 1) :=& \quad \frac{\sum_j \bar{p}_j(x_t)}{J}; \\ Q(x_t, 1) :=& \quad \frac{\sum_j \mathbb{1}(f_j(x_t)=1)}{J}. \end{aligned} \tag{5.6}$$

If $P(x_t, 1) < Q(x_t, 1)$, the "surprising" answer 1 will be considered as the true answer. The decision rule is similar for label 0. The procedure is illustrated in Algorithm 4.

To be noted, training the regressors to estimate the prior instead of directly using Eqn.(4) is necessary. Because, if we don't train the regressors and estimate the prior directly using Eqn.(4), prior will always be equal to posterior and we cannot use the decision rule mentioned above to obtain the "surprising" answer by comparing prior and postrior. For simplicity, the proof for binary classification (multiclass case is similar) is given as follows:

We set $J_1 = \sum_j \mathbb{1}(f_j(x_t) = 1)$ and $J_2 = \sum_j \mathbb{1}(f_j(x_t) = 0)$. Obviously, $J = J_1 + J_2$. Then we can get:

$$\begin{aligned} \bar{y}_t^j(1) =& \frac{\sum_{c1 \neq j} \mathbb{1}\left(f_{c_1}(x_t) = f_j(x_t) = 1\right)}{J-1} \\ \bar{y}_t^j(0) =& \frac{\sum_{c2 \neq j} \mathbb{1}\left(f_{c_2}(x_t) = f_j(x_t) = 0\right)}{J-1} \end{aligned}$$

The above two quantities further help us compute both the posterior and the "direct prior" as follows:

$$
\begin{aligned}
P_{direct}(x_t, 1) :=& \frac{\sum_j [\bar{y}_t^j(1) \cdot \mathbb{1}(f_j(x_t) = 1) + (1 - \bar{y}_t^j(0)) \cdot \mathbb{1}(f_j(x_t) = 0)]}{J}; \\
=& \frac{\sum_j [\bar{y}_t^j(1) \cdot \mathbb{1}(f_j(x_t) = 1)] + \sum_j [(1 - \bar{y}_t^j(0)) \cdot \mathbb{1}(f_j(x_t) = 0)]}{J}; \\
=& \frac{J_1 \cdot \frac{J_1 - 1}{J - 1} + J_2 \cdot (1 - \frac{J_2 - 1}{J - 1})}{J} = \frac{J_1}{J} = \frac{\sum_j \mathbb{1}(f_j(x_t) = 1)}{J}; \qquad (5.7) \\
Q(x_t, 1) :=& \frac{\sum_j \mathbb{1}(f_j(x_t) = 1)}{J}. \qquad (5.8)
\end{aligned}
$$

Therefore, the prior is equal to the postrior by comparing Eqn.(7) and (8). Based on this proof, learning the regressors to estimate the prior instead of directly using Eqn.(4) is necessary.

5.3.2 Discriminative Machine Truth Serum

The Heuristic Machine Truth Serum above relies on training models to predict the peer prediction information for each classifier (which will be used to compute the priors) and compare them to the posteriors, and then decide on whether to follow the minority opinion or not. HMTS closely mimicked the procedure of BTS method in the seed paper. But it is not the most efficient way due to the extra computational cost of regressors. Also, its performance is dependent on the quality of regression models. We notice the above task of determining whether to follow the minority or not is also a binary classification question. This observation inspires us to utilize a classification model to directly predict for each data point whether the minority should be chosen as the answer or not.

We propose Discriminative Machine Truth Serum (DMTS). Again, DMTS for binary classification will be introduced firstly and its multiclass extension is stated in Section 4.3. With DMTS, a new training dataset $D_D := \{x_l, \hat{y}_l\}_{l=1}^{N_L}$ about whether considering the minority as the final answer or not is constructed. Each data $D_D := (x_l, \hat{y}_l)$, for $l = 1, ..., N_L$, in this new training dataset is calculated as follows: for each $(x_l, y_l) \in D$

$$\hat{y}_l = \begin{cases} 1 \text{ if majority of } \mathcal{F} \text{ on } x_l \neq \text{the true label;} \\ 0 \text{ if majority of } \mathcal{F} \text{ on } x_l = \text{the true label.} \end{cases} \tag{5.9}$$

Now with above preparation, predicting whether majority is correct or not becomes a standard classification problem on $D_D := \{x_l, \hat{y}_l\}_{l=1}^{N_L}$. This is readily solvable by applying standard techniques. In our experiments, we will mainly use a Multi-Layer Perceptron (MLP) [53] denoted as $\overline{f}$. $\overline{f}$ is trained on this new training dataset and can directly predict whether we should adopt the minority as the answer or not. $\overline{f}$ does not restrict to MLP and can be other classifiers. We have tried several other methods, such as logistic regression and support vector machine, with similar conclusions obtained. The whole procedure of DMTS is illustrated in Algorithm 5.

5.3.3 Multiclass Extension of HMTS and DMTS

HMTS and DMTS can be extended to multiclass classification problem with the same ideas by modifying them accordingly. In the multiclass case, $k \in \mathcal{Y} = \{0, 1, ..., K-1\}$ is denoted as the class label of the dataset. Consider HMTS first. For each classifier j, we need to consider different class labels of regression models $\{p_j^k\}$, where $k \in \mathcal{Y} = \{0, 1, ..., K-1\}$. p_j^k is the belief regression model of classifier j and trained on the examples whose predicting labels are ks.

Algorithm 5 Discriminative Machine Truth Serum (Binary classification)

```
Require:
    Input:
    $\mathcal{D}_L = \{(x_1, y_1), ..., (x_{N_L}, y_{N_L})\}$: training data
    $\mathcal{T} = \{(x_1, y_1), ..., (x_{N_T}, y_{N_T})\}$: testing data
Ensure:
  for $l = 1$ to $N_L$ do
      Compute $\hat{y}_l$ according to Eqn.(7).
  end for
  Train DMTS classifier $\overline{f}$ on the dataset $\{x_l, \hat{y}_l\}_{l=1}^{N_L}$
  for $t = 1$ to $N_T$ do
      Compute the classification result $\overline{y}_t := \overline{f}(x_t)$
      if $\overline{y}_t = 0$ then
          Stay with the majority answer.
          if $\overline{y}_t = 1$ then
              Predict with the minority answer.
          end if
      end if
  end for
```

Again compute the following prior for each x_l

$$p_j(x_l, k) = \begin{cases} p_j^k(x_l) & \text{if } f_j(x_l) = k; \\ \big(1 - p_j^v(x_l)\big) \cdot r_k & \text{if } f_j(x_l) = v \neq k. \end{cases} \tag{5.10}$$

where $r_k = p_j^k(x_l)/(\sum_{c \in \mathcal{Y}: c \neq v} p_j^c(x_l))$ is defined as the ratio of the k's belief to the summation of all the other classes' beliefs except for class v. In the multi-class classification tasks, we cannot directly obtain the prior of class k — $p_j^k(x_l)$ as in the binary classification by using $(1 - p_j^v(x_l))$ if $f_j(x_l) = v \neq k$. Therefore, the prior regressors for other classes $\{p_j^c(x_l) \mid c \in \mathcal{Y} : c \neq v\}$ need to be utilized to calculate the prior of class k with a normalization parameter r_k.

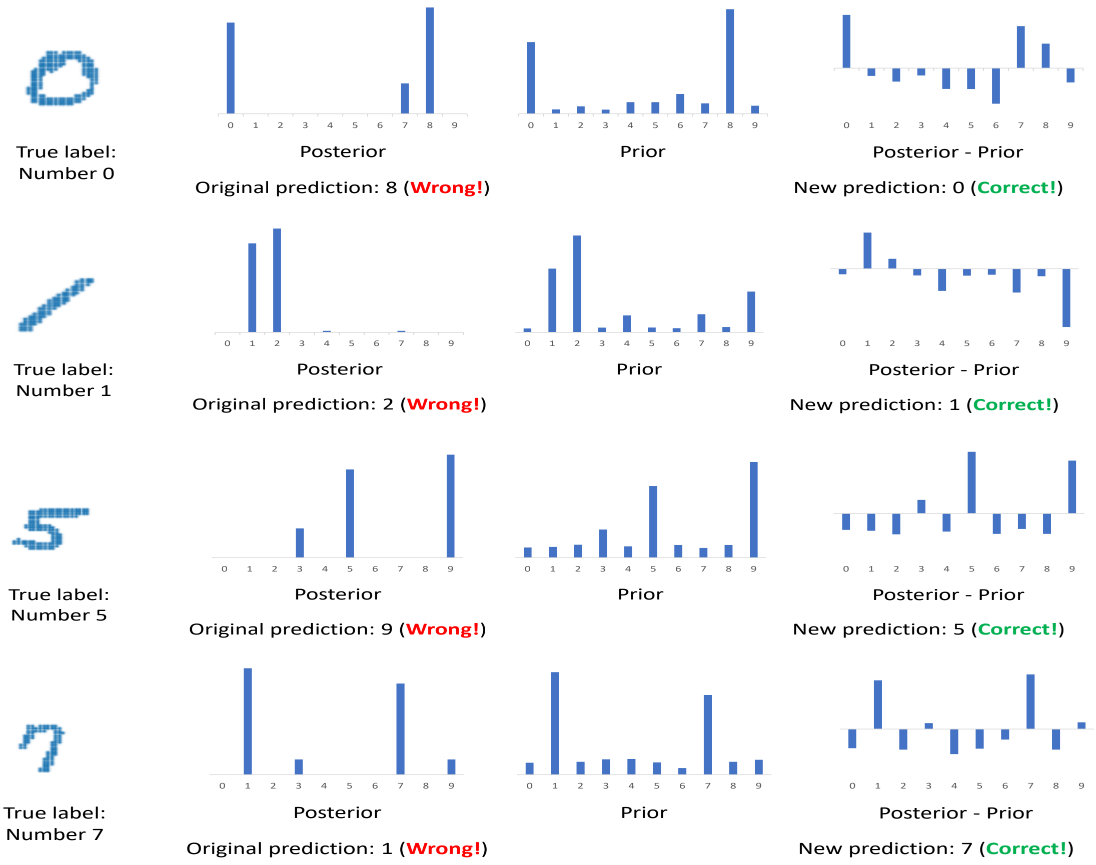


Figure 5.1: Four sample images (number 0, 1, 5, and 7) where HMTS corrects the wrong majority predictions of the majority voting baseline on Pendigits dataset (10 classes) testing dataset. Their posterior, prior, and posterior-prior information are listed

In HMTS, Eqn.(5.7) modify to the following:

$$\begin{aligned} P(x_l,k) &:= \frac{\sum_{j=1}^{J} p_j(x_l,k)}{J}, \\ Q(x_l,k) &:= \frac{\sum_{j=1}^{J} \mathbb{1}(f_j(x_l)=k)}{J} \end{aligned} \tag{5.11}$$

We then compute all the priors and posteriors of each class label based on Eqn.(5.11). It is possible that there exist more than one class labels whose posterior is larger than its prior. We define the set containing all these label classes as $\mathcal{Y}_{sat} = \{k \mid P(x_l, k) < Q(x_l, k)\}$. We then predict the class label which has the biggest improvement from its prior to posterior:

$$\text{argmax}_{k \in \mathcal{Y}_{sat}} \{Q(x_l, k) - P(x_l, k)\}$$

In DMTS, firstly we need to train a model that decides whether to apply the minority as the final answer which are very similar to the binary case. The difference is that we will then choose the minority answer as the predicted answer instead of using majority if i) it has the most votes in the minority answers and ii) the prediction result of classifier obtained in the training phase is 1 (we should use minority).

How does MTS work? In Fig. 5.1, we show four sample images to demonstrate how HMTS correct the wrong majority predictions. We show for these four cases even with high prediction on the wrong class, we are able to correct the prediction by introducing MTS to check on the priors. For example, in the first sample, the wrong prediction (number 8) is provided if we only look at posterior (number 0: 0.400; number 8: 0.467) following the majority rule. But the "surprising popular" correct minority (number 0) will be recovered if we predict based on Posterior - Prior (number 0: +0.117; number 8: +0.054).

5.3.4 HMTS and DMTS for Semi-supervised Classification

In this section, we describe the HMTS and DMTS for SSL classification problem. For simplicity, we consider binary classification and its multiclass extension can be inferred accordingly.

As we focus on applying ensemble methods on the pseudo labels' generation based on data augmentations for each unsupervised data, we first need to compute the m-th data augmentation classifier's "belief" of the fraction of other data augmentation classifiers that would "agree" with it. We first train M data augmentation classifiers $\{\bar{f}_m\}_{m=1}^M$ on the supervised training dataset $D_L = \{(x_l, y_l)\}_{l=1}^{N_L}$. Then we can compute the classification predictions denoted as $\bar{f}_m(x_{l,m}), m = 1, ..., M; l = 1, ..., N_L$ for the data augmentations of supervised training dataset generated by $x_{l,m} :=$ Augmentation$(x_l), m = 1, ..., M; l = 1, ..., N_L$. Denote the m-th data augmentation classifier's "belief" (the fraction of other data augmentation classifiers that would "agree" with it) as $\hat{y}_l^m$ for the data augmentations of each supervised training example $(x_{l,m}, y_l)$. $\hat{y}_l^m$ can be computed as follows:

$$\hat{y}_l^m = \frac{\sum_{c \neq m} \mathbb{1}\left(f_c(x_{l,c}) = f_m(x_{l,m})\right)}{M-1} \tag{5.12}$$

By above, we have pre-processed the supervised training data to obtain $D_{H|m}^L := \{(x_{l,m}, \hat{y}_l^m)\}_{l=1}^{N_L},\ m = 1, ..., M$, which can serve as the training data to predict the peer prediction information of data augmentation classifier m. We then train peer prediction regression models $\{\hat{p}_m\}_{m=1}^M$ on $D_{H|m}^S := \{(x_{l,m}, \hat{y}_l^m)\}_{m=1}^M,\ m = 1, ..., M$ respectively to map $x_{l,m}$ to $\hat{y}_l^m$. We consider different class labels[2] and will first train two regression models: $\hat{p}_{k,m}^-$ and $\hat{p}_{k,m}^+$ are two belief regression models of data augmentation classifier m and trained on the examples whose predicted labels are 0s ($D_{H|m}^{-L} := \{(x_{l,m}, \hat{y}_l^m) : \hat{f}_m(x_{l,m}) = 0\}_{l=1}^{N_L}$) and 1s ($D_{H|m}^{+L} := \{(x_{l,m}, \hat{y}_l^m) : \hat{f}_m(x_{l,m}) = 1\}_{l=1}^{N_L}$) respectively.

[2] In BTS, an agent predicts how many other agents agree with it depending on its own prediction.

Then compute the following prior of label 1 for the data augmentations of each $x_u, u = 1, ..., N_U$ in the unsupervised dataset:

$$\hat{p}_m(x_u) = \begin{cases} \hat{p}_m^+(x_{u,m}) & \text{if } \hat{f}_m(x_{u,m}) = 1; \\ 1 - \hat{p}_m^-(x_{u,m}) & \text{if } \hat{f}_m(x_{u,m}) = 0. \end{cases} \tag{5.13}$$

After obtaining these peer prediction regression results $\hat{p}_m(x_{u,m}), u = 1, ..., N_U$ for all unsupervised data, the prior and posterior of $(x_{u,m}, y_{u,m}) \in \mathcal{D}_U$ in the unsupervised dataset are then calculated by,

$$\begin{aligned} P(x_{u,m}, 1) :=& \quad \tfrac{\sum_m \hat{p}_m(x_{u,m})}{M}; \\ Q(x_{u,m}, 1) :=& \quad \tfrac{\sum_m \mathbb{1}(\hat{f}_m(x_{u,m})=1)}{M}. \end{aligned} \tag{5.14}$$

If $P(x_{u,m}, 1) < Q(x_{u,m}, 1)$, the "surprising" answer 1 will be considered as the true pseudo label in the semi-supervise classification. The decision rule is similar for answer 0.

As for the DMTS, $\{\hat{y}_l\}_{l=1}^{N_L}$ about whether considering the minority as the final pseudo label for each supervised training data x_l or not is constructed. Each data $D_D^L := (x_l, \hat{y}_l)$, for $l = 1, ..., N_L$, in this new training dataset is calculated as follows: for each $(x_l, y_l) \in D_L$

$$\hat{y}_l = \begin{cases} 1 \ \text{ if majority of predictions on } x_{l,m} (m = 1, ..., M) \neq \text{the true label;} \\ 0 \ \text{ if majority of predictions on } x_{l,m} (m = 1, ..., M) = \text{the true label.} \end{cases} \tag{5.15}$$

5.3.5 Theoretical Analysis

We performed a formal analysis of the correctness of our proposed algorithms via proofs adapted from proofs for BTS [119]. Similar to BTS, with MTS, each classifier (i.e., an agent), depending on its own predicted label, will use a different regression model to predict how many other classifiers agree with it. For simplicity, we only present the theorems for binary classification. The proofs of multiclass are similar to the binary case. The details of proofs are reported in Appendix A.3.

To set up for presenting the theorems, we restate our problem: we assume that each classifier $f_j(x)$ can take on any value in the discrete set $\{s_1, ..., s_S\}$ as its features for the simplicity of proof. In practice, conceptually each feature vector can be represented by an assigned (large-enough) categorical number. One can consider $s_i (i = 1, 2, ..., S)$ as a code for each feature vector. Our proof builds on similar assumptions made in [119]:

Assumption 1. *Conditional on each possible label k, $f_j(x), j = 1, 2, ..., J$ are independent from each other, and are identically distributed.*

Assumption 2. *The learner has access to the conditional distribution $\mathbb{P}(f_{-j}(x) \mid f_j(x))$, where $f_{-j}(x)$ denotes the prediction from a randomly selected classifier $r \neq j$.*

We reproduce the following theorems:

Theorem 1. *The correct answer (majority or minority) cannot be deduced by any algorithm if only relying on posterior probabilities, $Q(s_i, k), i = 1, ..., S; k = 0, 1$ because considering either 0 or 1 as the correct label can generate the same posterior probabilities based on the training dataset.*

Theorem 1 implies that any existing ensemble algorithm based on the majority voting rule cannot always infer the true answer no matter either majority or minority is the final true answer. In other words, we cannot decide whether majority or minority is correct if we only know the information of the posterior probabilities Q over all the possible labels. The majority rule applied by the existing ensemble methods is a special case of Theorem 1.

Theorem 2. *For input s_i, the estimate of the prior prediction for the correct classification label denoted as k^* will be strictly underestimated if the prediction probability of the true label is less than 1. We can express this as*

$$P(s_i, k^*) < Q(s_i, k^*) \quad \textit{if } \mathbb{P}(k^* \mid s_i) < 1.$$

We leave more details to the Appendix. Theorem 2 is applicable when the task is difficulty that the true label is only observed by a minority of the classifiers. A hidden assumption is that the minority but expert classifiers hold a stronger belief about the ground truth label than the majority classifiers who predicted wrongly. More formally we assume $\mathbb{P}(Y = k^* \mid f_j(s_i) = k^*) > \mathbb{P}(Y = k^* \mid f_j(s_i) = k)$ for all $k \neq k^*$. The high-level intuition is that the expert classifiers, though being minority, must retain a strong signal to classify a difficult task correctly. While for a non-expert one who predicted wrongly, would not "reason" specially how the hidden true label class. In Section 5.4.1, page 17-19, we have also provided an empirical observation and explanation.

Theorem 2 shows that having the prior information can help improve the robustness of models because the minority correct classification result can be recovered using the rule descried in the theorem when the minority is the true answer instead of the majority answer. In other words, having Theorem 2, the true minority answer can be revealed as correct if the prior probability is less than the posterior one. The existing ensemble methods always adopt the majority result as the final answer and cannot recover the minority correct answer.

As for the training complexity of our algorithm, the training time of HMTS is linear in the number of label classes because of the training of extra regressors. DMTS only needs to train one additional classifier and both the training and the running time are almost the same as the basic majority voting algorithm. Therefore our proposed methods are very practical to implement and run. Detailed discussions are described as follows:

Complexity Analysis of HMTS and DMTS For HMTS, for example in our experiments, another $J \cdot K$ (label classes $\{0, 1, ..., K-1\}$) simple regressors will be trained to predict others' beliefs based on K baseline classifiers. So the total training time is linear in the number of label classes. After training the extra regressors, running the algorithm only requires taking K averages (J of the $J \cdot K$ regressors each) and compare with average posterior. DMTS will only need to train one additional classifier based on K classifiers and both the training and the running time are almost the same as the basic majority voting algorithm. The above complexity analysis shows our methods are very practical.

5.4 Experimental Results

In this section, we present our experimental results. We test our proposed methods by applying in the ensemble final predictions step in supervised classification and in the ensemble data augmentations step in semi-supervised classification.

For supervised classification, we conducted the experiments on five binary and four multiclass real-world classification datasets. Experimental results show that consistently better classification accuracy can be obtained compared to always trusting the majority voting outcomes. As for the splitting of training and testing, the original setting are used when training and testing files are provided. The remaining datasets only give one data file. We adopt 50/50 spliting for the testing results' statistical significance as more data is distributed to testing dataset.

As for semi-supervised classification, we adopt recent methods - MixMatch [9] and MixText [24] as our ensemble baselines. We also used UDA [162] as the baseline but it isn't the ensemble baseline because UDA doesn't use ensemble data augmentation methods to generate the pseudo labels. Both of the ensemble baselines (MixMatch and MixText) mixed labeled and unlabeled datasets utilizing MixUp [173] by applying the recent data augmentations methods to generate low-entropy explicit pseudo labels for unlabeled examples. The difference is that [24] applied MixUp in hidden space so that it is more suitable for text tasks. We used MixMatch to conduct the image classification experiments on CIFAR-10 and CIFAR-100 datasets [83]. MixText is used as the ensemble baseline model for text classification tasks, where Yahoo! Answers [20] and AG News [174] datasets are performed. The experimental results show the effectiveness of our proposed methods by providing better pseudo labels for unsupervised data based on data augmentations than commly used ensemble method using the majority voting rule.

Datasets	Breast cancer	Movie Review	Spambase	Australian	German
Majority (ALL)	92.96% (264/284)	80.25%(321/400)	73.57%(1692/2300)	81.74%(282/345)	76.00%(380/500)
HMTS (ALL)	**95.42%** (264+7/284)	**82.00%** (321+7/400)	75.83% (1692+52/2300)	**84.06%** (282+8/345)	**77.20%** (380+6/500)
DMTS (ALL)	94.01% (264+4/284)	81.75% (321+6/400)	**76.30%** (1692+63/2300)	82.94% (282+4/345)	76.20% (380+1/500)
Majority (HA)	82.35% (42/51)	29.73% (11/37)	28.19% (42/149)	62.79% (27/43)	66.67% (30/45)
HMTS (HA)	**98.04%** (42+8/51)	**43.24%** (11+5/37)	60.40% (42+48/149)	**76.74%** (27+6/43)	**80.00%** (30+6/45)
DMTS (HA)	90.20% (42+4/51)	**43.24%** (11+5/37)	**75.17%** (42+70/149)	72.09% (27+4/43)	68.89% (30+1/45)
Majority (LA)	95.29% (222/233)	85.40% (310/363)	76.71% (1650/2151)	84.44% (255/302)	76.92% (350/455)
HMTS (LA)	94.85% (222-1/233)	**85.95%** (310+2/363)	**76.89%** (1650+4/2151)	**85.10%** (255+2/302)	**76.92%** (350+0/455)
DMTS (LA)	**95.29%** (222+0/233)	85.67% (310+1/363)	76.38% (1650-7/2151)	84.44% (255+0/302)	76.92% (350+0/455)

Table 5.1: Accuracy and the number of increased correct predictions for the three categories of cases, namely, Overall & “High disagreement (HA)” & “Low disagreement (LA)” cases’, using methods of Uniformly-weighted Majority Voting, HMTS, and DMTS on five binary classification datasets. The “high disagreement” means that the difference between the number of predicting 0 and 1 is small. We have 15 classifiers and the instance will be considered as having “high disagreement” if the vote number of majority class is 8 or 9. For other conditions the instance will be considered as having “low disagreement”. In MTS methods, the numbers of HA and LA instances we consider in five datasets are **51, 37, 149, 43, 45** and **233, 363, 2151, 302, 455** respectively. The numbers of five overall testing datasets are **284, 400, 2300, 345, 500**.

5.4.1 Experimental Setup and Results for Supervised Classification

In our binary classification experiments, we consider five commonly used binary classification algorithms which are Perceptron [125], Logistic Regression (LR) [116], Random Forest (RF), Support Vector Machine (SVM) [19], and MLP. In order to test the usefulness of our methods, we experiment with a noisy environment - we flipped the true class label with three noisy rates to construct three binary classifiers for each of the five methods which have mediocre performance on the test datasets. We wanted to diversify our classifiers by introducing different noisy rates (varying the data distribution). Our experiments used 0.06, 0.08, and 0.1 (probability of flipping the label) for each family of classifier. We also tried other values such as 0.1, 0.2, and 0.3, and we reached similar conclusions. In total, 15 different classifiers are obtained as the baseline classifiers.

In this subsection, we report the experimental results on five binary classification datasets and analyze when and why our proposed MTS methods perform better than majority voting.

Table 5.1 presents the experimental results of accuracy and the number of increased correct predictions for the three categories of cases, namely, Overall & “High disagreement (HA)” & “Low disagreement (LA)” cases’, using methods of Uniformly-weighted Majority Voting, HMTS, and DMTS on five binary classification datasets. Specifically, “HA” cases are the tasks/instances when the ensemble is least certain about. “LA” ones are the relatively easier tasks/instances that the ensemble is more certain about, which is also the cases when the majority opinion is likely to be correct.

Because the accuracy improvement from using our proposed MTS mainly occurred for the HA cases, in Table 5.1, we report the accuracy of the majority voted baseline and our proposed methods (HMTS and DMTS) on HA cases, LA cases, and all cases separately. From the results, we observe that our MTS methods significantly improve the performance on HA cases by 10%-50%. It is reasonable because the “high disagreement” instances, compared with “low disagreement”, are more difficult to classify. Hence, for HA cases, applying the majority voting rule leads to low accuracy and the majority answer is unreliable, when MTS is especially relevant because it was originally designed to address the issue of the majority being wrong. As such, our MTS methods can recover the correct minority answer when the majority is wrong, resulting in higher improvement in performance. For LA cases, that is, when the disagreement is low in the ensemble, accuracy achieved by trusting the majority labels is already high, as shown in the Majority (LA) row in Table 5.1. Such LA tasks leave us little room for our proposed methods to improve, as shown in the last three rows in Table 5.1 such that the accuracy is almost unchanged after applying our MTS methods as compared with the accuracy of the majority voting.

Another observation is that Heuristics Machine Truth Serum (HMTS) tends to have more robust and better performances than Discriminative Machine Truth Serum (DMTS) in most datasets, especially in the small-size datasets. These can be explained by the fact DMTS itself is a MLP classifier which needs a larger size of data to get good results. That HMTS can improve the classification accuracy in the small size of dataset is particularly useful in some fields such as healthcare in which collecting data is very time-consuming and expensive. As for the running time, DMTS is faster than HMTS as HMTS needs to compute the peer prediction results of all the 15 classifiers and DMTS only predicts once.

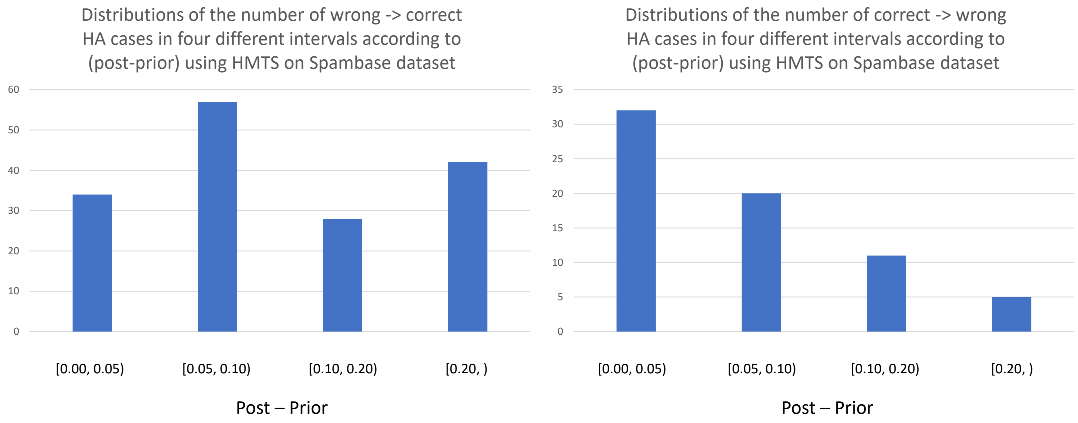


Figure 5.2: Distributions of the number of wrong->correct and correct->wrong cases (two subsets of HA cases) in four different intervals according to the value of (posterior - prior) using HMTS on Spambase binary classification dataset, where a larger value means a bigger difference between prior and posterior probabilities. Our proposed MTS methods can obtain more correct answers when there is a significant difference between prior and posterior.

To further demonstrate the conditions under which MTS Methods are expected to be effective, we compare the distributions of the difference between prior and posterior probabilities in two subsets of HA cases from the Spambase dataset. The first subset consists of the cases where the correct classifications are successfully recovered by applying the MTS methods. The other subset is constituted by the cases where MTS ends up recovering wrong answers (i.e., the majority is correct in the first place, but rejected by MTS). In Figure 5.2, for these two subsets of HA cases, we respectively present the distributions of the number of cases (wrong->correct and correct->wrong) in four different intervals according to the value of (posterior - prior), where a larger value means a bigger difference between prior and posterior probabilities. As Figure 5.2 shows, the MTS methods obtain more correct answers when there is a significant difference between prior and posterior. In other words, we are more likely to recover the correct answer successfully if the difference between prior and posterior is large.

Datasets	Waveform	Statlog	Optical	Pen-Based
# of class	3	6	10	10
Majority (ALL)	85.04% (2126/2500)	86.70% (1734/2000)	97.50% (1752/1797)	95.08% (3326/3498)
HMTS (ALL)	85.60% (+14/2500)	**87.05%** (+7/2000)	**97.66%** (+3/1797)	95.48% (+14/3498)
DMTS (ALL)	**85.64%** (+15/2500)	86.75% (+1/2000)	97.61% (+2/1797)	**95.54%** (+16/3498)
Majority (HA)	42.59% (23/54)	23.08% (15/65)	40.00% (6/15)	57.32% (90/157)
HMTS (HA)	62.96% (23+11/54)	**53.33%** (15+8/65)	53.33% (6+2/15)	**68.15%** (90+17/157)
DMTS (HA)	**68.52%** (23+14/54)	24.62% (15+1/65)	**60.00%** (6+3/15)	66.88% (90+15/157)
Majority (LA)	85.98% (2103/2446)	88.84% (1719/1935)	97.98% (1746/1782)	96.86% (3236/3341)
HMTS (LA)	**86.10%** (2103+3/2446)	88.79% (1719-1/1935)	**98.04%** (1746+1/1782)	96.77% (3236-3/3341)
DMTS (LA)	86.02% (2103+1/2446)	**88.84%** (1719+0/1935)	97.92% (1746-1/1782)	**96.89%** (3236+1/3341)

Table 5.2: Accuracy and the number of increased correct predictions for the three categories of cases, namely, Overall & “High disagreement (HA)” & “Low disagreement (LA)” cases’, using methods of Uniformly-weighted Majority Voting, HMTS, and DMTS on four multi-class classification datasets. We have 15 classifiers and the instance will be considered as having “high disagreement” if the vote number of majority class is less or equals to 6 for the 3-class dataset. The threshold number is 5 for 6-class and 3 for 10-class datasets. For other conditions the instance will be considered as having “low disagreement”. In MTS methods, the numbers of HA and LA instances we consider in four datasets are **54, 65, 15, 157** and **2446, 1935, 1782, 3341** respectively. The numbers of four overall testing datasets are **2500, 2000, 1797, 3498**.

We also tested our extension to multi-class classification problems. Experimental results on four multi-class classification datasets are reported in Table 5.2. We observe that HMTS and DMTS obtained similarly good performance in the accuracy metric because the size of multi-class classification datasets is larger and the MLP of DMTS can perform better than the binary case. In Table 5.2, we also noted that the similar significant improvement on the HA cases and almost unchanged performance on the LA cases after applying our MTS methods for four multi-class classification datasets.

We also observe that, in both binary & multi-class classification tasks, DMTS performs much worse than HMTS for some datasets. We analyze this phenomenon below.

Analysis on why DMTS performs much worse than HMTS in some datasets In some datasets (e.g., German and Statlog), compared with HMTS, DMTS performs much worse. After examining the cases in those datasets, we observe that in the cases where HMTS recovers the correct minority answers, there is an imbalance in the distribution of labels. For example, most corrected cases in the Statlog dataset have the same label. It makes sense because HMTS is a heuristic method and can compute for each data point individually and doesn't have the constraints of balanced distribution on labels. For DMTS, however, we found that the labels of the cases using DMTS are balanced, which suggests that it seems to be subject to a constraint of label balance. This could be because we trained the model on the dataset with a balanced distribution of labels. As a result, it enforces the balanced distribution of the labels when applied in the testing datasets.

Finally, we compare between several popular ensemble algorithms and our proposed approaches. We list the testing accuracy for Adaboost with 15 decision tree base estimators, Random Forest with 15 decision trees, Weighted Majority [50], Stacking with the same setting of 15 classifiers utilized in our two MTS algorithms and Logistic Regression or SVM as meta

Methods	Adaboost	Random Forest	Weighted Majority	Stacking	HMTS	DMTS
Breast Cancer	94.37%	94.37%	94.01%	94.72%	**96.13%**	94.01%
Movie Review	75.10%	77.20%	**81.60%**	70.30%	80.85%	80.60%
Spambase	74.74%	74.65%	74.17%	75.91%	76.87%	**77.35%**
Australian	82.03%	84.06%	84.06%	**85.22%**	83.44%	82.94%
German	72.20%	74.80%	73.80%	**77.20%**	**77.20%**	76.20%
Waveform	81.80%	82.60%	85.36%	84.00%	85.48%	**85.60%**
Statlog	85.85%	86.15%	86.85%	82.70%	**87.10%**	86.75%
Optical	93.99%	94.88%	92.21%	95.83%	97.61%	**97.66%**
Pen-Based	94.97%	95.45%	90.59%	95.43%	**95.57%**	95.51%
# of best	0	0	1	1	**4**	**3**
# of significant wins	0	1	0	0	**3**	**3**

Table 5.3: Comparison between popular ensemble and our proposed approaches. # of best means the number of datasets where the benchmark achieves the best performance. # of significant wins means winning number of comparisons between itself and other methods if they are significantly different (p-value$<$0.05) by doing paired t-test.

classifier, HMTS, and DMTS for all nine datasets in Table 5.3. As shown in the table, HMTS and DMTS outperform Adaboost, Random Forest, Weighted Majority, and Stacking in seven datasets and are very close to the best in two datasets. Compared to other weighted methods, we'd like to note that our aggregation operates on each single task separately - this means that our method will be more robust when the difficulty levels of tasks differ drastically in the dataset. None of the other weighted methods (with fixed and learned weights) has this feature. We also find that our method is robust to a smaller number of classifiers, in contrast to, say Adaboosting. We also conduct paired t-testing where all methods are compared to each other. If two methods are significantly different (p-value$<$0.05) and one method performs better, it means significant win or better. Random Forest is significantly better than Adaboost. HMTS and DMTS are significantly better than Adaboost, Random Forest, and Weighte Majority (almost for Stacking). Paired t-testing results show the effectiveness of our proposed approaches.

5.4.2 Experimental Setup and Results for Semi-supervised Classification

Methods	CIFAR-10	CIFAR-100
UDA	88.70%	75.23%
MixMatch (2-AUG)	90.68%	76.78%
MixMatch (5-AUG)	91.59%	78.20%
HMTS	**92.62%**	**80.75%**
DMTS	91.90%	79.52%

Table 5.4: Classification accuracy (%) in UDA, MixMatch (2-AUG), MixMatch (5-AUG), HMTS, and DMTS settings on the CIFAR-10 and CIFAR-100 testing dataset using MixMatch method. 2-AUG means that two data augmentation samples are constructed for each unsupervised data. HMTS and DMTS are based on 5-AUG setting.

Methods	Yahoo! Answers	AG News
UDA	65.6%	86.8%
MixText (2-AUG)	66.7%	87.6%
MixText (3-AUG)	67.1%	88.3%
HMTS	**67.8%**	**89.5%**
DMTS	67.3%	88.9%

Table 5.5: Classification accuracy (%) in UDA, MixText (2-AUG), MixText (3-AUG), HMTS, and DMTS settings on the Yahoo! Answers and AG News testing dataset using MixText method. 2-AUG means that two data augmentation samples are constructed for each unsupervised data. HMTS and DMTS are based on 3-AUG setting.

We adopt the recent SSL methods UDA [162], MixMatch [9], and MixText [24] as our baselines in SSL. Becuase UDA doesn't use ensemble data augmentation methods to generate the pseudo labels, we consider MixMatch and MixText as the ensemble baselines.

We applied UDA and MixMatch on image classification tasks (CIFAR-10 and CIFAR-100). In both CIFAR-10 and CIFAR-100 datasets, 14,000 data points are utilized as supervised dataset and the remaining as unsupervised dataset. For UDA, it performs worse than other methods because it doesn't use ensemble data augmentation methods to generate the pseudo labels. For MixMatch, we tried different data augmentation settings, where varying number of data augmented samples are constructed for each unsupervised data. As shown in Table 5.4, 2-AUG and 5-AUG settings are conducted. We observe that constructing more data augmented samples can improve the classification accuracy. HMTS and DMTS in Table 5.4 are applied on 5-AUG settings. We change the pseudo labels on the “high disagreement” cases, which are the ones when the ensemble is least certain about. In the image classification tasks, instances are considered as having “high disagreement” if three give the same classification results and the remaining two provide another consistent prediction results. HMTS and DMTS further improve the better performance than 5-AUG ensemble setting.

MixText utilized Mixup in the hidden states so that it is more suitable for text tasks. UDA can also be used in the text tasks. Therefore, we conducted the experiments on two text classification datasets - Yahoo! Answers and AG News using UDA and MixText. 100 labeled data and 5,000 unlabeled data per class in both datasets are used to train the model. For UDA, similar to image classification tasks, it performs worse than other methods because it doesn’t use ensemble data augmentation methods to generate the pseudo labels. For MixText, we also tried different data augmentation settings as in the MixMatch, where varying number of data augmented samples are constructed for each unsupervised data. In the 2-AUG setting, Russian and German machine translation models are utilized to generate data augmented samples for each unsupervised data. We add one more model - French machine translation model in the 3-AUG setting. We change the pseudo labels on the “high disagreement” cases which is defined in the above paragraph. In the text classification tasks, instances are considered as having “high disagreement” if two give the same classification results and the remaining one provide another prediction result. In Table 5.5, we observe the consistent improvement as the one in Table 5.4.

The reason that our MTS methods work in SSL is that better pseudo labels for unsuper-

Methods	HA accuracy (pseudo labels) in CIFAR-10	Improvement over 2-AUG (%)
2-AUG	86.34% (2308/2673)	-
5-AUG	89.45% (2391/2673)	+3.11%
HMTS	92.07% (2461/2673)	+5.73%
DMTS	90.35% (2415/2673)	+4.01%

Table 5.6: Pseudo labels accuracy (%) in high disagreement (HA) cases for 2-AUG, 5-AUG, HMTS, and DMTS settings on CIFAR-10 dataset. 2-AUG means that two data augmentation samples are constructed for each unsupervised data. HMTS and DMTS are based on the 5-AUG setting. The numbers of HA unsupervised cases and overall unsupervised cases are **2,673** and **36,000** respectively.

Methods	HA accuracy (pseudo labels) in CIFAR-100	Improvement over 2-AUG (%)
2-AUG	76.18% (2559/3359)	-
5-AUG	78.38% (2633/3359)	+2.20%
HMTS	81.30% (2731/3359)	+5.12%
DMTS	80.53% (2705/3359)	+4.35%

Table 5.7: Pseudo labels accuracy (%) in high disagreement (HA) cases for 2-AUG, 5-AUG, HMTS, and DMTS settings on CIFAR-100 dataset. 2-AUG means that two data augmentation samples are constructed for each unsupervised data. HMTS and DMTS are based on the 5-AUG setting. The numbers of HA unsupervised cases and overall unsupervised cases are **3,359** and **36,000** respectively.

Methods	HA accuracy (pseudo labels) in Yahoo! Answers	Improvement over 2-AUG (%)
2-AUG	64.1% (8452/13186)	-
3-AUG	66.2% (8729/13186)	+2.1%
HMTS	67.4% (8887/13186)	+3.3%
DMTS	66.9% (8821/13186)	+2.8%

Table 5.8: Pseudo labels accuracy (%) in high disagreement (HA) cases for 2-AUG, 3-AUG, HMTS, and DMTS settings on Yahoo! Answers dataset. 2-AUG means that two data augmentation samples are constructed for each unsupervised data. HMTS and DMTS are based on the 3-AUG setting. The numbers of HA unsupervised cases and overall unsupervised cases are **13,186** and **50,000** respectively.

vised data are obtained. For better analyzing why our MTS methods are effective, we show that the accuracy improvement on the high disagreement (HA) cases' pseudo labels for unsupervised data since we only applied our MTS methods on HA cases. The number of HA cases in the 36,000 unsupervised cases in CIFAR-10 dataset is 2,673. Because we actually have true labels

Methods	HA accuracy (pseudo labels) in AG News	Improvement over 2-AUG (%)
2-AUG	86.4% (3558/4118)	-
3-AUG	88.1% (3628/4118)	+1.7%
HMTS	90.7% (3735/4118)	+4.3%
DMTS	90.2% (3714/4118)	+3.8%

Table 5.9: Pseudo labels accuracy (%) in high disagreement (HA) cases for 2-AUG, 3-AUG, HMTS, and DMTS settings on AG News dataset. 2-AUG means that two data augmentation samples are constructed for each unsupervised data. HMTS and DMTS are based on the 3-AUG setting. The numbers of HA unsupervised cases and overall unsupervised cases are **4,118** and **20,000** respectively.

of unsupervised data in CIFAR-10, we can calculate the accuracy on HA cases' pseudo labels obtained by aggregating the predictions of data augmented cases for unsupervised data with ensemble methods. As shown in Table 5.6, our methods provide more correct pseduo labels and the improvement is significant. The similar improvements are observed on the experiments for other datasets (CIFAR-100, Yahoo! Answers, and AG News) in the SSL setting and the details are shown in Table 5.7, 5.8, and 5.9.

5.5 Related Work

5.5.1 Ensemble Methods

Wisdom of the crowd [144] often performs better than a few elite individuals in the applications such as decision making of public policy [110], answering the questions on general world knowledge [139]. The same idea has been also successfully applied in ML - ensemble methods combine multiple learning algorithms and usually performs better than any single method [42]. Ensemble methods are usually used where aggregating the predictions are needed such as ensemble final predictions in supervised learning and ensemble data augmentations in

SSL. In this paper, we focus on classification problem which is one of the most fundamental problems in ML community [37, 166, 69, 30, 165, 126, 169].

Ensemble Final Predictions for Supervised Classification In this part, we focus on describing the ensemble methods aggregating the final predictions for supervised classification which is the most commonly used scenario of ensemble methods. Ensemble methods consist of a rich family of algorithms. Popular ensemble methods include Boosting (e.g., AdaBoost [48]), Bootstrap aggregating (e.g., Random Forest [65]), and Stacking [11].

Ensemble Data Augmentation for Semi-supervised Classification Another important application of ensemble methods is to generate better pseudo labels for unsupervised data with the help of data augmentation in semi-supervised classification other than improving the performance of final predictions. There are a wide family of SSL algorithms [21, 114, 9, 162, 24]. In this paper, we mainly review the recent pseudo labeling based SSL methods [86, 123, 52, 92, 71, 9]. Pseudo labeling based SSL methods benefit from the unlabeled dataset by providing the high-quanlity explicit pseudo labels after applying data augmentation and ensemble methods. Some recent SSL methods such as UDA [162] conducted the consistency regularization training with implicit pseudo-labels and cannot be considered as our ensemble baseline because they don't use ensemble data augmentation methods to generate the pseudo labels. In this paper, we utilized MixMatch [9] and MixText [24] as the ensemble baseline methods in the SSL setting.

5.5.2 Bayesian Truth Serum

As mentioned in above sections, typical algorithms for aggregating human judgements and classical ensemble methods for combining classifiers' predictions have the same assumption that the majority answer is likely to be correct. Most works in these two settings, except for [118], would fail when the majority answer is instead likely to be wrong. But BTS only works in the setting of aggregating human judgements by collecting subjective judgment data. Inspired by the ideas proposed by [118, 119], we proposed two ML aided algorithms to discover the correct answer when it is minority instead of majority in the setting of classification problem. As our proposed methods are ML algorithms, they can be trained and the predictions will be made automatically instead of collecting subjective judgment data as the case in [118].

Chapter 6

The Rich Get Richer: Disparate Impact of Semi-Supervised Learning

6.1 Introduction

This chapter focuses on the fairness of semi-supervised learning (SSL). While SSL are widely applied in various kinds of real-world applications, fairness issue are drawing more attention. For example, even though the global model performance for the entire population of data is almost always improved by SSL, it is unclear how the improvements fare for different sub-populations which can leads to fairness issue, especially when sub-populations are defined by the demographic groups e.g., race and gender.

In this chapter, we aim to understand the disparate impact of SSL from both theoretical and empirical aspects, and propose to evaluate SSL from a different dimension. Specifically, based on classifications tasks, we study the disparate impact of model accuracies with respect to different sub-populations (such as label classes, feature groups and demographic groups) after applying SSL. Different from traditional group fairness [172, 7, 73] defined over one model, our study focuses on comparing the gap between two models (before and after SSL). To this end, we first theoretically analyze why and how disparate impact are generated. The theoretical results motivate us to further propose a new metric, *benefit ratio*, which evaluates the normalized improvement of model accuracy using SSL methods for different sub-populations. The benefit ratio helps reveal the *"Matthew effect"* of SSL: a high baseline accuracy tends to reach a high benefit ratio that may even be larger than 1 (the rich get richer), and a sufficiently low baseline accuracy may return a negative benefit ratio (the poor get poorer). The above revealed Matthew effect indicates that existing and popular SSL algorithms can be unfair. Aligning with recent literature on fair machine learning [60, 34, 150], we promote that a fair SSL algorithm should benefit different sub-populations equally, i.e., achieving Equalized Benefit Ratio which we will formally define in Definition 1. We then evaluate SSL using benefit ratios and discuss how two possible treatments, i.e., balancing the data and collecting more labeled data, might mitigate the disparate impact. We hope our analyses and discussions could encourage future contributions to promote the fairness of SSL.

6.2 Preliminaries

We summarize the key concepts and notations as follows.

6.2.1 Supervised Classification Tasks

Consider a K-class classification task given a set of N_L labeled training examples denoted by $D_L := \{(x_l, y_l)\}_{l=1}^{N_L}$, x_l is an input feature, $y_l \in \{0, 1, ..., K-1\}$ is a label. The clean data distribution with full supervision is denoted by $\mathcal{D}$. Examples (x_l, y_l) are drawn according to random variables $(X, Y) \sim \mathcal{D}$. The classification task aims to identify a classifier f that maps X to Y accurately. Let $\mathbb{1}\{\cdot\}$ be the indicator function taking value 1 when the specified condition is satisfied and 0 otherwise. Define the *0-1 loss* as $\mathbb{1}(f(X), Y) := \mathbb{1}\{f(X) \neq Y\}$. The optimal f is denoted by the Bayes classifier $f^* = \arg\min_f \ \mathbb{E}_{\mathcal{D}}[\mathbb{1}(f(X), Y)]$. One common choice is training a deep neural network (DNN) by minimizing the empirical risk: $\hat{f} = \arg\min_f \ \frac{1}{N} \sum_{l=1}^{N} {}_L\ell(f(x_l), y_l)$. Notation $\ell(\cdot)$ stands for the cross-entropy (CE) loss $\ell(f(x), y) := -\ln(\boldsymbol{f}_x[y]), y \in \{0, 1, ..., K-1\}$, where $\boldsymbol{f}_x[y]$ denotes the y-th component of $\boldsymbol{f}(x)$. Notations f and $\boldsymbol{f}$ stand for the same model but different outputs. Specifically, vector $\boldsymbol{f}(x)$ denotes the probability of each class that model f predicts given feature x. The predicted label $f(x)$ is the class with maximal probability, i.e., $f(x) := \arg\max_{k \in \{0,1,...,K-1\}} \boldsymbol{f}_x[k]$. We use notation f if we only refer to a model.

6.2.2 Semi-Supervised Classification Tasks

In the semi-supervised learning (SSL) task, there is also an unlabeled (a.k.a. unsupervised) dataset $D_U := \{(x_u, \cdot)\}_{u=1}^{N_U}$ drawn from $\mathcal{D}$, while the labels are missing or unobservable. Let $N := N_L + N_U$. Denote the corresponding unobservable supervised dataset by $D := \{(x_n, y_n)\}_{n=1}^{N}$. Compared with the supervised learning tasks, it is critical to leverage the unsupervised information in semi-supervised learning. To improve the model generalization ability, many recent SSL methods build consistency regularization with unlabeled data to ensure that the model output remains unchanged with randomly augmented inputs [9, 162, 8, 141, 164]. To proceed, we introduce soft/pseudo-labels $\boldsymbol{y}$.

Soft-labels Note the one-hot encoding of y_n can be written as $\boldsymbol{y}_n$, where each element writes as $\boldsymbol{y}_n[k] = \mathbb{1}\{k = y_n\}$. More generally, we can extend the one-hot encoding to soft labels by requiring each element $\boldsymbol{y}[k] \in [0, 1]$ and $\sum_{k\in[K]} \boldsymbol{y}[k] = 1$. The CE loss with soft $\boldsymbol{y}$ writes as $\ell(\boldsymbol{f}(x), \boldsymbol{y}) := -\sum_{k=0}^{K-1} \boldsymbol{y}[k] \ln(\boldsymbol{f}_x[k])$. If we interpret $\boldsymbol{y}[k] = \mathbb{P}(Y = k)$ as a probability [182] and denote by $\mathcal{D}_{\boldsymbol{y}}$ the corresponding label distribution, the above CE loss with soft labels can be interpreted as the expected loss with respect to a stochastic label $\widetilde{Y}$, i.e.,

$$\ell(\boldsymbol{f}(x), \boldsymbol{y}) := \sum_{k\in[K]} \mathbb{P}(\widetilde{Y} = k)\ell(f(x), k) = \mathbb{E}_{\widetilde{Y}\sim\mathcal{D}_{\boldsymbol{y}}} \left[\ell(f(x), \widetilde{Y})\right]. \tag{6.1}$$

Pseudo-labels In consistency regularization, by using model predictions, the unlabeled data will be assigned pseudo-labels either explicitly [9] or implicitly [162], where the pseudo-labels can be modeled as soft-labels. In the following, we review both the explicit and the implicit approaches and unify them in our analytical framework.

Consistency regularization with explicit pseudo-labels For each unlabeled feature x_n, pseudo-labels can be explicitly generated based on model predictions [9]. The pseudo-label is later used to evaluate the model predictions. To avoid trivial solutions where model predictions and pseudo-labels are always identical, independent data augmentations of feature x_n are often generated for M rounds. The augmented feature is denoted by $x'_{n,m} := \text{Augment}(x_n), m \in \{m = 1, 2, ..., M\}$. Then the pseudo-label $\boldsymbol{y}_n$ in epoch-t can be determined based on M model predictions as $\boldsymbol{y}_n^{(t)} = \text{Sharpen}\left(\frac{1}{M}\sum_{m=1}^{M} \bar{\boldsymbol{f}}^{(t)}(x'_{n,m})\right)$, where model $\bar{f}^{(t)}$ is a copy of the DNN at the beginning of epoch-t but without gradients. The function Sharpen$(\cdot)$ reduces the entropy of a pseudo-label, e.g., setting to one-hot encoding $\boldsymbol{e}_j, j = \arg\max_{k\in\{0,1,\dots,K-1\}} \boldsymbol{y}_n^{(t)}$ [141]. In epoch-t, with some consistency regularization loss $\ell_{\text{CR}}(\cdot)$, the empirical risk using pseudo-labels is:

$$L_1(f, D_L, D_U) = \frac{1}{N_L}\sum_{n=1}^{N_L} \ell(f(x_n), y_n) + \frac{1}{N_U}\sum_{n=N_L+1}^{N_L+N_U} \ell_{\text{CR}}(\boldsymbol{f}(x_n), \boldsymbol{y}_n^{(t)}).$$

Consistency regularization with implicit pseudo-labels Consistency regularization can also be applied without specifying particular pseudo-labels, where a divergence metric between predictions on the original feature and the augmented feature is minimized to make predictions consistent. For example, the KL-divergence could be applied and the data augmentation could be domain-specific [162] or adversarial [109]. In epoch-t, the total loss is:

$$L_2(f, D_L, D_U) = \frac{1}{N_L}\sum_{n=1}^{N_L} \ell(f(x_n), y_n) + \lambda \cdot \frac{1}{N_U}\sum_{n=N_L+1}^{N_L+N_U} \ell_{\text{CR}}(\boldsymbol{f}(x_n), \bar{\boldsymbol{f}}^{(t)}(x'_n)),$$

where λ balances the supervised loss and the unsupervised loss, $x'_n := \text{Augment}(x_n)$ stands for one-round data augmentation ($m = 1$ following the previous notation $x'_{n,m}$). Without loss of generality, we use $\lambda = 1$ in our analytical framework.

Consistency regularization loss $\ell_{\text{CR}}(\cdot)$ In the above two lines of works, there are different choices of $\ell_{\text{CR}}(\cdot)$, such as mean-squared error $\ell_{\text{CR}}(\boldsymbol{f}(x), \boldsymbol{y}) := \|\boldsymbol{f}(x) - \boldsymbol{y}\|_2^2/K$ [9, 85] or CE loss [109, 162] defined in Eq. (6.1). For a clean analytical framework, we consider the case when both supervised loss and unsupervised loss are the same, i.e., $\ell_{\text{CR}}(\boldsymbol{f}(x), \boldsymbol{y}) = \ell(\boldsymbol{f}(x), \boldsymbol{y})$. Note L_2 implies the entropy minimization [54] when both loss functions are CE and there is no augmentation, i.e., $x'_n = x_n$.

6.2.3 Analytical Framework

We propose an analytical framework to unify both the explicit and the implicit approaches. With Eqn. (6.1), the unsupervised loss in the above two methods can be respectively written as:

$$\frac{1}{N_U}\sum_{n=N_L+1}^{N_L+N_U}\mathbb{E}_{\widetilde{Y}\sim\mathcal{D}_{\boldsymbol{y}_n^{(t)}}}\left[\ell(f(x_n),\widetilde{Y})\right] \quad\text{and}\quad \frac{1}{N_U}\sum_{n=N_L+1}^{N_L+N_U}\mathbb{E}_{\widetilde{Y}\sim\mathcal{D}_{\bar{\boldsymbol{f}}^{(t)}(x'_n)}}\left[\ell(f(x_n),\widetilde{Y})\right].$$

Both unsupervised loss terms inform us that, for each feature x_n, we compute the loss with respect to the reference label $\widetilde{Y}$, which is a random variable following distribution $\mathcal{D}_{\boldsymbol{y}_n^{(t)}}$ or $\mathcal{D}_{\bar{\boldsymbol{f}}^{(t)}(x'_n)}$. Compared with the original clean label Y, the unsupervised reference label $\widetilde{Y}$ contains label noise, and the noise transition [185] depends on feature x_n. Specifically, we have

$$\mathbb{P}(\widetilde{Y}=k|X=x_n)=\boldsymbol{y}_n^{(t)}[k] \text{ (Explicit)} \quad\text{or}\quad \mathbb{P}(\widetilde{Y}=k|X=x_n)=\bar{\boldsymbol{f}}_{x'_n}^{(t)}[k] \text{ (Implicit)}.$$

Then with model $\bar{f}^{(t)}$, we can define a new dataset with instance-dependent noisy reference labels: $\widetilde{D}=\{(x_n,\tilde{\boldsymbol{y}}_n)\}_{n\in[N]}$, where $\tilde{\boldsymbol{y}}_n=\boldsymbol{y}_n,\forall n\in[N_L]$, and $\tilde{\boldsymbol{y}}_n=\boldsymbol{y}_n^{(t)}$ or $\bar{\boldsymbol{f}}_{x'_n}^{(t)}$, $\forall n\in\{N_L+1,\cdots,N\}$. The unified loss is:

$$L(f,\widetilde{D})=\frac{1}{N}\sum_{n=1}^{N}\ell(\boldsymbol{f}(x_n),\tilde{\boldsymbol{y}}_n)=\frac{1}{N}\sum_{n=1}^{N}\mathbb{E}_{\widetilde{Y}\sim\mathcal{D}_{\tilde{\boldsymbol{y}}_n}}\left[\ell(f(x_n),\widetilde{Y})\right]. \tag{6.2}$$

Therefore, with the pseudo-label as a bridge, we can model the popular SSL solution with consistency regularization as a problem of learning with noisy labels [112]. But different from the traditional class-dependent assumption [91, 158, 186], the instance-dependent pseudo-labels are more challenging [95].

6.3 Disparate Impacts of SSL

To understand the disparate impact of SSL, we study how supervised learning with D_L affects the performance of SSL with both D_L and D_U. In this section, the analyses are for one and an arbitrary sub-population, thus we did not explicitly distinguish sub-populations in notation.

Intuition The rich sub-population with a higher baseline accuracy (from supervised learning with D_L) will have higher-quality pseudo-labels for consistency regularization, which helps to further improve the performance. In contrast, the poorer sub-population with a lower baseline accuracy (again from supervised learning with D_L) can only have lower-quality pseudo-labels to regularize the consistency of unsupervised features. With a wrong regularization direction, the unsupervised feature will have its augmented copies reach consensus on a wrong label class, which leads to a performance drop. Therefore, when the baseline accuracy is getting worse, there will be more and more unsupervised features that are wrongly regularized, resulting in disparate impacts of model accuracies.

In the following, we first analyze the generalization error for supervised learning with labeled data, then extend the analyses to semi-supervised learning with the help of pseudo-labels.

Without loss of generality, we consider minimizing 0-1 loss $\mathbb{1}(f(X), Y)$ with infinite search space. Our analyses can be generalized to bounded loss $\ell(\cdot)$ and finite function space $\mathcal{F}$ following the generalization bounds that can be introduced using Rademacher complexity [6].

6.3.1 Learning with Clean Data

Denote the expected error rate of classifier f on distribution $\mathcal{D}$ by $R_{\mathcal{D}}(f) := \mathbb{E}_{\mathcal{D}}[\mathbb{1}(f(X), Y)]$. Let $\hat{f}_D$ denote the classifier trained by minimizing 0-1 loss with clean dataset D, i.e., $\hat{f}_D := \arg\min_f \ \hat{R}_D(f)$, where $\hat{R}_D(f) := \frac{1}{N}\sum_{n=1}^{N} \mathbb{1}(f(x_n), y_n)$. Denote by $Y^*|X := \arg\max_{k\in\{0,1,\dots,K-1\}} \mathbb{P}(Y|X)$ the Bayes optimal label on clean distribution $\mathcal{D}$. Theorem 3 shows the generalization bound in the clean case. Replacing D and N with D_L and N_L we have:

Theorem 3 (Supervised error)**.** *With probability at least $1-\delta$, the generalization error of supervised learning on clean dataset D_L is upper-bounded by $R_{\mathcal{D}}(\hat{f}_{D_L}) \leq \sqrt{\frac{2\log(2/\delta)}{N_L}} + \mathbb{P}(Y^* \neq Y)$.*

6.3.2 Learning with Semi-supervised Data

We further derive generalization bounds for learning with semi-supervised data. Following our analytical framework in Section 6.2.3, in each epoch-t, we can transform the semi-supervised data to supervised data with noisy supervisions by assigning pseudo-labels, where the semi-supervised dataset $D_L \cup D_U$ is converted to the dataset $\widetilde{D}$ given the model learned from the previous epoch.

Two-iteration scenario In our theoretical analyses, to find a clean-structured performance bound for learning with semi-supervised data and highlight the impact of the model learned from supervised data, we consider a particular *two-iteration scenario* where the model is first trained to convergence with the small labeled dataset D_L and get $\hat{f}_{D_L}$, then trained on the pseudo noisy dataset $\widetilde{D}$ labeled by $\hat{f}_{D_L}$. Noting assigning pseudo labels iteratively may improve the performance as suggested by most SSL algorithms [9, 162], our considered two-iteration scenario can be seen as a worst case for an SSL algorithm.

Independence of samples in $\widetilde{D}$ Recall x'_n denotes the augmented copy of x_n. The N instances in $\widetilde{D}$ may not be independent since pseudo-label $\tilde{\boldsymbol{y}}_n$ comes from $\hat{f}_{D_L}(x'_n)$, which depends on x_n. Namely, the number of independent instances N' should be in the range of $[N_L, N]$. Intuitively, with appropriate noise injection or data augmentation [162, 109] to x_n such that x'_n could be treated as independent of x_n, the number of independent samples in $\widetilde{D}$ could be improved to N. We consider the ideal case where all N instances are *i.i.d.* in the following analyses.

By minimizing the unified loss defined in Eq. (6.2), we can get classifier $\hat{f}_{\widetilde{D}} := \arg\min_f \hat{R}_{\widetilde{D}}(f)$, where $\hat{R}_{\widetilde{D}}(f) := \frac{1}{N}\sum_{n=1}^{N}\left(\sum_{k=0}^{K-1} \tilde{\boldsymbol{y}}[k] \cdot \mathbb{1}(f(x_n), k)\right)$. The expected error given classifier f is denoted by $R_{\widetilde{\mathcal{D}}}(f) := \mathbb{E}_{\widetilde{\mathcal{D}}}[\mathbb{1}(f(X), \widetilde{Y})]$, where the probability density function of distribution $\widetilde{\mathcal{D}}$ can be defined as $\mathbb{P}_{(X,\widetilde{Y})\sim\widetilde{\mathcal{D}}}(X = x_n, \widetilde{Y} = k) = \mathbb{P}_{(X,Y)\sim\mathcal{D}}(X = x_n) \cdot \tilde{\boldsymbol{y}}_n[k]$.

Decomposition With the above definitions, the generalization error (on the clean distribution) of classifier $\hat{f}_{\widetilde{D}}$ could be decomposed as $R_{\mathcal{D}}(\hat{f}_{\widetilde{D}}) = \underbrace{(R_{\mathcal{D}}(\hat{f}_{\widetilde{D}}) - R_{\widetilde{\mathcal{D}}}(\hat{f}_{\widetilde{D}}))}_{\text{Term-1}} + \underbrace{R_{\widetilde{\mathcal{D}}}(\hat{f}_{\widetilde{D}})}_{\text{Term-2}}$, where **Term-1** transforms the evaluation of $\hat{f}_{\widetilde{D}}$ from clean distribution $\mathcal{D}$ to the pseudo noisy distribution $\widetilde{\mathcal{D}}$. **Term-2** is similar to the generalization error in Theorem 3 but the model is trained and evaluated on noisy distribution $\widetilde{\mathcal{D}}$.

We first provide the related definitions about the upper and lower bounds for both terms. Let $\eta(X) := \frac{1}{2}\sum_{k=0}^{K-1} |\mathbb{P}(\widetilde{Y} = k|X) - \mathbb{P}(Y = k|X)|$, $e(X) := \mathbb{P}(Y \neq \widetilde{Y}|X)$ be the feature-dependent error rate, $\widetilde{A}_f(X) := \mathbb{P}(f(X) = \widetilde{Y}|X)$ be the accuracy of prediction $f(X)$ on noisy dataset $\widetilde{D}$. Denote their expectations (over X) by $\bar{\eta} := \mathbb{E}_X[\eta(X)]$, $\bar{e} := \mathbb{E}_X[e(X)]$, $\widetilde{A}_f = \mathbb{E}_X[\widetilde{A}_f(X)]$. To highlight that $\bar{\eta}$ and $\bar{e}$ depends on the noisy dataset $\widetilde{D}$ labeled by $\hat{f}_{D_L}$, we denote them as $\bar{\eta}(\hat{f}_{D_L})$ and $\bar{e}(\hat{f}_{D_L})$. Denote by $\widetilde{Y}^*|X := \arg\max_{i\in[K]} \mathbb{P}(\widetilde{Y} = i|X)$ the Bayes optimal label on noisy distribution $\widetilde{\mathcal{D}}$. Following the proof for Theorem 3, we have:

Theorem 4 (Semi-supervised learning error)**.** *Suppose the model trained with only D_L has generalization error $\bar{\eta}'(\hat{f}_{D_L})$. With probability at least $1-\delta$, the generalization error of semi-supervised learning on datasets $D_L \cup D_U$ is upper-bounded by*

$$R_{\mathcal{D}}(\hat{f}_{D_L\cup D_U}) \leq \underbrace{\bar{\eta}(\hat{f}_{D_L})}_{\textit{Disparity due to baseline}} + \underbrace{\mathbb{P}(\widetilde{Y} \neq \widetilde{Y}^*)}_{\textit{Sharpness of pseudo labels}} + \underbrace{\sqrt{\frac{2\log(2/\delta)}{N}}}_{\textit{Data dependency}},$$

where $\bar{\eta}(\hat{f}_{D_L}) := \bar{\eta}'(\hat{f}_{D_L}) \cdot N_U/N$ is the expected label error in the pseudo noisy dataset $\widetilde{D}$.

Takeaways Theorem 4 explains how disparate impacts in SSL are generated.

- Supervised error $\bar{\eta}'(\hat{f}_{D_L})$: the major source of disparity. The sub-population that generalizes well before SSL tends to have a lower SSL error. Namely, the rich get richer.
- Sharpness of noisy labels $\mathbb{P}(\widetilde{Y} \neq \widetilde{Y}^*)$: a minor source of disparity depends on how one processes pseudo-labels. This term is negligible if we sharpen the pseudo-label.
- Sample complexity $\sqrt{2\log(2/\delta)/N}$: disparity depends on the number of instances N and their independence. Recall we assume ideal data augmentations to get N in this term. There

will be much less than N *i.i.d.* instances with poor data augmentations. It would be a major source of disparity if data augmentations are poor.

6.4 Benefit Ratio: An Evaluation Metric

In our preliminary experiments, they demonstrates that SSL can lead to disparate impacts of different sub-populations' accuracies, but it is still not clear how much that SSL benefits a certain sub-population. To quantify the disparate impacts of SSL, we propose a new metric called *benefit ratio*.

Benefit Ratio The benefit ratio $\mathsf{BR}(\mathcal{P})$ captures the normalized accuracy improvement of sub-population $\mathcal{P}$ after SSL, which depends on three classifiers, i.e., $\hat{f}_{D_L}$: (baseline) supervised learning only with a small labeled data D_L, $\hat{f}_D$: (ideal) supervised learning if the whole dataset D has ground-truth labels, and $\hat{f}_{D_L \cup D_U}$: SSL with both labeled dataset D_L and unlabeled datasetD_U. The test/validation accuracy of the above classifiers are $a_{\text{baseline}}(\mathcal{P})$, $a_{\text{ideal}}(\mathcal{P})$, and $a_{\text{semi}}(\mathcal{P})$, respectively. As a posterior evaluation of SSL algorithms, the benefit ratio $\mathsf{BR}(\mathcal{P})$ is defined as:

$$\mathsf{BR}(\mathcal{P}) = \frac{a_{\text{semi}}(\mathcal{P}) - a_{\text{baseline}}(\mathcal{P})}{a_{\text{ideal}}(\mathcal{P}) - a_{\text{baseline}}(\mathcal{P})}. \tag{6.3}$$

Let $\mathcal{P}^\diamond := \{\mathcal{P}_1, \mathcal{P}_2, \cdots\}$ be the set of all the concerned sub-populations. We formally define the Equalized Benefit Ratio as follows.

Definition 1 (Equalized Benefit Ratio)**.** *We call an algorithm achieving equalized benefit ratio if all the concerned sub-populations have the same benefit ratio:* $\mathsf{BR}(\mathcal{P}) = \mathsf{BR}(\mathcal{P}'), \forall \mathcal{P}, \mathcal{P}' \in \mathcal{P}^\diamond$.

Intuitively, a larger benefit ratio indicates more benefits from SSL. We have $\mathsf{BR}(\mathcal{P}) = 1$

when SSL performs as well as the corresponding fully-supervised learning. A negative benefit ratio indicates the poor population is hurt by SSL such that $a_{\text{semi}}(\mathcal{P}) < a_{\text{baseline}}(\mathcal{P})$. It has the potential of providing guidance and intuitions for designing fair SSL algorithms on standard datasets with full ground-truth labels. Whether a fair SSL algorithm on one dataset is still fair on another dataset would be an interesting future work. In real scenarios without full supervisions, we may use some extra knowledge to estimate the highest achievable accuracy of each sub-population and set it as a proxy of the ideal accuracy $a_{\text{ideal}}(\mathcal{P})$.

Theoretical Explanation Recall we have error upper bounds for both supervised learning (Theorem 3) and semi-supervised learning (Theorem 4). Both bounds have similar tightness thus we can compare them and get a proxy of benefit ratio as

$$\widehat{\mathsf{BR}}(\mathcal{P}) := \frac{\sup\left(R_{\mathcal{D}}(\hat{f}_{D_L \cup D_U|\mathcal{P}})\right) - \sup\left(R_{\mathcal{D}}(\hat{f}_{D_L|\mathcal{P}})\right)}{\sup(R_{\mathcal{D}}(\hat{f}_{D|\mathcal{P}})) - \sup\left(R_{\mathcal{D}}(\hat{f}_{D_L|\mathcal{P}})\right)},$$

where $\sup(\cdot)$ denotes the upper bound derived in Theorem 3 and Theorem 4, $\mathcal{P}$ is a sub-population, and $D|\mathcal{P}$ denotes the set of *i.i.d.* instances in D that affect model generalization on $\mathcal{P}$. By assuming $\mathbb{P}(Y \neq Y^*) = \mathbb{P}(\widetilde{Y} \neq \widetilde{Y}^*)$ (both distributions have the same sharpness), we have:

Corollary 1. *The benefit ratio proxy for $\mathcal{P}$ is $\widehat{\mathsf{BR}}(\mathcal{P}) = 1 - \frac{\bar{\eta}(\hat{f}_{D_L|\mathcal{P}})}{\Delta(N_{\mathcal{P}}, N_{\mathcal{P}_L})}$, where $\Delta(N_{\mathcal{P}}, N_{\mathcal{P}_L}) = \sqrt{\frac{2\log(2/\delta)}{N_{\mathcal{P}_L}}} - \sqrt{\frac{2\log(2/\delta)}{N_{\mathcal{P}}}}$, $N_{\mathcal{P}}$ and $N_{\mathcal{P}_L}$ are the effective numbers of instances in $D|\mathcal{P}$ and $D_L|\mathcal{P}$.*

Corollary 1 shows the benefit ratio is negatively related to the error rate of baseline models and positively related to the number of *i.i.d.* instances after in SSL, which is consistent with our takeaways from Section 6.3. Note $N_{\mathcal{P}}$ and $N_{\mathcal{P}_L}$ may be larger than the corresponding sub-populations sizes if $\mathcal{P}$ shares information with other sub-population $\mathcal{P}'$ during training, e.g., a better classification of $\mathcal{P}'$ helps classify $\mathcal{P}$. It also informs us that SSL may have a negative effect on sub-population $\mathcal{P}$ if $\frac{\eta}{\Delta(N_{\mathcal{P}}, N_{\mathcal{P}_L})} > 1$, i.e., the benefit from getting more effective *i.i.d.* instances is less than the harm from wrong pseudo-labels. This negative effect indicates "the poor get poorer".

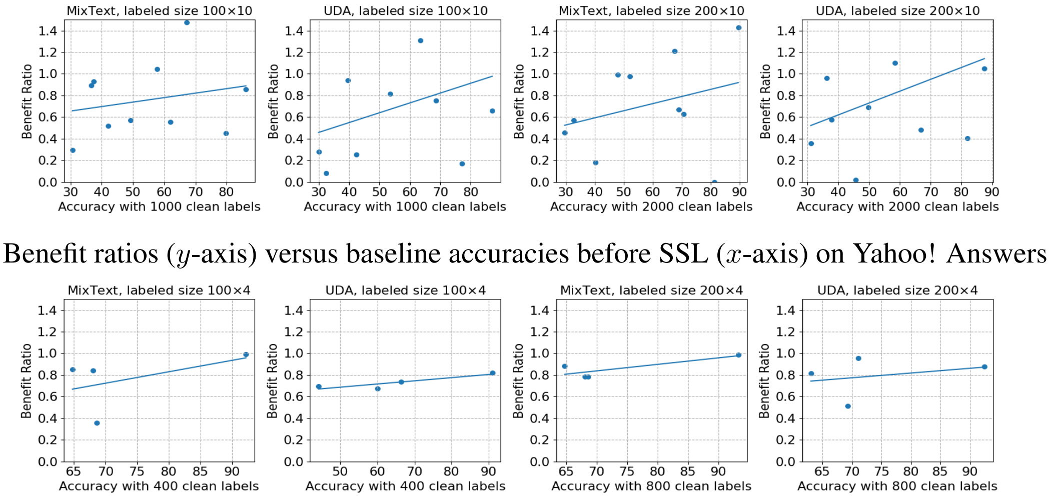
6.5 Experiments

In this section, we first show the existence of disparate impact on several representative SSL methods and then discuss the possible treatment methods to mitigate this disparate impact.

Settings Two representative SSL methods, i.e., UDA [162], and MixText [24], are tested on several text classification tasks. In our text classification tasks, we employ three datasets: Yahoo! Answers [20], AG News [174] and Jigsaw Toxicity [74]. Jigsaw Toxicity dataset contains both classification labels (one text comment is toxic or non-toxic) and a variety of sensitive attributes (e.g., race and gender information) of the identities that are mentioned in the text comment, which are fairness concerns in real-world applications.

6.5.1 Disparate Impact Exists in Popular SSL Methods

We show that, even though the size of each sub-population is equal, disparate impacts exist in the model accuracy of different sub-populations, i.e., 1) *explicit sub-populations* such as classification labels in Figure 6.1, and 2) *implicit sub-populations* such as demographic



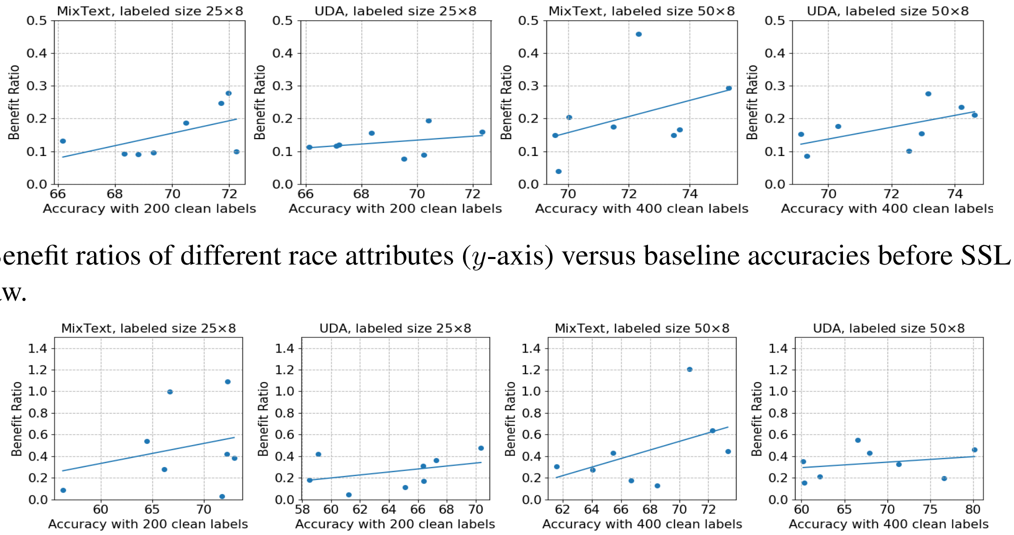
(a) Benefit ratios (y-axis) versus baseline accuracies before SSL (x-axis) on Yahoo! Answers

(b) Benefit ratios of different class labels (y-axis) versus baseline accuracies before SSL (x-axis) on AG News

Figure 6.1: Benefit ratios across explicit sub-populations. Dot: Result of each label class. Line: Best linear fit of dots.

groups including race & gender in Figure 6.2a & Figure 6.2b. All the experiments in this subsection adopt both a balanced labeled dataset and a balanced unlabeled dataset. Note we sample a balanced subset from the raw (unbalanced) Jigsaw dataset. Other datasets are originally balanced.

Disparate impact across explicit sub-populations In this part, we show the disparate impact on model accuracy across different classification labels on Yahoo! Answers, and AG News datasets. In Figure 6.1a, and 6.1b, we show the benefit ratios (y-axis) versus baseline accuracies before SSL (x-axis). From left to right, we show results with different sizes of labeled data: 100 per class to 200 on Yahoo! Answers. Figure 6.1a, and 6.1b utilized two SSL methods (Yahoo! Answers and AG News: MixText & UDA).We statistically observe that the class labels with higher baseline accuracies have higher benefit ratios on Yahoo! Answers datasets. It means



(a) Benefit ratios of different race attributes (y-axis) versus baseline accuracies before SSL on Jigsaw.

(b) Benefit ratios of different gender attributes (y-axis) versus baseline accuracies before SSL (x-axis) on Jigsaw.

Figure 6.2: Benefit ratios across implicit sub-populations.

that "richer" classes benefit more from applying SSL methods than the "poorer" ones. We also observe that for some models with low baseline accuracy (left side of the x-axis), applying SSL results in rather low benefit ratios that are close to 0.

Disparate impact across implicit sub-populations We demonstrate the disparate impacts on model accuracy across different sub-populations on Jigsaw (race and gender) datasets. In Figure 6.2, we can statistically observe the disparate impact across different sub-populations on Jigsaw (race and gender) datasets for two baseline SSL methods. We again observe very similar disparate improvements as presented in Figure 6.2a, and 6.2b.Note the disparate impact on the demographic groups in Jigsaw raises fairness concerns in real-world applications.

6.5.2 Mitigating Disparate Impact

Our analyses in Section 6.3 and Section 6.4 indicate the disparate impacts may be mitigated by balancing the supervised error $\bar{\eta}(\hat{f}_{D_L|\mathcal{P}})$ and the number of effective *i.i.d.* instances for different populations. We perform preliminary tests to check the effectiveness of the above findings in this subsection. We hope our analyses and experiments could inspire future contributions to disparity mitigation in SSL.

Balancing and Collecting more labeled data We firstly sample 400 *i.i.d.* instances from the raw (unbalanced) Jigsaw dataset and get the setting of Unbalanced (400). To balance the supervised error and effective instances for different sub-populations, an intuitive method is to balance the labeled data by reweighting, e.g., if the size of two sub-populations is 1:2, we simply sample instances from two sub-populations with a probability ratio of 2:1 to ensure all sub-population have the same weights in each epoch during training. Balance labeled (400) denotes only 400 labeled instances are reweighted. Balance both (400) means both labeled and unlabeled instances are balanced. Table 6.1 shows a detailed comparison on the benefit ratios between different race identities of the Jigsaw dataset and their standard deviations, where the first three rows denote the above three settings. We can observe that both the standard deviation and the number of negative benefit ratios become lower with more balanced settings, which demonstrates the effectiveness of the balancing treatment strategy, although there still exists a sub-population that has a negative benefit ratio, indicating an unfair learning result. To further improve fairness, as suggested in our analyses, we “collect" (rather add) another 400 *i.i.d.* labeled instances (800 in total) from the Jigsaw dataset, and show the result after balancing both labeled

[Jigsaw] Settings	t_asian	t_black	t_latin	t_white	nt_asian	nt_black	nt_latin	nt_white	*SD*
Unbalanced (400)	24.71	21.76	28.21	**-9.57**	27.27	**-8.33**	**-13.82**	29.47	19.28
Balance labeled (400)	28.18	20.00	25.23	33.93	14.33	**-11.29**	**-10.37**	3.70	17.29
Balance both (400)	28.57	0.00	11.11	40.63	15.38	14.29	6.90	**-7.41**	15.29
Balance both (800)	13.04	25.00	7.69	24.00	3.85	10.71	16.67	20.00	7.64

Table 6.1: Comparison on the benefit ratio (%) of all race identities & standard deviation (%) between different settings. t_asian (nt_asian) is the benefit ratio of asian identity with "toxic" labels ("non-toxic" labels). *SD* denotes standard deviation. Negative benefit ratios are highlighted in red colors.

and unlabeled data in the last row of Table 6.1 (Balance both (800)). Both the standard deviation and the number of negative benefit ratios can be further reduced with more labeled data. More experimental results are on Jigsaw (gender) datasets shown in Table 6.2 (balancing) and Table 6.3 (more data).

Datasets	*Mean*	*Standard Deviation*
Jigsaw (race) (1.4:4.5:1:7.5) Accuracy	66.71 → 67.25 → **67.88**	25.64 → 21.33 → 17.84
Jigsaw (gender) (13.7:12.6:4.8:1) Accuracy	66.03 → 66.74 → 67.17	12.90 → 9.75 → **8.68**
Jigsaw (race) (1.4:4.5:1:7.5) Benefit Ratio	12.46 → 13.09 → **13.67**	19.28 → 17.29 → **15.29**
Jigsaw (gender) (13.7:12.6:4.8:1) Benefit Ratio	11.90 → 12.18 → 12.54	88.52 → 50.67 → 34.24

Table 6.2: Balance samples with reweighting. Jigsaw is naturally unbalanced. For race, we consider asian, black, latin, and white identities. For gender, we consider male, female, male female, and transgender identities. The rounded ratios on the number of samples between different identities in race and gender are listed in the table.

Datasets	*Mean*	*Standard Deviation*
Jigsaw (race) Accuracy	64.88 → 67.88 → 72.25 → 73.50	29.55 → 17.84 → 9.93 → 6.86
Jigsaw (gender) Accuracy	64.50 → 67.17 → 73.50 → **74.83**	20.15 → 8.68 → 6.41 → **5.99**
Jigsaw (race) Benefit Ratio	12.09 → 13.67 → 14.52 → 15.12	18.29 → 15.29 → 11.69 → **7.64**
Jigsaw (gender) Benefit Ratio	7.80 → 12.54 → 22.46 → **25.81**	41.29 → 34.24 → 27.30 → 13.08

Table 6.3: Collect more data. $a \to b \to c \to d$: stands for the change of mean or standard deviation with different sizes of labeled dataset. For Jigsaw (both race and gender), the sizes are $25 \times 8, 50 \times 8, 100 \times 8, 150 \times 8$.

6.6 Related Works

Semi-supervised learning SSL is popular in various communities [37, 69, 30, 165, 126, 55, 28, 156, 100, 70, 5, 157, 90]. We briefly review recent advances in SSL. See comprehensive overviews by [21, 181] for traditional methods. Recent works focus on assigning pseudo-labels generated by the supervised model to unlabeled dataset [86, 71, 9, 8], where the pseudo-labels are often confident or with low-entropy [141, 178, 106]. There are also many works on minimizing entropy of predictions on unsupervised data [54] or regularizing the model consistency on the same feature with different data augmentations [146, 109, 127, 173, 108, 162]. In addition to network inputs, augmentations can also be applied on hidden layers [24]. Besides, some works [117, 39, 56, 27, 168] first conduct pre-training on the unlabeled dataset then fine-tune on the labeled dataset, or use ladder networks to combine unsupervised learning with supervised learning [123].

Disparate impact Even models developed with the best intentions may introduce discriminatory biases [120]. Researchers in various fields have found the unfairness issues, e.g., vision-and-language representations [153], model compression [4], differential privacy [67, 68], recommendation system [51], information retrieval [49], image search [152], machine translation [75], message-passing [72], and learning with noisy labels [94, 183, 98]. There are also some treatments considering fairness without demographics [84, 40, 61], minimax Pareto fairness [105], multiaccuracy boosting [76], and fair classification with label noise [151]. Most of these works focus on supervised learning. To our best knowledge, the unfairness of SSL has not been sufficiently explored.

Chapter 7

A New Weakly Supervised Learning Dataset — Research Replication Prediction

7.1 Introduction

This chapter focuses on proposed a new weakly supervised learning dataset to contribute the community.

Non-reproducible scientific results will negatively impact the development of science and lead to the replication crisis. In recent years, several systematic large-scale direct replication projects in contemporary published social science studies have been conducted based on the concerns of research credibility in contemporary published social science studies [14, 15, 44, 79, 80, 33]. However, direct replication is expensive and time-consuming. A much more efficient alternative that uses ML methods emerged for predicting research replication [43, 167, 3, 101]. Nonetheless, ML-based results for RRP failed to improve with new machine learning algorithms.

The primary cause is due to the lack of publicly available and standardized datasets for researchers to develop and test the more advanced ML methods. Our goal is to provide this community with one to facilitate the development of automated solutions for this replication prediction task.

Our efforts include two types of data collected with different costs. The first set of data is collected via existing direct replication efforts. This data is collected by professional individual volunteers or volunteer teams throughout direct replication and therefore contains the strong supervisions which has high quality. As we discussed earlier, due to the nature of a direct replication being expensive and time-consuming [47], we only obtained a small size of directly replicated dataset. A large and diverse dataset is required in order to apply modern deep learning approaches. To complement the above data, we crowdsource to obtain a large size of annotated dataset from an ordinary population of participants. Despite being a more cost-efficient way for collecting data, the crowdsourced labels are often inaccurate or noisy.

Our dataset covers nicely research results published in psychology, social science, and economics journals. Altogether, we have assembled 3081 articles. We benchmark the performance of several populations weakly supervised learning approaches for predicting research replicability. We report several commonly used metrics (accuracy, precision, recall, and F1) to evaluate the considered methods. We also build a RRP website where users can upload their own research paper (PDF format) and obtain the replication prediction probabilities as well as highlighted sentences obtained by the variational contextual consistency sentence masking (VCCSM) method described in the Chapter 4.

7.2 Our Proposed Research Replication Prediction (RRP) dataset

RRP dataset includes two types of data (directly replicated and crowdsourced datasets) with different costs. The first type of data includes 399 directly replicated data with higher costs and strong supervision. The second type of data has 2,682 crowdsourced samples where the label collecting methods is more economical than direct replication but the labels are noisy.

7.2.1 Directly replicated dataset

7.2.1.1 Dataset collection

In total, we obtained 399 directly replicated labeled data which are collected by professional individuals or team throughout direct replication and therefore contain the strong supervisions. There are different standard and definitions on decide whether one research paper is replicable or not. In this paper, we consider a research paper as replicable if the results of an independent replication can produce the same statistically significant effect in the direction from the original paper. To collect more papers, we define it produces a statistically significant effect when p-value ≤ 0.05 [3].

Based on the treating standard, we collect the directly replicated labeled dataset from eight research replication projects which are the Registered Replication Report (RRR) [135], Many Labs 1 [78], Many Labs 2 [80], Many Labs 3 [44], Social Sciences Replication Project (SSRP) [15], PsychFileDrawer [115], Experimental Economics Replication Project [14], and Reproducibility Project: Psychology (RPP) [32].

	Direct replication projects	Domains	# of labeled samples
1	Registered Replication Report (RRR) [135]	Psychology	19
2	Many Labs 1 [78]	Psychology	15
3	Many Labs 2 [80]	Psychology	28
4	Many Labs 3 [44]	Psychology	20
5	Social Sciences Replication Project (SSRP) [15]	Social sciences	21
6	PsychFileDrawer [115]	Psychology	72
7	Experimental Economics Replication Project [14]	Economics	206
8	Reproducibility Project: Psychology (RPP) [32]	Psychology	18

Table 7.1: Distribution of research papers' number and domains by eight direction replication projects in the directly replicated labeled dataset

7.2.1.2 Dataset statistics and observation

The fields of these eight research replication projects are included in social science and are mainly about economics and psychology. The distribution of research papers' number and domains by eight direction replication projects in the directly replicated labeled dataset are showed in Table 7.1. The distribution of research papers' number by different fields is listed in Table 7.2. We observe that the number of economical research publications is larger than the ones in the psychology and other social science fields.

Dataset	# of docs	Econ	Psyc	Other
Directly replicated	399	206	172	21

Table 7.2: Distribution of research papers' number by fields (economics, psychology, and other social science fields) in the directly replicated labeled dataset

Among 399 annotated samples, 201 samples are labeled as '1' (replicable). The remaining 198 samples are annotated as '0' (non-replicable). From the distribution of class labels, we observe that this annotated dataset is balanced. Table 7.3-7.4 show two samples from the directly replicated labeled data and list the features of title, abstract, body, sample size, effect size and p-value.

Throughout looking into the research papers in the directly replicated labeled dataset by hand, we can observe that the replicable cases show more accurate description in the abstract, larger sample size, and larger ratio of pages on experiments than the non-replicable ones.

Label	1 (replicalbe)
Title	The Economics of Credence Goods: An Experiment on the Role of Liability, Verifiability, Reputation, and Competition
Abstract	Credence goods markets are characterized by asymmetric ... ... does not lead to higher efficiency as long as liability is violated.
Body	Repair services, medical treatments, the provision of software ... ... preferences on the performance of markets for credence goods.
Sample size[1]	936
Effect size[2]	0.29
P-value[3]	0.01

Table 7.3: A **replicable** sample in American Economic Review (Directly replicated labeled dataset)

Label	0 (non-replicalbe)
Title	Reference Points and Effort Provision
Abstract	A key open question for theories of reference-dependent ... ... If expectations are high, subjects work longer and earn more money than if expectations are low.
Body	Imagine two identical workers. One expected a salary increase of ... ... to stop at the two fixed payments but not at the mean.
Sample size[1]	238
Effect size[2]	0.22
P-value[3]	0.015

Table 7.4: A **non-replicable** sample in American Economic Review (Directly replicated labeled dataset)

[1] Number of observations in this research paper.

[2] Number measuring the strength of the relationship between two variables in a population, or a sample-based estimate of that quantity.

[3] Probability of obtaining test results at least as extreme as the results actually observed, under the assumption that

7.2.2 Crowdsourced dataset

The first type of data (directly replicated labeled) has high quality but is expensive. In reality, even such small-size good-quality training data can be hard to obtain due to its prohibitive cost. An alternative inexpensive and commonly used way to obtain labeled training data is crowdsourcing. The power of human prediction has been studied in many contexts [147]. We used crowdsourcing method to collect the rates about the credibility of social and behavioral science research papers in the Replication Markets, part of the larger DARPA-funded program on Systematizing Confidence in Open Research and Evidence (SCORE) [96].

7.2.2.1 Data collection

The whole data collecting process lasted for 10 months. In each month, 300 of the 3000 papers were divided into batches of 10 of the same field, e.g., psychology or economics, and opened for prediction. Each participant was assigned one or more batches according to their interests specified at the beginning of the data collecting process. For each paper, the participants were given complete information about the paper, including the title, abstract, authors, the published journal, DOI, and a link to a pdf file of the full paper. Meanwhile, the participants were also presented with summarized information related to the paper's main claim, including the extracted main claim, the corresponding hypothesis test, the experiment method, the results, and the main statistics, i.e., the sample size, effect size, and p-value.

The participants were asked to predict how likely (a probability between 0 and 1) the main claim is replicable with statistical significance. Participants were assigned accuracy scores

the null hypothesis is correct.

for their predictions in each batch, and the top four participants in each batch would receive a monetary prize. The accuracy scores were determined by the surrogate scoring rules [99], a strategy-proof scoring rule that works in the absence of ground truth. Over the whole data collecting process, we collected around 57,000 predictions in total. On average, each participant provided 127 predictions and each paper received 19 predictions. A more detailed statistics (average, standard variance, maximium, minimium, and media) on the number of predictions per paper is shown in Table 7.5.

# of predictions (per paper)	Avg	Std	Max	Min	Med
Crowdsourced	19.48	8.54	41	4	19

Table 7.5: Statistics (average, standard variance, maximium, minimium, and median) on the number of predictions per paper

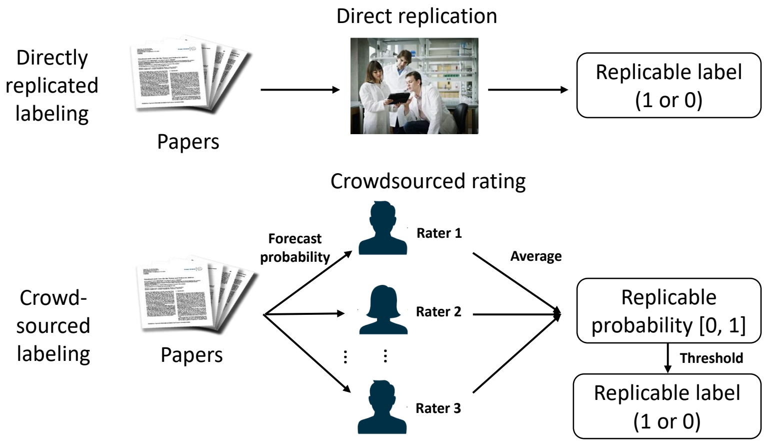


Figure 7.1: Data collection process for directly replicated and crowdsourced datasets

The comparison between the data collection processes for directly replicated and crowdsourced datasets are shown in Figure 7.1.

7.2.2.2 Dataset statistics and observation

Dataset	# of docs	Econ	Psyc	Other
Crowdsourced	2682	674	1763	245

Table 7.6: Distribution of research papers' number by fields (economics, psychology, and other social science fields) in the crowdsourced dataset

The distribution of research papers' number by different fields in the crowdsourced dataset is listed in Table 7.6. We observe that there are more psychological research publications than the ones in the economics and other social science fields.

As demonstrated in Table 7.6, the fields in the crowdsourced labeled data are mainly economics and psychology. Table 7.7 shows one replicable sample from the crowdsourced labeled data and list the features of publish year, digital object identifiers (DOI), title, abstract, claims, sample size, effect size and p-value.

7.3 Weakly Supervised Research Replication Prediction

Research Replication Prediction (RRP) task We model RRP task as a binary classification problem. We aim to build a model f that takes each research `paper` as input and predicts whether the research `paper` is replicable or not $f(\texttt{paper}) \in \{0 \text{ (non-replicable)}, 1 \text{ (replicable)}\}$. A research paper is considered as replicable if an independent replication can produce a statistically significant effect in the direction claimed in the original paper.

Label	1 (replicalbe)
Number of raters	9
Scores of raters	[0.6, 0.92, 0.81, 0.3, 0.6, 0.65, 0.75, 0.6, 0.4]
Label process	Average score: 0.527 > 0.50 (threshold)
Publish year	2015
DOI	10.1177/0956797615581491
Title	Variability modifies life satisfaction's association with mortality risk in older adults
Abstract	Greater life satisfaction is associated with greater longevity, but its variability across time has not been examined ... ... intraindividual variability provides additional insight into associations between psychological characteristics and health.
Claim1 abstract	These findings were qualified by a significant interaction such that individuals with low mean satisfaction and high variability in satisfaction had the greatest risk of mortality over the follow-up period.
...	
ClaimN abstract	The interaction between mean life satisfaction and variability in life satisfaction on proportional hazard was statistically significant.
Sample size	4458
Effect size	0.91
P-value	$p \leq 0.001$

Table 7.7: A **replicable** sample in Psychological Science (Crowdsourced dataset)

7.3.1 Feature Extraction

In this subsection, we explain how to extract the text features as the input of our baseline methods. Specifically, we utilized PDFMiner [133] to extract the text information from the raw pdf files of the research papers. Tf-idf features are extracted as the input of bag-of-words. As for the sequential models, BERT is used to obtain the input word embeddings. BERT can provide the context-aware word embeddings and can further improve the classification accuracy. We used the directly replicated and crowdsourced datasets to pretrain the model first and then fine-tuned using the directly replicated supervised data based on a published BERT pretrained model ("bert-base-uncased"[1]). Then the fine-tuned pretrained model is utilized as the embedding layer of bi-directional GRU [29]. BERT has the maximum limitation - 512 tokens for the input sequence. BERT contains two special tokens in the input sequence. The first token is always [CLS] which indicates the embedding containing the information of whole sequence. The other special token is [SEP] which is used to separate segments. For text classification tasks, the final hidden state of the first token [CLS] is usually extracted as the representation of the whole sequence.

We set the maximum length of documents L to 10,000 in the BERT model because the average length of all the documents in the labeled dataset is about 10,000. Since BERT takes an input of a sequence of no more than 512 tokens and outputs the representation of the sequence, we adopted the trick described in the [142]. We first divide the input text into $L = L/510$ sections. Then each section is fed into the BERT to obtain its representation. Then we used mean pooling to obtain the final representation of the overall document.

We showed the feature extraction process in Figure 7.2. Figure 7.3 shows a more detailed of feature extraction process and architecture of Model (neural network) in Figure 7.2.

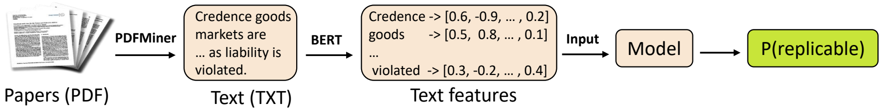

PDFMiner
Credence goods markets are ... as liability is violated.
BERT
Credence -> [0.6, -0.9, ... , 0.2]
goods -> [0.5, 0.8, ... , 0.1]
...
violated -> [0.3, -0.2, ... , 0.4]
Input
Model
P(replicable)
Papers (PDF)
Text (TXT)
Text features

Figure 7.2: Feature extraction process and architecture of the overall model

[1] https://huggingface.co/bert-base-uncased

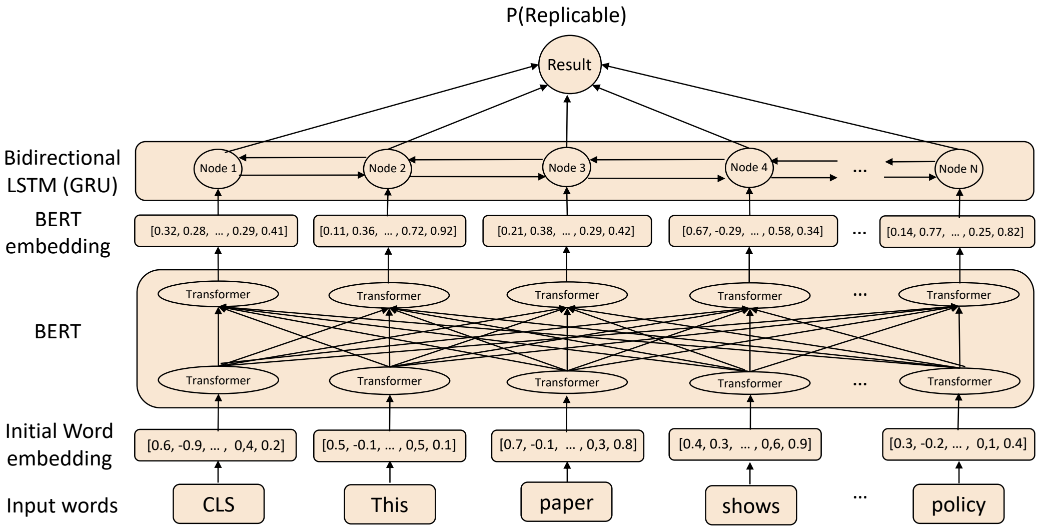


Figure 7.3: Detailed feature extraction process and architecture of Model in Figure 7.2

7.3.2 Supervised baseline methods

Our problem is formulated as a binary classification problem to predict whether a research paper can be replicated or not. We first used five commonly used binary classification algorithms including Logistic Regression (LR) [116], Random Forest (RF) [65], Support Vector Machine (SVM) [19], Multilayer Perceptron (MLP) [53], and BERT [39] using only the 300 labeled training samples.

7.3.3 Weakly supervised baseline methods

To better leverage the crowdsourced training samples, we also test how weakly supervised learning approaches fare for our task. Our weakly supervised learning approaches tie close to learning with noisy labels [17, 13, 131, 130, 149], as well as semi-supervised learning [37, 166, 162, 24]. Our considered baselines can be categorized as follows:

Loss correction A main research line of learning with noisy labels are loss correction methods require estimating the error rates. A representative method is variational inference (VI) aided weakly supervised method [101]. In VI, several basic classifiers (only for error rates' estimation) are first trained on the directly replicated dataset and then predictions of crowd sourced samples are obtained using the trained basic classifiers. Then noisy label of crowd sourced samples as well as the predictions of basic classifiers are used to estimate the error rates using the variational inference methods proposed by Liu et al. [89]. Finally, the noisy training is conducted on the crowdsourced dataset with the proxy loss function [112] as shown below:

$$\sum_{u=1}^{N_U} \frac{(1-\rho_{1-y_u})\ell(y_u^p, y_u) - \rho_{y_u}\ell(y_u^p, 1-y_u)}{1-\rho_1-\rho_0}$$

where y_u^p is the u-th sample's prediction of final LSTM model and y_u is the corresponding noisy label. ℓ is the standard cross entropy loss function. N_U is the total number of unlabeled training dataset and σ_0 , σ_1 are two classes' estimated error rates estimated using variational inference method mentioned in the Preliminary section.

Peer loss Instead of estimating the noise rates in VI (may introduce the extra errors), Liu and Guo [97] provided an alternative, peer loss, to deal with noisy labels without requiring an additional estimation step for the noise rates. To apply peer loss, we first construct peer samples for each sample in the unlabeled training dataset. More specifically, for the u-th training sample (x_u, y_u) in the unlabeled dataset, we randomly choose two other samples $(x_{u_1}, y_{u_1}), (x_{u_2}, y_{u_2})$ such that $u_1 \neq u_2$ and $u_1, u_2 \neq u$. Then we can construct the peer sample $(x_{u_1}, y_{u_1}), (x_{u_2}, y_{u_2})$ for

(x_u, y_u). Then we can calculate peer loss function as shown below:

$$\ell_{noise\_peer} = \sum_{u=1}^{N_U} \ell(y_u^p, y_u) - \alpha \cdot \ell(y_{u_1}^p, y_{u_2})$$

where $\ell(y_u^p, y_u)$ is a standard cross entropy loss function. y_u^p is the u-th sample's prediction and y_u is the corresponding noisy label. α is an important hyperparameter that need to be tuned with in the peer loss function. N_U is the number of unlabeled training dataset.

Semi-supervised learning We also applied the recent semi-supervised method in Natural Language Processing (NLP), MixText [24], as the other weakly supervised baseline method. More specifically, we drop all the labels in the crowdsourced dataset and consider it as an unlabeled dataset. A model is first trained on the directly replicated labeled dataset. Then the trained model provides low-entropy labels for two data augmentations generated by Russian and German machine translation trained models for each unlabeled data. Through ensemble methods, we can obtain the pseduo label for each unlabeled data. In the final training, MixText mixed the labeled and unlabeled using MixUp [173] by interpolating text in hidden space which is more suitable for NLP tasks.

7.4 Experiments

To provide the researchers with baselines to compare, we conducted the experiments on our RRP dataset using baseline methods.

7.4.1 Setup

We have 399 directly replicated labeled and 2,682 crowdsourced samples. Randomly selected 300 (150 replicable and 150 non-replicable) labeled and the 2,682 crowdsourced samples are considered as the training dataset. We test our proposed framework on the remaining 99 (51 replicable and 48 non-replicable) directly replicated samples.

For our five supervised models, TF-IDF features extracted by scikit-learn[2] are used as the input of LR, RF, SVM, and MLP models. The input of BERT model is obtained after pretraining on the crowdsourced dataset and fine-tuning on the directly replicated dataset. “Bert-base-uncased” pretrained model is used in this paper. It is on English language with a masked language modeling objective and the vocabulary size is 30,522. “Bert-base-uncased” contains an encoder with 12 Transformer blocks, 12 self-attention heads, and the hidden size of 768. Therefore, it has 110M parameters.

As for the weakly supervised learning methods, we tried MixText, VI, and PL which are described in Section 7.3.3. For directly replicated data, they conducted the same normal training. Their differences are shown in the training on the crowdsourced data. The basic classifiers for VI are LR, RF, SVM, MLP, and LSTM. For PL methods, we need to tune an important hyperparameter α. Introducing the hyperparameter α can make this weakly supervised model be more robust to class-imbalanced dataset. When PL is trained on the directly replicated labeled dataset, the hyperparameter α is always set to 0. When training on crowdsourced dataset, α is set to 0 in the first 30 epochs and will be continually increased with more epochs. These weakly supervised learning models are trained on three GeForce RTX 2080 GPUs. Other

[2] https://scikit-learn.org/stable/index.html

parameters setting are shown in Table 7.8.

Parameters	MixText	VI	PL
Epochs	50	500	500
Batch size	4 (L) & 24 (U)	128	128
Learning rate	e-5	5e-5	5e-5
Dropout	0.25	0.25	0.25

Table 7.8: Parameters for MixText, VI, and PL. ‘L’ and ‘U’ denotes labeled and unlabeled respectively.

Model	Train Setting	Accuracy	Precision	Recall	F1
LR	300 (Direct)	57.58	61.90	50.98	55.91
RF	300 (Direct)	51.52	54.05	39.22	45.45
SVM	300 (Direct)	58.59	63.04	56.86	59.79
MLP	300 (Direct)	59.60 ± 1.00	65.00 ± 1.00	50.98 ± 1.96	57.14 ± 1.31
BERT	300 (Direct)	65.66 ± 1.00	64.71 ± 0.65	67.35 ± 0.74	66.00 ± 1.30
MixText	[300 (Direct) + 2,682 (Crowd)]	66.67 ± 1.00	70.59 ± 1.32	69.02 ± 1.97	69.47 ± 0.67
VI	2,682 (Crowd)	65.66 ± 2.00	78.43 ± 1.89	63.49 ± 3.92	70.17 ± 1.93
VI	[300 (Direct) + 2,682 (Crowd)]	67.68 ± 2.00	69.23 ± 2.54	71.04 ± 1.96	70.12 ± 1.89
VI	[300 (Direct)] + [2,682 (Crowd)]	69.70 ± 1.00	73.22 ± 1.34	70.47 ± 0.69	71.84 ± 1.23
PL	2,682 (Crowd)	69.70 ± 2.00	72.55 ± 2.04	69.81 ± 1.96	71.15 ± 2.00
PL	[300 (Direct) + 2,682 (Crowd)]	70.71 ± 1.00	**82.35± 1.55**	67.74 ± 0.55	74.33 ± 1.29
PL	[300 (Direct)] + [2,682 (Crowd)]	**73.74 ± 1.00**	78.43 ± 1.77	**72.73 ± 1.96**	**75.47± 1.34**

Table 7.9: Comparison on Train setting, Test accuracy (%), Precision (%), Recall (%), and F1 (%) between Logistic Regression, Random Forest, SVM, BERT, MixText, Variational Inferance based method, and Peer Loss based method. Particularly, [300 (Direct) + 2,682 (Crowd)] means that model is trained on 300 directly replicated labeled samples and 2,682 crowdsourced samples labeled by trained model using 300 directly replicated labeled samples.

7.4.2 Results

The results (accuracy, precision, recall, and F1) of the five supervised ML models are reported in the first five lines in Table 7.9. The results (test accuracy, precision, recall, and F1) of seven weakly supervised models with different settings are reported in the last seven lines in Table 7.9.

As for different settings in Table 7.9, 300 (Direct) means that 300 directly replicated labeled samples are used to train. 2,682 (Crowd) means that 2,682 crowdsourced labeled samples are used to train. [300 (Direct) + 2,682 (Crowd)] means that 300 directly replicated labeled samples and 2,682 crowdsourced samples labeled by trained model on 300 directly replicated labeled samples are used to train. [300 (Direct)] + [2,682 (Crowd)] means that 300 directly replicated labeled samples and 2,682 crowdsourced labeled samples are used to train.

From Table 7.9, we first observe that all the weakly supervised ML methods (lines after the fifth line) using both of the two types of data are better than the classical supervised ML methods (first five lines) trained only on directly replicated labeled dataset. It shows that weakly supervised methods can make use of crowdsourced dataset with weak supervisions to improve the predict performance. The average results including the variance are reported in Table 7.9 after running the models for 10 times.

In the experiments with crowdsourced dataset included, we have two observations. Firstly, we only used crowdsourced dataset to train the model and aim to further test the effectiveness of PL weakly supervised learning with imperfectly labeled dataset. We found that the prediction accuracy of research replication is better than the performance in the previous experiments where the high quality directly replicated data were used. This finding suggests a very promising solution to the issue of research reproducibility because the cost of obtaining crowdsourced dataset is much less than that to get a labeled dataset through direct replication. Second, to examine the benefits of crowdsourced labels to the training, we compared [300 (direct) + 2,682 (crowd)] with [300 (direct)] + 2,682 (crowd) settings. The labels of 2,682 crowdsourced samples in the former setting were obtained by the model trained on 300 directly replicated labeled samples. We found that the performance of the latter setting is better. It suggests that the labels of crowdsourced data can be best leveraged with the frameworks of learning with noisy labels.

7.4.3 Ablation Studies

We conducted the ablation studies to show the effectiveness of the information included in our RRP datasets.

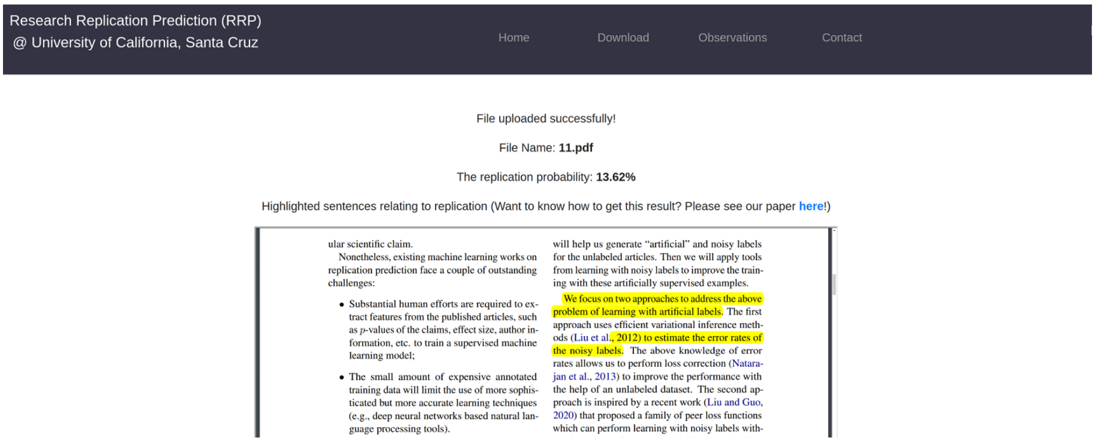
7.4.3.1 Remove different labels from RRP

Model	Train setting	Accuracy
PL	[300 (Direct)] + [2,682 (Crowd)]	73.74%
PL	w/o Crowd's labels: [300 (Direct) + 2,682 (Crowd)]	70.71%
PL	w/o Direct's labels: 2,682 (Crowd)	69.70%

Table 7.10: Ablation studies of PL method utilizing all label information of RRP dataset on accuracy.

Based on the results using PL weakly supervised learning in Table 7.9, we compare the settings of PL with [300 (Direct)] + [2,682 (Crowd)], PL with [300 (Direct) + 2,682 (Crowd)], and PL with 2,682 (Crowd) in Table 7.10. We observe that a larger drop on the accuracy without the training on directly replicated labeled dataset. It is reasonable because directly replicated labeled dataset is of high quality with higher cost than crowdsourced dataset. A promising finding is that the drop on the accuracy without training on crowdsourced labeled dataset is almost the same as the one without training on directly replicated labeled dataset. It means that we can utilized weakly supervised learning methods training on the crowdsourced dataset to obtain the almost same performance as the one training on high-quality directly replicated dataset, but with a much less cost.

7.4.4 RRP website


Research Replication Prediction (RRP)
@ University of California, Santa Cruz
Home
Download
Observations
Contact

File uploaded successfully!

File Name: **11.pdf**

The replication probability: **13.62%**

Highlighted sentences relating to replication (Want to know how to get this result? Please see our paper here!)

ular scientific claim.

Nonetheless, existing machine learning works on replication prediction face a couple of outstanding challenges:

- Substantial human efforts are required to extract features from the published articles, such as p-values of the claims, effect size, author information, etc. to train a supervised machine learning model;
- The small amount of expensive annotated training data will limit the use of more sophisticated but more accurate learning techniques (e.g., deep neural networks based natural language processing tools).

will help us generate "artificial" and noisy labels for the unlabeled articles. Then we will apply tools from learning with noisy labels to improve the training with these artificially supervised examples.

We focus on two approaches to address the above problem of learning with artificial labels. The first approach uses efficient variational inference methods (Liu et al., 2012) to estimate the error rates of the noisy labels. The above knowledge of error rates allows us to perform loss correction (Natarajan et al., 2013) to improve the performance with the help of an unlabeled dataset. The second approach is inspired by a recent work (Liu and Guo, 2020) that proposed a family of peer loss functions which can perform learning with noisy labels with-

Figure 7.4: Page showing the prediction probability as well as the highlighted important sentences in our RRP website

We also built a RRP website where users can upload their own research paper (PDF format) and obtain the replication prediction probabilities as well as highlighted sentences obtained by the variational contextual consistency sentence masking (VCCSM) method described in the Chapter 4. The screenshots of our RRP website is shown in Figure 7.4.

7.5 Related Work

Researchers have conducted direct replication projects in contemporary published social science studies to alleviate the replication crisis [14, 15, 44, 79, 80, 33]. As such direct replication is extremely time-consuming and expensive, ML methods serve as a much more efficient method to predict the reproducibility of a scientific finding [43, 167, 3, 101].

The labeled data for RRP obtained by direct replication is of high quality, but is also very expensive to obtain in practice. One alternative way to obtain labeled training data is crowdsourcing, which is inexpensive and commonly used [58, 59, 161, 96]. The downside of crowdsourcing methods is that the labels generated by human raters are often inaccurate. Therefore, we need to seek the help of weakly supervised learning methods that have been utilized to make use of a large size of weakly supervised data to improve the prediction performance [9, 162, 24, 101]. Therefore, we applied the recent weakly supervise learning methods as the baselines for RRP task.

Chapter 8

Conclusion and Future Directions

8.1 Conclusion

In this PhD thesis, we show several studies of weakly supervised learning methods in text classification which is a fundamental task in NLP community. More specifically, we first proposed several new weakly supervised learning methods to further improve the accuracy and interpretability of neural network models on two main research directions: learning with noisy labels and semi-supervised learning. As for learning with noisy labels, two weakly supervised learning methods are proposed to further improve the accuracy in the supervised-starved task – Research Replication Prediction. For semi-supervised learning, we proposed a new weakly supervised learning method to improve the model interpretability with the help of unlabeled dataset. In addition, we proposed a new ensemble method to further improve the quality of pseudo or noisy labels for the unlabeled dataset comparing with the existing ensemble methods (apply the majority rule to obtain the answer) since our model can reveal the minority correct answer when the majority answer is wrong.

Furthermore, we also found the fairness issue about the imbalanced improvement among different sub-populations such as race and gender in semi-supervised learning for text classification tasks. Finally, we contributes a new weakly supervised learning dataset (Research Replication Prediction) to the community for facilitate the researchers to develop the weakly supervised learning models more efficiently.

8.2 Future Directions

My long-term research goal is to propose new and general weakly supervised learning methods to sufficiently make use of the unlabeled dataset to further improve all aspects of performance in NLP such as accuracy and interpretability. Following the goal, several future directions are shown in the remaining chapter.

8.2.1 Weakly Supervised Learning in Sequential Tasks in NLP

Sequential tasks in NLP such as natural language generation [155, 154, 62, 93], machine translation [45], and sequence tagging [64] are different from text classification. The output of sequential tasks are sequence instead of assigning one class label once to a text unit in the text classification. Therefore the sequential tasks have different properties. For example, the errors will accumulate in the subsequent sequence generation if the current prediction is wrong. Therefore, the loss functions of learning with noisy labels methods for the sequential tasks should be different from the ones applied in the text classification.

8.2.2 Multimodel Weakly Supervised Learning

Multimodel draws a lot of attention in recent years [180, 171]. Many tasks has evolved into more complex models of multimodal beyond the model utilizing only single source of data. For example, sentiment analysis is a traditional text classification task. In the social media, different sources of data such as image and audio are accompanied with text information. Therefore, sentiment analysis can be evolved into multimodel sentiment analysis task. The new weakly supervised learning methods which are specially designed for the new multimodel tasks needed to be explored.

8.2.3 Fusing Different Types of Research Methods in Weakly Supervised Learning

There are different types of weakly supervised learning methods such as learning with noisy label [89, 112, 130, 149, 97] and semi-supervised learning methods [86, 71, 9, 8]. They are different but have their respect advantages. Semi-supervised learning methods focus on providing the high-quality labels and use the normal loss functions e.g., standard Cross-Entropy Loss. Learning with noisy labels aims to apply the modified loss functions including the error rates information to against the noisy labels. Fusing these different types of weakly supervised learning methods is a promising research direction.

Appendix A

Omitted Proofs and Additional Results

A.1 Detailed Derivation of Lower Bound for VCCSM in Section 4.2

In this section, we provided the complete details on the derivation of lower bound for Variational Contextual Consistency Sentence Masking (VCCSM) in Section 4.2.

Assuming that the true joint distribution is $P(X, Y, Z)$ and X, Y, Z are random variables which have the following conditional dependency: $Y \leftrightarrow X \leftrightarrow Z$. And x, y, z are instances of ramdom variables. We can have

$$P(X, Y, Z) = P(Z|X, Y)P(Y|X)P(X) = P(Z|X)P(Y|X)P(X). \tag{A.1}$$

According to the definition of $I(Z; Y)$, we have

$$I(Z; Y) = \sum_{z,y} P_{Z,Y}(z, y) \log \frac{P_{Z,Y}(z, y)}{P_Z(z)P_Y(y)} = \sum_{z,y} P_{Z,Y}(z, y) \log \frac{P_{Y|Z}(y|z)}{P_Y(y)}. \tag{A.2}$$

And we also have

$$\begin{aligned} P_{Y|Z}(y|z) = \sum_x P_{X,Y|Z}(x,y|z) &= \sum_x P_{Y|X}(y|x) P_{X|Z}(x|z) \\ &= \sum_x \frac{P_{Y|X}(y|x) P_{Z|X}(z|x) P_X(x)}{P_Z(z)}. \end{aligned} \tag{A.3}$$

Since $P(Y|Z)$ can be intractable, $Q(Y|Z)$ is considered as a variational approximation to $P(Y|Z)$. $Q(Y|Z)$ is our decoder and a neural network. Because the Kullback Leibler divergence is non-negative, we have

$$\mathrm{KL}[P(Y|Z)||Q(Y|Z)] \geq 0 \Rightarrow \sum_y p(y|z) \log p(y|z) \geq \sum_y p(y|z) \log q(y|z). \tag{A.4}$$

Therefore, we can obtain the lower bound of $I(Z;Y)$ as follows:

$$I(Z;Y) \geq \sum_{z,y} P_{Z,Y}(z,y) \log \frac{Q_{Y,Z}(y|z)}{P_Y(y)} = \sum_{z,y} P_{Z,Y}(z,y) \log Q_{Y|Z}(y|z) + H(Y). \tag{A.5}$$

where $H(Y) = -\sum_y P_Y(y) \log P_Y(y)$ is entropy. According to Equation A.1, we have

$$\begin{aligned} P(Y|Z) = \sum_x P_{X,Y,Z}(x,y,z) &= \sum_x P_{X,Y,Z}(x,y,z) \\ &= \sum_x P_X(x) P_{Y|X}(y|x) P_{Z|X}(z|x). \end{aligned} \tag{A.6}$$

Hence, we obtain the lower bound of $I(Z, Y)$ as follows:

$$\sum_{x,y,z} P_X(x) P_{Y|X}(y|x) P_{Z|X}(z|x) \log Q_{Y|Z}(y|z).$$

As for $I(Z; X)$, similar to Equation A.2 in the derivation of $I(Z; Y)$, we first obtain

$$\begin{aligned} I(Z; X) &= \sum_{z,x} P_{Z,X}(z, x) \log \frac{P_{Z|X}(z|x)}{P_Z(z)} \\ &= \sum_{z,x} P_{Z,X}(z, x) \log P_{Z|X}(z|x) - \sum_{z} P_Z(z) \log P_Z(z). \end{aligned} \tag{A.7}$$

Because the marginal distribtuion of Z, $P(Z) = \sum_x P_{Z|X}(z|x) P_X(x)$ in which the computation might be difficult, we replace $P(Z)$ by a variational approximation of $Q(Z)$. Since $\text{KL}[P(Z)||Q(Z)] \geq 0 \Rightarrow \sum_z P_Z(z) \log P_Z(z) \geq \sum_z P_Z(z) \log Q_Z(z)$, we can get the upper bound of $I(Z; X)$ as follows:

$$\begin{aligned} I(Z; X) &\leq \sum_{z,x} P_{Z,X}(z, x) \log P_{Z|X}(z|x) - \sum_{z,x} P_{Z,X}(z, x) \log Q_Z(z) \\ &\leq \sum_{z,x} P_X(x) P_{Z|X}(z|x) \log \frac{P_{Z|X}(z|x)}{Q_Z(z)}. \end{aligned} \tag{A.8}$$

Combining Equation A.5 and A.8, we can get the lower bound of $I(Z; Y) - \beta I(Z; X)$ as

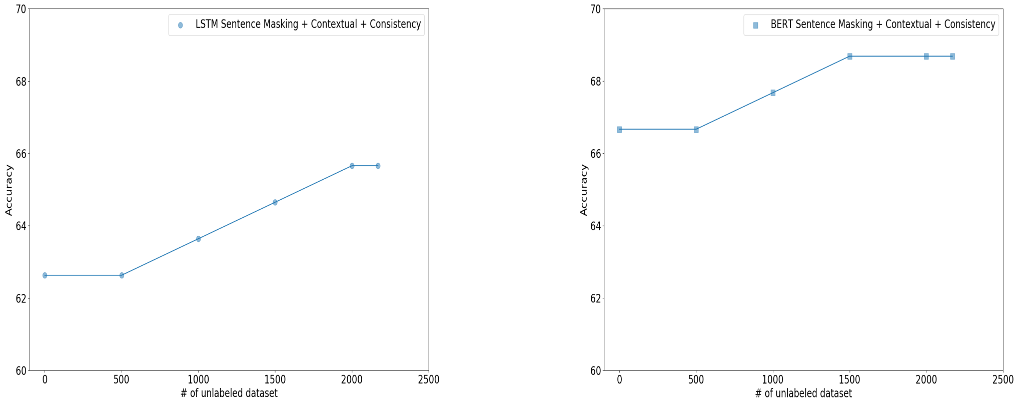


Figure A.1: Testing accuracy (%) on RRP dataset with varying number of unlabeled dataset for VCCSM applied on two neural text classifiers (LSTM and BERT)

follows:

$$\sum_{x,y,z} P_X(x)P_{Y|X}(y|x)P_{Z|X}(z|x)\log Q_{Y|X}(y|z)$$
$$-\beta\sum_{z,x} P_X(x)P_{Z|X}(z|x)\log\frac{P_{Z|X}(z|x)}{Q_Z(z)}.$$

A.2 Performance with Varying Number of Unlabeled Data

We conducted the experiments to test our model's effectiveness by varying number of unlabeled data for VCCSM applied on two neural text classifiers (LSTM and BERT). From Figure A.1, we can observe that, with more unlabeled data, the testing accuracy become higher on both LSTM Sentence Masking + Contextual + Consistency and BERT Sentence Masking + Contextual + Consistency models, which validates the effectiveness of using unlabeled data.

A.3 Proof of Theorems in Section 5.3.5

In this part, we provided the detailed proof of two theorems which are the analytical evidences for the correctness of our proposed approaches. For simplicity, we only show the proof details of binary classification. The proof of multi-class classification is similar to the binary case. This proof is largely adapted from [119]. Nonetheless we reproduce the details for completeness.

Theorem 1. *The correct answer (majority or minority) cannot be deduced by any algorithm if only relying on posterior probabilities,* $Q(s_i, k), i = 1, ..., S; k = 0, 1$ *because considering either* 0 *or* 1 *as the correct label can generate the same posterior probabilities based on the training dataset.*

Proof. In this proof, for any arbitrarily selected class label as the answer, we can generate the same posterior probabilities. Therefore, we cannot decide which label (majority or minority) is the true class label if only relying on posterior probabilities.

Denote by k^* as the true class label. Given the training dataset, $\mathbb{P}(s_i \mid k^*), i = 1, ..., S$ is known. Based on the description of theorem, the posterior probabilities $Q(s_i, k) = \mathbb{P}(k \mid s_i), i = 1, ..., S; k = 0, 1$ is also known.

But we don't know which class label is the truth label. We arbitrarily selected one class label l as the true label. We denote the corresponding model is $K(s_i, k)$. We will prove that $K(s_i, k)$ can generate the same $\mathbb{P}(s_i \mid k^*), i = 1, ..., S$ and $Q(s_i, k) = \mathbb{P}(k \mid s_i), i = 1, ..., S; k = 0, 1$ for any arbitrarily selected class label k.

Because the known parts don't constrain the prior over the feature vector s_i. In particular, we can set the prior of model $K(s_i, k)$ to:

$$\mathbb{K}(s_i) = \frac{\mathbb{P}(s_i \mid k^*)}{\mathbb{P}(k \mid s_i)} \left(\sum_r \frac{\mathbb{P}(s_r \mid k^*)}{\mathbb{P}(k \mid s_r)} \right)^{-1}$$

Because the posteriors in the corresponding model $K(s_i, k)$ must equal to the known posteriors, we have $\mathbb{K}(k \mid s_i) = \mathbb{P}(k \mid s_i)$, for $i = 1, ..., S; k = 0, 1$. So we can get the joint distribution of label k and the feature vector s_i:

$$\begin{aligned} \mathbb{K}(k, s_i) &= \mathbb{K}(k \mid s_i)\mathbb{K}(s_i) = \mathbb{P}(k \mid s_i)\mathbb{K}(s_i) \\ &= \mathbb{P}(s_i \mid k^*) \left(\sum_r \frac{\mathbb{P}(s_r \mid k^*)}{\mathbb{P}(k \mid s_r)} \right)^{-1} \end{aligned}$$

Then we can get the marginal distribution k by summing over i:

$$\mathbb{K}(k) = \sum_i \mathbb{P}(s_i \mid k^*) \left(\sum_r \frac{\mathbb{P}(s_r \mid k^*)}{\mathbb{P}(k \mid s_r)} \right)^{-1} = \left(\sum_r \frac{\mathbb{P}(s_r \mid k^*)}{\mathbb{P}(k \mid s_r)} \right)^{-1}$$

After getting the marginal distributions $\mathbb{K}(s_i), \mathbb{K}(k)$, and the posteriors, $\mathbb{K}(k \mid s_i)$, for $i = 1, ..., S$, the feature vector distribution s_i of the arbitrarily selected class label k, $\mathbb{K}(s_i \mid k)$ can be calculated by:

$$\mathbb{K}(s_i \mid k) = \frac{\mathbb{K}(k \mid s_i)\mathbb{K}(s_i)}{\mathbb{K}(k)} = \mathbb{P}(s_i \mid k^*)$$

Because k was arbitrarily chosen, this theorem is proved.

□

Theorem 1 implies that any existing ensemble algorithm based on the majority voting rule cannot always infer the true answer no matter either majority or minority is the final true answer. In other words, we cannot decide whether majority or minority is correct if we only know the information of the posterior probabilities Q over all the possible labels. The majority rule applied by the existing ensemble methods is a special case of Theorem 1.

In the following part, we are considering the extra information which is the estimation of other classifiers' prediction results. We use $\mathbb{P}(v_k \mid s_i), k \in \{0, 1\}$ to represent the how many percentage of basic classifiers will predict label k given s_i.

We also define two possible learnt final classification functions ω_0^i and ω_1^i which decide the final label for each s_i. ω_0^i is the function which finally predict s_i as 0 and ω_1^i is the function which finally predict s_i as 1. If the true label is 1, ω_1^i is defined as the actual final classifier and ω_0^i is counterfactural final classifier. For simplicity, we ignore the input index of $\omega_k^i, k \in \{0, 1\}$ for each s_i and write it as $\omega_k, k \in \{0, 1\}$ in the proof of Theorem 2.

Theorem 2. *For input s_i, the estimate of the prior prediction for the correct classification label denoted as k^* will be strictly underestimated if the prediction probability of the true label is less than 1. We can express this as*

$$P(s_i, k^*) < Q(s_i, k^*) \quad \textit{if } \mathbb{P}(k^* \mid s_i) < 1.$$

Proof. For each s_i, we set k^* as the true label. We first prove that the actual percentage of predicted labels for the true label in the actual final classifier exceeds counterfactual classifier's percentage for the true label, $\mathbb{P}(v_{k^*} \mid w_{k^*}) > \mathbb{P}(v_{k^*} \mid w_k), k \neq k^*$.

Based on the description of ω_k and v_k mentioned above and a BTS's hidden assumption that the minority but expert classifiers hold a stronger belief about the ground truth label than the majority classifiers who predicted wrongly, for the true label k_*, the probability of ω_{k_*} being the actual final classifier for the expert classifiers predicting correctly is higher than the one for the non-expert classifiers predicting the other wrong label k. Therefore, we can get $\mathbb{P}(w_{k^*} \mid v_{k^*}) > \mathbb{P}(w_{k^*} \mid v_k)$. Then we have $\mathbb{P}(w_{k^*} \mid v_{k^*})\mathbb{P}(v_k) > \mathbb{P}(w_{k^*} \mid v_k)\mathbb{P}(v_k)$ by timing the same factor $P(v_k)$ on both sides. So we have:

$$\mathbb{P}(w_{k^*} \mid v_{k^*}) > \mathbb{P}(w_{k^*} \mid v_{k^*})\mathbb{P}(v_{k^*}) + \mathbb{P}(w_{k^*} \mid v_k)\mathbb{P}(v_k) = \mathbb{P}(w_{k^*}) \tag{A.9}$$

According to Bayesian rule, we have the following deduction:

$$\frac{\mathbb{P}(v_{k^*} \mid w_{k^*})}{\mathbb{P}(v_{k^*} \mid w_k)} = \frac{\mathbb{P}(w_{k^*} \mid v_{k^*})\mathbb{P}(w_k)}{\mathbb{P}(w_k \mid v_{k^*})\mathbb{P}(w_{k^*})} = \frac{\mathbb{P}(w_{k^*} \mid v_{k^*})}{1 - \mathbb{P}(w_{k^*} \mid v_{k^*})} \frac{1 - \mathbb{P}(w_{k^*})}{\mathbb{P}(w_{k^*})} \tag{A.10}$$

Based on equation A.9, equation A.10 is greater than one. So $\mathbb{P}(v_{k^*} \mid w_{k^*}) > \mathbb{P}(v_{k^*} \mid w_k), k \neq k^*$ is proved.

The estimate of classification prediction given the feature value s_i can be computed by marginalizing the actual and counterfactual final classifiers, $\mathbb{P}(v_{k^*} \mid s_i) = \mathbb{P}(v_{k^*} \mid w_{k^*})\mathbb{P}(w_{k^*} \mid s_i) + \mathbb{P}(v_{k^*} \mid w_k)\mathbb{P}(w_k \mid s_i)$. And we proved that $\mathbb{P}(v_{k^*} \mid w_{k^*}) > \mathbb{P}(v_{k^*} \mid w_k), k \neq k^*$. Therefore, $\mathbb{P}(v_{k^*} \mid s_i) \leq \mathbb{P}(v_{k^*} \mid w_{k^*})$. It will be the strict inequality unless $\mathbb{P}(w_{k^*} \mid s_i) = 1$. If the prediction probability is less than 1, the prior prediction for each s_i will be strictly underestimated. So we can get $P(s_i, k^*) < Q(s_i, k^*)$ if the prediction probability is less than 1.

This theorem is proved.

□

Theorem 2 shows that having the prior information can help improve the robustness of models because the minority correct classification result can be recovered using the rule descried in the theorem when the minority is the true answer instead of the majority answer. In other words, having Theorem 2, the true minority answer can be revealed as correct if the prior probability is less than the posterior one. The existing ensemble methods always adopt the majority result as the final answer and cannot recover the minority correct answer.

Bibliography

[1] Vibhu Agarwal, Tanya Podchiyska, Juan M Banda, Veena Goel, Tiffany I Leung, Evan P Minty, Timothy E Sweeney, Elsie Gyang, and Nigam H Shah. 2016. Learning statistical models of phenotypes using noisy labeled training data. *Journal of the American Medical Informatics Association*, 23(6):1166–1173.

[2] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. 2016. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*.

[3] Adam Altmejd, Anna Dreber, Eskil Forsell, Juergen Huber, Taisuke Imai, Magnus Johannesson, Michael Kirchler, Gideon Nave, and Colin Camerer. 2019. Predicting the replicability of social science lab experiments. *PloS one*, 14(12).

[4] Eugene Bagdasaryan, Omid Poursaeed, and Vitaly Shmatikov. 2019. Differential privacy has disparate impact on model accuracy. *Advances in Neural Information Processing Systems*, 32:15479–15488.

[5] Yingbin Bai, Erkun Yang, Bo Han, Yanhua Yang, Jiatong Li, Yinian Mao, Gang Niu, and Tongliang Liu. 2021. Understanding and improving early stopping for learning with noisy labels. *Advances in Neural Information Processing Systems*, 34.

[6] Peter L Bartlett and Shahar Mendelson. 2002. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482.

[7] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, et al. 2018. Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *arXiv preprint arXiv:1810.01943*.

[8] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. 2019. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*.

[9] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. 2019. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

[10] Marouane Birjali, Mohammed Kasri, and Abderrahim Beni-Hssane. 2021. A comprehensive survey on sentiment analysis: Approaches, challenges and trends. *Knowledge-Based Systems*, 226:107134.

[11] Christopher M Bishop. 2006. *Pattern recognition and machine learning*. springer.

[12] Abel Brodeur, Nikolai Cook, and Anthony Heyes. 2020. Methods matter: P-hacking and publication bias in causal analysis in economics. *American Economic Review*, 110(11):3634–60.

[13] Tom Bylander. 1994. Learning linear threshold functions in the presence of classification noise. In *Proceedings of the seventh annual conference on Computational learning theory*, pages 340–347.

[14] Colin F Camerer, Anna Dreber, Eskil Forsell, Teck-Hua Ho, Jürgen Huber, Magnus Johannesson, Michael Kirchler, Johan Almenberg, Adam Altmejd, Taizan Chan, et al. 2016. Evaluating replicability of laboratory experiments in economics. *Science*, 351(6280):1433–1436.

[15] Colin F Camerer, Anna Dreber, Felix Holzmeister, Teck-Hua Ho, Jürgen Huber, Magnus Johannesson, Michael Kirchler, Gideon Nave, Brian A Nosek, Thomas Pfeiffer, et al. 2018. Evaluating the replicability of social science experiments in nature and science between 2010 and 2015. *Nature Human Behaviour*, 2(9):637–644.

[16] Davide Castelvecchi. 2016. Can we open the black box of ai? *Nature News*, 538(7623):20.

[17] Nicolo Cesa-Bianchi, Shai Shalev-Shwartz, and Ohad Shamir. 2011. Online learning of noisy data. *IEEE Transactions on Information Theory*, 57(12):7907–7931.

[18] Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. 2019. Neural legal judgment prediction in english. *arXiv preprint arXiv:1906.02059*.

[19] Chih-Chung Chang and Chih-Jen Lin. 2011. Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):27.

[20] Ming-Wei Chang, Lev-Arie Ratinov, Dan Roth, and Vivek Srikumar. 2008. Importance of semantic representation: Dataless classification. In *Aaai*, volume 2, pages 830–835.

[21] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. 2006. Semi-supervised learning. *MIT Press*.

[22] Hanjie Chen, Song Feng, Jatin Ganhotra, Hui Wan, Chulaka Gunasekara, Sachindra Joshi, and Yangfeng Ji. 2021. Explaining neural network predictions on sentence pairs via learning word-group masks. *arXiv preprint arXiv:2104.04488*.

[23] Hanjie Chen and Yangfeng Ji. 2020. Learning variational word masks to improve the interpretability of neural text classifiers. *arXiv preprint arXiv:2010.00667*.

[24] Jiaao Chen, Zichao Yang, and Diyi Yang. 2020. Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification. *arXiv preprint arXiv:2004.12239*.

[25] Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan. 2018. Learning to explain: An information-theoretic perspective on model interpretation. In *International Conference on Machine Learning*, pages 883–892. PMLR.

[26] Kay-Yut Chen, Leslie R Fine, and Bernardo A Huberman. 2004. Eliminating public knowledge biases in information-aggregation mechanisms. *Management Science*, 50(7):983–994.

[27] Mingda Chen, Qingming Tang, Karen Livescu, and Kevin Gimpel. 2019. Variational sequential labelers for semi-supervised learning. *arXiv preprint arXiv:1906.09535*.

[28] Hao Cheng, Zhaowei Zhu, Xingyu Li, Yifei Gong, Xing Sun, and Yang Liu. 2021.

Learning with instance-dependent label noise: A sample sieve approach. In *International Conference on Learning Representations*.

[29] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

[30] Kevin Clark, Minh-Thang Luong, Christopher D Manning, and Quoc V Le. 2018. Semi-supervised sequence modeling with cross-view training. *arXiv preprint arXiv:1809.08370*.

[31] Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. *arXiv preprint arXiv:1804.05685*.

[32] Open Science Collaboration. 2012. An open, large-scale, collaborative effort to estimate the reproducibility of psychological science. *Perspectives on Psychological Science*, 7(6):657–660.

[33] Open Science Collaboration et al. 2015. Estimating the reproducibility of psychological science. *Science*, 349(6251):aac4716.

[34] Sam Corbett-Davies and Sharad Goel. 2018. The measure and mismeasure of fairness: A critical review of fair machine learning. *arXiv preprint arXiv:1808.00023*.

[35] Peng Cui and Le Hu. 2021. Sliding selector network with dynamic memory for extractive summarization of long documents. In *Proceedings of the 2021 Conference of the North*

American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 5881–5891.

[36] Emmanuel Gbenga Dada, Joseph Stephen Bassi, Haruna Chiroma, Adebayo Olusola Adetunmbi, Opeyemi Emmanuel Ajibuwa, et al. 2019. Machine learning for email spam filtering: review, approaches and open research problems. *Heliyon*, 5(6):e01802.

[37] Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. *Advances in neural information processing systems*, 28:3079–3087.

[38] Pieter-Tjerk De Boer, Dirk P Kroese, Shie Mannor, and Reuven Y Rubinstein. 2005. A tutorial on the cross-entropy method. *Annals of operations research*, 134(1):19–67.

[39] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[40] Emily Diana, Wesley Gill, Michael Kearns, Krishnaram Kenthapadi, Aaron Roth, and Saeed Sharifi-Malvajerdi. 2021. Multiaccurate proxies for downstream fairness. *arXiv preprint arXiv:2107.04423*.

[41] Tobias Dienlin, Niklas Johannes, Nicholas David Bowman, Philipp K Masur, Sven Engesser, Anna Sophie Kümpel, Josephine Lukito, Lindsey M Bier, Renwen Zhang, Benjamin K Johnson, et al. 2021. An agenda for open science in communication. *Journal of Communication*, 71(1):1–26.

[42] Thomas G Dietterich. 2000. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer.

[43] A Dreber, T Pfeiffer, E Forsell, D Viganola, M Johannesson, Y Chen, B Wilson, BA Nosek, and J Almenberg. 2019. Predicting replication outcomes in the many labs 2 study. *Journal of Economic Psychology*.

[44] Charles R Ebersole, Olivia E Atherton, Aimee L Belanger, Hayley M Skulborstad, Jill M Allen, Jonathan B Banks, Erica Baranski, Michael J Bernstein, Diane BV Bonfiglio, Leanne Boucher, et al. 2016. Many labs 3: Evaluating participant pool quality across the academic semester via replication. *Journal of Experimental Social Psychology*, 67:68–82.

[45] Qingkai Fang and Yang Feng. 2022. Neural machine translation with phrase-level universal visual representations. *arXiv preprint arXiv:2203.10299*.

[46] Erin D Foster and Ariel Deardorff. 2017. Open science framework (osf). *Journal of the Medical Library Association: JMLA*, 105(2):203.

[47] Leonard P Freedman, Iain M Cockburn, and Timothy S Simcoe. 2015. The economics of reproducibility in preclinical research. *PLoS Biol*, 13(6):e1002165.

[48] Yoav Freund and Robert E Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139.

[49] Ruoyuan Gao and Chirag Shah. 2021. Addressing bias and fairness in search systems. In

Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 2643–2646.

[50] Pascal Germain, Alexandre Lacasse, Francois Laviolette, Mario Marchand, and Jean-Francis Roy. 2015. Risk bounds for the majority vote: From a pac-bayesian analysis to a learning algorithm. *The Journal of Machine Learning Research*, 16(1):787–860.

[51] Elizabeth Gómez, Ludovico Boratto, and Maria Salamó. 2021. Disparate impact in item recommendation: A case of geographic imbalance. In *European Conference on Information Retrieval*, pages 190–206. Springer.

[52] Chen Gong, Dacheng Tao, Xiaojun Chang, and Jian Yang. 2017. Ensemble teaching for hybrid label propagation. *IEEE transactions on cybernetics*, 49(2):388–402.

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

[54] Yves Grandvalet, Yoshua Bengio, et al. 2005. Semi-supervised learning by entropy minimization. *CAP*, 367:281–296.

[55] Lan-Zhe Guo, Zhen-Yu Zhang, Yuan Jiang, Yu-Feng Li, and Zhi-Hua Zhou. 2020. Safe deep semi-supervised learning for unseen-class unlabeled data. In *International Conference on Machine Learning*, pages 3897–3906. PMLR.

[56] Suchin Gururangan, Tam Dang, Dallas Card, and Noah A Smith. 2019. Variational pretraining for semi-supervised text classification. *arXiv preprint arXiv:1906.02242*.

[57] Xiaochuang Han, Byron C Wallace, and Yulia Tsvetkov. 2020. Explaining black box

predictions and unveiling data artifacts through influence functions. *arXiv preprint arXiv:2005.06676*.

[58] Robin Hanson. 1995. Could gambling save science? encouraging an honest consensus.

[59] Robin Hanson. 2003. Combinatorial information market design. *Information Systems Frontiers*, 5(1):107–119.

[60] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29:3315–3323.

[61] Tatsunori Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. 2018. Fairness without demographics in repeated loss minimization. In *International Conference on Machine Learning*, pages 1929–1938. PMLR.

[62] Tatsunori B Hashimoto, Hugh Zhang, and Percy Liang. 2019. Unifying human and statistical evaluation for natural language generation. *arXiv preprint arXiv:1904.02792*.

[63] Yotam Hechtlinger. 2016. Interpretation of prediction models using the input gradient. *arXiv preprint arXiv:1611.07634*.

[64] Benjamin Heinzerling and Michael Strube. 2019. Sequence tagging with contextual and non-contextual subword representations: A multilingual evaluation. *arXiv preprint arXiv:1906.01569*.

[65] Tin Kam Ho. 1995. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE.

[66] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

[67] Sara Hooker, Aaron Courville, Gregory Clark, Yann Dauphin, and Andrea Frome. 2019. What do compressed deep neural networks forget? *arXiv preprint arXiv:1911.05248*.

[68] Sara Hooker, Nyalleng Moorosi, Gregory Clark, Samy Bengio, and Emily Denton. 2020. Characterising bias in compressed models. *arXiv preprint arXiv:2010.03058*.

[69] Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

[70] Zhuo Huang, Chao Xue, Bo Han, Jian Yang, and Chen Gong. 2021. Universal semi-supervised learning. *Advances in Neural Information Processing Systems*, 34.

[71] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. 2019. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5070–5079.

[72] Zhimeng Jiang, Xiaotian Han, Chao Fan, Zirui Liu, Na Zou, Ali Mostafavi, and Xia Hu. 2022. Fmp: Toward fair graph message passing against topology bias.

[73] Zhimeng Jiang, Xiaotian Han, Chao Fan, Fan Yang, Ali Mostafavi, and Xia Hu. 2022. Generalized demographic parity for group fairness. In *International Conference on Learning Representations*.

[74] Kaggle. 2018. Jigsaw Toxicity dataset: Toxic comment classification challenge. `https://www.kaggle.com/c/`

jigsaw-unintended-bias-in-toxicity-classification. Accessed: 2021-11-15.

[75] Falaah Arif Khan, Eleni Manis, and Julia Stoyanovich. 2021. Translation tutorial: Fairness and friends. In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*.

[76] Michael P Kim, Amirata Ghorbani, and James Zou. 2019. Multiaccuracy: Black-box post-processing for fairness in classification. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 247–254.

[77] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

[78] Richard A Klein, Kate A Ratliff, Michelangelo Vianello, Reginald B Adams Jr, Štěpán Bahník, Michael J Bernstein, Konrad Bocian, Mark J Brandt, Beach Brooks, Claudia Chloe Brumbaugh, et al. 2014. Investigating variation in replicability. *Social psychology*.

[79] Richard A Klein, Kate A Ratliff, Michelangelo Vianello, Reginald B Adams Jr, Štěpán Bahník, Michael J Bernstein, Konrad Bocian, Mark J Brandt, Beach Brooks, Claudia Chloe Brumbaugh, et al. 2014. Theory building through replication: Response to commentaries on the “many labs” replication project.

[80] Richard A Klein, Michelangelo Vianello, Fred Hasselman, Byron G Adams, Reginald B Adams Jr, Sinan Alper, Mark Aveyard, Jordan R Axt, Mayowa T Babalola, Štěpán Bahník,

et al. 2018. Many labs 2: Investigating variation in replicability across samples and settings. *Advances in Methods and Practices in Psychological Science*, 1(4):443–490.

[81] Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pages 1885–1894. PMLR.

[82] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. 2019. Text classification algorithms: A survey. *Information*, 10(4):150.

[83] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. Technical report, Citeseer.

[84] Preethi Lahoti, Alex Beutel, Jilin Chen, Kang Lee, Flavien Prost, Nithum Thain, Xuezhi Wang, and Ed H Chi. 2020. Fairness without demographics through adversarially reweighted learning. *arXiv preprint arXiv:2006.13114*.

[85] Samuli Laine and Timo Aila. 2016. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*.

[86] Dong-Hyun Lee et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896.

[87] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al.

2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195.

[88] Junnan Li, Richard Socher, and Steven CH Hoi. 2020. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*.

[89] Qiang Liu, Jian Peng, and Alexander T Ihler. 2012. Variational inference for crowdsourcing. In *Advances in neural information processing systems*, pages 692–700.

[90] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. 2020. Early-learning regularization prevents memorization of noisy labels. *Advances in neural information processing systems*, 33:20331–20342.

[91] Tongliang Liu and Dacheng Tao. 2015. Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3):447–461.

[92] Xialei Liu, Joost Van De Weijer, and Andrew D Bagdanov. 2019. Exploiting unlabeled data in cnns by self-supervised learning to rank. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1862–1878.

[93] Xin Liu, Baosong Yang, Dayiheng Liu, Haibo Zhang, Weihua Luo, Min Zhang, Haiying Zhang, and Jinsong Su. 2021. Bridging subword gaps in pretrain-finetune paradigm for natural language generation. *arXiv preprint arXiv:2106.06125*.

[94] Yang Liu. 2021. Understanding instance-level label noise: Disparate impacts and treatments. In *International Conference on Machine Learning*, pages 6725–6735. PMLR.

[95] Yang Liu. 2022. Identifiability of label noise transition matrix. *arXiv e-prints*, pages arXiv–2202.

[96] Yang Liu, Michael Gordon, Juntao Wang, Michael Bishop, Yiling Chen, Thomas Pfeiffer, Charles Twardy, and Domenico Viganola. 2020. Replication markets: Results, lessons, challenges and opportunities in ai replication. *arXiv preprint arXiv:2005.04543*.

[97] Yang Liu and Hongyi Guo. 2020. Peer loss functions: Learning from noisy labels without knowing noise rates. *International Conference on Machine Learning*.

[98] Yang Liu and Jialu Wang. 2021. Can less be more? when increasing-to-balancing label noise rates considered beneficial. *Advances in Neural Information Processing Systems*, 34.

[99] Yang Liu, Juntao Wang, and Yiling Chen. 2020. Surrogate scoring rules. In *Proceedings of the 21st ACM Conference on Economics and Computation*, pages 853–871.

[100] Huixiang Luo, Hao Cheng, Yuting Gao, Ke Li, Mengdan Zhang, Fanxu Meng, Xiaowei Guo, Feiyue Huang, and Xing Sun. 2021. On the consistency training for open-set semi-supervised learning. *arXiv preprint arXiv:2101.08237*.

[101] Tianyi Luo, Xingyu Li, Hainan Wang, and Yang Liu. 2020. Research replication prediction using weakly supervised learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 1464–1474.

[102] Tianyi Luo, Rui Meng, Xin Eric Wang, and Yang Liu. 2022. Interpretable research

replication prediction via variational contextual consistency sentence masking. *arXiv preprint arXiv:2203.14474*.

[103] Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

[104] David M Markowitz, Hyunjin Song, and Samuel Hardman Taylor. 2021. Tracing the adoption and effects of open science in communication research. *Journal of Communication*.

[105] Natalia Martinez, Martin Bertran, and Guillermo Sapiro. 2020. Minimax pareto fairness: A multi objective perspective. In *International Conference on Machine Learning*, pages 6755–6764. PMLR.

[106] Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2018. Weakly-supervised neural text classification. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 983–992.

[107] Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. 2021. Deep learning–based text classification: a comprehensive review. *ACM Computing Surveys (CSUR)*, 54(3):1–40.

[108] Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2016. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*.

[109] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. 2018. Virtual

adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993.

[110] M Granger Morgan. 2014. Use (and abuse) of expert elicitation in support of decision making for public policy. *Proceedings of the National Academy of Sciences*, 111(20):7176–7184.

[111] W James Murdoch, Peter J Liu, and Bin Yu. 2018. Beyond word importance: Contextual decomposition to extract interactions from lstms. *arXiv preprint arXiv:1801.05453*.

[112] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. 2013. Learning with noisy labels. In *Advances in neural information processing systems*, pages 1196–1204.

[113] Dong Nguyen. 2018. Comparing automatic and human evaluation of local explanations for text classification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1069–1078.

[114] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. 2018. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems*, pages 3235–3246.

[115] H Pashler, B Spellman, S Kang, and A Holcombe. 2019. Psychfiledrawer: archive of replication attempts in experimental psychology. *Online< http://psychfiledrawer. org/view_ article_list. php*.

[116] Chao-Ying Joanne Peng, Kuk Lida Lee, and Gary M Ingersoll. 2002. An introduction to logistic regression analysis and reporting. *The journal of educational research*, 96(1):3–14.

[117] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

[118] D. Prelec. 2004. A bayesian truth serum for subjective data. *Science*, 306(5695):462–466.

[119] Dražen Prelec, H Sebastian Seung, and John McCoy. 2017. A solution to the single-question crowd wisdom problem. *Nature*, 541(7638):532.

[120] Reilly Raab and Yang Liu. 2021. Unintended selection: Persistent qualification rate disparities and interventions. *Advances in Neural Information Processing Systems*, 34.

[121] Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

[122] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

[123] Antti Rasmus, Harri Valpola, Mikko Honkala, Mathias Berglund, and Tapani Raiko. 2015. Semi-supervised learning with ladder networks. *arXiv preprint arXiv:1507.02672*.

[124] Vikas C Raykar, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. 2010. Learning from crowds. *Journal of Machine Learning Research*, 11(Apr):1297–1322.

[125] Frank Rosenblatt. 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

[126] Devendra Singh Sachan, Manzil Zaheer, and Ruslan Salakhutdinov. 2019. Revisiting lstm networks for semi-supervised text classification via mixed objective function. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6940–6948.

[127] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. 2016. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Advances in neural information processing systems*, 29:1163–1171.

[128] Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. 2016. Evaluating the visualization of what a deep neural network has learned. *IEEE transactions on neural networks and learning systems*, 28(11):2660–2673.

[129] Sowmya Sanagavarapu, Sashank Sridhar, and S Chitrakala. 2021. News categorization using hybrid bilstm-ann model with feature engineering. In *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 0134–0140. IEEE.

[130] Clayton Scott. 2015. A rate of convergence for mixture proportion estimation, with application to learning from noisy labels. In *Artificial Intelligence and Statistics*, pages 838–846.

[131] Clayton Scott, Gilles Blanchard, and Gregory Handy. 2013. Classification with asymmetric label noise: Consistency and maximal denoising. In *Conference On Learning Theory*, pages 489–511.

[132] Sofia Serrano and Noah A Smith. 2019. Is attention interpretable? *arXiv preprint arXiv:1906.03731*.

[133] Yusuke Shinyama. 2014. Pdfminer.

[134] Joseph P Simmons, Leif D Nelson, Jeff Galak, and Shane Frederick. 2010. Intuitive biases in choice versus estimation: Implications for the wisdom of crowds. *Journal of Consumer Research*, 38(1):1–15.

[135] Daniel J Simons, Alex O Holcombe, and Barbara A Spellman. 2014. An introduction to registered replication reports at perspectives on psychological science. *Perspectives on Psychological Science*, 9(5):552–555.

[136] Uri Simonsohn, Joseph P Simmons, and Leif D Nelson. 2015. Better p-curves: Making p-curve analysis more robust to errors, fraud, and ambitious p-hacking, a reply to ulrich and miller (2015).

[137] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *In Workshop at International Conference on Learning Representations*. Citeseer.

[138] Chandan Singh, W James Murdoch, and Bin Yu. 2018. Hierarchical interpretations for neural network predictions. *arXiv preprint arXiv:1806.05337*.

[139] Push Singh, Thomas Lin, Erik T Mueller, Grace Lim, Travell Perkins, and Wan Li Zhu. 2002. Open mind common sense: Knowledge acquisition from the general public. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 1223–1237. Springer.

[140] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. 2017. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*.

[141] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*.

[142] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer.

[143] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR.

[144] James Surowiecki. 2005. *The wisdom of crowds*. Anchor.

[145] Aiham Taleb, Winfried Loetzsch, Noel Danz, Julius Severin, Thomas Gaertner, Benjamin Bergner, and Christoph Lippert. 2020. 3d self-supervised methods for medical imaging. *Advances in Neural Information Processing Systems*, 33:18158–18172.

[146] Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-

averaged consistency targets improve semi-supervised deep learning results. *arXiv preprint arXiv:1703.01780*.

[147] Philip E Tetlock and Dan Gardner. 2016. *Superforecasting: The art and science of prediction*. Random House.

[148] Naftali Tishby, Fernando C Pereira, and William Bialek. 2000. The information bottleneck method. *arXiv preprint physics/0004057*.

[149] Brendan Van Rooyen, Aditya Menon, and Robert C Williamson. 2015. Learning with symmetric label noise: The importance of being unhinged. In *Advances in Neural Information Processing Systems*, pages 10–18.

[150] Sahil Verma and Julia Rubin. 2018. Fairness definitions explained. In *2018 ieee/acm international workshop on software fairness (fairware)*, pages 1–7. IEEE.

[151] Jialu Wang, Yang Liu, and Caleb Levy. 2021. Fair classification with group-dependent label noise. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 526–536.

[152] Jialu Wang, Yang Liu, and Xin Eric Wang. 2021. Are gender-neutral queries really gender-neutral? mitigating gender bias in image search. *arXiv preprint arXiv:2109.05433*.

[153] Jialu Wang, Yang Liu, and Xin Eric Wang. 2022. Assessing multilingual fairness in pre-trained multimodal representations. In *the Findings of ACL 2022*, Dublin, Ireland. Association for Computational Linguistics.

[154] Qixin Wang, Tianyi Luo, and Dong Wang. 2016. Can machine generate traditional chinese poetry? a feigenbaum test. In *International Conference on Brain Inspired Cognitive Systems*, pages 34–46. Springer.

[155] Qixin Wang, Tianyi Luo, Dong Wang, and Chao Xing. 2016. Chinese song iambics generation with neural attention-based model. *arXiv preprint arXiv:1604.06274*.

[156] Xiang Wang, Shiwei Zhang, Zhiwu Qing, Yuanjie Shao, Changxin Gao, and Nong Sang. 2021. Self-supervised learning for semi-supervised temporal action proposal. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1905–1914.

[157] Zhuowei Wang, Jing Jiang, Bo Han, Lei Feng, Bo An, Gang Niu, and Guodong Long. 2020. Seminll: A framework of noisy-label learning by semi-supervised learning. *arXiv preprint arXiv:2012.00925*.

[158] Jiaheng Wei and Yang Liu. 2021. When optimizing \$f\$-divergence is robust with label noise. In *International Conference on Learning Representations*.

[159] Jiaheng Wei, Zhaowei Zhu, Hao Cheng, Tongliang Liu, Gang Niu, and Yang Liu. 2021. Learning with noisy labels revisited: A study using real-world human annotations. *arXiv preprint arXiv:2110.12088*.

[160] Jiaheng Wei, Zhaowei Zhu, Hao Cheng, Tongliang Liu, Gang Niu, and Yang Liu. 2022. Learning with noisy labels revisited: A study using real-world human annotations. In *International Conference on Learning Representations*.

[161] Justin Wolfers and Eric Zitzewitz. 2006. Prediction markets in theory and practice. Technical report, national bureau of economic research.

[162] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. 2019. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*.

[163] Fei Xu and Joshua B Tenenbaum. 2007. Word learning as bayesian inference. *Psychological review*, 114(2):245.

[164] Yi Xu, Lei Shang, Jinxing Ye, Qi Qian, Yu-Feng Li, Baigui Sun, Hao Li, and Rong Jin. 2021. Dash: Semi-supervised learning with dynamic thresholding. In *International Conference on Machine Learning*, pages 11525–11536. PMLR.

[165] Diyi Yang, Jiaao Chen, Zichao Yang, Dan Jurafsky, and Eduard Hovy. 2019. Let's make your request more persuasive: Modeling persuasive strategies via semi-supervised neural nets on crowdfunding platforms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3620–3630.

[166] Diyi Yang, Miaomiao Wen, and Carolyn Rose. 2015. Weakly supervised role identification in teamwork interactions. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1671–1680.

[167] Yang Yang. 2018. The replicability of scientific findings using human and ma-

chine intelligence. https://www.metascience2019.org/presentations/yang-yang/ Metascience 2019.

[168] Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. 2017. Improved variational autoencoders for text modeling using dilated convolutions. In *International conference on machine learning*, pages 3881–3890. PMLR.

[169] Yao Yao, Jiehui Deng, Xiuhua Chen, Chen Gong, Jianxin Wu, and Jian Yang. 2020. Deep discriminative cnn with temporal ensembling for ambiguously-labeled image classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12669–12676.

[170] Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. *arXiv preprint arXiv:1909.00161*.

[171] Xin Yuan, Zhe Lin, Jason Kuen, Jianming Zhang, Yilin Wang, Michael Maire, Ajinkya Kale, and Baldo Faieta. 2021. Multimodal contrastive training for visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6995–7004.

[172] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. 2017. Fairness constraints: Mechanisms for fair classification. In *Artificial Intelligence and Statistics*, pages 962–970. PMLR.

[173] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup:

Beyond empirical risk minimization. In *International Conference on Learning Representations*.

[174] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28:649–657.

[175] Yuchen Zhang, Xi Chen, Dengyong Zhou, and Michael I Jordan. 2014. Spectral methods meet em: A provably optimal algorithm for crowdsourcing. In *Advances in neural information processing systems*, pages 1260–1268.

[176] Dengyong Zhou, Sumit Basu, Yi Mao, and John C Platt. 2012. Learning from the wisdom of crowds by minimax entropy. In *Advances in neural information processing systems*, pages 2195–2203.

[177] Dengyong Zhou, Qiang Liu, John Platt, and Christopher Meek. 2014. Aggregating ordinal labels from crowds by minimax conditional entropy. In *International conference on machine learning*, pages 262–270.

[178] Zhi-Hua Zhou. 2018. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53.

[179] Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. 2021. Retrieving and reading: A comprehensive survey on open-domain question answering. *arXiv preprint arXiv:2101.00774*.

[180] Junnan Zhu, Haoran Li, Tianshang Liu, Yu Zhou, Jiajun Zhang, and Chengqing Zong.

2018. Msmo: Multimodal summarization with multimodal output. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 4154–4164.

[181] Xiaojin Zhu, Zoubin Ghahramani, and John D Lafferty. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International conference on Machine learning (ICML-03)*, pages 912–919.

[182] Zhaowei Zhu, Zihao Dong, and Yang Liu. 2022. Detecting corrupted labels without training a model to predict.

[183] Zhaowei Zhu, Tongliang Liu, and Yang Liu. 2021. A second-order approach to learning with instance-dependent label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10113–10123.

[184] Zhaowei Zhu, Tianyi Luo, and Yang Liu. 2021. The rich get richer: Disparate impact of semi-supervised learning. *arXiv preprint arXiv:2110.06282*.

[185] Zhaowei Zhu, Yiwen Song, and Yang Liu. 2021. Clusterability as an alternative to anchor points when learning with noisy labels. *arXiv preprint arXiv:2102.05291*.

[186] Zhaowei Zhu, Jialu Wang, and Yang Liu. 2022. Beyond images: Label noise transition matrix estimation for tasks with lower-quality features. *arXiv preprint arXiv:2202.01273*.