Distribution Agreement

In presenting this thesis or dissertation as a partial fulfillment of the requirements for an advanced degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis or dissertation in whole or in part in all forms of media, now or hereafter known, including display on the world wide web. I understand that I may select some access restrictions as part of the online submission of this thesis or dissertation. I retain all ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Signature:

_____ _____

Payam Karisani Date

Mining User Generated Content: Addressing Data Scarcity in Filtering Tasks

By

Payam Karisani
Doctor of Philosophy

Computer Science and Informatics

Li Xiong, Ph.D.
Supervisor

Phillip M. Wolff, Ph.D.
Committee Member

Carl Yang, Ph.D.
Committee Member

Liang Zhao, Ph.D.
Committee Member

Accepted:

Kimberly J. Arriola, Ph.D.
Dean of the James T. Laney School of Graduate Studies

Date

Mining User Generated Content: Addressing Data Scarcity in Filtering Tasks

By

Payam Karisani

Supervisor: Li Xiong, Ph.D.

Computer Science Department
James T. Laney School of Graduate Studies of Emory University
2021

Abstract

Filtering tasks have a broad range of applications in mining user-generated data. Examples include public health monitoring, product monitoring, user satisfaction analysis, crisis management, and hate speech detection. This dissertation proposes methods and techniques to overcome one of the primary challenges of these tasks, i.e., the lack of enough training data. It has four main contributions.

First, it employs semi-supervised learning and proposes a novel method based on self-training and pseudo labeling to use unlabeled data. Our model uses the pretraining-finetuning paradigm in a semi-supervised setting to use unlabeled data for model initialization. It also employs a novel learning rate schedule to exploit noisy pseudo-labels as a means to explore the loss surface. We empirically demonstrate the efficacy of these strategies.

Second, it exploits unlabeled documents in a multi-view model. We propose a novel algorithm for one of the most challenging filtering tasks in social media, i.e., the adverse drug reaction monitoring task. Here, we propose a pair of loss functions to pretrain and then finetune the classifier in each view by the pseudo-labels obtained in the other view. Therefore, we effectively transfer the knowledge obtained in one view to the classifier in the other view. We empirically demonstrate that this model is the first known algorithm that outperforms the multi-layer transformer models pretrained on domain specific data.

Third, it proposes a novel active learning model when additional labels can be obtained for a range of tasks. Specifically, we use a multi-view model to extract two views from documents, and then, we propose a novel acquisition function to aggregate the informativeness and the representativeness metrics for querying additional labels. We analytically argue that our acquisition function incorporates document contexts into the active learning query process. We also treat the highly informal language of users in social media as a factor that manifests itself in the output of learners and causes a high variance. Therefore, we employ a query-by-committee model as a variance reduction technique to combat this undesired effect. Our experiments show that our model significantly outperforms existing models.

Finally, we observe that although in many cases labeled data is not available, annotated data for semantically similar tasks is available. Motivated by this, we formulate a new problem and propose an algorithm for single-source domain adaptation. We assume that in addition to the source and target data, we can access a set of unlabeled auxiliary domains. We empirically show that existing state-of-the-art models are unable to effectively use this type of data. We then propose a novel algorithm based on the uncertainty in output predictions to decompose the target data into two sets. Then, we show that training using the set of confidently labeled target documents along the auxiliary unlabeled data yields a classifier that is highly effective in the regions close to the classification decision boundaries. The experiments testify that our algorithm outperforms the state-of-the-art in this new problem setting.

Mining User Generated Content: Addressing Data Scarcity in Filtering Tasks

By

Payam Karisani
B.Sc., Iran University of Science and Technology (IUST), Tehran, Iran, 2008
M.Sc., University of Tehran (UT), Tehran, Iran, 2011

Supervisor: Li Xiong, Ph.D.

A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Computer Science and Informatics
2021

Contents

1 Introduction **1**

1.1 Applications of User-Generated Content 2

1.2 Existing Challenges for Filtering Information 3

1.3 Thesis Scope 5

1.4 Overview of Existing Studies 6

1.5 Contributions 7

1.6 Broader Impact 10

2 Semi-Supervised Learning **12**

2.1 Introduction 12

2.2 Related Work 14

2.3 Semi-Supervised Learning via Self-Pretraining 17

2.3.1 Hypothesis Transfer and Iterative Distillation 19

2.3.2 Two-Stage Semi-Supervised Learning 20

2.3.3 Right Trapezoidal Learning Rates 22

2.3.4 Inertial Class Distributions 23

2.4 Experimental Setup 26

2.4.1 Datasets 26

2.4.2 Baselines 27

2.4.3 Experimental Details 29

2.5 Results and Discussion 30
2.5.1 Main Results 30
2.5.2 Empirical Analysis 32
2.6 Conclusions 36

3 Multi-View Learning **37**
3.1 Introduction 37
3.2 Related Work 38
3.3 Proposed Method 39
3.4 Experimental Setup 42
3.5 Results and Analysis 44
3.6 Conclusions 45

4 Active Learning **47**
4.1 Introduction 47
4.2 Background and Related Work 49
4.3 COCOBA : Model Description 51
4.3.1 Extracting Two Contextual Representations from User Postings 51
4.3.2 Incorporating Context in Co-testing 52
4.3.3 Increasing Robustness to Noise 54
4.3.4 Overview of Algorithm 56
4.4 COCOBA : Implementation Details 56
4.5 Experimental Setup 58
4.5.1 Datasets 58
4.5.2 Baselines 60
4.5.3 Experimental Details 61
4.6 Results and Analysis 61
4.6.1 Results 61

4.6.2 Empirical Analysis 63
4.7 Conclusions 66
4.8 Ethical Considerations 66

5 Domain Adaptation **67**
5.1 Introduction 67
5.2 Related Work 69
5.3 Proposed Model 70
5.4 Experiments 73
5.4.1 Setup 73
5.4.2 Main Results 74
5.4.3 Expectations 74
5.4.4 Empirical Analysis 75
5.5 Conclusions 77

6 Conclusions and Future Work **78**

Bibliography **80**

List of Figures

2.1 The Self-Pretraining learning rate schedule. The dashed horizontal line is the learning rate of the network during the training by the pseudo-labels, and the slanted line is the learning rate during the training by the labeled documents. 24

2.2 F1 of the resulting classifier in every iteration of Self-Pretraining with different values of k–the number of randomly selected pseudo-labels. The middle values are interpolated. The results are in the test set of ADR dataset. 33

3.1 The illustration of the document and drug views in our model. We have used BERT as an encoder. See Devlin et al. [33] for the format of input tokens. 40

4.1 The document and word level views in COCOBA query strategy. The regular co-testing algorithm queries the contention point with the largest distance from the classifier decision boundary in two views (the yellow triangle). COCOBA queries the contention point which is closest to the set of other contention points and also has a large distance from the decision boundary in two views (the black triangle). Figure best viewed in color. 54

4.2 F1 of the models at varying training set sizes during the active learning iterations in Illness dataset. 62
4.3 F1 of the models at varying training set sizes during the active learning iterations in Observation dataset. 62
4.4 F1 of the models at varying training set sizes during the active learning iterations in Product dataset. 62
4.5 Ablation study of COCOBA by deactivating the two modules of our method, i.e., COBA (COCOBA without context) and COCO (COCOBA with no bagging) at varying training set sizes in Illness dataset. 65

List of Tables

2.1 Summary of ADR , Earthquake , and Product datasets. 27

2.2 F1, precision, and recall of Self-Pretraining in ADR , Earthquake , and Product datasets compared to the baseline models. The models were trained on 300 and 500 labeled user postings. 31

2.3 Results of Self-Pretraining with different values of k–the number of randomly selected pseudo-labels–in the test set of ADR dataset. The models began with 500 labeled user postings. 32

2.4 Results of Self-Pretraining in the test set of ADR dataset at varying values of the temperature (T) for iterative distillation. 33

2.5 Results of Self-Pretraining in the test set of ADR dataset at varying values of the hyper-parameter (λ) for our two-stage learning–see Equation 2.2. 34

2.6 Results of Self-Pretraining in the test set of ADR dataset at varying values of the hyper-parameter (α) for the inertial transformation of the pseudo-labels–see Equation 2.3. 35

2.7 Results of Self-Pretraining in the test set of ADR dataset after deactivating the distillation (Section 2.3.1), the two-stage learning (Section 2.3.2), the trapezoidal learning rate (Section 2.3.3), and the inertial transformation (Section 2.3.4). 35

3.1 F1, Precision, and Recall of our model (VID) in comparison with the baselines. 44
3.2 F1, Precision, and Recall of VID in comparison to the performance of the classifiers trained on the document, drug, and combined views. . . 45
3.3 Performance of VID in comparison to the performance of the classifiers pretrained on the document or drug pseudo-labels (indicated by P-{•}) and finetuned on the document or drug training examples (indicated by F-{•}). 45

4.1 The number of tweets, and the percentage of the positive and negative tweets across the topics in Illness dataset. 59
4.2 F1 of the models at 25%, 50%, 75%, and 100% of the training set sizes during the active learning iterations. The improvements indicated by * are statistically significant–using paired t-test (adjusted $P < 0.05$). 64

5.1 Average results for single source unsupervised domain adaptation in the presence of unlabeled data from multiple source domains. 74
5.2 Performance at varying number of available auxiliary domains. 76
5.3 Performance at varying percentage of available documents in every auxiliary domain. 76
5.4 Performance of individual classifiers compared to DAVUD . This experiment is equivalent to an ablation study. 76

List of Algorithms

1 Overview of Vanilla Self-Pretraining 18
2 Overview of VID 42
3 One Iteration of COCOBA 57
4 Overview of DAVUD 72

Chapter 1

Introduction

Users are the key factors on the Internet and their communication is an asset. Interaction between users produces data, and this has created a whole new ecosystem of products and services. The data generated by the users is typically public, it usually requires a certain amount of effort to produce, and it is often created outside of their profession [81]. This data is denoted as user-generated content (UGC) [81]. Blog posts, forums, tweets, podcasts, and user images are categorized as user-generated data. In this work, we focus on textual UGC (particularly the data on **microblogs**, i.e., Twitter, Tumblr, or Facebook); we propose novel methods and techniques to address the challenges in processing this data. We specifically employ Semi-Supervised Learning, Active Learning, Multi-View Learning, and Domain Adaptation to overcome the data **scarcity challenge** in classification tasks for information filtering [105].

In this chapter, we first provide a summary of the applications of UGC (Section 1.1), then, we discuss the technical challenges in mining this data (Section 1.2). Then, we provide the scope of this dissertation and discuss the primary challenges that we tackle (Section 1.3); after that, we provide a brief summary of existing studies in this area (Section 1.4). Then, we present an overview of the contributions of this

dissertation (Section 1.5), and finally, we discuss the potentially broader impact of our research in other areas (Section 1.6).

1.1 Applications of User-Generated Content

The large amount of user-generated data publicly available on the websites such as Twitter or Facebook tremendously benefits [35] the industry, public healthcare, and public safety. Below we briefly discuss the existing applications in these areas.

Businesses and corporations can integrate social media data into their models to improve their brand awareness, customer service, and customer engagement. Examples of the real world applications in this area are recommender systems, reputation management, and online marketing. Recommender systems are perhaps the most prominent applications, where the websites analyze user profiles and based on the user traits direct their advertisements.

Thanks to social media, news agencies can communicate with their audiences faster and easier than before. In the past, these institutions would use traditional surveys and polling methods to collect information, however, with the large amount of data available online, this procedure is almost entirely transformed. Major applications in this area include public opinion mining, sentiment analysis, or crisis monitoring.

Public health is another area that has benefited from user-generated data. Public health surveillance[1] involves "continuous, systematic collection, analysis and interpretation of health data". Traditionally, this procedure is performed either through surveys or monitoring clinical visits [86]. Surveys for a long time has been the mainstream means of collecting data. Phone calls, in-person forms, or online web-pages are the methods of conducting surveys. Clinical visits have been also exploited, however, this method requires a significant coordination between hospitals and clinics.

[1] https://www.emro.who.int/health-topics/public-health-surveillance/index.html

Even though the traditional ways of collecting health data are reliable, there are still disadvantages in employing these methods. Biases in the population that still use land-line phones, the high cost of in-person forms, and the slow process of collecting data from the clinical visits are the major drawbacks. These challenges in using the traditional methods, have created opportunities to exploit the user-generated data available on social media. As opposed to the traditional methods, this data is cheap and easy to obtain. Additionally, in the cases that timely response matters, i.e., the early detection of outbreaks, the user-generated data is a valuable resource.

Political institutions and organizations can exploit the social media data to monitor the public opinion about politicians and their parties. This data can help political campaigns to detect the public needs and develop strategies to gain more followers. Applications such as opinion mining and hate speech detection can aid political parties to become familiar with the reality of the society and enhance their position compared to their rivals. Intelligence agencies can use the social media data to glean information about individuals or groups that can potentially pose threats to the national security. This data can be utilized to alert the law enforcement agencies in a timely manner to prevent crimes. The information required for these purposes is typically buried under large volume data and sophisticated multi-modal applications are developed to address the needs.

1.2 Existing Challenges for Filtering Information

Even though user-generated data is a valuable resource, extracting information from this data is challenging. The major challenges include: biases [84], which mainly question the reliability of the resources; low data quality [4], which is due to the used inventive lexicons and the typically short length of messages; and the lack of sufficient training data and imbalanced class distributions [58], which demand designing ma-

chine learning models with specific considerations.[2] Below we discuss each challenge in more detail.[3]

The data collected from social media websites exhibits numerous biases [84], therefore, the conclusions drawn from this data is subsequently affected. This data is subject to the population bias, i.e., there might be systematic discrepancies between the population of users that form the dataset and the target population. There might be also behavioral biases in the collected data, i.e., there might be systematic discrepancies between the user traits in the dataset and the target user traits. The behavioral biases can be also reflected in the content generated by users. For instance, user postings may be drastically different in their lexicons and semantics across two platforms. There can be also temporal biases in the data, i.e., the content may dramatically vary over time. The redundancy in the data is also problematic, because users can re-publish exiting content and this may impact the underlying distribution of the data.

Another major problem is the typically low quality of data [4]. User postings are generally written in informal language. Non-standard spellings are commonly used. Punctuation is often absent, and capitalization is rarely practiced. These factors pose significant challenges to natural language processing tools. Additionally, the length of documents in the social media websites is usually short, either due to the website restrictions or due to user preferences. This results in sparse representations for documents. This sparsity along the discussed noisy content are the bottlenecks in effectively processing this data.

Finally, existing machine learning models typically rely on large amount of labeled training data. Obtaining this data is often time consuming and expensive. Addition-

[2]In some cases, e.g., sarcasm detection [11], there are creative ways to automatically construct big datasets –e.g., using hashtags, but such techniques do not always apply. For example, we have observed that generally hashtags are poor indicators of personal health reports.

[3]There are also privacy challenges that we do not discuss here. In this work, we assume that the users have made their data publicly available. We also assume that no information about the users is stored or inferred.

ally, in many applications the class distributions are highly imbalanced, therefore, to have a good representation for each category, even more data should be collected and labeled. An example is the disease mining domain, where on average, only 20% of collected user postings are true reports of health conditions [58]. The above challenges substantially hinder the progress in this area.

1.3 Thesis Scope

In this work, we are not concerned with a particular application or domain, instead, we mainly focus on a general class of problems for information filtering [105]. These are typically classification tasks defined by certain criteria to filter out irrelevant user postings and to collect desired ones for downstream applications. Examples include adverse drug reaction mining [122], online crisis management [53], disease mining [8], and mining user sentiment on products [52]. We take the online crisis management task as a more specific example. In this task, given a real world incident, e.g., an earthquake, the aim is to detect user postings that report any specific information about the incident. User postings such as *"Million flee homes on Chile coast after earthquake triggers tsunami"* or *"8 dead in Mount Everest avalanche after Nepal quake"* are labeled relevant, on the other hand, user postings such as *"Nepal Earthquake Spurs Fashion Designer"* or *"There are over 1000 buildings that are not quake reinforced"* are irrelevant and should be filtered out.

We particularly propose methods and techniques to address the data scarcity problem and to enhance model performance in the settings that labeled data is limited or non-existent. As the primary methods to increase model generalization, in Chapter 2 we focus on Semi-Supervised Learning and in Chapter 3 we focus on Multi-View Learning. Both of these techniques are well-known tools for enhancing model performance in low-resource settings. As the auxiliary methods, in Chapter 4, we focus on

Active Learning and in Chapter 5, we focus on Domain Adaptation. Both of these techniques are currently trending topics in the Web and NLP related venues.

1.4 Overview of Existing Studies

Existing studies for filtering information in user-generated data employ a wide range of tools and techniques. Employing domain knowledge [66], using document context [41], employing Semi-Supervised Learning [7], incorporating Active Learning [89], using Domain Adaptation [5], using model pretraining [114], and multi-modal learning [54] are examples of such studies. In addition to these studies, there are also works that explore hybrid models that integrate multiple techniques into a single algorithm. For instance, Cui et al. [31] incorporate Active Learning into a semi-supervised model, and Burkhardt et al. [19] aggregate Active Learning and Crowd Sourcing.

In this work, we focus on addressing the lack of labeled training data in filtering tasks. A methodical approach to address the data scarcity problem is to develop models that generalize better. Traditionally, such models either exploit unlabeled data [24] or use inductive bias [80]. There are also auxiliary techniques, such as Active Learning [99] and Domain Adaptation [12], that provide learning algorithms with additional signals from external resources. In each chapter, we provide a separate overview of related studies. However, below we focus on the common practices in each research area that we seek to modify and improve.

Self-training as one of the primary methods in Semi-Supervised Learning suffers from numerous problems, including the reliance on a confidence threshold and the deterioration in performance as noisy pseudo-labels are added to the training pool. Existing models based on self-training [7, 61] inherit these problems. In Chapter 2 we introduce a variant of self-training to address these drawbacks.

Documents in social media are short, and researcher traditionally use single view

models to process them [114, 40]. In Chapter 3, we investigate Multi-View Learning, and propose a variant of co-training to transfer the knowledge from the classifier in one view to the classifier in the other view using unlabeled data. We empirically show that this algorithm enhances model performance in highly imbalanced settings.

The majority of existing active learning models for filtering data in social media [106, 19] rely on the uncertainty-based model [70]. In Chapter 4, we propose an algorithm that uses a multi-view model and incorporates the distribution of documents into its query strategy to effectively use document contexts.

Finally, existing domain adaptation models primarily focus on exploiting labeled data from source domains [74, 107]. There is no study to investigate the application of unlabeled source data in the adaptation process. In Chapter 5, we present a model to incorporate this data into the process.

Apart from the aforementioned categorization of existing studies, there are also studies that explicitly explore the efficacy of hybrid models. For instance, Hajmohammadi et al. [44] aggregate Semi-Supervised Learning and Active Learning, and Rai et al. [92] aggregate Domain Adaptation and Active Learning. Here, we focus on each technique individually.

1.5 Contributions

The contributions of this work are mainly made in social media filtering tasks under low-resource settings. The efficacy of the techniques proposed here, are demonstrated in numerous publicly available benchmarks. The contributions are made in four chapters. In Chapter 2, we propose a novel semi-supervised learning model based on self-training to use unlabeled data.[4] In this chapter, our main contributions are as follows:

[4]The work in this chapter was accepted as a full paper to WSDM 2021 [59].

- Our model uses the self-training paradigm, however, it does not suffer from the problems specific to self-training–i.e., the semantic drift problem [32] or the reliance on a confidence threshold [93].[5] To our knowledge, our method is the first model that addresses these problems in a unified framework.

- We propose a novel learning rate schedule to use unlabeled data as a means for exploration in the optimization process.

- To reduce the noise in pseudo-labels, we model the class distribution of pseudo-labels as a stochastic process across the bootstrapping iterations, and propose a novel approach to transform the class distributions.

- We carry out experiments across multiple public Twitter datasets and show that our model outperforms existing baselines.

In Chapter 3, we propose a new variant of the co-training algorithm for the notoriously difficult task of classifying adverse drug effect reports on Twitter.[6] Here, the core idea is to use the knowledge obtained from one view to initialize the base learner in the other view. In this chapter our main contributions are as follows:

- We exploit unlabeled data to transfer knowledge across models in multi-view settings, and propose a new variant of co-training that is robust against the semantic drift problem.

- We evaluate our model in the largest publicly available ADR dataset, and show that it yields an additive improvement to the common practice of language model pretraining in this task. To our knowledge, our work is the first study that reports such an achievement.

[5] Semantic drift is the deterioration in model performance as the self-training iterations are carried out. See Chapter 2 for more information in this regard.

[6] The work in this chapter was accepted as a workshop paper to SMM4H workshop co-located with NAACL 2021 [62].

In Chapter 4, we propose a novel active learning model. Our main contributions in this chapter are as follows:

- Regular active learning models for classifying user-generated data use single-view algorithms. As opposed to the current trend, we propose a novel unified multi-view model to address the tasks tailored for query words.
- We carry out an extensive set of experiments and show that our model is applicable to at least three representative tasks.
- We show that our model consistently outperforms existing active learning models.
- We constructed a relatively large dataset of manually annotated tweets for PHM task that is publicly available. Our dataset consists of 18,000 tweets across three topics: Parkinson's, cancer, and diabetes.

Finally, in Chapter 5, we propose a new problem setting in Domain Adaptation, and then, we propose a novel model for this task.[7] The main contributions in this chapter are:

- We present a new research problem. Our new problem is an alternative view to the multi-target domain adaptation [28] and is an extension to the unsupervised single-source domain adaptation [74]. Here, we assume that in addition to the labeled source domain and the unlabeled target domain, we also have a set of unlabeled auxiliary domains. Our research problem can potentially lead to enhancing the performance of existing domain adaptation models.
- We propose a novel model based on the concept of the noisy regions. Noisy regions are the areas that a base learner is uncertain about.

[7] The work in this chapter is ready to submit.

- We empirically show that our model outperforms existing models in this task.

To summarize, in this dissertation we explore four directions to address data scarcity and to enhance model performance when there is small or no labeled data available for filtering information in user-generated data. We explore Semi-Supervised Learning, Multi-View Learning, Active Learning, and Domain Adaptation.

In practice, and in developing real-world systems, these techniques can be integrated and used simultaneously. In fact, in each one of these areas we present a novel model that uses the tools and the ideas proposed in the other areas. For instance, in Chapter 3 to develop our multi-view model, we use self-training and unlabeled data. In Chapter 4, to develop our active learning model we use a multi-view model, and in Chapter 5, to develop our domain adaptation model we use unlabeled data from semantically related domains.

1.6 Broader Impact

We highlighted the importance and the applications of user-generated data in Section 1.1. Due to the wide range of applications that user-generated data has, in this dissertation we focus on this type of data. Nonetheless, in certain cases we solely use general properties of learning algorithms and do not restrict ourselves to user-generated data, therefore, we expect that our models *potentially* generalize to other types of data, e.g., images. Future work may explore this direction.

Our semi-supervised learning model, proposed in Chapter 2, exploits unlabeled data in a self-training framework. The self-training algorithm is widely used in the NLP and in the computer vision communities [121, 67]. The algorithm presented in Chapter 2 aims at addressing the semantic drift problem and also at reducing the reliance on classification confidence scores.[8] The techniques that we employ in this

[8]The scores are used to select the best candidate documents to be aggregated with the set of labeled data.

regard include a hypothesis transfer, a two-step learning algorithm, and a new learning rate schedule.[9] These methods are general, and don't use any specific property of social media data.

Our model proposed in Chapter 3 mainly relies on the properties of user-generated data to extract two views from documents. While, the extraction of two views is only applicable to this type of data, we expect that the underlying principle of our model be applicable to other scenarios. Indeed, the model proposed in this chapter is a variant of co-training algorithm which is less sensitive to the semantic drift problem. Co-training by itself is not domain specific [120]. Correspondingly, our active learning model, proposed in Chapter 4, is a variant of the co-testing algorithm. Multi-view active learning is not domain specific and has been applied to images before [29].

In Chapter 5, we formulate a new research problem for using unlabeled data from auxiliary domains in Domain Adaptation and also propose a novel model based on the prediction uncertainty to use this data. Existing literature on Domain Adaptation [12] typically don't restrict themselves to any type of data, and our study in this area is no exception. While in the experimental section of this chapter we follow the theme of this dissertation and evaluate our model in social media data, there is no theoretical or practical arguments to suggest that our model cannot be used in other domains. To summarize, due to the importance of user-generated content, in this dissertation we focus on this type of data. Nonetheless, as we argued above, in numerous cases our techniques are developed for general scenarios. The efficacy of our models in such cases remains to be explored as future work.

[9] See Chapter 2 for details of these techniques.

Chapter 2

Semi-Supervised Learning

2.1 Introduction

Semi-supervised text classifiers have achieved remarkable success in the past few years due to the high capacity of neural networks in generalization. Even though modern classifiers usually rely on large training sets, the introduction of contextual word embeddings and language model pretraining [88, 33, 91] has tremendously reduced the need for manual data annotation. However, the state-of-the-art neural models are still prone to overfitting, particularly in the areas with sparse and specialized language models. These areas include, but are not limited to: legal domain [50], medical domain [68], and social media domain [6].

Depending on the task at hand, one solution to address this issue is to automatically construct a large–and perhaps noisy–dataset [41], however, this is not always feasible [124]. A more methodical approach is to employ the techniques that improve generalization. These techniques include exploiting neural word embeddings [58], data augmentation [11], and domain adaptation [5]. Exploiting unlabeled data [121, 15] is also a complementary approach. In this chapter, we add to the body of literature on semi-supervised learning by employing the properties of neural networks

and proposing a novel way to utilize unlabeled data.

Our algorithm, termed Self-Pretraining , is inspired by the self-training paradigm [121]. Similar to self-training, our algorithm is iterative and in each iteration selects a set of unlabeled documents to label. However, as opposed to self-training, our algorithm is threshold-free. Thus, it does not rank the unlabeled documents based on their prediction confidences. This makes our algorithm particularly suitable for the neural network models due to their poorly calibrated outputs [69]. Additionally, our algorithm is able to cope with the semantic drift problem [20]. That is, it is resilient to the noise in the *pseudo-labels* as the number of iterations increases and the error rate of the underlying classifier rises. Furthermore, Self-Pretraining is able to potentially revise the labels of the previously labeled documents. To achieve these, our model employs an iterative distillation process, i.e., in each iteration, the information obtained in the previous iterations is distilled into the classifier. It transfers a hypothesis across iterations, and utilizes a two-stage learning model, where the set of pseudo-labels is used to initialize the classifier, and the set of labeled documents is used to finetune the classifier. Additionally, Self-Pretraining adapts a novel learning rate schedule to efficiently integrate the two sets of noisy and noise-free training examples. Finally, in order to further mitigate the impact of noisy pseudo-labels in every iteration, our model transforms the distribution of pseudo-labels such that it reflects the distribution of the labels in the previous iterations.

Our experiments in three publicly available Twitter datasets show that Self-Pretraining outperforms the state of the art in multiple settings where only a few hundred labeled documents are available. This is significant, considering that the underlying classifier of our algorithm and all the baseline models is BERT [33] which already uses the language model pretraining, and therefore, makes any improvement over the baselines very challenging. We also carry out a comprehensive set of experiments to better understand the qualities of Self-Pretraining . Particularly, we

demonstrate the robustness of our model against the noise in the pseudo-labels. The contributions of this chapter are as follows: **1)** We propose a novel semi-supervised learning framework termed Self-Pretraining . Our model is based on the self-training paradigm, however, it is threshold-free, it can cope with the semantic drift problem, and can also revise the previously labeled documents. To our knowledge, Self-Pretraining is the first model that addresses these drawbacks in a unified framework. **2)** We propose a novel learning rate schedule to effectively integrate the optimization procedure with our two-stage semi-supervised learning process. **3)** In order to further mitigate the semantic drift problem, we model the class distribution of the pseudo-labels as a stochastic process across the bootstrapping iterations, and propose a novel approach to transform the class distributions. **4)** We carry out a comprehensive set of experiments across three publicly available Twitter datasets, and demonstrate that our model outperforms several state-of-the-art baselines in multiple settings.

Our research clearly pushes the state of the art in semi-supervised text classification. We believe the ideas presented in this chapter can be applied to other domains, e.g., image classification. Future work may explore this direction. In the next section, we provide an overview of the related studies and highlight the qualities of Self-Pretraining .

2.2 Related Work

Unlabeled data in semi-supervised learning. Unlabeled data can be exploited in multiple ways. It can be used as a meta-source of information [42], it can be used as a regularizer [118], or it can be used in a domain adaptation setting to correlate the source and target data [107]. A more recent interest in literature is *self-supervision*, where a self-contained task is defined such that no manual annotation is required. Instances of such tasks are language model pretraining [88, 33] in NLP,

and contrastive learning in image processing [95, 26]. From a different perspective, self-supervision studies can be categorized into task-agnostic [16] and task-specific [43] approaches. This has given rise to the notion of "pretrain, then finetune" the model. We integrate this paradigm into the self-training algorithm.

Bootstrapping in semi-supervised learning. Self-training is the oldest approach to semi-supervised learning [24] dating back to 1965 [98]. This idea re-emerged in the seminal work of Yarowsky [121] for NLP tasks in 1995, and also once more in the computer vision community in 2013 as *pseudo-labeling* [67]. This algorithm is a wrapper that repeatedly uses a supervised algorithm as the underlying model. There are multiple assumptions under which self-training–and in general semi-supervised learning–is expected to perform well. For instance, the *smoothness assumption* that states if the two data points x_1 and x_2 are close, then their predictions y_1 and y_2 should be also close–this assumption has been the basis of algorithms such as MixUp [125] and MixMatch [14]. As we discuss in the next section, one unsatisfactory aspect of self-training is that it relies on the properties of the underlying predictive model, e.g., the model output distributions. There have been attempts to address this drawback. For instance, throttling [2] can be used to dampen the effect of noisy candidates, or in the context of transductive learning, the density of the unlabeled data points can be incorporated to mitigate this issue [101].

In the past few years, studies have explored the efficacy of the neural networks as the underlying predictive model in self-training. A neural network variant of co-training [15] is proposed in [90]. In [61], the authors propose a framework to integrate human knowledge with co-training. In [117], a reinforcement learning variant of co-training is proposed. In [93], a neural network variant of tri-training with disagreement [103] is presented, and it is shown that the combination is a surprisingly strong baseline in the domain adaptation setting. The authors in [21] propose to use percentile scores instead of the confidence scores to select the best pseudo-labels;

and the authors in [82] employ Bayesian neural networks to select the most and the least confident pseudo-labels in every iteration. In [7], a new document sampling strategy for self-training is proposed. The model, in addition to the classifier confidence, employs the training epochs in which the unlabeled documents are approximately correctly labeled. In [9], the authors propose to integrate MixUp [125] with the oversampling of the labeled training examples. They show that self-training is indeed a very strong baseline comparing to the common regularization and data augmentation techniques. In comparison to these studies, Self-Pretraining is the first model that employs model distillation [48] along a hypothesis to transfer information across iterations, enabling it to potentially revise the pseudo-labels. It integrates the pretraining/finetuning paradigm with self-training, utilizes an efficient optimization procedure along a perturbation technique to mitigate the negative impact of noisy pseudo-labels.

Other closely related studies. In addition to the studies above, Self-Pretraining is also related to the studies on model distillation [48] and temporal ensembling [65]. Model distillation was proposed in [17, 48] to transfer the knowledge from one model to another model. In [27], the authors show that transferring the knowledge of a big network, trained by a self-supervised task, to a small network improves generalization. Their main contribution is to show that big models are trained easier, and therefore, can be used as a proxy to train small networks. Their model is not iterative, and does not explore the unlabeled data to extract new information. Born-again networks were proposed in [36], the authors show that simply distilling a neural network into itself improves performance. Their model is not a semi-supervised algorithm, and is not proposed to exploit unlabeled data. The authors in [119] show that the regular neural self-training algorithm can be improved by adding noise to the model. Similar to our work, they allow the pseudo-labels evolve over iterations. Beyond this step, they don't propose any modification to the self-training algorithm. Additionally, the

efficacy of their model is not explored in the semi-supervised setting. A very close approach to this study is presented in [45], where the authors again show that adding noise to the inner representation of the model enhances the self-training performance. Temporal ensembling was proposed in [65]. The authors propose to maintain the per-sample prediction average of the unlabeled data across the epochs and constrain the prediction variance. Their model is not based on self-training, has no strategy to separate labeled from unlabeled data, and becomes unwieldy when using large datasets. The authors in [109] resolve the high complexity of temporal ensembling by updating the weights of the model across the epochs, instead of storing the predictions.

2.3 Semi-Supervised Learning via Self-Pretraining

We begin this section by providing an overview of Self-Pretraining and highlighting its differences from the self-training algorithm. Then we introduce a series of strategies to overcome the drawbacks of the vanilla Self-Pretraining [1].

In the self-training algorithm [121], a small set of labeled documents L and a large set of unlabeled documents U are available for training. The algorithm is iterative and in each iteration the predictive model M is trained on the current set L, and is used to probabilistically label the current set U. Given the hyper-parameter θ as the minimum confidence threshold, the most confidently labeled documents in U and their associated *pseudo-labels* are selected to be augmented with the set L. This procedure is repeated till a certain criterion is met. There are three drawbacks with this algorithm: 1) The semantic drift problem [20], where the increasingly negative impact of noisy pseudo-labels overshadows the benefit of incorporating unlabeled data. 2) Reliance on the model calibration. If the underlying classifier is unable to accurately model the class distributions, then, it will fail to properly rank the candidate documents, e.g., in the case of neural networks [69]. 3) Being unable

[1] We focus on the binary classification problems.

to revise the pseudo-labels once they are assigned to the unlabeled documents and augmented with the set of labeled documents. Even though there exist techniques to address these challenges under certain conditions, e.g., throttling [2] for the poor model calibration or mutual exclusive bootstrapping [32] for the semantic drift, to our knowledge, Self-Pretraining is the first unified framework to address all three.

Our algorithm is iterative and utilizes two neural networks as the underlying classifiers. Algorithm 4 illustrates Self-Pretraining in its basic form. Initially, the set L is used to train the network M_1 (Line 2), then the parameters of M_1 are copied to the network M_2 (Line 5). In the next step, a set of unlabeled documents are randomly drawn from U (Line 7). This set is labeled by M_2 and used along the set L to retrain[2] M_1 (Line 8). The role of the two networks is reversed in the next iteration. In each iteration, the sample size is increased by k (Line 6), and the algorithm stops when the sample set covers the entire set U. Finally, the ensemble of M_1 and M_2 can be used to label the unseen documents–we used the mean of their class predictions.

Algorithm 1 Overview of Vanilla Self-Pretraining

```
1:  function SELF_PRETRAINING(L, U, k)
2:    M1 ← train_model(L)
3:    sample_size ← 0
4:    repeat
5:      M2 ← copy_model(M1)
6:      sample_size ← sample_size + k
7:      C ← random_sample(U, sample_size)
8:      M1 ← train_model({(C, M2(C)) ∪ L})
9:    until sample_size < |U|
10:   return M1, M2
```

Algorithm 1 has two advantages: 1) To select the pseudo-labels the class distribution is not taken into account, therefore, there is no constraint on the classifier capacity in ranking the unlabeled documents. Additionally, this prevents the model from repeatedly selecting a fixed set of unlabeled documents in every iteration–i.e.,

[2]Note that by definition, the neural self-training requires reinitialization and retraining in every iteration [93], thus our algorithm is comparable to other self-training models in terms of runtime.

the set of highly confident pseudo-labels. 2) The information that is transferred across the iterations is in the form of a hypothesis rather than a set of fixed pseudo-labels. Therefore, the model belief about the pseudo-labels can evolve over time–the pseudo-labels are not augmented with the set of labeled documents. On the other hand, this algorithm has one substantial disadvantage, and that is the problem of semantic drift. In fact, randomly sampling from the set of unlabeled documents exacerbates this problem by introducing noisy labels and pushing the transferred hypothesis towards a sub-optimal point. In the following, we exploit the neural network properties and introduce a series of strategies to cope with this problem and also to enhance the flow of information across the iterations.

2.3.1 Hypothesis Transfer and Iterative Distillation

Self-Pretraining transfers a hypothesis–a learned function–from one iteration to the next iteration. In each iteration, this hypothesis is used to form a new one by creating a set of pseudo-labels and augmenting them with the set of labeled documents. Even though the ultimate criterion is maximizing the model utility, the short term goal in each iteration is not necessarily making accurate predictions but to carefully transfer the knowledge from one model to another. These two processes are not necessarily in accordance with each other, since the former may rely on the learner outcome and the latter may rely on the learning procedure itself. Thus, the classifier labels, even though informative, are not expressive enough to transfer the entire knowledge from one iteration to the next one.

The authors in [17, 48] propose an algorithm called model distillation to transfer the knowledge from a large model (called *teacher*) to a small model (called *student*). Model distillation is based on the argument that the class distribution carries a significant amount of information regarding the classifier decision boundary. For instance, given a document d that is labeled positive, it is nontrivial information to know that

if the class prediction was 95% positive or 65% positive. The authors in [48] use model distillation to transfer knowledge from one network to another by modifying the softmax layer as follows:

$$a_i = \frac{\exp \frac{z_i}{T}}{\sum_j \exp \frac{z_j}{T}} \tag{2.1}$$

where z_i is the last layer *i-th* logit, j is the number of classes, and a_i is the class prediction. The hyper-parameter T is called temperature and is introduced to smooth the class predictions. A higher temperature results in a higher entropy in predictions. This is particularly desirable, since neural networks are known to have a low entropy in their predictions [69].

Given the argument above, we employ model distillation in Self-Pretraining , and effectively distill the previous iterations into the student network M_1. Thus, in each iteration, instead of using the teacher–M_2–hard predictions on unlabeled documents, we use the soft predictions along the set L to train the student network–Algorithm 4, Line 8.

2.3.2 Two-Stage Semi-Supervised Learning

As we mentioned earlier, self-training suffers from the semantic drift problem. This problem occurs when the errors primarily caused by the pseudo-labels accumulate across iterations and ultimately distort the classifier boundary. Even though the minimum confidence threshold θ can potentially prevent spurious pseudo-labels from entering the training set, as the set L grows in size the probability of mislabeling documents increases correspondingly. This problem is even severer in our model, since it is threshold-free. One naive solution is to assign a lower weight to the pseudo-labels, however, we observed in our experiments that this approach is not effective enough to resolve the underlying problem.

To mitigate this problem, one solution is to process the set of pseudo-labels and

decouple the information that contradicts the information stored in the set L. Erasing this section of the pseudo-labels can lower the error rate and subsequently improve the hypothesis in the current iteration. To accomplish this, we exploit the catastrophic forgetting phenomenon in the neural networks [77, 63]. Catastrophic forgetting occurs in the continual learning setting where a network is trained on a series of tasks. Each training procedure updates the parameters of the model to meet the requirements of the objective function, and the updates in the current task may contradict and erase the information related to the previous tasks. This effect is typically undesirable, however, in the context of Self-Pretraining, we use this mechanism as a proxy to build a hierarchy of information in the network. Therefore, we make a small modification in Algorithm 4. Instead of aggregating the set of pseudo-labels with the set of labeled documents–Line 8–we first use the set of pseudo-labels to initialize–train–the current network M_1, and then further train it using the set of labeled documents.

Decomposing the training procedure into two stages introduces a new challenge, and that is the possibility of completely updating the network parameters in order to learn the regularities in the set of labeled documents. To avoid this, we propose to use the following objective function while training the model M_1 using the set L:

$$\begin{aligned}\mathcal{L} =&(1-\lambda)(-\sum_{i=1}^{N}[y_i \log a_i + (1-y_i)\log(1-a_i)])+ \\ &\lambda(-\sum_{i=1}^{N}[q_i \log a'_i + (1-q_i)\log(1-a'_i)])\end{aligned} \tag{2.2}$$

where N is the number of the documents in the set L, y_i is the true label of the document d_i, a_i is the class prediction of M_1 for d_i, a'_i is the class prediction of M_1 for d_i with a high temperature as described in Section 2.3.1, and q_i is the class prediction of M_2 for d_i with the same temperature as that of M_1. λ is a hyper-parameter to govern the relative weight of the two terms ($0 \leq \lambda \leq 1$). Since the gradients of the second term in Equation 2.2 scale by $\frac{1}{T^2}$, in order to balance the impact of the two

terms in back-propagation, we multiply these gradients by T^2–see Equation 2.1.

The first term in Equation 2.2 is the cross entropy between the ground truth labels and the class probabilities of M_1. The second term is the cross entropy between the class probabilities of M_2 and M_1. This objective function is an effort to keep a balance between the information that is transferred from the previous iterations and the information that is extracted from the set of labeled documents L.

In Section 4.6 we demonstrate that the ideas proposed in this section greatly mounts the resistance of Self-Pretraining to the noise in the pseudo-labels. These ideas are related to two categories of studies: 1) The studies on pretraining neural networks [47, 51]. 2) The studies on curriculum learning [13]. Researchers [47, 51] in both NLP and the vision community have shown that *pretraining* a neural network with out-of-domain data and then *finetuning* it with the target data can significantly contribute to the performance. These two steps are analogous to the two stages that we described in this section. Additionally, our work is also closely related to the idea of curriculum learning [13], where it is shown that a learner can leverage the order of the training examples to learn more efficiently. Even though Self-Pretraining employs this mechanism, the criterion to determine the order of the training examples is not based on the properties of the data points but is based on the source of the labels.

2.3.3 Right Trapezoidal Learning Rates

In the previous section, we employed an approach to mitigate the semantic drift problem by exploiting the catastrophic forgetting phenomenon. This two-stage strategy creates a suitable opportunity for enhancing the optimization process. Since the pseudo-labels are potentially noisy, we propose to use this set to explore the hypothesis space and detect the region that contains a better local-optima. Thereafter, the set of labeled documents, which are noise-free, can be used to detect the target local-optima.

Given the argument above, we propose to use a *right trapezoidal* learning rate–illustrated in Figure 2.1–as follows:

$$\eta_t = \begin{cases} R & batch_t \subset C \\ R \; - \; R \; \frac{t-b_C}{b_L} & batch_t \subset L \end{cases}$$

where t denotes the current time step, and $batch_t$ is the current batch of documents being processed. η_t is the current learning rate, R is the initial learning rate, C is the set of pseudo-labels, L is the set of labeled documents, b_C is the number of pseudo-label batches, and b_L is the number of labeled batches.

Our proposed learning rate is composed of two phases: 1) A fixed learning rate–the dashed line in Figure 2.1–where the pseudo-labels are used to train the model M_1–see Algorithm 4. In this stage, the network parameters can freely update, and therefore, the learner can essentially explore the hypothesis space. 2) A gradually decreasing learning rate–the solid slanted line in Figure 2.1–where the labeled documents are used to further train the network. In this stage, the optimizer settles down, therefore, we use the noise-free labeled documents, since even a small perturbation in the data may cause a significant loss. Having a two-phase learning rate also organically integrates with our two-stage semi-supervised learning procedure. Since the gradual reduction in the learning rate, prevents the objective of the second task from completely erasing the knowledge transferred from the previous iterations.

2.3.4 Inertial Class Distributions

Semi-supervised learning models rely on unlabeled data as their primary source of information. While these methods have obtained promising results, they are inherently prone to overfitting on the irregularities in the unlabeled data. Introducing an inductive bias [80] into the semi-supervised learning algorithms is a common approach to increase their robustness. For instance metric regularization [97] or temporal en-

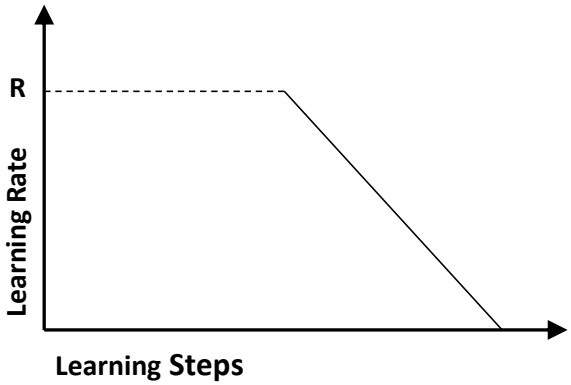

Figure 2.1: The Self-Pretraining learning rate schedule. The dashed horizontal line is the learning rate of the network during the training by the pseudo-labels, and the slanted line is the learning rate during the training by the labeled documents.

sembling [65] are a few examples. While these techniques can be integrated into Self-Pretraining , in this section, we opt to explore a new direction.

We hypothesize that the class probability distribution of the randomly selected set of unlabeled documents–Algorithm 4, Line 8–should evolve slowly and avoid abrupt transitions across iterations. This is a harsh assumption, since this probability distribution also depends on the drawn samples. However, we argue that an abrupt change in this distribution can be the sign of an influx of noisy pseudo-labels in the previous iterations. Thus, we aim to prohibit such changes. To achieve this, we assume the distribution of the class probabilities is a random process dynamically evolving across the iterations, and the class probability distribution of the selected unlabeled documents in every iteration–$M_2(C)$ in Algorithm 4–is a sample from the underlying random variables.

For simplicity, we assume the process consists of only a family of two Gaussian random variables S^+ and S^-, where S^+ is the state of the positive pseudo-labels, and S^- is the state of the negative pseudo-labels. The sample mean and variance of S^+

in the iteration t (i.e., S_t^+) are given by:

$$\mu_t = \frac{\sum_{i=1}^{n} p_i^t}{n}$$
$$\sigma_t^2 = \frac{\sum_{i=1}^{n} (p_i^t - \mu_t)^2}{n}$$

where n is the number of positive pseudo-labels in the iteration t, and p_i^t is the probability of the *i-th* positive pseudo-label belonging to the positive class–it is clear that $0.5 \leq p_i^t$, because the sample is positive. Correspondingly, the sample mean and variance of S^- in the iteration t (i.e., S_t^-) are given by:

$$\gamma_t = \frac{\sum_{i=1}^{m} q_i^t}{m}$$
$$\varphi_t^2 = \frac{\sum_{i=1}^{m} (q_i^t - \gamma_t)^2}{m}$$

where m is the number of negative pseudo-labels in the iteration t, and q_i^t is the probability of the *i-th* negative pseudo-label belonging to the negative class–note that $0.5 \leq q_i^t$ and also note that for every pseudo-label $p_i^t + q_i^t = 1$.

In the iteration $t+1$, the sample distributions of the random variables S^+ and S^- proceed to $S_{t+1}^+ \sim \mathcal{N}(\mu_{t+1}, \sigma_{t+1}^2)$ and $S_{t+1}^- \sim \mathcal{N}(\gamma_{t+1}, \varphi_{t+1}^2)$. These updates can be due to the randomness in the model initialization, the randomness in the selected set of unlabeled documents in the iteration t, or partially due to the noisy pseudo-labels introduced in the iteration t. More specifically, the misclassifications of the model M_2 in the iteration t–see Algorithm 4–which were subsequently used to pretrain the model M_1, and ultimately distorted the class distribution of the set of pseudo-labels in the iteration $t+1$. To dampen the impact of this noise, we define two Gaussian distributions $\hat{S}_{t+1}^+$ and $\hat{S}_{t+1}^-$ as the linear combination of the class distributions in the iterations t and $t+1$, and project[3] the pseudo-labels in S_{t+1}^+ into $\hat{S}_{t+1}^+$, and the

[3]No projection is performed in the first iteration.

pseudo-labels in S^-_{t+1} into $\hat{S}^-_{t+1}$. Thus:

$$\begin{aligned} \hat{S}^+_{t+1} &= \alpha\ S^+_t + (1-\alpha)\ S^+_{t+1} \\ \hat{S}^-_{t+1} &= \alpha\ S^-_t + (1-\alpha)\ S^-_{t+1} \end{aligned} \tag{2.3}$$

where α is a hyper-parameter to govern the rate at which the probability distributions can evolve in every iteration. The new distributions $\hat{S}^+_{t+1}$ and $\hat{S}^-_{t+1}$ are defined between the class distributions in the iteration t and $t+1$. The hyper-parameter α determines the degree at which the pseudo-labels in the iteration $t+1$ are perturbed to resemble the pseudo-labels in the iteration t. By employing this mechanism, the sudden abrupt changes in the distribution of pseudo-labels are avoided. We perform this step after we generate the pseudo-labels using the model M_2, and before using this set to pretrain the model M_1–Algorithm 4, Line 8.

In Section 4.6, we show that Self-Pretraining algorithm, along the techniques that we introduced in the sections 2.3.1, 2.3.2, 2.3.3, and 2.3.4 achieves the state-of-the-art results in multiple settings. In the next section, we describe our datasets, baselines, and training setup.

2.4 Experimental Setup

We begin this section by describing the datasets that we used, then we provide a brief overview of the baseline models, and finally review the detail of the experiments.

2.4.1 Datasets

We evaluate Self-Pretraining on three Twitter text classification tasks[4]: 1) Adverse Drug Reaction monitoring (ADR). In this task, the goal is to detect the tweets that

[4]Please refer to the cited articles for the analysis and discussion on the difficulties of these tasks, we skip this subject.

	Training			Test		
Dataset	Tweets	Neg	Pos	Tweets	Neg	Pos
ADR	20624	91%	9%	4992	92%	8%
Earthquake	8166	53%	47%	3502	53%	47%
Product	4503	69%	31%	2114	78%	22%

Table 2.1: Summary of ADR , Earthquake , and Product datasets.

report an adverse drug effect. We used the dataset introduced in [114] prepared for the ACL 2019 SMM4H Shared Task. 2) Crisis Report Detection (CRD). In this task, the goal is to detect the tweets that mention an event related to natural disasters. We used the dataset introduced in [5] about the 2015 Nepal earthquake. 3) Product Consumption Pattern identification (PCP). In this task, the goal is to identify the tweets that report the usage of a product. We used the dataset introduced in [52], which is about receiving an influenza vaccine.

The ADR and Earthquake datasets are released with pre-specified training and test sets. In Product dataset we used the tweets published in 2013 and 2014 for the training set, and the tweets published in 2015 and 2016 for the test set. Table 2.1 summarizes the datasets. We see that Earthquake dataset is balanced and ADR dataset is highly imbalanced. The Earthquake dataset is released along a set of unlabeled tweets. For the other two datasets, we used the Twitter API and crawled 10,000 related tweets for each one to be used as the unlabeled sets (the set U in Algorithm 4). For ADR dataset we used the drug names to collect the unlabeled set and for Product dataset we used the query "flu AND (shot OR vaccine)" to collect the set.

2.4.2 Baselines

We compare our model with six baselines.

Baseline. The setting for evaluating semi-supervised learning models should be realistic. Pretrained contextual language models are the primary ingredient of the state-of-the-art text classifiers. Thus, we used BERT [33] as the naive baseline, and

also as the underlying classifier for all the other baselines. Note that this makes any improvement over the base classifier very challenging, since the improvement should be additive. We train this model on the set of labeled documents, and evaluate on the test set. We used the published pretrained *base* variant, followed by one fully connected layer and one softmax layer. We used the Pytorch implementation [115] of BERT; the settings are identical to the suggestions in [33].

Self-training. We included the regular self-training algorithm [121], where in each iteration the top pseudo-labels, subject to a minimum threshold confidence, are selected and added to the labeled set. We used one instance of *Baseline* in this algorithm.

Tri-training+. We included a variant of tri-training algorithm called tri-training with disagreement [103]. In [93], the authors show that this model is a very strong baseline for semi-supervised learning. We used three instances of *Baseline* in this algorithm.

Mutual-learning. We included the model introduced in [127]. This model is an ensemble, and is based on the idea that increasing the entropy of the class predictions improves generalization [87]. We used two instances of *Baseline* in this model–in the parallel setting.

Spaced-rep. We included the model introduced in [7]. This model employs a queuing technique along a validation set to select the unlabeled documents that are easy and also informative for the task. We used our own implementation of this model.

Co-Decomp. We included the framework introduced in [61]. This model uses domain knowledge to decompose the task into a set of subtasks to be solved in a multi-view setting. We used the keyword level representations and sentence level representations as the two views. We used two instances of *Baseline* in this algorithm.

Self-Pretraining . The model that we introduced in Section 4.3. We used two instances of *Baseline* as M_1 and M_2.

2.4.3 Experimental Details

To evaluate the models in the semi-supervised setting, we sampled a small subset of the training sets[5] and did not use the rest of the tweets. Note that the remaining set was not used as the unlabeled data either–see Section 2.4.1 for the description of the unlabeled sets. To sample the data, we used a stratified random sampling to preserve the ratio of the positive to the negative documents. We also ensured that the initial labeled set is identical for all the models. We repeated all the experiments 3 times with different random seeds. We will report the average across the runs. All the baseline models use throttling [2] with confidence thresholding ($\theta = 0.9$). We also linearly increased the size of the sample set [94], however, did not add more than 10% of the current training set in each iteration.

In our experiments we observed that the performances of *self-training* and *Co-Decomp* degrade if we use the entire set of unlabeled data–due to the semantic drift problem. Thus, we assumed the number of the iterations in these algorithms is a hyper-parameter and used 20% of the labeled set as the validation set to find the best value. *Tri-training+* has an internal stopping criterion. *Mutual-learning* uses the unlabeled data as a regularizer. *Spaced-rep* requires a validation set for the stopping criterion and also for the candidate selection. Thus, in this model we used 20% of the labeled set as the validation set. We also set the number of queues to 6, the rest of the settings are identical to what is used in [7].

Since we are experimenting in the semi-supervised setting, we did not do full hyper-parameter tuning. We used the training set in Product dataset and searched for the optimal values of λ in Equation 2.2 and α in Equation 2.3. Their best values are 0.3 and 0.1 respectively. We set the step size k in Algorithm 4 to 2,000 and the temperature T in Equation 2.1 to 3. In our two-stage training procedure the goal

[5]Using the entire set of labeled tweets turns the classification task into a supervised problem, which is not the subject of our study.

of the first step is the model initialization, thus we trained the network for only 1 epoch. In the rest of the cases, including in our model and the baselines we trained the models for 3 epochs. The only exception is *Space-rep*, which requires a certain number of training epochs with early stopping. To train BERT in all of the cases we used a setting identical to that of the reference [33]–we set the batch-size to 32. Following the argument in [78], we used F1 in the positive class to tune the models. In the next section, we report average F1, Precision, and Recall of the models across the runs.

2.5 Results and Discussion

We begin this section by reporting the main results. Then we present a series of experiments that we carried out to better understand the properties of Self-Pretraining.

2.5.1 Main Results

Table 2.2 reports the performance of Self-Pretraining in comparison to the baselines under two sampling quantities–i.e, 300 and 500 initial random tweets–in the three datasets. We see that in all of the cases Self-Pretraining is either the top model or on a par with the top model. The difference in ADR dataset is substantial, however, in Earthquake dataset the difference is very small. ADR is an imbalanced dataset. Our case by case inspections also showed that the positive tweets in this dataset are very diverse, which makes the models very susceptible to the number of training examples. We also see that *Mutual-learning* completely fails in this dataset. Our experiments showed that this is due the the skewed class distributions in this dataset[6]. Surprisingly, we see that *Spaced-rep* is performing poorly in the experiments, even though this model was evaluated on social media tasks before [7]. We believe the reason is as

[6]We built two imbalanced datasets by subsampling from Earthquake and Product datasets, this model also failed in these cases.

# Tweets	Model	ADR dataset			Earthquake dataset			Product dataset		
		F1	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**
300	*Baseline*	0.238	0.237	0.342	0.715	0.692	0.749	0.728	0.696	0.770
	Self-training	0.303	0.269	0.350	0.728	0.697	0.762	0.731	0.675	0.798
	Tri-training+	0.306	0.236	**0.448**	0.735	0.680	0.799	0.734	0.659	**0.828**
	Mutual-learning	0.024	**0.707**	0.012	**0.743**	0.685	**0.814**	0.753	**0.778**	0.730
	Spaced-rep	0.258	0.248	0.277	0.721	0.650	0.811	0.727	0.701	0.760
	Co-Decomp	0.310	0.288	0.356	0.728	**0.722**	0.735	0.754	0.756	0.758
	Self-Pretraining	**0.397**	0.370	0.440	0.737	0.704	0.772	**0.766**	0.757	0.777
500	*Baseline*	0.312	0.253	0.411	0.746	0.735	0.760	0.740	0.704	0.782
	Self-training	0.335	0.300	0.387	0.737	**0.765**	0.714	0.741	0.739	0.745
	Tri-training+	0.365	0.298	0.480	0.747	0.707	**0.793**	0.758	0.697	**0.833**
	Mutual-learning	0.108	**0.638**	0.059	0.751	0.730	0.773	0.767	**0.811**	0.728
	Spaced-rep	0.295	0.274	0.417	0.728	0.694	0.775	0.737	0.693	0.788
	Co-Decomp	0.345	0.313	0.388	0.749	0.746	0.752	0.766	0.771	0.764
	Self-Pretraining	**0.420**	0.376	**0.483**	**0.752**	0.718	0.789	**0.787**	0.784	0.792

Table 2.2: F1, precision, and recall of Self-Pretraining in ADR , Earthquake , and Product datasets compared to the baseline models. The models were trained on 300 and 500 labeled user postings.

k	F1	Precision	Recall
1000	0.395	0.306	0.565
2000	0.420	0.376	0.483
3000	0.428	0.386	0.485
4000	0.413	0.347	0.537

Table 2.3: Results of Self-Pretraining with different values of k–the number of randomly selected pseudo-labels–in the test set of ADR dataset. The models began with 500 labeled user postings.

follows: This model relies on the number of training epochs to construct its internal data structure for ranking the candidate tweets. When the underlying classifier is a pretrained language model, e.g., bert, increasing the number of epochs may result in overfitting and therefore, contradicts the purpose. On the other hand, early stopping also prevents the model from separating the informative from uninformative tweets.

2.5.2 Empirical Analysis

We begin this section by reporting the effect of the step size k on Self-Pretraining–see Algorithm 4. Table 2.3 reports F1, precision, and recall of Self-Pretraining at varying step sizes in the test set of ADR dataset. Since this datset is the largest one, we report all of the experiments in this dataset. We see that the performance improves up to the step size of 3000 unlabeled tweets per iteration. We still do not have a concrete explanation to justify this trend, since it is natural to expect the smaller step sizes yield better results. One reason may be that if the set of pseudo-labels is small, the network can perfectly learn the noise in the set during the pretraining. In Section 2.3.2 we argued that the two-stage training can cope with the semantic drift problem. To support this argument, we report the performance of the middle classifiers M_1 at the end of every iteration. Figure 2.2 reports the performances during the training for varying step sizes. We see that for none of the step sizes the performance drops as the number of unlabeled tweets grows–the typical symptom of semantic drift.

Self-Pretraining relies on iterative distillation–Section 2.3.1–to transfer knowledge

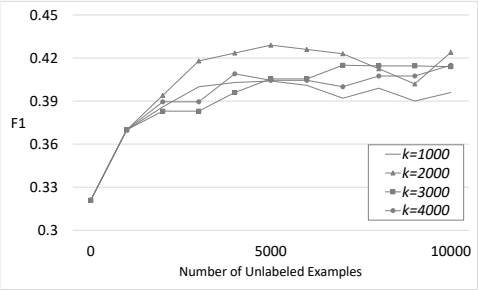

Figure 2.2: F1 of the resulting classifier in every iteration of Self-Pretraining with different values of k–the number of randomly selected pseudo-labels. The middle values are interpolated. The results are in the test set of ADR dataset.

T	F1	Precision	Recall
2	0.422	0.361	0.514
3	0.420	0.376	0.483
4	0.421	0.356	0.517
5	0.433	0.382	0.506
6	0.422	0.370	0.491

Table 2.4: Results of Self-Pretraining in the test set of ADR dataset at varying values of the temperature (T) for iterative distillation.

from one iteration to the next one. Model distillation leverages the temperature T in the softmax layer, see Equation 2.1. It is informative to find the degree at which this hyper-parameter can affect the learning performance. Table 2.4 reports the model performance at varying values of the hyper-parameter T. We see that the performance peaks at $T = 5$. In section 2.3.2 we proposed an objective function and argued that the second term of the function prevents the hard labels of the training set from erasing the information transferred from the previous iteration. To demonstrate the impact of the second term, in Table 2.5 we report the model performance at varying values of the hyper-parameter λ–the weight of the second term. We see that the performance almost gradually improves as we increase λ and peaks at $\lambda = 0.4$. This is primarily due to the improvement in precision.

In Section 2.3.4 we proposed to transform the class probability distribution in the iteration $t + 1$ into a new distribution that resembles the distribution in the iteration t. We argued that this transformation can help to mitigate the semantic drift problem

λ	**F1**	**Precision**	**Recall**
0.1	0.425	0.357	0.529
0.2	0.428	0.355	0.541
0.3	0.420	0.376	0.483
0.4	0.438	0.377	0.531
0.5	0.421	0.350	0.534

Table 2.5: Results of Self-Pretraining in the test set of ADR dataset at varying values of the hyper-parameter (λ) for our two-stage learning–see Equation 2.2.

via constraining the degree at which the pseudo-labels can evolve in every iteration, therefore, can potentially limit the negative impact of noisy pseudo-labels. In Table 2.6 we report the model performance at varying values of the hyper-parameter α in Equation 2.3. This hyper-parameter governs the degree of the transformation. We see that the performance noticeably improves as we increase the value of α. Finally, we report an ablation study in Table 2.7. In the previous experiments we showed that a better performance in ADR dataset is achievable by a dataset specific hyper-parameter tuning. Nonetheless, we still expect that, with the current hyper-parameters in ADR , the ablation study can reveal the relative importance of the Self-Pretraining modules in general. In this experiment, we replaced the two-stage training model (Section 2.3.2) with the simple data augmentation of the labeled and pseudo-labels. Additionally, we replaced our right trapezoidal learning rate (Section 2.3.3) with the default slanted learning rate [33]. We replaced our iterative distillation process (Section 2.3.1) with simply using the hard labels in every iteration. Finally, we deactivated our pseudo-label transformation step (Section 2.3.4). We see that the two-stage training model and the inertial transform have the highest and the lowest contributions.

In summary, we showed that Self-Pretraining is the state-of-the-art in multiple settings. The authors in [7] show that semi-supervised models–although under domain shift–typically fail when they are evaluated on a different task from what they are initially proposed for. Thus, they conclude that these models should be evaluated in

α	**F1**	**Precision**	**Recall**
0.1	0.420	0.376	0.483
0.2	0.422	0.353	0.530
0.3	0.413	0.345	0.518
0.4	0.424	0.355	0.532
0.5	0.429	0.363	0.527

Table 2.6: Results of Self-Pretraining in the test set of ADR dataset at varying values of the hyper-parameter (α) for the inertial transformation of the pseudo-labels–see Equation 2.3.

Deactivated Step	**F1**	**Precision**	**Recall**
two-stage learning	0.339	0.373	0.333
trapezoidal lr	0.360	0.235	0.770
iterative distillation	0.389	0.320	0.495
inertial transform	0.420	0.365	0.497

Table 2.7: Results of Self-Pretraining in the test set of ADR dataset after deactivating the distillation (Section 2.3.1), the two-stage learning (Section 2.3.2), the trapezoidal learning rate (Section 2.3.3), and the inertial transformation (Section 2.3.4).

at least two datasets. In this chapter we evaluated Self-Pretraining in three Twitter datasets. We selected strong baselines, i.e., Tri-training with disagreement [103], Mutual Learning [127], Spaced Repetition [7], and Co-Decomp [61], and showed that some of them fail under certain cases. As opposed to these models, we demonstrated that Self-Pretraining is either the best model or on a par with the best model in every setting. We also reported an extensive set of experiments that we carried out to reveal the qualities of Self-Pretraining . These experiments empirically supported the claims that we made throughout.

Our study is not flawless. To avoid imposing any constrain on the underlying classifier, we proposed to randomly draw the unlabeled documents–Algorithm 4, Line 7. However, if one can guarantee certain classifier properties, then perhaps a sophisticated selection policy will be more effective. The application of our framework in other modalities, e.g., image classification, is also an unexplored topic. Future work may investigate these directions.

2.6 Conclusions

In this chapter, we proposed a semi-supervised learning model called Self-Pretraining. Our model is inspired by the traditional self-training algorithm. Self-Pretraining employs the properties of neural networks to cope with the inherent problems of self-training. Particularly, it employs an iterative distillation procedure to transfer information across the iterations. It also utilizes a two-stage training model to mitigate the semantic drift problem. Additionally, Self-Pretraining uses an efficient learning rate schedule and a pseudo-label transformation heuristic. We evaluated our model in three publicly available Twitter datasets, and compared with six baselines, including pretrained BERT. The experiments show that our model consistently outperforms the existing baselines.

Chapter 3

Multi-View Learning

3.1 Introduction

One of the well-studied areas in online public health monitoring is the extraction of adverse drug reactions (ADR) from social media data. ADRs are the unintended effects of drugs for prevention, diagnosis, or treatment. Duh et al. [34] report that consumers, on average, report the negative effect of drugs on social media 11 months earlier than other platforms. This highlights the importance of this task. Another team of researchers [39] reviewed more than 50 studies and report that the prevalence of ADRs across multiple platforms ranges between 0.2% and 8.0%, which justifies the difficulty of this task. In fact, despite the long history of this task in the research community [122], for various reasons, the performance of the state-of-the-art models is still unsatisfactory. Social media documents are typically short and their language is informal [60]. Additionally, the imbalanced class distributions in ADR task has exacerbated the problem.

In this chapter we propose a novel model for extracting ADRs from Twitter data. Our model which we call View Distillation (VID) relies on the existence of two views in the tweets that mention drug names. We use unlabeled data to transfer the knowledge

from the classifier in each view to the classifier in the other view. Additionally, we use a finetuning technique to mitigate the impact of noisy pseudo-labels after the initialization [59]. As straightforward as it is to implement, our model achieves the state-of-the-art performance in the largest publicly available ADR dataset, i.e., SMM4H dataset. Our contributions are as follows: 1) We propose a novel algorithm using unlabeled data to transfer knowledge across models in multi-view settings, 3) We evaluate our model in the largest publicly available ADR dataset, and show that it yields an additive improvement to the common practice of language model pretraining in this task. To our knowledge, our work is the first study that reports such an achievement. Next, we provide a brief overview of the related studies.

3.2 Related Work

Researchers have extensively explored the applications of ML and NLP models in extracting ADRs from user-generated data. Perhaps one of the early reports in this regard is published in Yates and Goharian [122], where the authors utilize the related lexicons and extraction patterns to identify ADRs in user reviews. With the surge of neural networks in text processing, subsequently, the traditional models were aggregated with these techniques to achieve better generalization [111]. The recent methods for extracting ADRs entirely rely on neural network models, particularly on multi-layer transformers [112].

In the shared task of SMM4H 2019 [114], the top performing run was BERT model [33] pretrained on drug related tweets. Remarkably, one year later in the shared task of SMM4H 2020 [40], again a variant of pretrained BERT achieved the best performance [73]. Here, we propose an algorithm to improve on pretrained BERT in this task. Our model relies on multi-view learning and exploits unlabeled data. To our knowledge, our model is the first approach that improves on the domain-specific

pretrained BERT.

3.3 Proposed Method

Our model for extracting the reports of adverse drug effects rely on the properties of contextual neural word embeddings. Previous research on Word Sense Disambiguation (WSD) [96] has demonstrated that contextual word embeddings can effectively encode the context in which words are used. Although the representations of the words in a sentence are assumed to be distinct, they still possess shared characteristics. This is justified by the observation that the techniques such as self-attention [112], which a category of contextual word embeddings employ [33], rely on the interconnected relations between word representations.

This property is particularly appealing when documents are short, therefore, word representations, if are adjusted accordingly, can be exploited to extract multiple representations for a single document. In fact, previous studies have demonstrated that word contexts can be used to process short documents, e.g., see the models proposed in Liao and Grishman [72] and Karisani et al. [61] for event extraction using handcrafted features and contextual word embeddings respectively. Therefore, we use the word representations of drug mentions in user postings as the secondary view along the document representations of user postings in our model. As a concrete example, from the hypothetical tweet *"this seroquel hitting me"*, we extract one representation from the entire document and another representation from the drug name[1] Seroquel. In continue, we call these two views the document and drug views. Figure 3.1 illustrates these two views using BERT [33] as an encoder.

Given the two views we can either concatenate the two sets of features and train a classifier on the resulting feature vector or use a co-training framework as described

[1]We assume every user posting contains only one drug name, in cases that there are multiple names we can use the first occurrence.

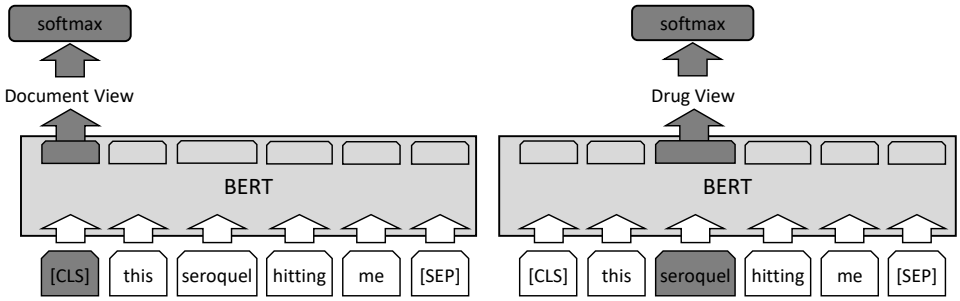

Figure 3.1: The illustration of the document and drug views in our model. We have used BERT as an encoder. See Devlin et al. [33] for the format of input tokens.

in Karisani et al. [61]. However, the former is not exploiting the abundant amount of unlabeled data, and the latter is resource intensive, because it is iterative, and also it has shown to be effective only in semi-supervised settings where there are only a few hundred training examples available. Therefore, below we propose an approach to effectively use the two views along the available unlabeled data in a supervised setting.

In the first step, we assume the classifier in each view is a student model and train this classifier using the pseudo-labels generated by the counterpart classifier. Since the labeled documents are already annotated, we carry out this step using the unlabeled documents. More concretely, let L and U be the sets of labeled and unlabeled user postings respectively. Moreover, let L_d and L_g be the sets of representations extracted from the document and drug views of the training examples in the set L; and let U_d and U_g be the document and drug representations of the training examples in the set U. To carry out this step, we train a classifier C_d on the representations in L_d and probabilistically, with temperature T in the softmax layer, label the representations in U_d. Then we use the association between the representations in U_d and U_g to construct a pseudo-labeled dataset of U_g. This dataset along its set of probabilistic pseudo-labels is used in a distillation technique [48] to train a classifier called $\widehat{C_g}$. Correspondingly, we use the set L_g to train a classifier C_g, then label the set U_g and use the association between the data points in U_g and U_d to construct a pseudo-labeled

dataset in the document view to train the classifier $\widehat{C_d}$.

The procedure above results in two classifiers $\widehat{C_d}$ and $\widehat{C_g}$. The classifier in each view is *initialized* by the knowledge transferred from the other view. However, the pseudo-labels that are used to train each classifier can be noisy. Thus, in order to reduce the negative impact of this noise, in the next step, we use the training examples in the sets L_d and L_g to further finetune these two classifiers respectively. To finetune $\widehat{C_d}$ we use the objective function below:

$$\mathcal{L}_d = \frac{1}{|L_d|} \sum_{v \in L_d} (1-\lambda) J(\widehat{C_d}(v), y_v) + \lambda J(\widehat{C_d}(v), C_d(v)), \tag{3.1}$$

where J is the cross-entropy loss, y_v is the ground-truth label of the training example v, and λ is a hyper-parameter to govern the impact of the two terms in the summation. The first term in the summation, is the regular cross-entropy between the output of $\widehat{C_d}$ and the ground-truth labels. The second term is the cross-entropy between the outputs of $\widehat{C_d}$ and C_d. We use the output of C_d as a regularizer to train $\widehat{C_d}$ in order to increase the entropy of this classifier for the prediction phase. Previous studies have shown that penalizing low entropy predictions increases generalization [87]. We argue that this is particularly important in the ADR task, where the data is highly imbalanced. Note that, even though C_d is trained on the training examples in L_d, the output of this classifier for the training examples is not sparse–particularly for the examples with uncommon characteristics. Thus, we use these soft-labels[2] along the ground-truth labels to train $\widehat{C_d}$. Respectively, we use the objective function below to finetune $\widehat{C_g}$:

$$\mathcal{L}_g = \frac{1}{|L_g|} \sum_{v \in L_g} (1-\lambda) J(\widehat{C_g}(v), y_v) + \lambda J(\widehat{C_g}(v), C_g(v)), \tag{3.2}$$

where the notation is similar to that of Equation 3.1. Here, we again use the output of

[2]Again, we use temperature T in the softmax layer to train using the soft-labels.

C_g as a regularizer to train $\widehat{C_g}$. In the evaluation phase, to label the unseen examples, we take the average of the outputs of the two classifiers $\widehat{C_d}$ and $\widehat{C_g}$.

Algorithm 2 Overview of VID

1: **procedure** VID
2: **Given:**
3: L : Set of labeled documents
4: U : Set of unlabeled documents
5: **Return:**
6: Two classifiers $\widehat{C_d}$ and $\widehat{C_g}$
7: **Execute:**
8: Derive two sets of representations L_d and L_g from L
9: Derive two sets of representations U_d and U_g from U
10: Use L_d to train classifier C_d
11: Use L_g to train classifier C_g
12: Use C_d to probabilistically label U_d
13: Transfer labels of U_d to U_g and use them to train $\widehat{C_g}$
14: Finetune $\widehat{C_g}$ using Equation 3.2
15: Use C_g to probabilistically label U_g
16: Transfer labels of U_g to U_d and use them to train $\widehat{C_d}$
17: Finetune $\widehat{C_d}$ using Equation 3.1
18: **Return** $\widehat{C_d}$ and $\widehat{C_g}$

Algorithm 4 illustrates our model (VID) in Structured English. On Lines 8 and 9 we derive the document and drug representations from the sets L and U. On Lines 10 and 11 we use the labeled training examples in the two views to train C_d and C_g. On Lines 12-14 we train and finetune $\widehat{C_g}$, and on Lines 15-17 we train and finetune $\widehat{C_d}$. Finally, we return $\widehat{C_d}$ and $\widehat{C_g}$. In the next section, we describe our experimental setup.

3.4 Experimental Setup

We evaluated our model in the largest publicly available ADR dataset, i.e., the SMM4H dataset. This dataset consists of 30,174 tweets. The training set in this dataset consists of 25,616 tweets of which 9.2% are positive. The labels of the test

set are not publicly available. The evaluation in the dataset must be done via the CodaLab website. We compare our model with two sets of baselines: 1) a set of baselines that we implemented, 2) the set of baselines that are available on the CodaLab website[3].

Our own baseline models are: **BERT**, the base variant of the pretrained BERT model [33], as published by Google. **BERT-D**, a domain-specific pretrained BERT model. This model is similar to the previous baseline, however, it is further pretrained on 800K unlabeled drug-related tweets that we collected from Twitter. We pretrained this model for 6 epochs using the next sentence prediction and the masked language model tasks. **BERT-D-BL**, a bi-directional LSTM model. In this model we used BERT-D followed by a bi-directional LSTM network [49].

We also compare our model with all the baselines available on the CodaLab webpage. These baselines include published and unpublished models. They also cover models that purely rely on machine learning models and those that heavily employ medical resources; see Weissenbacher and Gonzalez-Hernandez [114] for the summary of a subset of these models.

We used the Pytorch implementation of BERT [115]. we used two instances of BERT-D as the classifiers in our model–see Figure 3.1. Please note that using domain-specific pretrained BERT in our framework makes any improvement very difficult, because the improvement in the performance should be additive. We used the training set of the dataset to tune for our two hyper-parameters T and λ. The optimal values of these two hyper-parameters are 2 and 0.5 respectively. We trained all the models for 5 epochs[4]. During the tuning, we observed that the finetuning stage in our model requires much fewer training steps, therefore, we finetuned for

[3]Available at: https://competitions.codalab.org/SMM4H. The 2020 edition of the shared task is not online anymore. Therefore, for a fair comparison with the baselines, we do not use RoBERTa in our model, and instead use pre-trained BERT model.

[4]We used 20% of the training set for validation, and observed that the models overfit if we train more than 5 epochs.

Type	Method	F1	Precision	Recall
Our Impl.	BERT	0.57	0.669	0.50
	BERT-D	0.62	0.736	0.54
	BERT-D-BL	0.61	**0.749**	0.52
CodaLab	Sarthak	0.65	0.661	0.65
	leebean337	0.67	0.600	**0.76**
	aab213	0.67	0.608	0.75
	VID	**0.70**	0.678	0.72

Table 3.1: F1, Precision, and Recall of our model (VID) in comparison with the baselines.

only 1 epoch. In our model, we used the same set of unlabeled tweets that we used to pretrain BERT-D. This verifies that, indeed, our model extracts new information that cannot be extracted using the regular language model pretraining. As required by SMM4H we tuned for F1 measure. In the next section, we report the F1, Precision, and Recall metrics.

3.5 Results and Analysis

Table 3.1 reports the performance of our model in comparison with the baseline models–only the top three CodaLab baselines are listed here. We see that our model significantly outperforms all the baseline models. We also observe that the performances of our implemented baseline models are lower than that of the CodaLab models. This difference is mainly due to the gap between the size of the unlabeled sets for the language model pretraining in the experiments–ours is 800K, but the top CodaLab model used a corpus of 1.5M examples. This suggests that our model can potentially achieve a better performance if there is a larger unlabeled corpus available.

Table 3.2 reports the performance of VID in comparison to the classifiers trained on the document and drug representations. We also concatenated the two representations and trained a classifier on the resulting feature vector, denoted by *Combined-View*. We see that our model substantially outperforms all three models. Table

Method	F1	Precision	Recall
Document-View	0.62	0.736	0.54
Drug-View	0.63	0.706	0.570
Combined-View	0.63	0.745	0.543
VID	0.70	0.678	0.72

Table 3.2: F1, Precision, and Recall of VID in comparison to the performance of the classifiers trained on the document, drug, and combined views.

Method	F1	Precision	Recall
P-Doc-F-Doc	0.69	0.658	0.71
P-Drug-F-Drug	0.68	0.681	0.68
P-Doc-F-Drug	0.70	0.674	0.72
P-Drug-F-Doc	0.69	0.655	0.72
VID	0.70	0.678	0.72

Table 3.3: Performance of VID in comparison to the performance of the classifiers pretrained on the document or drug pseudo-labels (indicated by P-{•}) and finetuned on the document or drug training examples (indicated by F-{•}).

3.3 compares our model with the classifiers with different pretraining and finetuning resources. Again, we see that VID is comparable to the best of these models. We also observe 2 percent absolute improvement by comparing *P-Drug-F-Drug* and *P-Doc-F-Drug*, which signifies the efficacy of View Distillation.

In summary, we evaluated our model in the largest publicly available ADR dataset and compared with the state-of-the-art baseline models that use domain specific language model pretraining. We showed that our model outperforms these models, even though it uses a smaller unlabeled corpus. We also carried out a set of experiments and demonstrated the efficacy of our proposed techniques.

3.6 Conclusions

In this chapter we proposed a novel model for extracting adverse drug effects from user generated content. Our model relies on unlabeled data and a novel technique called view distillation. We evaluated our model in the largest publicly available ADR

dataset, and showed that it outperforms the existing BERT-based models.

Chapter 4

Active Learning

4.1 Introduction

In this chapter, we focus on Active Learning. The distinctive characteristics of active learning models make them especially appealing to the researchers in social media mining. Being robust towards the initial training set and addressing noisy labels [38], overcoming class imbalance challenge [30], and compensating for the lack of training data [31] are the well-understood qualities of Active Learning.

Here, we tackle the classification tasks tailored for query words. The applications of such tasks are abundant. In Online Public Health Monitoring where given the variants of a disease name we want to extract the positive report cases [86]. In Customer Satisfaction Monitoring where given a product or brand name we want to extract the true mentions of the product and visualize the outcome [3]. In Observation Extraction where given a real-world phenomenon we want to extract the relevant reported observations [31]. Or in Entity Filtering where given an entity name we want to filter out non-relevant user postings for the down-stream tasks–e.g., for Online Reputation Management [104]. Here, we exploit this shared quality and propose a novel unified active learning model for a range of tasks.

Our model, which we call COCOBA (Context-aware Co-testing with Bagging), is based on the idea that the content of user postings can be used in a context sensitive multi-view active learning model to resolve the disagreement over similar use cases. To achieve this, we use the properties of the problem and derive two contextual representations from user postings. Then we modify a multi-view active learning model to effectively use these representations. And finally, we use a query-by-committee model to increase robustness to the noise in user postings. We show that COCOBA is applicable to at least three important representative problems[1]. Namely we focus on: Personal Health Mention detection (PHM) [58] where given an illness name the goal is to detect the positive reports of the illness; Observation Extraction (OE) [124] where given a real-world event the goal is to extract the relevant reported observations; and Product Consumption Pattern identification (PCP) [52] where given a product the goal is to detect the number of usages of the product to calculate its penetration rate. Our experiments testify that our novel unified model consistently outperforms existing models.

The contributions of this chapter are as follows: **1)** We propose a novel unified multi-view active learning model to address the tasks tailored for a query in user-generated data. **2)** We carry out an extensive set of experiments and show that our model is applicable to at least three representative tasks. **3)** We show that our model consistently outperforms existing active learning models. **4)** We constructed a relatively large dataset of manually annotated tweets for PHM task that is publicly available. Our dataset consists of 18,000 tweets across three topics[2]: Parkinson's, cancer, and diabetes.

We believe our novel model, our detailed experiments, and our new dataset significantly push the state of the art, and also help practitioners to develop better systems

[1]Please see the cited articles for the discussion on the challenges of the selected tasks.

[2]Based on published reports [123] our dataset is the largest manually annotated dataset on this topic.

with smaller training sets. In the next section, we contrast COCOBA with existing models.

4.2 Background and Related Work

Background. In a typical active learning classification scenario, there is a small set of labeled data and a large set of unlabeled data available[3]. A predictive model is trained on the set of labeled data, and based on a criterion–either labeling cost or model performance–one data point from the set of unlabeled data is *queried* for annotation[4]. The annotated data point is added to the set of labeled data, and the procedure is iterated. The initial state in which the model has access to a small set of labeled data is called the *cold start* state. The learning algorithm that the model employs to explore the hypothesis space is called the ***base learner***; and the algorithm that the model uses to select the next unlabeled data point is called the *query strategy*. Majority of the active learning models rely on *informativeness*, *representativeness*, and *diversity* metrics to select their candidate data points [22]. Despite the significant advances in Active Learning over the last decades, the uncertainty-based sampling model [70] remains one of the most widely used and studied models [10, 56]. There are multiple methods to identify uncertainty in the base learner: the amount of entropy in the model prediction [99], the magnitude of gradients in back propagation [126], or the variance in successive predictions of the model [37] are a few examples.

Active Learning for user-generated content. Given the stability and usually satisfactory performance of the uncertainty-based sampling model, the majority of the successful applications of Active Learning in user-generated data rely on this model. [89] proposes a model for Crisis Report monitoring, [110] integrates Active

[3]The survey by Settles [99] and the article by Lowell et al., [75] provide a complete overview of Active Learning.

[4]Our criterion in this article is the model performance, and we assume that the annotation cost is uniform.

Learning with Semi-supervised Learning for entity recognition, and [104] proposes to combine the informativeness and representativeness metrics for entity recognition. [71] experiments with Active Learning for detecting symptoms in Chinese tweets, [106] reports the application of Active Learning in Adverse Drug Reaction monitoring (ADR) task, and [19] combines Active Learning with crowd sourcing for the ADR task. The authors in [57] propose a query diversity criterion for spam filtering on Twitter, and the authors in [128] combine Active Learning with crowd-sourcing to develop a pipeline for detecting job-related posts in social media. All of these studies use the uncertainty-based sampling model.

In this study, we focus on a multi-view contention reduction model [1] called co-testing [83]. The main idea of co-testing algorithm is to construct two views from input data and train a base learner on each view. Then query a data point from the set of unlabeled points that are assigned to the opposite classes by two base learners– these points are called ***contention*** points. To be able to use multi-view models, we derive two contextual representations from user postings. Then we modify the co-testing query strategy to utilize this contextual information and increase the gain in user annotations. We aim at a category of social media tasks tailored for a query word–or a closely related set of query words. Such tasks have many applications, ranging from Entity Filtering and Disease Mining to Crisis Management and Customer Satisfaction Monitoring. We show that our model, which we call COCOBA , is applicable to at least three representative tasks from different domains. Namely we focus on: Personal Health Mention detection (PHM), Observation Extraction (OE), and Product Consumption Pattern identification (PCP).

In summary, to our knowledge, our study is the first that proposes a unified active learning model for a range of social media tasks. It is also the first study that proposes to use a multi-view model to address these tasks. It is one of the very few works that

step beyond applying the traditional uncertainty-based model[5], and to our knowledge, it is the only work that extends an active learning model to effectively exploit the properties of the user-generated data.

4.3 COCOBA : Model Description

We begin this section by discussing the approach for extracting two contextual representations from user postings. Given two views, we can employ co-testing algorithm, however, the default co-testing algorithm is context independent. Therefore, we will modify the default co-testing query strategy to use the contextual information. Finally, we try to tackle the typically noisy language of user postings via a variance reduction technique.

4.3.1 Extracting Two Contextual Representations from User Postings

Our approach to construct two views from the user postings is inspired by the research on Word Sense Disambiguation (WSD) and their mainstream solutions, i.e., the contextual word embeddings. The neural contextual word embeddings are proven to encode the information required to effectively characterize the context in which the words occur [96]. To extract two contextual representations from the user postings, we extract one representation on the document level to capture the overall information of the user postings, and extract another representation on the word level to capture the context that the query words are used in. Because by definition the user postings always contain at least one of the query words then this task is always feasible. This approach is a derivation of the algorithm that we proposed in [61].

[5]The study by [31] employs the expected error reduction technique along a semi-supervised learning model.

We demonstrate this by outlining the task of extracting the true reports of diabetes on Twitter. Given the query words "diabetes" and "diabetic", we may observe the hypothetical tweet: *"Right now the only complication I've got with my **diabetes** is neuropathy, which isn't fun"*. Given this tweet, we can extract a feature vector on the tweet level which encodes the overall information of the tweet. Additionally, we can extract another feature vector on the word level to capture the context of the search term[6], i.e., the vector representation of the search term in: *"...my **diabetes** is neuropathy..."*.

Even though the feature vectors of the tweet level and word level views are not fully orthogonal, we argue that they still focus on different aspects of the text to represent the context of the tweet. Local and global feature sets have shown to be effective in other scenarios [38]. In the next section, we exploit this motif in an active learning framework.

4.3.2 Incorporating Context in Co-testing

Having two separate contextual representations for every user posting allows us to employ co-testing algorithm. However, the default co-testing query strategy and its variants [83, 38] are unable to fully utilize the contextual information that is stored in the representations. These variations mostly rely on the confidence of base learners to score the candidate data points, e.g., most confident disagreement between base learners. We argue that the contextual representations that we extract contain enough information to detect similar user postings, and this information can be used to resolve the disagreement over a set of user postings, rather than one single user posting. This can potentially lead to a better annotation choice during the active learning iterations. Based on this argument, we propose the following query strategy.

Let $\vec{d}$ and $\vec{w}$ be the document and word level representations of the user posting

[6]In the case that multiple search terms are used to collect the data, all the occurrences of the search terms in the tweets can be mapped to a single synthesized token.

t, and given t, let $Conf_D(\vec{d}|t)$ and $Conf_W(\vec{w}|t)$ be the confidence of the base learners for classification in the document level and word level views respectively. We define the score of the contention user posting t as follows:

$$score(t) = P_D(\vec{d}|t) \times Conf_D(\vec{d}|t) + P_W(\vec{w}|t) \times Conf_W(\vec{w}|t) \tag{4.1}$$

where $P_D(\vec{d}|t)$ and $P_W(\vec{w}|t)$ are the probabilities of the user posting t being generated by the distribution of the contention points in the document and word level views respectively. The terms $Conf_D(\vec{d}|t)$ and $Conf_W(\vec{w}|t)$ can be estimated by the output of the classifiers in the document and word level views respectively. To estimate $P_D(\vec{d}|t)$ and $P_W(\vec{w}|t)$, we first fit two density estimators on the vectors of the contention data points in each view to extract the empirical distribution of the population, and then use these estimators to calculate the probability of observing the data points[7].

Intuitively, Equation 4.1 assigns a higher score to the user postings that are confidently assigned to the opposite classes in two views, and are also close to the other set of contention points in each view. There are two advantages in employing this scoring function. First, scaling the confidence of the base learners by the probability densities naturally aggregates the benefits of contention reduction and density based query strategies. Second, assuming that the data points that are close to each other in the feature space are similar and likely to have the same label [23], by promoting the user postings that are close to the cluster of the contention points, we can effectively use the contextual information to resolve the disagreement over a set of similar user postings. This is particularly the case when a candidate data point and its adjacent points are projected into the same regions of the input feature space in both views.

Figure 4.1 demonstrates our query strategy. Each data point in the document representation space (the left panel) is associated to one data point in the keyword

[7]For the theoretical discussion regarding the density estimators see [102].

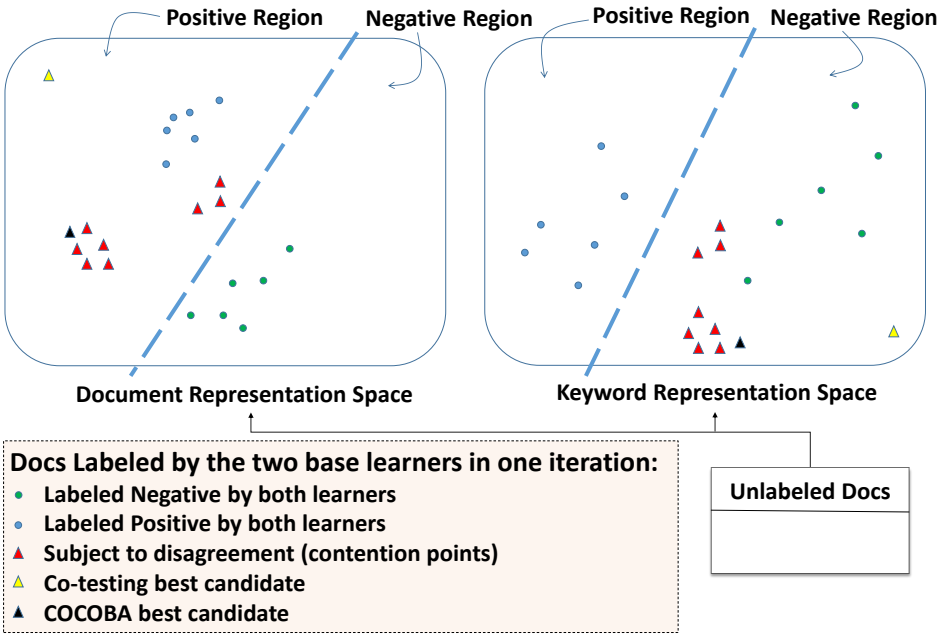

Figure 4.1: The document and word level views in COCOBA query strategy. The regular co-testing algorithm queries the contention point with the largest distance from the classifier decision boundary in two views (the yellow triangle). COCOBA queries the contention point which is closest to the set of other contention points and also has a large distance from the decision boundary in two views (the black triangle). Figure best viewed in color.

representation space (the right panel). The triangular data points are the set of contention tweets, i.e., the tweets that are assigned to the opposite classes by the classifiers in two views. The regular co-testing algorithm selects the data point with the largest distance from the classifier decision boundary–the dashed lines–i.e., the yellow data point. However, we select the data point which is close to the cluster of contention data points and also has a large distance from the classifier decision boundary, i.e., the black data point.

In the next sections, we use Equation 4.1 as the ranking function in our model.

4.3.3 Increasing Robustness to Noise

As pointed out by [58], the user postings in social media–particularly on the Twitter website–are highly noisy. They tend to be short, and suffer from inventive lexicons.

For instance, in our early example of extracting the reports of diabetes, a user posting may be added to the set of contention points and selected for annotation due to its unique figurative language. However, selecting another user posting for annotation might be a better choice to have a more diverse and representative training set. If we assume the relatively uninformative user postings are noise–which due to their unique characteristics may receive a high score by Equation 4.1–then we may be able to dampen their effect through variance reduction algorithms.

To address this issue we propose to employ bagging technique, which is empirically shown to reduce model variance [18]. In the discussed example, bagging can influence the score of the mentioned user posting, either through affecting the distribution of the contention user postings, or reducing the disagreement rate between two base learners. We use bagging as follows: In each iteration, we sample multiple subsets of user postings from the set of labeled data. On each subset, we train a pair of base learners as described in Section 4.3.1. For each pair of base learners, we use the model described in Section 4.3.2 to assign a score to all unlabeled user postings. Finally, the ultimate ranking list is constructed by aggregating the scores of the unlabeled data across the models.

Our approach for employing bagging is slightly different from the regular query-by-committee model [1]. In the regular query-by-committee model, one estimator is trained on each subset of data, and the best candidate data point is the data point which is subject to the most *disagreement* among the estimators. In our model, the candidate data points, for each subset, are the data points that are assigned to opposite classes by the base learners. Then, each predictive model votes for these contention points, and the best candidate data point is the one that is subject to the most *agreement* among the models.

4.3.4 Overview of Algorithm

Algorithm 4 summarizes *one iteration* of COCOBA . Lines 10-21 describe the training procedure, and Lines 22-31 describe the labeling procedure. The training stage begins by sampling from the set of labeled tweets; then two base learners are trained on two views of the sampled set. Next, two base learners are used to label the set of unlabeled tweets. The contention tweets are detected, and in each view one density estimator is fitted. The density models are used to approximate the probability mass values of every contention tweet. These steps are repeated for each sub-sample. To rank the set of unlabeled tweets, the prediction confidences and probability mass values are used in Equation 4.1 to score all the contention tweets. The top tweet is queried and added to the labeled set and all the sampled sets–Line 19 and Line 20. Finally, all the base learners are re-trained on the updated sampled sets. In the labeling stage, each pair of the base learners is used to label the test tweets–Line 25. To predict the final label a majority voting algorithm is employed–Lines 28-31. In the next section, we discuss the implementation details of COCOBA .

4.4 COCOBA : Implementation Details

In this section, first we discuss the feature vectors that we used in COCOBA . Then, we discuss the base learners and the density estimation models that we implemented. Finally, we explain the details of the bagging step.

Feature vectors (Section 4.3.1): We used neural contextual word embeddings to represent the two contextual representations discussed in Section 4.3.1. We used the BERT pre-trained base model [33], to extract the document level and word level views–the size of the vectors in this model is 768. For simplicity, if a task had multiple query words we assumed their contexts is comparable[8]–even though the approach in

[8]Recall that by definition, our tasks are defined for closely related search keywords.

Algorithm 3 One Iteration of COCOBA

procedure COCOBA
  Given:
    L : Set of labeled tweets
    U : Set of unlabeled tweets
    T : Set of test tweets
    K : Number of estimators
  Return:
    Labeled set of test tweets, and updated training set
  Execute:
    for $i \leftarrow 1$ to K **do**
      Sample a subset of L and store in $S[i]$
      Train two base learners on $S[i]$ and store in $BL[i][0]$ and $BL[i][1]$
      Use $BL[i][0]$ and $BL[i][1]$ to label the set U
      Store the contention tweets in $C[i]$, and their prediction confidences in $Conf[i][0]$ and $Conf[i][1]$
      Fit two density estimation models on two views of $C[i]$ and store them in $DS[i][0]$ and $DS[i][1]$
      Use $DS[i][0]$ and $DS[i][1]$ to calculate the probability mass values for all the tweets in $C[i]$ and store them in $P[i][0]$ and $P[i][1]$
    Plug the arrays $Conf$ and P into Equation (4.1) to calculate the aggregated score for tweets in C
    Rank all the tweets in C based on their score, and store the top one in W
    Query the label of W
    Add W to L and all the tweet sets stored in S
    Use the updated S to retrain the base learners of BL
    for t in T **do**
      $PCount \leftarrow 0$
      for $pair$ in BL **do**
        $label \leftarrow conf_{pair[0]}(t) + conf_{pair[1]}(t)$
        if $label \geq 0$ **then**
          $PCount \leftarrow PCount + 1$
      if $PCount \geq K/2$ **then**
        t is Positive
      else
        t is Negative
    Return T, L

[100] could have been leveraged to create a canonical term. Additionally, If a user posting contained more than one search term, we selected the first occurrence to construct the word level view.

Base learners (Section 4.3.1): We used a one-layer fully connected network as the base learner. To account for the increasing size of the training set during the active learning iterations, we also updated the BERT vectors every few hundred iterations by fine-tuning–see Section 4.5.3 for detail.

Density estimators (Section 4.3.2): We used a Parzen density estimator to approximate the density of the contention points [46]. For simplicity, we opted for a linear kernel model. We set the bandwidth hyper-parameter in the document level view to 30, and in the word level view to 45–these values were determined based on the average distance of the data points in each view which is independent of the labeled data.

Bagging details (Section 4.3.3): There is no widely accepted number of estimators for the models based on bagging [99]. We used 15 estimators in our implementation. For each estimator, we randomly sub-sampled 60% of the labeled set with replacement to be used as the training data.

4.5 Experimental Setup

We begin this section by describing the datasets, then we discuss the baselines, and finally, explain the experiments.

4.5.1 Datasets

We show that our model is applicable to three tasks: Personal Health Mention detection (PHM), Observation Extraction (OE), and Product Consumption Pattern identification (PCP). Below we describe the datasets.

	Training			Test		
Topic	Size	Neg	Pos	Size	Neg	Pos
Parkinson's	4096	84%	16%	2120	85%	15%
Cancer	3915	80%	20%	2091	79%	21%
Diabetes	4318	82%	18%	2097	86%	14%

Table 4.1: The number of tweets, and the percentage of the positive and negative tweets across the topics in Illness dataset.

Illness dataset: For PHM task, we constructed a dataset of English tweets across three different topics: Parkinson's disease, cancer, and diabetes. To collect the tweets related to diabetes, we used the search terms "diabetes" and "diabetic". We used the Twitter search API and retrieved a set of tweets–excluding retweets and replies– over the span of one year between 2018 and 2019. To create the training sets, we randomly sampled about 4,000 tweets for each topic from the 2018 data. To create the test sets, we randomly sampled about 2,000 tweets per topic from the 2019 data. To annotate the sampled sets, we followed the definition of Personal Health Mention detection problem (PHM), proposed in [58]. That is, the tweets that mention the health condition and contain a health report were labeled positive, otherwise, they were labeled negative. We hired one annotator to annotate the tweets. In order to validate the annotations, we randomly sub-sampled 10% of the labeled tweets, and hired another annotator to re-annotate the set. We found the inter-agreement rate to be 0.81 with Cohen Kappa test, which represents a substantial agreement between the two annotators [113]. Table 4.1 summarizes Illness dataset. We see that on average about 18% of the tweets are positive in each topic.

Observation dataset: For OE task, we used the dataset introduced in [124] on reporting flood incidents, which contains 4,000 tweets[9]. Each tweet is categorized as Direct-Observation, Indirect-Observation, or None. We assumed the tweets that make a direct observation are positive–which account for 17% of the dataset. With

[9]Available at https://crisisnlp.qcri.org/

preserving the original distribution, we sampled 1,000 tweets for the test set. Query keywords used to collect the dataset are "flood", "rain", and "overflow".

Product dataset: For PCP task, we used the dataset introduced in [52]. This dataset[10] consists of the tweets related to a medical product–influenza vaccine. A tweet is labeled positive if it reports receiving the medical product. There are 6,617 tweets in this dataset. We used the tweets posted in 2013 and 2014 in the training set, and the tweets posted in 2015 and 2016 in the test set. In the training set, we found 4,503 tweets for which 31% of them were positive. In the test set, we found 2,114 tweets for which 22% were positive.

4.5.2 Baselines

In this section, we describe the baseline models that we included in the experiments. We included one naive baseline (random sampling), one classic baseline (uncertainty sampling), one learning-from-data model (LAL), and one self-paced learning model (SPAL). In Section 5.4.4 we also compare our model with the co-testing algorithm. The input features were identical between all the models–as described in Section 4.4.

random: This baseline is without Active Learning. In each iteration, we randomly selected one tweet from the set of unlabeled tweets, and added to the labeled set.

uncertainty: We included the most widely used uncertainty-based model described in [99]. The output probability of the base learner was used as the confidence score.

lal: We included the model proposed in [64][11]. This model is an error reduction algorithm, which models the query sampling problem as a regression task. We report the *Iterative* variant, which is a stronger baseline and performed better. We used the suggested settings in the reference to set-up the model.

spal: We included the model proposed in [108][12]. This model is a self-paced method,

[10]Publicly available via the organizers of SMM4H workshop: https://aclweb.org/portal/content/smm4h

[11]Available at https://github.com/ksenia-konyushkova/LAL

[12]Available at https://github.com/NUAA-AL/ALiPy

which tries to maintain a balance between the informativeness and the easiness of queries through an objective function. We used the settings proposed in the reference to set-up the model.

4.5.3 Experimental Details

We trained and evaluated all of the models in each topic of Illness , Observation , and Product datasets separately. Following the argument in [78], we report the F1 of the models in the positive set. The rest of the experimental setup was identical to what is adopted in the active learning literature [99, 75]. In the cold start state, we randomly sampled 50 labeled tweets, and assumed that the rest of the labeled data is unlabeled. We report F1 measure in the test set as the training set is augmented with new labeled tweets. We fixed the initial set of labeled tweets across all the experiments, ensuring that all of the models have access to an identical set of tweets in their cold start state. Additionally, we repeated all the experiments 5 times and report the average of the experiments. In order to account for the increasing size of the training sets during the active learning iterations, every 350 iterations we fine-tuned the BERT model–mentioned in Section 4.4–and updated the entire set of tweet and word representations in all the baseline models.

4.6 Results and Analysis

In this section we report the main results, and then we provide an empirical analysis.

4.6.1 Results

Figures 4.2, 4.3, and 4.4 report the performance of the models in Illness , Observation , and Product datasets respectively. Additionally, Table 4.2 compares the performances at four different ratios of the training set sizes, i.e., 25%, 50%, 75%, and 100%. The

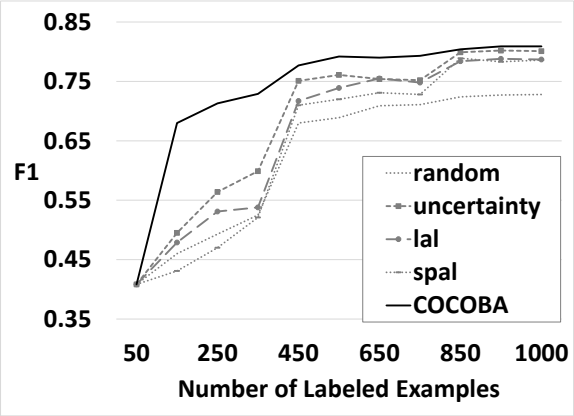

Figure 4.2: F1 of the models at varying training set sizes during the active learning iterations in Illness dataset.

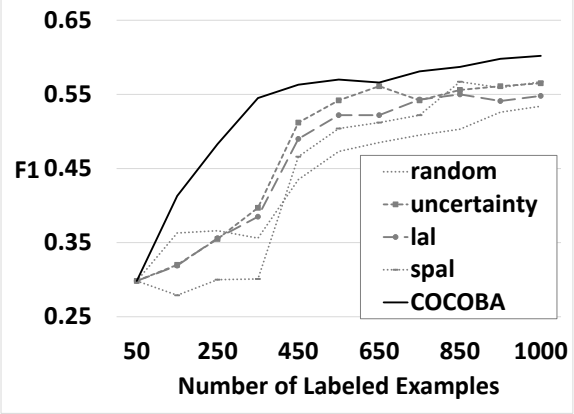

Figure 4.3: F1 of the models at varying training set sizes during the active learning iterations in Observation dataset.

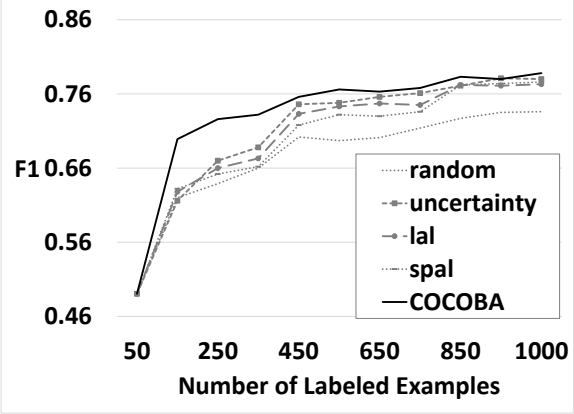

Figure 4.4: F1 of the models at varying training set sizes during the active learning iterations in Product dataset.

results confirm that–except in a few cases–all the models outperform *random* baseline, confirming that Active Learning is an effective strategy to approach these tasks. The results signify that our model COCOBA is consistently outperforming the baselines. This is particularly the case over the initial iterations. During these iterations our model employs two views to issue the queries, whereas the other models rely on one view. As more training data becomes available, and the pool of unlabeled data shrinks, the models converge–except in Observation dataset. Finally, the experiments show that *uncertainty* model is performing strikingly well, confirming the consistency of this model–discussed in Section 4.2. The authors in [10] report that under different problem settings state-of-the-art active learning models may be inferior to the uncertainty model.

4.6.2 Empirical Analysis

In Section 4.3.2 we argued that the regular co-testing algorithm can be further improved by exploiting the density of the contention points. We also proposed a method to incorporate this information using a Parzen density estimator. In Section 4.3.3 we argued that a variance reduction technique can mitigate the problem caused by the noisy language model. To support these arguments we report an ablation study by deactivating the two modules. Figure 4.5 reports the results of this experiment. We see that the performance of our model is noticeably higher than that of the new models.

A closer look at the graphs in Figures 4.2, 4.3, and 4.4 shows the existence of an elbow point in the early iterations. The improvement rate before reaching this point is dramatic and after this point it is slower. Our case by case inspection revealed that during the early iterations our scoring function–described in Section 4.3.2–can effectively use the density of the contention points which is coupled by the knowledge obtained by the two views. However, as the algorithm proceeds, the set of contention

	F1 in Illness dataset				F1 in Observation dataset				F1 in Product dataset			
Model	**25%**	**50%**	**75%**	**100%**	**25%**	**50%**	**75%**	**100%**	**25%**	**50%**	**75%**	**100%**
random	0.513	0.688	0.679	0.728	0.360	0.468	0.403	0.533	0.651	0.706	0.715	0.736
uncertainty	0.584	0.757	0.790	0.801	0.359	0.541	0.551	0.565	0.682	0.750	**0.774**	0.780
lal	0.530	0.731	0.778	0.787	0.340	0.514	**0.566**	0.547	0.669	0.734	0.763	0.772
spal	0.503	0.718	0.778	0.786	0.312	0.492	0.506	0.567	0.656	0.722	0.762	0.776
COCOBA	**0.723***	**0.788***	**0.804***	**0.809**	**0.522***	**0.573***	0.559	**0.602***	**0.738***	**0.761**	**0.774**	**0.788**

Table 4.2: F1 of the models at 25%, 50%, 75%, and 100% of the training set sizes during the active learning iterations. The improvements indicated by * are statistically significant–using paired t-test (adjusted P < 0.05).

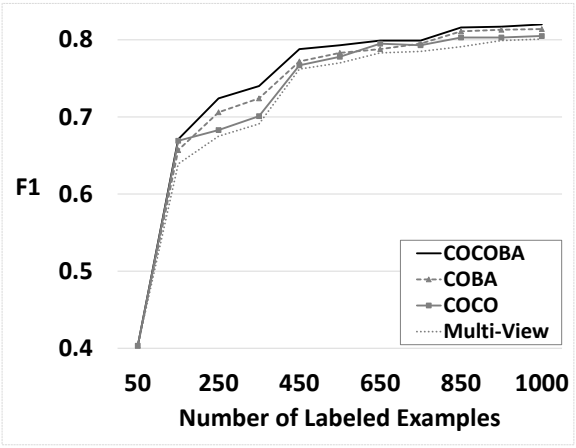

Figure 4.5: Ablation study of COCOBA by deactivating the two modules of our method, i.e., COBA (COCOBA without context) and COCO (COCOBA with no bagging) at varying training set sizes in Illness dataset.

points is exhausted and our model converges to a regular contention reduction algorithm. Thus, we conjecture that in the presence of larger set of unlabeled data COCOBA may yield even better results[13]. One particularly interesting quality of our model is the absence of critical hyper-parameters to tune. Excluding the hyper-parameters of base learners, in our experiments COCOBA was not sensitive to the number of estimators in the bagging step or the value of the bandwidth in the kernel density estimators[14].

In summary, we showed that our active learning model outperforms the state of the art in multiple settings. The authors in [10] report that active learning models typically show mixed results and fail to generalize to new scenarios. Thus, we selected three datasets and also included two state-of-the-art and two traditional baselines and showed that our model consistently perfroms well. The results suggest that our model can be potentially applied to a broader set of query-based classification tasks. This claim is to be further investigated. Additionally, there is still a set of social media tasks that are not based on queries e.g., sarcasm detection, hate speech detection,

[13]In terms of runtime, COCOBA is comparable to *lal*–which is also an ensemble. In the experiments, *spal* performed much slower.

[14]We tried {10,15,20} estimators, the results were consistent.

and fake news identification. Future work may explore these areas.

4.7 Conclusions

In this chapter we proposed a novel active learning model for short text classification tasks in user-generated data. Our model utilizes the contextual information of user postings in a multi-view active learning model, exploits the density of the contention points to increase the gain per query, and employs a query-by-committee step to address the usually noisy language of social media posts. Through an extensive set of experiments we showed that our model, COCOBA , is applicable to multiple tasks. Our code and a relatively large dataset that we constructed along the way are publicly available.

4.8 Ethical Considerations

We have used the Twitter API to collect a publicly available set of user postings. According to the US federal law[15] our dataset does not require an IRB review, because the data is public. Furthermore, our study does not violate the Twitter developer terms of service[16], because we do not store or use any personal identifier.

[15]Electronic Code of Federal Regulations in effect on February 26, 2021.
[16]Available at https://developer.twitter.com/en/dt

Chapter 5

Domain Adaptation

5.1 Introduction

In certain real-world scenarios there may not be enough annotated data available to train a classifier, but there may be sufficient training data available for a semantically similar task. For instance, in the task of detecting true reports of natural disasters, there may not be enough training data available for detecting earthquake incident reports, however, there may be a training set available for detecting the reports of a wildfire incident. In such cases Domain Adaptation is a promising direction.

In Domain Adaptation a classifier is trained in one domain (the *source* domain) and evaluated in another domain (the *target* domain). One of the fundamental assumptions of classification is that training and test data follow an identical distribution [85]. Therefore, a classifier trained in one domain, typically is a poor predictor for another domain. Thus, Domain Adaptation primarily tackles the domain shift challenge.

Domain adaptation models can be deployed in highly dynamic environments. Therefore, having techniques that can fully utilize available resources is of great value. This is particularly crucial in mining user-generated data, because the models trained

on this type of data already face considerable challenges–e.g., short document length, specialized and noisy language model, and imbalanced class distribution [4]. Existing techniques, while very effective, are still unable to fully exploit the vast amount of available *unlabeled* data from various domains. For instance, in the case of classifying documents related to an earthquake incident, existing approaches are able to use the labeled data collected for similar incidents, e.g., wildfire or flood incidents. However, there are no methods capable of using unlabeled data from these domains. Such scenarios are abundant. Product mining, disease mining, and mining documents related to rumours are a few examples. In the case of product mining, we may be interested in training a classifier for the documents related to Apple, and we may have a labeled dataset about Microsoft. With existing domain adaptation methods, we can train such a classifier for Apple, however, these models are unable to exploit the large amount of available unlabeled data for other tech companies in their training procedure–e.g., documents related to Oracle or Google. This is a major limitation and the subject of this work.

We hypothesize that in the single-source domain adaptation setting, where there is data available in a labeled source and an unlabeled target domains, the classifier performance can be enhanced by incorporating unlabeled data form additional semantically similar domains–which we call the auxiliary domains. Below we formally define our problem statement. Then, in the next section we provide an overview of related works. Afterwards, we present our model, then, we discuss our experimental setup and report the results.

Problem statement. We explore a domain adaptation setting in which a labeled source domain S and an unlabeled target domain T are available. The aim is to train a model on the source domain with a low prediction error in the target domain. As opposed to the previous work, we also assume that there is unlabeled data available from an additional set of *auxiliary* domains $\{A_i\}_{i=1}^{M}$. Therefore, the goal of our

research is to exploit this additional data to enhance the adaptation procedure from the domain S to the domain T.

5.2 Related Work

Our research problem is not domain specific. Our problem statement targets a common real world scenario, where in addition to the labeled source domain we have access to unlabeled data from various domains. Existing domain adaptation models are unable to exploit this unlabeled data, here, we aim to close this gap by proposing a model to incorporate this resource. Below we describe the areas that have inspired our research, and also discuss the techniques to incorporate unlabeled data in existing models.

Our proposed approach is categorized as a single source domain adaptation model. Existing models in this category are not able to use the unlabeled data from auxiliary domains. However, in order to be able to use these models as baseline models, one approach is to augment the unlabeled data from auxiliary domains with the unlabeled data from the target domain, then, align the source and target data. We particularly include a model termed JDDAC [25], which employs the correlation alignment metric [107] along a regularization term for clustering the data points in the feature space to enhance class discrimination. In the results section, in addition to this model, we also report a variant of JDDAC that combines the target and auxiliary data–we call this variant JDDAC-C.

Our work is also closely related to multiple-source domain adaptation [76]. In the multiple-source setting, it is assumed that the source data consists of labeled documents across multiple source domains, and the goal is to train a model for the target domain. As opposed to this setting, here we assume that there is only one source domain available and the remainder of auxiliary data is unlabeled and obtained

from multiple domains. Multiple-source domain adaptation models are unable to exploit the unlabeled data from auxiliary domains.

Our research provides an alternative view to multiple-target domain adaptation setting [28]. In the multiple-target setting, we assume that there is one labeled source domain and multiple unlabeled target domains available. The goal is to train a model that on average performs well across the target domains. While there is a clear connection between our setting and the multiple-target setting, here we aim to popularize the application of unlabeled data from multiple domains in the single source domain adaptation setting. Additionally, in our setting there is no necessity to explicitly make predictions in multiple target domains. Nonetheless, we use a model from this category in our comparisons. We include a model termed CCL [55], which consists of an ensemble of single source models that are collaboratively trained using a regularizer term based on the Kullback-Leibler (KL)-divergence.

5.3 Proposed Model

We begin this section by describing our notations, and then continue by providing an overview of our model.

Notations. We denote the labeled source domain S by $\{(\mathrm{x}_i^s, y_i^s)\}_{i=1}^{n_s}$, where n_s is the number of the documents in this domain and (x_i^s, y_i^s) is the *i-th* source document and its corresponding label. We also denote the unlabeled target domain T by $\{\mathrm{x}_i^t\}_{i=1}^{n_t}$, where n_t is the number of the documents in this domain and x_i^t is the *i-th* target document. In addition to the domains T and S we are also given a set of M unlabeled auxiliary domains $A_1, A_2, \ldots, A_M$ denoted by $\{(\mathrm{x}_i^{a_j})\}_{i=1}^{n_{a_j}}$, where n_{a_j} is the number of the documents in the *j-th* auxiliary domain and $\mathrm{x}_i^{a_j}$ is the *i-th* document in this domain. Similar to previous works [74, 107, 116] classifiers in our model consist of two modules: 1) an encoder module denoted by E, which takes a document as

input and projects it into a low dimensional space,[1] 2) a prediction module denoted by θ, which takes the projected document representation as input and performs the classification task.[2]

Our model consists of two classifiers: 1) the main classifier C_{main}, which is trained on the data in the source and target domains, 2) the auxiliary classifier C_{aux}, which is trained on the data in the source, the target, and the auxiliary domains.

The core idea of our model is that we can enhance the prediction in the regions that the main classifier is expected to perform weakly by training an auxiliary model that can participate in the classification along the main classifier. We assume these regions are those that the main classifier is most uncertain about.[3]

Algorithm 4 summarizes the training procedure of our model, called DAVUD (Domain Adaptation via Unlabeled auxiliary Data). The algorithm begins (Line 9) by training the new classifier C_{main} on the source and target data using Equation 5.1. Afterwards, the classifier C_{main} is used to label the documents in the set T (Line 10). Then, the documents with the most uncertain labels are removed from T (Line 11). Finally, the new classifier C_{aux} is trained (Line 12) on the data in S, $A_{\bullet}$, and the revised set T using Equation 5.2.

To train the classifier C_{main}, we use a discrepancy reduction metric [107] to eliminate the divergence between the distribution of the documents in the source and the target domains:

$$\mathcal{L} = \frac{1}{n_s} \sum_{i=1}^{n_s} J(C_{main}(\mathrm{x}_i^s), y_i^s) \; + \; \lambda \mathcal{D}(E(\mathrm{x}^s), E(\mathrm{x}^t)), \tag{5.1}$$

[1] an example the encoder is BERT [33], where a document can be encoded into a vector of 768 dimensions.

[2] an example the prediction module is one layer fully connected network followed by a softmax layer.

[3] In the experiments we used a threshold on the entropy of the main classifier output to detect the documents that are located in uncertain regions. This method is computationally efficient. However, one can also use Monte Carlo dropout [37] which adds slightly more computational overhead.

Algorithm 4 Overview of DAVUD

procedure DAVUD
  Given:
    S : Labeled source domain
    T : Unlabeled target domain
    $A_1, A_2, \ldots, A_M$: Unlabeled auxiliary domains
  Return:
    Trained classifiers C_{main} and C_{aux}
  Execute:
    Use Equation 5.1 and train a new classifier C_{main} on S and T
    Use C_{main} to generate pseudo-labels for the documents in T
    Remove the documents with the highest prediction entropy from T
    Use Equation 5.2 to train a new classifier C_{aux} on S, T, and $A_\bullet$
    Return C_{main} and C_{aux}

where the model is parameterized by C_{main}, J is the cross-entropy function, $E(\mathrm{x}^s)$ and $E(\mathrm{x}^t)$ are the output of the encoder for all source and target documents respectively, $\mathcal{D}$ is the discrepancy term, and $\lambda > 0$ is a scaling factor. We set λ to 10 in all of the experiments. The discrepancy term $\mathcal{D}$ governs the degree in which the parameters of the encoder must update to reduce the divergence between source and target representations. In this work we use the correlation alignment term [107] as the discrepancy term $\mathcal{D}$, which aligns the co-variance of the representations in the source and target domains.

To train the auxiliary classifier we use the objective function below:

$$\mathcal{L} = \frac{1}{n_s}\sum_{i=1}^{n_s} J(C_{main}(\mathrm{x}_i^s), y_i^s) \;+\; \frac{1}{n_t}\sum_{i=1}^{n_t} J(C_{aux}(\mathrm{x}_i^t), \overline{y}_i^t) \;+\; \lambda \sum_{i=1}^{M} \mathcal{D}(E(\mathrm{x}^s), E(\mathrm{x}^{A_i})), \tag{5.2}$$

where the auxiliary classifier is parameterized by C_{aux}, J is the cross-entropy function, $\overline{y}_\bullet^t$ denote the pseudo-labels generated by the main classifier for the target documents, $E(\mathrm{x}^s)$ and $E(\mathrm{x}^{A_\bullet})$ are the output of the encoder for all source and auxiliary documents respectively, $\mathcal{D}$ is the correlation alignment term, and $\lambda > 0$ is

a scaling factor. Equation 5.2 ensures that the auxiliary classifier yields the same outputs as those of the main classifier for the documents that are not located on the decision boundary–due to the first and the second terms in this equation. However, the outputs in the remaining regions is impacted by the data in the auxiliary domains, due to the third term in this equation. To label unseen documents, the outputs of C_{main} and C_{aux} are aggregated.

5.4 Experiments

In this section we evaluate our model and empirically analyze its properties. In Section 5.4.1, we describe the details of our experiments including the dataset, the metric, and the baselines. In Section 5.4.2 we compare our model with several recent baselines. Then, in Section 5.4.3, we describe our expectations regarding the experiments that we carry out to analyze our model. These experiments are reported in Section 5.4.4.

5.4.1 Setup

We used the dataset published by Zubiaga et al. [129] on rumour detection in our experiments. This dataset consists of 5 different domains, therefore, there are 20 pairs of source-target domains in the experiments. Following the argument made by Mccreadie et al. [78] about imbalanced datasets, we report the F1 measure. We repeated the experiments five times and report the average performance across the domain pairs. We used four baselines: JDDAC and JDDAC-C [25], CCL [55], and a source-only model. JDDAC uses CORAL [107] as the discrepancy term and a clustering regularization to increase discrimination between classes. JDDAC-C is the same as JDDAC, however, it aggregates the unlabeled target and auxiliary data. CCL is a multi-target model that uses an ensemble of classifiers and a KL divergence term to simultaneously predict in multiple domains. The source-only model uses the classifier

Method	F1	Precision	Recall
source-only	0.545	0.692	0.449
JDDAC	0.618	0.676	0.570
JDDAC-C	0.610	0.678	0.555
CCL	0.633	**0.664**	0.605
DAVUD	**0.650**	0.662	**0.638**

Table 5.1: Average results for single source unsupervised domain adaptation in the presence of unlabeled data from multiple source domains.

trained in the source domain for prediction in the target domain. We used BERT [33] in all of the baselines as the encoder, and used one layer fully connected network followed by a softmax layer as the prediction module in all of the models. We used this setting to factor-out the gain that can be obtained by out-of-the-box language model pretraining. This provides us with a more realistic platform for evaluation and makes improvement over the source-only model difficult. In our model, we set the entropy threshold for selecting uncertain documents to 0.95. Nonetheless, we observed almost no sensitivity to this hyper-parameter.

5.4.2 Main Results

Table 5.1 summarizes the results. We see that all of the models outperform the source-only model. This signifies the effectiveness of Domain Adaptation in this task. We also see that JDDAC outperforms JDDAC-C. This means that simply aggregating the the target and the auxiliary data is not the best solution. Finally, we observe that our model DAVUD outperforms the baselines. We particularly see that there is a noticeable improvement over the recent multi-target model CCL.

5.4.3 Expectations

In the next section we report an experiment on the impact of the number of auxiliary domains on model performance. We expect to observe an improvement in performance

as we add more domains. The documents in each domain follow the language model of the domain and therefore, their representations have a distinct distribution. Adding more domains adds more diversity to the training set of the auxiliary classifier. With a larger training set, we will have less bias and will find a better optima with Equation 5.2. In other words, the model becomes aware of the regions that it wouldn't be otherwise.

In the next section, we report an experiment on the impact of the availability of the documents in auxiliary domains on model performance. Again we expect to observe an improvement in performance as we add more documents to auxiliary domains. The documents in each auxiliary domain are a sample of the entire documents in the set. Therefore, they form an empirical distribution and not the theoretical distribution. Having more documents in each auxiliary domain increases the similarity between the empirical and the theoretical distributions. This, in turn, reduces model bias and therefore as argued above, the reduction in bias helps to find a better local optima with Equation 5.2.

In the next section, we report an ablation study to evaluate the impact of the auxiliary classifier. We expect that by removing the auxiliary classifier we observe a degradation in model performance. The auxiliary classifier uses the third term in Equation 5.2 to explore the noisy regions–note that the decision boundary is fixed in the other regions by the first two terms. Such information is not given to the main classifier, therefore, we expect to see a degradation if we remove the auxiliary classifier from the model.

5.4.4 Empirical Analysis

We report the impact of the number of available auxiliary domains on model performance in Table 5.2. We see that as more auxiliary domains become available, the performance slightly improves. We particularly see that increasing the number of

# Auxiliary Domains	F1	Precision	Recall
1	0.640	0.657	0.623
2	0.647	0.667	0.628
3	0.650	0.662	0.638

Table 5.2: Performance at varying number of available auxiliary domains.

Available Auxiliary Data	F1	Precision	Recall
30%	0.645	0.654	0.637
60%	0.648	0.657	0.640
100%	0.650	0.662	0.638

Table 5.3: Performance at varying percentage of available documents in every auxiliary domain.

auxiliary domains, increases the the model recall.

We report the impact of the availability of unlabeled documents in the auxiliary domains on model performance in Table 5.3. Again, we see the same pattern, as more unlabeled data becomes available the performance improves.

Finally, we report the contribution of each individual classifier in our model in Table 5.4. We see that the auxiliary classifier has a better performance than the main classifier. Considering that the two classifiers yield the same outputs in non-noisy regions, the improvement in the auxiliary classifier indicates that this classifier can make better predictions in the noisy regions. This experiment verifies our initial hypothesis regarding the application of unlabeled data in Domain Adaptation.

Model	F1	Precision	Recall
C_{main}	0.639	0.644	0.635
C_{aux}	0.647	0.659	0.637
DAVUD	0.650	0.662	0.638

Table 5.4: Performance of individual classifiers compared to DAVUD . This experiment is equivalent to an ablation study.

5.5 Conclusions

In this work, we presented a novel research problem and also proposed a new model to address the task. Our model consists of two classifiers. A main classifier which is trained regularly, and an auxiliary classifier which is trained on the labeled data in the source domain, pseudo-labeled data in the target domain, and unlabeled data from a set of auxiliary domains. We showed that our model outperforms existing state-of-the-art approaches in a rumour detection dataset. We also demonstrated the effect of available unlabeled data on model performance and empirically verified our initial hypothesis regarding the application of unlabeled data in Domain Adaptation.

Chapter 6

Conclusions and Future Work

In this work, we presented various novel ideas and techniques to mitigate the data scarcity problem for filtering information in user-generated data. We explored Semi-Supervised Learning and proposed a novel algorithm based of self-training to use unlabeled data. We focused on one of the most challenging classification tasks in Twitter, i.e., the adverse drug reaction monitoring, and proposed a new algorithm to transfer the knowledge from one view to another view using unlabeled data in Multi-View Learning. Then, we investigated Active Learning and presented a new query strategy for a range of classification tasks. Finally, we formulated a new research problem in Domain Adaptation, and aimed to develop a model using unlabeled data from related domains to enhance model performance.

As future work, one can investigate the efficacy of our semi-supervised learning model (Chapter 2) in supervised settings. Our algorithm relies on the pretraining and finetuning paradigm, previous studies demonstrated the effectiveness of this paradigm in general natural language processing tasks [33]. In Chapter 4, we focused on query-based classification tasks. However, a large set of filtering tasks are not query based and active learning models to effectively use their properties are yet to be developed. For instance, hate speech detection or sarcasm detection do not admit to the prop-

erties of the query-based tasks. In Chapter 5, we opened an entirely new avenue for research in Domain Adaptation. As such, we may observe more research studies in this area in the future. One particular direction that we may explore, is the application of noisy labels in the adaptation process, i.e., distant supervision [79] in domain adaptation.

Bibliography

[1] Naoki Abe and Hiroshi Mamitsuka. Query learning strategies using boosting and bagging. In *Proc of the 5th ICML*, pages 1–9, 1998. ISBN 1-55860-556-8.

[2] Steven Abney. *Semisupervised Learning for Computational Linguistics*. Chapman & Hall/CRC, 1st edition, 2007. ISBN 1584885599.

[3] Raj Agnihotri, Rebecca Dingus, Michael Y. Hu, and Michael T. Krush. Social media: Influencing customer satisfaction in b2b sales. *Industrial Marketing Management*, 53:172 – 180, 2016.

[4] Mohammad Akbari, Xia Hu, Liqiang Nie, and Tat-Seng Chua. From tweets to wellness: Wellness event detection from twitter streams. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 87–93, 2016.

[5] Firoj Alam, Shafiq Joty, and Muhammad Imran. Domain adaptation with adversarial training and graph embeddings. In *Proceedings of the 56th ACL*, pages 1077–1087, Melbourne, Australia, 2018. Association for Computational Linguistics.

[6] Thayer Alshaabi, David R Dewhurst, and et al. The growing echo chamber of social media: Measuring temporal and social contagion dynamics for over 150 languages on twitter for 2009–2020. *arXiv preprint arXiv:2003.03667*, 2020.

[7] Hadi Amiri. Neural self-training through spaced repetition. In *Proceedings of the 2019 Conference of NAACL*, pages 21–31, Minneapolis, Minnesota, June 2019.

[8] Eiji Aramaki, Sachiko Maskawa, and Mizuki Morita. Twitter catches the flu: Detecting influenza epidemics using twitter. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1568–1576, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-937284-11-4.

[9] Eric Arazo, Diego Ortego, Paul Albert, and et al. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks, IJCNN, July 19-24, 2020*, pages 1–8. IEEE, 2020.

[10] Josh Attenberg and Foster Provost. Inactive learning?: Difficulties employing active learning in practice. *KDD Exp. News.*, 12:36–41, 2011. ISSN 1931-0145.

[11] David Bamman and Noah A. Smith. Contextualized sarcasm detection on twitter. In *Proceedings of the Ninth International Conference on Web and Social Media, ICWSM 2015, University of Oxford, Oxford, UK, May 26-29, 2015*, pages 574–577, 2015.

[12] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

[13] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th ICML*, ICML '09, page 41–48, New York, NY, USA, 2009. Association for Computing Machinery.

[14] David Berthelot, Nicholas Carlini, Ian J. Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *NeurIPS 2019, 8-14 Vancouver, BC, Canada*, pages 5050–5060, 2019.

[15] Avrim Blum and Tom M. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh COLT, 1998, Madison, Wisconsin, USA, July 24-26, 1998.*, pages 92–100, 1998.

[16] Tom B Brown, Benjamin Mann, and et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[17] Cristian Buciluundefined, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD*, KDD '06, page 535–541, 2006.

[18] Peter Buhlmann and Bin Yu. Analyzing bagging. *Ann. Statist.*, 30(4):927–961, 08 2002.

[19] Sophie Burkhardt, Julia Siekiera, Josua Glodde, Miguel A Andrade-Navarro, and Stefan Kramer. Towards identifying drug side effects from social media using active learning and crowd sourcing. In *Pacific Symposium of Biocomputing (PSB)*, pages 319–330, 2020.

[20] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka, and Tom M. Mitchell. Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fourth AAAI*, page 1306–1313, 2010.

[21] Paola Cascante-Bonilla, Fuwen Tan, Yanjun Qi, and Vicente Ordonez. Curriculum labeling: Self-paced pseudo-labeling for semi-supervised learning. *arXiv preprint arXiv:2001.06001*, 2020.

[22] Haw-Shiuan Chang, Shankar Vembu, Sunil Mohan, Rheeya Uppaal, and Andrew McCallum. Overcoming practical issues of deep active learning and its applications on named entity recognition. *arXiv preprint arXiv:1911.07335*, 2019.

[23] Olivier Chapelle, Jason Weston, and Bernhard Schölkopf. Cluster kernels for semi-supervised learning. In *NIPS 15*, pages 601–608. MIT Press, 2003.

[24] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, editors. *Semi-Supervised Learning*. The MIT Press, 2006. ISBN 9780262033589.

[25] Chao Chen, Zhihong Chen, Boyuan Jiang, and Xinyu Jin. Joint domain alignment and discriminative feature learning for unsupervised deep domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3296–3303, 2019.

[26] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[27] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020.

[28] Ziliang Chen, Jingyu Zhuang, Xiaodan Liang, and Liang Lin. Blending-target domain adaptation by adversarial meta-adaptation networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[29] Jian Cheng and Kongqiao Wang. Active learning for image retrieval with co-svm. *Pattern recognition*, 40(1):330–334, 2007.

[30] Jongwon Choi, Kwang Moo Yi, Jihoon Kim, Jincho Choo, Byoungjip Kim, Jin-Yeop Chang, Youngjune Gwon, and Hyung Jin Chang. Vab-al: Incorporating class imbalance and difficulty with variational bayes for active learning. *arXiv preprint arXiv:2003.11249*, 2020.

[31] Hang Cui, Tarek Abdelzaher, and Lance Kaplan. A semi-supervised active-learning truth estimator for social networks. In *The World Wide Web Conference*, WWW '19, page 296–306, New York, NY, USA, 2019. Association for Computing Machinery.

[32] James R Curran, Tara Murphy, and Bernhard Scholz. Minimising semantic drift with mutual exclusion bootstrapping. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*, volume 6, pages 172–180. Bali, 2007.

[33] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc of the 2019 NAACL*, pages 4171–4186, 2019.

[34] Mei Sheng Duh, Pierre Cremieux, Marc Van Audenrode, Francis Vekeman, Paul Karner, Haimin Zhang, and Paul Greenberg. Can social media data lead to earlier detection of drug-related adverse events? *Pharmacoepidemiology and Drug Safety*, 25(12):1425–1433, 2016.

[35] Anna Atefeh Farzindar and Diana Inkpen. Natural language processing for social media. *Synthesis Lectures on Human Language Technologies*, 13(2):1–219, 2020.

[36] Tommaso Furlanello, Zachary Chase Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born-again neural networks. In *Proceedings of the*

35th ICML, Stockholm, Sweden, July 10-15, 2018, volume 80, pages 1602–1611, 2018.

[37] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *Proc of the 34th ICML*, pages 1183–1192, 2017.

[38] Rayid Ghani, Rosie Jones, Tom Mitchell, and Ellen Riloff. Active learning for information extraction with multiple view feature sets. In *Proc of the 20th ICML*, pages 26–34, 2003.

[39] Su Golder, Gill Norman, and Yoon K Loke. Systematic review on the prevalence, frequency and comparative value of adverse events data in social media. *British Journal of Clinical Pharmacology*, 80(4):878–888, 2015.

[40] Graciela Gonzalez-Hernandez, Ari Z. Klein, Ivan Flores, Davy Weissenbacher, Arjun Magge, Karen O'Connor, Abeed Sarker, Anne-Lyse Minard, Elena Tutubalina, Zulfat Miftahutdinov, and Ilseyar Alimova, editors. *Proceedings of the Fifth Social Media Mining for Health Applications Workshop & Shared Task*, Barcelona, Spain (Online), December 2020. Association for Computational Linguistics.

[41] Roberto Gonzalez-Ibaez, Smaranda Muresan, and Nina Wacholder. Identifying sarcasm in twitter: A closer look. In *Proceedings of the 49th ACL*, HLT '11, page 581–586, 2011.

[42] Suchin Gururangan, Tam Dang, Dallas Card, and Noah A. Smith. Variational pretraining for semi-supervised text classification. In *Proceedings of the 57th ACL*, pages 5880–5894, Florence, Italy, July 2019.

[43] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of ACL*, 2020.

[44] Mohammad Sadegh Hajmohammadi, Roliana Ibrahim, Ali Selamat, and Hamido Fujita. Combination of active learning and self-training for cross-lingual sentiment classification with density analysis of unlabelled samples. *Inf. Sci.*, 317:67–77, 2015.

[45] Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. Revisiting self-training for neural sequence generation. In *8th International Conference on Learning Representations, ICLR 2020, April 26-30, 2020*. OpenReview.net, 2020.

[46] Nils-Bastian Heidenreich, Anja Schindler, and Stefan Sperlich. Bandwidth selection for kernel density estimation: a review of fully automatic selectors. *AStA Advances in Statistical Analysis*, 97(4):403–433, 2013.

[47] Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In *Proceedings of the 36th ICML, California, USA*, volume 97, pages 2712–2721, 2019.

[48] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

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

[50] Nils Holzenberger, Andrew Blair-Stanek, and Benjamin Van Durme. A dataset for statutory reasoning in tax law entailment and question answering. *arXiv preprint arXiv:2005.05257*, 2020.

[51] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th ACL*, pages 328–339, 2018.

[52] Xiaolei Huang, Michael C Smith, Michael J Paul, Dmytro Ryzhkov, Sandra C Quinn, David A Broniatowski, and Mark Dredze. Examining patterns of influenza vaccination in social media. In *Workshops at the 31st AAAI*, 2017.

[53] Muhammad Imran, Carlos Castillo, Fernando Diaz, and Sarah Vieweg. Processing social media messages in mass emergency: A survey. *ACM Comput. Surv.*, 47(4):67:1–67:38, June 2015. ISSN 0360-0300.

[54] Muhammad Imran, Ferda Ofli, Doina Caragea, and Antonio Torralba. Using AI and social media multimodal content for disaster response and management: Opportunities, challenges, and future directions. *Inf. Process. Manag.*, 57(5): 102261, 2020.

[55] Takashi Isobe, Xu Jia, Shuaijun Chen, Jianzhong He, Yongjie Shi, Jianzhuang Liu, Huchuan Lu, and Shengjin Wang. Multi-target domain adaptation with collaborative consistency learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[56] Khaled Jedoui, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Deep bayesian active learning for multiple correct outputs. *arXiv preprint arXiv:1912.01119*, 2019.

[57] Zhuoren Jiang, Zhe Gao, Yu Duan, Yangyang Kang, Changlong Sun, Qiong Zhang, and Xiaozhong Liu. Camouflaged Chinese spam content detection with semi-supervised generative active learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3080–3085, Online, July 2020. Association for Computational Linguistics.

[58] Payam Karisani and Eugene Agichtein. Did you really just have a heart attack?: Towards robust detection of personal health mentions in social media.

In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, pages 137–146, 2018.

[59] Payam Karisani and Negin Karisani. Semi-supervised text classification via self-pretraining. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, WSDM '21, page 40–48. Association for Computing Machinery, 2021.

[60] Payam Karisani, Farhad Oroumchian, and Maseud Rahgozar. Tweet expansion method for filtering task in twitter. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 55–64, 2015.

[61] Payam Karisani, Joyce C. Ho, and Eugene Agichtein. Domain-guided task decomposition with self-training for detecting personal events in social media. In *Proceedings of The Web Conference 2020*, WWW '20, page 2411–2420, 2020.

[62] Payam Karisani, Jinho D. Choi, and Li Xiong. View distillation with unlabeled data for extracting adverse drug effects from user-generated data. In *Proceedings of the Sixth Social Media Mining for Health (#SMM4H) Workshop and Shared Task*, pages 7–12, Mexico City, Mexico, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.smm4h-1.2.

[63] James Kirkpatrick, Razvan Pascanu, and et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

[64] Ksenia Konyushkova, Raphael Sznitman, and Pascal Fua. Learning active learning from data. In *Advances in Neural Information Processing Systems 30*, pages 4225–4235. Curran Associates, Inc., 2017.

[65] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learn-

ing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[66] Alex Lamb, Michael J Paul, and Mark Dredze. Separating fact from fear: Tracking flu infections on twitter. In *Proc of the 2012 NAACL: Human Language Technologies*, pages 789–795, 2013.

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

[68] Jinhyuk Lee, Wonjin Yoon, and et al. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36 (4):1234–1240, 09 2019.

[69] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems 31*, pages 7167–7177. 2018.

[70] David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In *Proc of the 17th SIGIR*, pages 3–12, 1994. ISBN 0-387-19889-X.

[71] Chao Li, Xin Kang, and Fuji Ren. Medweb task: Identify multi-symptoms from tweets based on active learning and semantic information. In *Proc of the 13th NTCIR*, pages 5–8, 2017.

[72] Shasha Liao and Ralph Grishman. Using prediction from sentential scope to build a pseudo co-testing learner for event extraction. In *Proc of 5th IJCNLP*, pages 714–722, 2011.

[73] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen,

Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[74] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 97–105.

[75] David Lowell, Zachary C. Lipton, and Byron C. Wallace. Practical obstacles to deploying active learning. In *Proc of the 2019 EMNLP*, pages 21–30, 2019.

[76] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation with multiple sources. In *Advances in neural information processing systems*, pages 1041–1048, 2009.

[77] Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of *Psychology of Learning and Motivation*, pages 109 – 165. Academic Press, 1989.

[78] R. Mccreadie, C. Buntain, and I. Soboroff. Trec incident streams: Actionable information on social media. In *Proc of the 16th ISCRAM*, 2019.

[79] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, 2009.

[80] Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., 1 edition, 1997. ISBN 0070428077, 9780070428072.

[81] Marie-Francine Moens, Juanzi Li, and Tat-Seng Chua. *Mining user generated content*. CRC press, 2014.

[82] Subhabrata Mukherjee and Ahmed Hassan Awadallah. Uncertainty-aware self-training for text classification with few labels, 2020.

[83] Ion Muslea, Steven Minton, and Craig A. Knoblock. Active learning with multiple views. *J. Artif. Int. Res.*, 27(1):203–233, October 2006. ISSN 1076-9757.

[84] Alexandra Olteanu, Carlos Castillo, Fernando Diaz, and Emre Kıcıman. Social data: Biases, methodological pitfalls, and ethical boundaries. *Frontiers in Big Data*, 2:13, 2019. ISSN 2624-909X. doi: 10.3389/fdata.2019.00013.

[85] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, 22(10):1345–1359, 2010. doi: 10.1109/TKDE.2009.191. URL https://doi.org/10.1109/TKDE.2009.191.

[86] Michael J. Paul and Mark Dredze. *Social Monitoring for Public Health*. Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan & Claypool Publishers, 2017.

[87] Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey E. Hinton. Regularizing neural networks by penalizing confident output distributions. In *5th ICLR 2017, Toulon, France, April 24-26, 2017*, 2017.

[88] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 NAACL*, pages 2227–2237, New Orleans, Louisiana, June 2018.

[89] Daniela Pohl, Abdelhamid Bouchachia, and Hermann Hellwagner. Batch-based

active learning: Application to social media data for crisis management. *Expert Systems with Applications*, 93:232 – 244, 2018.

[90] Siyuan Qiao, Wei Shen, Zhishuai Zhang, Bo Wang, and Alan Yuille. Deep co-training for semi-supervised image recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[91] Colin Raffel and et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

[92] Piyush Rai, Avishek Saha, Hal Daumé III, and Suresh Venkatasubramanian. Domain adaptation meets active learning. In *Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing*, pages 27–32, 2010.

[93] Sebastian Ruder and Barbara Plank. Strong baselines for neural semi-supervised learning under domain shift. In *Proceedings of the 56th ACL*, pages 1044–1054. Association for Computational Linguistics, 2018.

[94] Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. Asymmetric tri-training for unsupervised domain adaptation. In *Proceedings of the 34th ICML*, ICML'17, page 2988–2997, 2017.

[95] M. Sajjadi, M. Javanmardi, and T. Tasdizen. Mutual exclusivity loss for semi-supervised deep learning. In *2016 IEEE (ICIP)*, pages 1908–1912, Sep. 2016.

[96] Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. Sensembert: Context-enhanced sense embeddings for multilingual word sense disambiguation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8758–8765. AAAI Press, 2020.

[97] Dale Schuurmans and Finnegan Southey. Metric-based methods for adaptive model selection and regularization. *Machine Learning*, 48(1):51–84, 2002.

[98] H Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371, 1965.

[99] Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[100] Peng Shi and Jimmy Lin. Simple bert models for extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*, 2019.

[101] Weiwei Shi, Yihong Gong, Chris Ding, Zhiheng MaXiaoyu Tao, and Nanning Zheng. Transductive semi-supervised deep learning using min-max features. In *Proceedings of (ECCV)*, pages 299–315, 2018.

[102] Bernard W Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

[103] Anders Søgaard. Simple semi-supervised training of part-of-speech taggers. In *Proceedings of the ACL 2010*, page 205–208, USA, 2010.

[104] Damiano Spina, Maria-Hendrike Peetz, and Maarten de Rijke. Active learning for entity filtering in microblog streams. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, page 975–978, New York, NY, USA, 2015. Association for Computing Machinery.

[105] Bharath Sriram, Dave Fuhry, Engin Demir, Hakan Ferhatosmanoglu, and Murat Demirbas. Short text classification in twitter to improve information filtering. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 841–842, 2010.

[106] Gabriel Stanovsky, Daniel Gruhl, and P Mendes. Recognizing mentions of adverse drug reaction in social media using knowledge-infused recurrent models. In *Proc of the 15th EACL*, pages 142–151, 2017.

[107] Baochen Sun and Kate Saenko. Deep CORAL: correlation alignment for deep domain adaptation. In *Computer Vision - ECCV 2016 Workshops - Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III*, volume 9915 of *Lecture Notes in Computer Science*, pages 443–450, 2016.

[108] Ying-Peng Tang and Sheng-Jun Huang. Self-paced active learning: Query the right thing at the right time. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 5117–5124. AAAI Press, 2019.

[109] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems 30*, pages 1195–1204. 2017.

[110] Van Cuong Tran, Ngoc Thanh Nguyen, Hamido Fujita, Dinh Tuyen Hoang, and Dosam Hwang. A combination of active learning and self-learning for named entity recognition on twitter using conditional random fields. *Knowledge-Based Systems*, 132:179 – 187, 2017.

[111] Elena Tutubalina and Sergey Nikolenko. Combination of deep recurrent neural networks and conditional random fields for extracting adverse drug reactions from user reviews. *Journal of healthcare engineering*, 2017, 2017.

[112] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference*

on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pages 5998–6008, 2017.

[113] A. J. Viera and J. M. Garrett. Understanding interobserver agreement: the kappa statistic. *Fam Med*, 37(5):360–363, May 2005.

[114] Davy Weissenbacher and Graciela Gonzalez-Hernandez, editors. *Proceedings of the Fourth Social Media Mining for Health Applications SMM4H Workshop & Shared Task*, Florence, Italy, August 2019. Association for Computational Linguistics.

[115] Thomas Wolf, Lysandre Debut, and et al. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019.

[116] Dustin Wright and Isabelle Augenstein. Transformer based multi-source domain adaptation. *arXiv preprint arXiv:2009.07806*, 2020.

[117] Jiawei Wu, Lei Li, and William Yang Wang. Reinforced co-training. In *Proceedings of the 2018 NAACL*, pages 1252–1262, New Orleans, Louisiana, June 2018.

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

[119] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[120] Chang Xu, Dacheng Tao, and Chao Xu. A survey on multi-view learning, 2013.

[121] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd ACL*, pages 189–196, Cambridge, Massachusetts, USA, June 1995.

[122] Andrew Yates and Nazli Goharian. Adrtrace: Detecting expected and unexpected adverse drug reactions from user reviews on social media sites. In *Advances in Information Retrieval*, pages 816–819, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[123] Zhijun Yin, Lina M Sulieman, and Bradley A Malin. A systematic literature review of machine learning in online personal health data. *J. of American Medical Informatics Association*, 26:561–576, 2019.

[124] Kiran Zahra, Muhammad Imran, and Frank O. Ostermann. Automatic identification of eyewitness messages on twitter during disasters. *Information Processing & Management*, 57(1):102107, 2020.

[125] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *6th ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[126] Ye Zhang, Matthew Lease, and Byron C. Wallace. Active discriminative text representation learning. In *Proc of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pages 3386–3392, 2017.

[127] Ying Zhang, Tao Xiang, Timothy M. Hospedales, and Huchuan Lu. Deep mutual learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[128] Yunpeng Zhao, Mattia Prosperi, Tianchen Lyu, Yi Guo, and Jing Bian. Integrating crowdsourcing and active learning for classification of work-life events from tweets. *arXiv preprint arXiv:2003.12139*, 2020.

[129] Arkaitz Zubiaga, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Peter Tolmie. Analysing how people orient to and spread rumours in social media by looking at conversational threads. *PloS one*, 11(3):e0150989, 2016.