Domain-Guided Task Decomposition with Self-Training for Detecting Personal Events in Social Media

Payam Karisani
Emory University
payam.karisani@emory.edu

Joyce C. Ho
Emory University
joyce.c.ho@emory.edu

Eugene Agichtein
Emory University
eugene.agichtein@emory.edu

Abstract

Mining social media content for tasks such as detecting personal experiences or events, suffer from lexical sparsity, insufficient training data, and inventive lexicons. To reduce the burden of creating extensive labeled data and improve classification performance, we propose to perform these tasks in two steps: 1. Decomposing the task into domain-specific sub-tasks by identifying key concepts, thus utilizing human domain understanding; and 2. Combining the results of learners for each key concept using co-training to reduce the requirements for labeled training data. We empirically show the effectiveness and generality of our approach, Co-Decomp, using three representative social media mining tasks, namely Personal Health Mention detection, Crisis Report detection, and Adverse Drug Reaction monitoring. The experiments show that our model is able to outperform the state-of-the-art text classification models–including those using the recently introduced BERT model–when small amounts of training data are available.

CCS Concepts

• **Information systems** → **Search results deduplication**; **Social networks**; **Document filtering**; **Information extraction**; **Clustering and classification**; *Nearest-neighbor search.*

Keywords

classification, semi-supervised learning, social media analysis, event detection

ACM Reference Format:
Payam Karisani, Joyce C. Ho, and Eugene Agichtein. 2020. Domain-Guided Task Decomposition with Self-Training for Detecting Personal Events in Social Media. In *Proceedings of The Web Conference 2020 (WWW '20), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3366423.3380304

1 Introduction

Social networks, such as Twitter and Facebook, have become inseparable parts of societies. A broad spectrum of topics are shared and discussed in the networks every day, and this has turned them into a suitable means for the online public monitoring. The applications include, but not limited to, consumer opinion mining [18], stock market prediction [7], sarcasm detection [12], and user reputation management [3]. These cases signify that social networks, e.g., Twitter, went beyond their initial purpose years ago–which was being simple personal messaging tools[1]. Personal Event Detection is an example of the online public monitoring. For instance, in the case of Personal Health Mention detection [30], the aim is to mine and track any individual health event. Scalability, real-time surveillance, and rapid response to potential outbreaks are the main advantages of this task when it is used inside a public health monitoring system. Another example is Crisis Report detection [16] through social media, which aims to mine user postings and alert humanitarian institutions and agencies during natural disasters.

Even though social networks are a valuable source of information, mining user postings comes with several challenges. For instance, the tasks usually suffer from the lack of enough training data [21]. Even in the cases that there is enough resources to construct a training set, the class distributions might be highly imbalanced [1, 33]. Thus, having machine learning models to perform well in this data scarce environment is of great value.

In classification tasks a common practice is to first extract a set of features, either manually or through representation learning, and then train a classifier over the resulting feature vectors. While training a single classifier over the entire content is a standard practice, an end-to-end classifier may require substantial amount of annotated data. Instead, for a subset of tasks, we can use domain knowledge to decompose the problem into a set of sub-tasks, and use a separate learner to tackle each one individually. This can lead to the development of models which are equipped with domain understanding and require less training data. For instance, if the task is cancer surveillance on the Twitter website, in the tweet *"I Just went to my Oncology appointment at the Hospital!!! Praying that it's not cancer"*, we might be able to infer the class label from the contextual information of either the word "I" or "cancer". Therefore, we can solve each classification problem individually and aggregate the results.

We propose Co-Decomp, a semi-supervised model that can classify short text for problems with a set of sub-tasks. While our model can be potentially applied to any problem that is centered around a group of concepts or entities, we focus on three personal event detection tasks; because they usually suffer from the lack of training data and imbalanced class distributions, as mentioned earlier. Namely, we focus on Personal Health Mention detection [21], Crisis Report detection [16], and Adverse Drug Reaction monitoring [33], and show that Co-Decomp can outperform state-of-the-art classifiers in semi-supervised settings. In summary, our contributions are:

[1] https://www.nytimes.com/2010/10/31/technology/31ev.html

WWW '20, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7023-3/20/04.
https://doi.org/10.1145/3366423.3380304

- We propose Key Concept Sets to decompose a particular category of text classification problems, referred to as decomposable problems, into a set of sub-tasks.
- We introduce a co-training model to effectively utilize the problem decomposition, and reduce the need for training data.
- We show that a category of personal event detection tasks fall into the class of decomposable problems. We carry out comprehensive experiments on four datasets, and show that our model reduces the need for training data, and can outperform state-of-the-art classifiers in the low data regime.

Together, these contributions significantly advance the state of the art in the personal event detection and related tasks. Next, we review the related work to place our contributions in context.

2 Related Work

Our model falls into the category of divide-and-conquer algorithms, and this family of algorithms have been employed in text classification before. For example, a pipeline of filtering steps have been applied to documents in order to filter out the confidently negative ones [1]. The main difference between our model and the pipelining approach is that we initially decompose the task into a set of subtasks that can be complementary, whereas in the case of pipelining, the final classifier still needs to tackle the same initial task. Additionally, our decomposition reduces the need for training data such that the task can be solved in semi-supervised settings. Our model is also deeply connected to the information extraction [26], relation classification [41], and semantic role labeling [35] tasks in natural language processing. In addition to be agnostic towards the number of entities and their relation type, which are pivotal in the mentioned tasks, our proposal is mainly a new perspective on tackling text classification problems in semi-supervised settings. Thus, in contrast to these tasks, we are not concerned about entity extraction or relation classification, but our focus is on how to decompose the classification problem such that the resulting pieces are good representations.

Another related topic, which has inspired our work, is Annotator Rationale technique introduced in [40]. The authors use manual annotations within documents to derive new training examples. To take into account the possible biases in the synthesized examples, they also adjust the classification model accordingly. Similar to their approach, our model also relies on the annotations within each document. The manual annotation of the sentences within each document raises efficiency concerns about the cost of preparing the training data. However, they carry out a set of extensive experiments and show that the effort of labeling the sentences within each document is not significant. Specifically, they show that when the classification task is predetermined but the set of candidate sentences and words is open and unknown, human annotators can rapidly scan the text and highlight the important sections. In our model, this issue is even less concerning, because once the set of Key Concept Sets is defined, they can be automatically *discovered and highlighted*; and ready to *annotate*. The main difference between Co-Decomp and Annotator Rationale is that our model relies on domain-guided problem decompositions to derive new training examples. Consequently, Co-Decomp is able to divide the initial problem into potentially smaller tasks, and tackle each one individually.

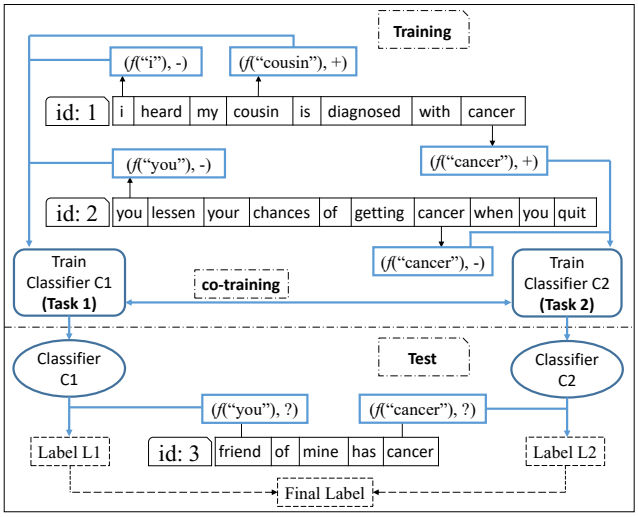

Figure 1: Illustration of Co-Decomp method for detecting personal health mentions (cancer), where the task is decomposed into detecting positive human mentions (Class C1) and actual health event (cancer) mentions (Class C2). In the training phase, classifiers for C1 and C2 are trained over the labeled instances of C1 and C2. To label the unseen examples in the test phase, the predictions of classifiers for C1 and C2 are aggregated.

In the context of the personal health event detection, the closest work to ours is the WESPAD model introduced in [21]–We have included the model as a baseline. The underlying assumption of WESPAD is that there is enough data to extract good lexical features. Even though this model works well in supervised settings, in Section 6 we will show that it performs poorly in semi-supervised settings. Finally, in contrast to general semi-supervised learning models such as transductive [19], graph-based [42], generative [29], or hybrid models [5], our model is a novel method to incorporate domain knowledge into the learning process. Therefore, our solution can be still implemented in any of the machine learning frameworks which can regulate the interaction between multiple learners, e.g., [6, 14, 32]. In summary, our work advances the state of the art by identifying the problem decomposition in text classification tasks, proposing an effective co-training model to utilize the technique, and showing the superiority of the model in semi-supervised settings across multiple tasks.

3 Co-Decomp: Method Description

We begin this section by presenting an example, and explaining the intuition behind Co-Decomp. Consider the task of cancer surveillance in Twitter. The common practice is to extract a set of feature vectors from user postings–manually or automatically–and train a classifier over the extracted vectors. However, this approach has some drawbacks. First, the classifier needs to learn a mapping function from the linguistic patterns that appear in tweets to the class

labels. Even if the patterns are not semantically and directly related to the task, the classifier still needs to learn to discard them. Second, no domain understanding is used to tackle the problem. With sufficient training data, classifiers can ultimately discover the right feature set, and detect the correct mapping function. But this is not the case in semi-supervised settings with insufficient labels. To address these issues, our proposal is to decompose the task into a set of complementary sub-tasks, and tackle each one individually.

For instance, in the case of cancer surveillance, as shown in Figure 1, the original task can be decomposed into (1) detecting positive mentions of humans (marked by "Task 1" in Figure 1) and (2) detecting positive mentions of the word cancer (marked by "Task 2" in Figure 1). A tweet may contain multiple human mentions and cancer mentions, as shown in the case of the tweet "id: 1" in Figure 1. The mentions that refer to the human with the reported cancer are labeled positive, while the remaining mentions are labeled as negative. Two separate classifiers are trained over the mentions of humans and the mentions of cancer, respectively. The two classifiers are then aggregated in a co-training framework to result a robust model. In the following subsections, we define Key Concept Sets and decomposable problems. Then, we describe our model Co-Decomp, which utilizes the problem decomposition in a co-training framework.

3.1 Decomposable Text Classification Tasks

In this section, we introduce Key Concept Sets, which allow us to decompose a problem into a set of sub-tasks. Let π be the distribution over document and class pairs π: $(d, c) \in D \times \{0, 1, \cdots\}$, and V be the vocabulary set. Also let $f : (w, d, i) \mapsto \mathbb{R}^{\kappa}$ be a vector-valued function which captures the contextual information of the *i-th* occurrence of term w in document d, and maps it into an n-dimension space of real values. Given threshold γ, we define K to be a Key Concept Set if: **1)** $K \subseteq V$ **2)** $\forall w, v \in K$: $\|f(w, :, :) - f(v, :, :)\| \leq \gamma$ **3)** There exists distribution φ over the value of f and class pairs φ: $(f, c) \in f \times \{0, 1, \cdots\}$ such that $\forall d \in D, \exists\, w \in K, \exists\, (w, d, i)$: $(d, c_k) \sim \pi \Leftrightarrow (f(w, d, i), c_k) \sim \varphi$.

Thus, a Key Concept Set is a subset of the vocabulary set–attribute (1)–in which its members are contextually similar–governed by γ in attribute (2)–and if we train a classifier on the context vectors of its members, there is at least one term in every document where its label is the same as the document label–attribute (3). We call a classification problem decomposable, if there exists at least one Key Concept Set in the vocabulary set.

Key Concept Sets simplify the classification inference, since the classification over the documents can be replaced with the classification over the key-concept-set terms in the documents. More specifically the advantages are: First, the dimension of the context function f is usually much smaller than the size of the vocabulary set V, thus feature selection becomes easier. Second, since intuitively there are limited ways of using a word in context, there is less variance in distribution φ in comparison to distribution π, which can virtually model the entire language. Third, as we will discuss in the next section, we can rely on our domain understanding to identify Key Concept Sets, and therefore, equip the model with a knowledge that otherwise it would need to learn through more training data. This will help the model to generalize better with smaller number of training examples.

3.2 Domain-Guided Key Concept Set Identification

To identify Key Concept Sets we rely on human knowledge. Our model is proposed for the tasks which are tailored for specific entities or concepts. Therefore, we assume once the problem statement is defined, the identification of the subject entities will be straightforward. To demonstrate that this assumption holds in some real-world scenarios, in Section 4 we present three tasks that follow this motif. Namely, we discuss Personal Health Mention detection [21], Crisis Report detection [16], and Adverse Drug Reaction monitoring [33] tasks. We show that, even though there is a large body of work behind each one, they can be viewed as decomposable problems and addressed similarly. This is striking, since to the best of our knowledge so far no connection has been made between these three tasks. We conjecture that there may be an even larger set of tasks that have the same attributes and can be potentially decomposable–one particularly interesting case which we may explore in the future is the product review task in social media.

A short note on the role of human knowledge in our model. Our model is not a human-in-the-loop algorithm. Once the training stage begins, no human supervision is required. In the regular learning, the learner mines the entire feature space to detect the conclusive subset of features. To do so, the model requires enough training data. We are in fact eliminating this step, and reducing document level classification to word level classification. In other words, we rely on human knowledge to relocate one of the data exploration steps from the learning stage to the design stage. Thus, the learning procedure still occurs, however, in a smaller feature space with less variation. The idea of reliance on human knowledge is not novel. For instance, the distant supervision model [26], assumes the user has enough domain expertise to introduce a large noisy dataset. Co-training model [6], assumes the user has enough information about the task to introduce two subsets of features. And the data programming model [32], assumes the user has enough knowledge to provide the learner with a set of heuristics. Interestingly, all of these models are proposed for the low data regime.

3.3 Co-Decomp: Exploiting Task Decomposition for Semi-Supervised Learning

The contextual similarity between the members of a Key Concept Set, that was introduced in the previous section, insures that the sets that can potentially capture different aspects of documents are not combined [2]. Being able to capture multiple views of the same problem–even loosely–is shown to be effective in models such as co-training [6, 28]. Thus, we propose to use co-training to utilize the problem decomposition[3]. Algorithm 1 illustrates the training

[2]The similarity condition–introduced by γ–does not by itself guarantee orthogonality of the features. However, if two subsets of vocabularies are contextually different, and their context vectors are indicators of the document class, then, we assume they can capture different aspects of the document.

[3]We consider the binary classification problems, however, our model can also generalize to multi-label classification problems.

procedure of Co-Decomp. Since there could be multiple occurrences of the members of a Key Concept Set in a document, the problem is viewed as a multiple instance learning problem [9], where each document is called an example, and each set member occurrence in the document is called an instance. The procedure is iterative, and in every iteration the set of labeled instances of every example are used to train a classifier. Then the classifiers are used to label the instances of the unlabeled data, and according to the multiple instance learning selection metric the examples are labeled–e.g, based on their most confident positive instance. Finally, the most confident positive and negative examples of each Key Concept Set are added to the pool of the labeled training data.

Algorithm 1 Training Procedure of Co-Decomp

procedure TRAIN
  Given:
    L : Set of labeled examples
    U : Set of unlabeled examples
    J : Number of key concept sets
    K : Number of iterations
  Return:
    $C[1 \dots J]$: array of classifiers trained on instances of each key concept set in L and U
  Execute:
    for $i \leftarrow 1$ to K **do**
      for $j \leftarrow 1$ to J **do**
        Train C_j on **instances** of key concept set j in L
        Use C_j and multiple instance learning metric to label the **examples** in U
          Store the most confident positive and negative **examples** in EP_j and EN_j
      for $j \leftarrow 1$ to J **do**
        Delete EP_j and EN_j in U and add them to L
    Return $C[1 \dots J]$

Algorithm 2 illustrates the test procedure. The array of classifiers trained in Algorithm 1 are used to label the unseen examples. To label every example, each classifier is used to calculate the probability of the example being positive, and then a simple criterion similar to the one proposed in [6] is used to label the example. In a more complicated scenario, each classifier could have a prior reliability score, however, for simplicity we opted for the model proposed in [6].

A short note on the orthogonality of Key Concept Sets. Multi-view learning techniques [39] are effective even in the presence of correlated views. Particularly in the case of co-training algorithm, numerous studies have shown that the initial assumption of orthogonality between the views was over-strong. For instance, Balcan, Blum, and Yang [4] propose a theoretical framework and argue that if the classifiers in each view are sufficiently strong PAC-learners, then the initial constraint on the views can be substantially relaxed. In the application domain, Nigam and Ghani [28] show that by randomly splitting lexical features, one can construct two separate views for co-training algorithm. Jones et al., [20], propose Co-EM algorithm for information extraction. Their two feature sets

Algorithm 2 Test Procedure of Co-Decomp

procedure TEST
  Given:
    J : Number of key concept sets
    $C[1 \dots J]$: array of classifiers
    $Test$: Test set
  Return:
    Labeled test set
  Execute:
    for *exmpl* **in** $Test$ **do**
      for $j \leftarrow 1$ to J **do**
        Use C_j and multiple instance learning metric to find the probability of *exmpl* being positive
          Store the corresponding probability in P_j
      if $\prod_{i=1}^{J} P_i \geq \prod_{i=1}^{J} (1 - P_i)$ **then**
        exmpl is positive
      else
        exmpl is negative
    Return $Test$

are noun phrases and their surrounding contexts. They show that even though these two feature sets are highly correlated, they can be still effective in a co-training model.

In the next section, we use Co-Decomp to propose a solution to a set of personal event detection tasks in social media.

4 Applications: Personal Event Detection

In this section, we show that Co-Decomp is applicable to three important real-world scenarios: Personal Health Mention detection (PHM), Crisis Report detection (CR), and Adverse Drug Reaction monitoring (ADR). We show that these three tasks are decomposable problems and have a unified solution.

4.1 Personal Health Mention Detection

Personal Health Mention detection (PHM) is described in [21], and concerns *"identifying postings in social data, which not only contain a specific **disease**, but also mention a **person** who is affected"*. To employ Co-Decomp, we regard the two entities that are present in the problem statement as the Key Concept Sets: 1) The set of all human mentions. 2) The disease keyword mentioned in the task. We argue that both of the sets loosely follow the conditions which are described in Section 3.1. Intuitively, all the human mentions have similar contextual vectors (condition (2)); and by construction, there is at least one human mention that determines the label of the user posting (condition (3)). The same reasoning applies to the second Key Concept Set; there must be at least one occurrence of the disease keyword which determines the label of the user posting (condition (3)).

After identifying the Key Concept Sets, the next step is to prepare the training set. We implemented a tool to automatically extract the human mentions and highlight the mentions for manual annotation–similar to Annotators Rationale method [40]. Since user postings are short, we assumed all the disease mentions in the positive user postings were positive instances of the second Key Concept Set. All the human mentions and disease mentions of

the negative user postings were assumed to be negative instances. Thus, the extraction and annotation of the disease mentions, the extraction of the human mentions, and also the annotation of the negative human mentions are all fully automatic. Only the annotation of the positive human mentions is manual–after a tweet is labeled positive, the user is asked to highlight the affected human mention.

We followed Algorithm 1 for training the classifiers, and augmented the labeled data with unlabeled data. To add positive instances of Key Concept Sets to the labeled data, we selected the most confidently labeled instance and its most probable counterpart in the other Key Concept Set–we effectively stored the set of instances as labeled data. For example, assume the classifier trained over disease mentions confidently labeled the word "cancer" positive in the tweet *"a friend of me is diagnosed with cancer"*. Then, we added this instance to the set of labeled data, and also used the classifier trained over the human mentions to label the mentions of human in the tweet, i.e., "friend" and "me", and selected the most confident one and added to the labeled data. To add negative instances of Key Concept Sets to the labeled data, we selected the example which all of its instances were confidently labeled negative, and added to the labeled data. To test our model, we followed Algorithm 2.

4.2 Crisis Report Detection

Crisis Report detection (CR) as defined in [17] concerns[4] *"detecting reports of casualties and/or injured **people** due to the **crisis**. Or reports and/or questions about missing or found **people**"*. We regard the two entities mentioned in the problem statement as the Key Concept Sets: 1) The set of all human mentions. 2) The crisis keyword mentioned in the task. In this study, we focus on the reports which were posted during an earthquake. To prepare the training set and evaluate our model, we followed the same procedure that we used for the PHM problem.

4.3 Adverse Drug Reaction Monitoring

Adverse Drug Reaction monitoring (ADR) is defined in [11], and is meant for *"detecting **personal** injuries resulting from medical **drug** use"*. We regard the two entities mentioned in the problem statement as the Key Concept Sets: 1) The set of all human mentions. 2) The set of all drug mentions. To prepare the training set and evaluate our model, we re-implemented all the decisions that we made for the PHM problem.

4.4 Implementation Details

In this section we provide a detailed explanation of the modules and components used in Co-Decomp to address the tasks mentioned earlier. Specifically, we discuss the context function described in Section 3.1, the classifiers described in Section 3.3, the extraction of the Key Concept Sets mentioned in Sections 4.1, 4.2, and 4.3; and finally the learning representation of the Key Concept Sets.

Context Function. We used contextual embeddings as the context function described in Section 3.1. We used the BERT model [8], even though other models such as ELMO could be also used [31]. We used the base variant, and pre-trained it on Twitter data–see below for the details about pre-training.

Used Classifiers. We used logistic regression classifier as the learners mentioned in Section 3.3. Thus, after fine-tuning the embeddings using the training data, we used the contextual features to train the logistic regression classifiers[5]. The Mallet implementation of logistic regression [24] was used in this step.

Key Concept Set Extraction. To detect human mentions we used a weak rule-based classifier. The accurate detection of human mentions is out of our research scope; here, we aim to show that even a weak human mention detector can contribute to the performance. The rules for human mention detection were as follows: Using the Stanford Named Entity Recognition (NER) tagger [10] we labeled all of the "PERSON" tags. Using the Stanford Parts of Speech (POS) tagger [36] we labeled all of the personal pronoun tags except for the word "it". We also labeled all of the Twitter mentions–indicated by the sign "@". Finally, we used a dictionary of 240 words manually collected from the Web to cover the remaining cases. Since not all of the human mentions are explicitly referred in user postings, we also used a simple noisy rule based human mention synthesizer: If a sentence started with a past tense verb we inserted the word "i" at the beginning. If a sentence started with an adjective we inserted "i am" at the beginning. If a sentence started with a past participle verb we inserted "i have" at the beginning. If a sentence started with a present continuous verb we inserted "i am" at the beginning. And finally, if a sentence started with "is", we replaced it with "i am". We empirically developed these rules, and as mentioned earlier, to achieve a better performance they can be replaced with more sophisticated models.

The model relies on the positive mentions of the humans in the positive tweets–described in Section 4.1. One of the authors of the article supplied the annotations. The rules for the annotation were as follows: The explicit mentions of the humans which are associated with the event (either disease, or disaster, or drug injury) should be annotated. If the explicit mention does not exist, the implicit mentions which are associated with the event should be annotated.

To extract the disease Key Concept Set mentioned in Section 4.1, we conducted a keyword search for the disease name in the task description. For instance if the task is about Parkinson's disease surveillance, the disease Key Concept Set contains the word {Parkinson's}. To extract the crisis Key Concept Set mentioned in Section 4.2, we also performed a keyword search for the incident in the task description. As mentioned earlier, in this study we focused on an earthquake incident. Thus, the crisis Key Concept Set contains the keywords {earthquake, quake}. To extract the drug Key Concept Set described in Section 4.3, we used the list of drug names published in [33], and conducted a keyword search for the drug names in the list.

Learning Key Concept Set Representations. Since the human mentions are lexically different–although we expect them to be contextually similar–we replaced all of them with a mask token HUM_TOK and learned the representation. To do so, we collected a set of 7,598,545 random tweets by Twitter API in October 2018, replaced all the human mentions with this token, and pre-trained

[4]There are also other variations of this task, e.g., displacing or evacuating people, during different incidents [2].

[5]We made this decision based on implementation considerations.

the base variant of the BERT model for 10 epochs–with default hyperparameters as mentioned in [8]. The word vectors used in the personal health mention detection and crisis report detection tasks are the output of this model. To unify the representations of the drug mentions, we used the list of drug names published in [33] to collect a set of 28,710 tweets containing the drug names [6], replaced the names with DRUG_TOK and further pre-trained the above mentioned model for 10 epochs. The word vectors used in the adverse drug reaction monitoring task are the output of this model.

5 Experimental Setup

In this section we first describe the datasets that we used in the experiments, and then, we review the baselines that we implemented, and finally discuss the training procedure.

5.1 Datasets

For personal health mention detection task we used two datasets. First, the dataset introduced in [23], which we call FLU dataset [7]. At the time of downloading this dataset, there were still 2,837 tweets available to crawl, in which 49% of them are negative–awareness tweets–and 51% of them are positive–report actual cases of flu. Second, the dataset introduced in [21], which we call PHM dataset. At the time of downloading this dataset, there were 7,192 tweets available to crawl. This dataset consists of 6 diseases: Alzheimer's, heart attack, Parkinson's, cancer, depression, and stroke. All of these sub-datasets are highly imbalanced, positive examples span between 11% to 40% of the cases. For crisis report detection task, we used the earthquake related dataset introduced in [17], which we call CRISIS dataset. This dataset contains a set of 2,013 tweets which were posted during the California earthquake in 2014 [8]. Only 11% of the tweets in this dataset are positive cases of injured or missing people. For adverse drug reaction monitoring task, we used the dataset introduced in [33], which we call ADR dataset. At the time of crawling the dataset, there were 4,355 tweets available. This dataset is also highly imbalanced, only 10% of the tweets are positive cases of drug injures. Table 1 summarizes the 4 datasets and their target prediction tasks.

5.2 Baselines

To compare the performance of our method, we implemented the following methods and classifiers. Model hyperparameters were tuned based on the training folds and datasets, and in most cases their optimal values were dependent on the training data.

NB. A Naive Bayes classifier is trained over unigrams and bigrams, as it has been shown to perform well with small training sets [27].

EM. We implemented the Expectation Maximization algorithm proposed by [29], which is known to work well in semi-supervised settings. We experimented with the set of {10,20,50,100} for the number of unlabeled documents.

[6]We used the Twitter streaming API for four weeks, and collected about 300K tweets, however, found that the majority of them were duplicates.

[7]We used the infection vs awareness version of FLU dataset, for detailed information about the datasets please refer to the cited articles.

[8]Reference [17] also introduces a few more datasets. We used the California earthquake version, and split by the injured and missing vs other categories.

Name	Target	# Tweets	% Positive
FLU [23]	Positive flu cases	2837	51
PHM [21]	Alzheimer	1256	18
PHM [21]	Heart attack	1219	13
PHM [21]	Parkinson's	1040	11
PHM [21]	Cancer	1242	21
PHM [21]	Depression	1213	40
PHM [21]	Stroke	1222	14
CRISIS [17]	Injured or missing	2013	11
ADR [33]	Drug injuries	4355	10

Table 1: Summary of FLU [23], PHM [21], CRISIS [17], and ADR [33] datasets and their associated prediction tasks. The third and fourth columns report the size of the dataset and percentage of the positive tweets respectively.

FastText. We trained the shallow neural network classifier introduced in [13], which can update word embeddings during the training. We experimented with {0.05,0.1,0.25,0.5} for the learning rate, and {2,4} for the window size.

WESPAD. We trained the PHM model introduced in [21], which is specifically designed for Personal Health Mention detection. We experimented with {3,4,5} for the number of clusters, and {0.05,0.15,0.3} for threshold values.

BERT-BASE. We included the model introduced in [8], which is named BERT and uses a multi-layer transformer encoder followed by one layer of a fully connected neural network for binary classification problems. In the experiments we observed that the large variant shows poor performance when the training data is small, thus we report the results of the base variant *BERT-BASE*–which has fewer layers. We followed the parameter settings suggested in [8]; but empirically observed that if we set the number of epochs for fine-tuning to 15, the model is more stable and performs better.

BERT-TW. Since we experimented with Twitter data, we also pre-trained BERT in order to adjust the language model. Thus, we used the set of 7 million tweets described in Section 4.4 to further pre-train *BERT-BASE* for 10 epochs–without replacing human mentions. The hyperparameters were set to what is suggested in [8], and by the time the pre-training was done, the performance of the internal language modelling tasks for sample tweets was similar to the performance of *BERT-BASE* for sample Wikipedia pages.

BERT-DR. We also used the set of drug related tweets mentioned in Section 4.4–without replacing the drug mentions–to further pre-train *BERT-TW* to be used in ADR task. We used the same parameter setting as *BERT-TW*.

Co-BE-LE. In order to boost the BERT model with Bootstrapping, we also included a *co-training model* with two learners: One Naive Bayes classifier trained over unigrams and bigrams, and one logistic regression classifier trained over the *BERT-TW* or *BERT-DR* representation of the tweets–depending on the task. We experimented with {13,25,50} as the number of iterations in co-training model.

Co-Decomp. Our method described in Section 4. We empirically set the number of iterations in the co-training model to 25–based on the training and development folds in the FLU dataset–and did not do any further tuning beyond what we did for *BERT-TW*. We report all the results with this setting unless stated otherwise.

5.3 Training Details

We used standard 10-fold cross validation to train, validate, and test all of the models. To evaluate the models in semi-supervised settings, we did not use the entire training and validation data, but randomly sampled a few examples and used the rest of the examples as unlabeled data. In the next section, we report the results when we have 100 training examples, however, we also show that our model still performs well when the number of available training examples increases. To split the datasets into the folds, we used stratified sampling to preserve the original class distribution in the datasets. We also preserved the folds and samples identical across the experiments to ensure that all of the models use exactly the same training and test data. Since there is a natural randomness in neural network initialization and regularization techniques, we carried out all of the experiments 5 times, and averaged the performance results.

Because the datasets are highly imbalanced, following the argument in [25], we used the F1 measure in the positive class to tune the models. In the next section we report F1, Precision, and Recall in the positive class–averaged over the test folds.

6 Results and Discussion

In this section, we first report the performance results in FLU, PHM, CRISIS, and ADR datasets, and then analyze our model through a series of experiments.

6.1 Performance Results

Table 2 summarizes the F1, precision, and recall of the models in FLU and PHM datasets–the results in PHM dataset are averaged over the topics. Table 3 summarizes the results in CRISIS dataset, and Table 4 reports the results in ADR dataset. We also report the performance of the models in PHM dataset across all the topics in Table 5. The experiments show that Co-Decomp outperforms state-of-the-art classifiers across the majority of the tasks. We can see that the improvements in the imbalanced datasets (PHM and ADR) are more noticeable than the improvements in the balanced dataset (FLU). We can also see that the semi-supervised learning model *Co-BE-LE* performs relatively well, although it has a low precision. In contrast, our model maintains a high precision. We attribute this advantage to the easier tasks that Co-Decomp is tackling–i.e., selecting the most confident unlabeled instances via the context representations versus via the document representations. Finally, the results suggest that crisis report detection is an easier problem than adverse drug reaction monitoring, because even though both CRISIS and ADR have about 10% positive examples, the performance of the models in the ADR dataset is much lower. We will discuss this dataset in more detail in the next section.

6.2 Discussion

To better understand the impact of each component in our model, we report the results of the ablation study in Table 6. Since PHM dataset was the most diverse dataset (it constitutes 6 sub-topics), we carried out the experiment in this dataset. The results show that the weak human mention classifier is clearly contributing to the performance when it is combined with the disease mention classifier. Then a further improvement is achieved when co-training iterations are performed. However, the improvement after 50 iterations comes at the cost of dramatic deterioration in precision, which might not be desirable.

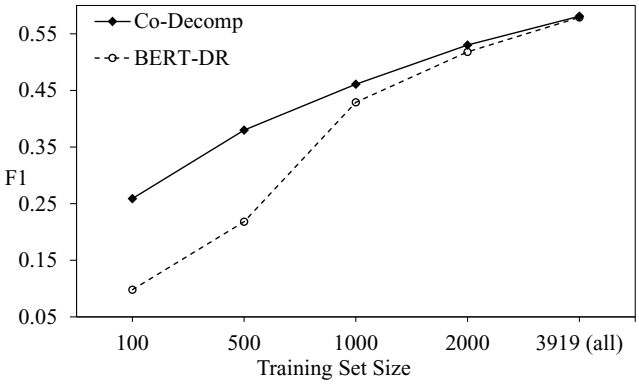

Figure 2: F1 at different training set size cut-offs for *BERT-DR* and *Co-Decomp* models in ADR dataset. There are 3,919 examples in the training folds of ADR dataset–excluding the test folds in 10 fold cross validation.

In Section 6.1, we observed that the performance of the models in ADR dataset was very low. To investigate the performance of the models as the function of the training set size, in Figure 2 we report the performance of Co-Decomp in comparison to the state-of-the-art *BERT-DR* classifier at different training set size cut-offs in this dataset. The results show that even in supervised settings our model is on par with strong classifiers–for this dataset and with manual feature engineering the F1 of 0.538 is reported in [33][9].

Finally, often in the real world situations, practitioners who try to tackle a classification problem, may have a small training set for the task and a larger diverse training set in the similar domains. We tried to evaluate our model in such a scenario. Thus, we assumed FLU dataset was the small training set which was available to do influenza surveillance in social media, and PHM dataset was the bigger diverse dataset which was available for similar domains. In Table 7, we report the results of domain adaptation in FLU dataset, when we use PHM dataset as the out-of-domain training data. We randomly sampled 500 positive and 500 negative examples from PHM dataset and fine-tuned the models; then further fine-tuned them using the training folds of FLU dataset, and finally used for labeling the FLU test folds–we used this approach to prevent from the catastrophic forgetting phenomenon in neural networks [22]. The results signify that even with a moderately large balanced training set, a supervised model cannot outperform Co-Decomp.

In this study we defined problem decomposition, and showed that it has at least three important real-world applications in social media. Our model is defined for the tasks that are centered around a set of entities or concepts. Co-Decomp can be also regarded as an approach to incorporate domain knowledge into the machine learning models. In Section 3.1, we presented three arguments that explain why our model is effective: 1) The vector representation of words is smaller than the vector representation of documents.

[9]The ADR task has been extensively explored in supervised settings [34, 37, 38]. However, the studies on semi-supervised ADR are limited [15]

	FLU dataset			PHM dataset		
Model	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**
NB	0.752	0.712	0.800	0.304	0.616	0.255
EM	0.766	0.708	**0.843**	0.407	0.528	0.414
FastText	0.747	0.728	0.772	0.278	0.626	0.215
WESPAD	0.763	0.728	0.805	0.336	0.668	0.272
BERT-BASE	0.757	0.739	0.790	0.572	0.682	0.537
BERT-TW	0.786	0.782	0.800	0.563	**0.698**	0.512
Co-BE-LE	0.771	0.715	0.838	0.577	0.627	0.593
Co-Decomp	**0.809**	**0.800**	0.822	**0.630**	0.674	**0.617**

Table 2: F1, precision, and recall in FLU and PHM datasets for all the models.

	CRISIS dataset		
Model	**F1**	**Precision**	**Recall**
NB	0.545	0.865	0.400
EM	0.568	0.625	0.535
FastText	0.382	0.815	0.258
WESPAD	0.607	**0.932**	0.458
BERT-BASE	0.710	0.818	0.676
BERT-TW	0.732	0.859	0.678
Co-BE-LE	0.609	0.615	0.614
Co-Decomp	**0.765**	0.880	**0.694**

Table 3: F1, precision, and recall in CRISIS dataset for all the models.

	ADR dataset		
Model	**F1**	**Precision**	**Recall**
NB	0.020	0.267	0.011
EM	0.072	0.168	0.052
FastText	0.004	0.100	0.002
WESPAD	0.016	0.300	0.008
BERT-BASE	0.082	0.274	0.054
BERT-DR	0.098	0.290	0.066
Co-BE-LE	0.184	0.183	0.202
Co-Decomp	**0.259**	**0.302**	**0.236**

Table 4: F1, precision, and recall in ADR dataset for all the models.

Thus, classification is easier over the words. 2) There are limited ways of using a word in a context. 3) Equipping the model with domain knowledge. The last argument, is based on the fact that we use domain understanding to impose a new inductive bias on the learner, through removing less important word features and targeting the pivotal entities in the task.

7 Conclusions and Future Work

We proposed a novel semi-supervised model for classification tasks that are centered around specific entities or concepts. Our model is based on: (1) decomposing the problem into a set of sub-tasks, and (2) combining the results in a co-training framework. By leveraging domain knowledge to decompose problems, and employing co-training framework to reinforce the underlying classifiers, our model Co-Decomp is able to generalize well and outperform state-of-the-art classifiers in semi-supervised settings. We showed that our model is applicable to at least three important personal event detection problems, namely, Personal Health Mention detection, Crisis Report detection, and Adverse Drug Reaction monitoring. We also carried out extensive experiments and reported the performance of the model in various settings. The results indicate that Co-Decomp is able to consistently and significantly outperform state-of-the-art classifiers in the three mentioned tasks.

Our current research introduces three potential future work directions. First, investigating other tasks which may be decomposable. As we discussed in Section 3.2, the tasks that are centered around entities and concepts can be potential targets. For instance, our model can be applied to the customer satisfaction task–where the mentions of human and the product can serve as candidate Key Concept Sets. The next two future directions are on the theory aspect of our method. One direction is to investigate the extent in which the choice of Key Concept Sets can impact the model performance. This will help us to understand whether our model can be applied to the tasks that the domain understanding is incomplete. Even though our experiments with a weak human mention detector showed promising results, we believe further investigation is required to understand if noisy Key Concept Sets can still be beneficial. And finally, the last future direction is to investigate the ways of automatically discovering Key Concept Sets.

Acknowledgments

This work was funded by Emory University; also partially by NIH grant LM013014-02, NSF award IIS- #1838200, and Google Cloud Platform research credits.

References

[1] Mohammad Akbari, Xia Hu, Liqiang Nie, and Tat-Seng Chua. 2016. 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*. 87–93.

[2] Firoj Alam, Shafiq Joty, and Muhammad Imran. 2018. Graph Based Semi-Supervised Learning with Convolution Neural Networks to Classify Crisis Related Tweets. In *Twelfth International AAAI Conference on Web and Social Media*.

[3] Abolfazl AleAhmad, Payam Karisani, Maseud Rahgozar, and Farhad Oroumchian. 2016. OLFinder: Finding opinion leaders in online social networks. *J. Information Science* 42, 5 (2016), 659–674.

[4] Maria-Florina Balcan, Avrim Blum, and Ke Yang. 2004. Co-training and Expansion: Towards Bridging Theory and Practice. In *Proceedings of the 17th International Conference on Neural Information Processing Systems (NIPS'04)*. MIT Press, Cambridge, MA, USA, 89–96.

[5] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. 2019. MixMatch: A Holistic Approach to Semi-Supervised Learning. *arXiv preprint arXiv:1905.02249* (2019).

[6] Avrim Blum and Tom M. Mitchell. 1998. Combining Labeled and Unlabeled Data with Co-Training. In *Proceedings of the Eleventh Annual Conference on*

	Alzheimer's			Heart attack		
Model	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**
NB	0.534	0.859	0.403	0.058	0.400	0.032
EM	0.617	0.663	0.618	0.072	0.500	0.039
FastText	0.418	**0.890**	0.284	0.048	0.400	0.025
WESPAD	0.535	0.837	0.421	0.058	0.400	0.032
BERT-BASE	**0.698**	0.723	0.701	0.366	0.586	0.309
BERT-TW	0.660	0.728	0.634	0.425	0.675	0.332
Co-BE-LE	0.682	0.674	**0.721**	0.378	0.647	0.298
Co-Decomp	0.676	0.694	0.682	**0.534**	**0.684**	**0.451**
	Parkinson's			**Cancer**		
Model	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**
NB	0.155	0.563	0.096	0.278	0.661	0.181
EM	0.356	0.521	0.301	0.429	0.492	0.424
FastText	0.076	0.350	0.043	0.219	**0.706**	0.134
WESPAD	0.188	**0.683**	0.113	0.335	0.682	0.227
BERT-BASE	0.451	0.631	0.387	0.570	0.679	0.515
BERT-TW	0.452	0.597	0.405	0.534	0.700	0.466
Co-BE-LE	0.518	0.551	0.546	0.569	0.632	0.573
Co-Decomp	**0.560**	0.520	**0.630**	**0.627**	0.704	**0.581**
	Depression			**Stroke**		
Model	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**
NB	0.670	0.617	0.742	0.130	0.597	0.076
EM	0.671	0.563	**0.841**	0.298	0.431	0.259
FastText	0.702	**0.744**	0.671	0.208	0.663	0.129
WESPAD	0.713	0.715	0.722	0.187	0.688	0.117
BERT-BASE	0.729	0.727	0.745	0.617	0.746	0.564
BERT-TW	**0.737**	0.740	0.747	0.569	**0.752**	0.490
Co-BE-LE	0.718	0.662	0.791	0.596	0.595	0.630
Co-Decomp	0.711	0.715	0.717	**0.673**	0.732	**0.643**

Table 5: F1, precision, and recall of the models across the topics in PHM dataset.

Model	**F1**	**Precision**	**Recall**
BERT-TW	0.803	0.782	0.836
Co-Decomp	0.810	0.813	0.813

Table 7: Domain adaptation results in FLU dataset. 1000 training examples from PHM dataset were randomly sampled–500 positives and 500 negatives–as the out-of-domain data.

Model	**F1**	**Precision**	**Recall**
Human-cl	0.390	0.326	0.521
Disease-cl	0.541	0.707	0.469
Combined	0.557	0.733	0.489
+13-itr co-train	0.608	0.705	0.565
+25-itr co-train	0.630	0.674	0.617
+50-itr co-train	0.637	0.587	0.715
+75-itr co-train	0.627	0.545	0.768

Table 6: Improvement analysis in PHM dataset. The performance of human mention classifier (*Human-cl*), disease mention classifier (*Disease-cl*), their combination in co-training framework without adding unlabeled data (*Combined*), and when unlabeled data is added per co-training iteration (4 unlabeled documents are added in every iteration).

Computational Learning Theory, COLT 1998, Madison, Wisconsin, USA, July 24-26, 1998. 92–100. https://doi.org/10.1145/279943.279962

[7] Johan Bollen, Huina Mao, and Xiao-Jun Zeng. 2011. Twitter mood predicts the stock market. *J. Comput. Science* 2, 1 (2011), 1–8.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018). arXiv:1810.04805

[9] Thomas G. Dietterich, Richard H. Lathrop, and Tomás Lozano-Pérez. 1997. Solving the Multiple Instance Problem with Axis-Parallel Rectangles. *Artificial Intelligence* 89, 1-2 (1997), 31–71.

[10] Jenny Rose Finkel, Trond Grenager, and Christopher D. Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*. 363–370.

[11] Rachel Ginn, Pranoti Pimpalkhute, Azadeh Nikfarjam, Apurv Patki, Karen O'Connor, Abeed Sarker, Karen Smith, and Graciela Gonzalez. 2014. Mining Twitter for adverse drug reaction mentions: a corpus and classification benchmark. In *Proceedings of the fourth workshop on building and evaluating resources for health and biomedical text processing*. 1–8.

[12] Roberto I. González-Ibáñez, Smaranda Muresan, and Nina Wacholder. 2011. Identifying Sarcasm in Twitter: A Closer Look. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA - Short Papers*. 581–586.

[13] Edouard Grave, Tomas Mikolov, Armand Joulin, and Piotr Bojanowski. 2017. Bag of Tricks for Efficient Text Classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*. 427–431.

[14] Melody Y. Guan, Varun Gulshan, Andrew M. Dai, and Geoffrey E. Hinton. 2018. Who Said What: Modeling Individual Labelers Improves Classification. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*. 3109–3118.

[15] Shashank Gupta, Manish Gupta, Vasudeva Varma, Sachin Pawar, Nitin Ramrakhiyani, and Girish Keshav Palshikar. 2018. Co-training for extraction of adverse drug reaction mentions from tweets. In *European Conference on Information Retrieval*. Springer, 556–562.

[16] Muhammad Imran, Carlos Castillo, Fernando Diaz, and Sarah Vieweg. 2015. Processing Social Media Messages in Mass Emergency: A Survey. *ACM Comput. Surv.* 47, 4, Article 67 (June 2015), 38 pages.

[17] Muhammad Imran, Prasenjit Mitra, and Carlos Castillo. 2016. Twitter as a Lifeline: Human-annotated Twitter Corpora for NLP of Crisis-related Messages. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)* (23-28). European Language Resources Association (ELRA), Paris, France.

[18] Bernard J. Jansen, Mimi Zhang, Kate Sobel, and Abdur Chowdury. 2009. Twitter power: Tweets as electronic word of mouth. *JASIST* 60, 11 (2009), 2169–2188.

[19] Thorsten Joachims. 1999. Transductive Inference for Text Classification using Support Vector Machines. In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML 1999), Bled, Slovenia, June 27 - 30, 1999*. 200–209.

[20] Rosie Jones, Rayid Ghani, Tom Mitchell, and Ellen Riloff. 2003. Active learning for information extraction with multiple view feature sets. *Proc. of Adaptive Text Extraction and Mining, EMCL/PKDD-03, Cavtat-Dubrovnik, Croatia* (2003), 26–34.

[21] Payam Karisani and Eugene Agichtein. 2018. 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*. 137–146.

[22] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences* 114, 13 (2017), 3521–3526.

[23] Alex Lamb, Michael J. Paul, and Mark Dredze. 2013. Separating Fact from Fear: Tracking Flu Infections on Twitter. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*. 789–795.

[24] Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. (2002).

[25] Richard Mccreadie, Cody Buntain, and Ian Soboroff. 2019. TREC Incident Streams: Finding Actionable Information on Social Media. In *Proceedings of the 16th International Conference on Information Systems for Crisis Response and Management (ISCRAM), 2019*.

[26] Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore*. 1003–1011.

[27] Andrew Y. Ng and Michael I. Jordan. 2001. On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes. In *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]*. 841–848.

[28] Kamal Nigam and Rayid Ghani. 2000. Analyzing the Effectiveness and Applicability of Co-training. In *Proceedings of the 2000 ACM CIKM International Conference on Information and Knowledge Management, McLean, VA, USA, November 6-11, 2000*. 86–93.

[29] Kamal Nigam, Andrew McCallum, Sebastian Thrun, and Tom M. Mitchell. 2000. Text Classification from Labeled and Unlabeled Documents using EM. *Machine Learning* 39, 2/3 (2000), 103–134.

[30] Michael J. Paul and Mark Dredze. 2017. *Social Monitoring for Public Health*. Morgan & Claypool Publishers.

[31] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*. 2227–2237.

[32] Alexander J. Ratner, Christopher De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. 2016. Data Programming: Creating Large Training Sets, Quickly. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*. 3567–3575.

[33] Abeed Sarker and Graciela Gonzalez. 2015. Portable automatic text classification for adverse drug reaction detection via multi-corpus training. *Journal of Biomedical Informatics* 53 (2015), 196–207.

[34] Gabriel Stanovsky, Daniel Gruhl, and Pablo Mendes. 2017. Recognizing mentions of adverse drug reaction in social media using knowledge-infused recurrent models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. 142–151.

[35] Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-Informed Self-Attention for Semantic Role Labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*. 5027–5038.

[36] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2003, Edmonton, Canada, May 27 - June 1, 2003*.

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

[38] Davy Weissenbacher and Graciela Gonzalez-Hernandez (Eds.). 2019. *Proceedings of the Fourth Social Media Mining for Health Applications (#SMM4H) Workshop & Shared Task*. Association for Computational Linguistics, Florence, Italy. https://www.aclweb.org/anthology/W19-3200

[39] Chang Xu, Dacheng Tao, and Chao Xu. 2013. A survey on multi-view learning. *arXiv preprint arXiv:1304.5634* (2013).

[40] Omar Zaidan, Jason Eisner, and Christine D. Piatko. 2007. Using "Annotator Rationales" to Improve Machine Learning for Text Categorization. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, April 22-27, 2007, Rochester, New York, USA*. 260–267. http://www.aclweb.org/anthology/N07-1033

[41] Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*.

[42] Xiaojin Zhu, Zoubin Ghahramani, and John D. Lafferty. 2003. Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions. In *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA*. 912–919.