DeMis: Data-efficient Misinformation Detection using Reinforcement Learning

Kornraphop Kawintiranon ✉ and Lisa Singh

Georgetown University
Washington DC, USA
{kk1155,lisa.singh}@georgetown.edu

Abstract. Deep learning approaches are state-of-the-art for many natural language processing tasks, including misinformation detection. To train deep learning algorithms effectively, a large amount of training data is essential. Unfortunately, while unlabeled data are abundant, manually-labeled data are lacking for misinformation detection. In this paper, we propose DeMis, a novel reinforcement learning (RL) framework to detect misinformation on Twitter in a resource-constrained environment, i.e. limited labeled data. The main novelties result from (1) using reinforcement learning to identify high-quality weak labels to use with manually-labeled data to jointly train a classifier, and (2) using fact-checked claims to construct weak labels from unlabeled tweets. We empirically show the strength of this approach over the current state of the art and demonstrate its effectiveness in a low-resourced environment, outperforming other models by up to 8% (F1 score). We also find that our method is more robust to heavily imbalanced data. Finally, we publish a package containing code, trained models, and labeled data sets.

Keywords: reinforcement learning · misinformation detection

1 Introduction

Social media sites allow users to share different types of online content. Unfortunately, there is no requirement that the content be true. As a result, we are seeing varying levels of accuracy in shared content. False information (fake information, misinformation, and disinformation) detection is not a new problem, and a significant amount of research has emerged (see [7,1] for surveys). Most research studies focus on detecting the spread of fake news by news sources [16,17], e.g. CNN and Washington Post. Some researchers have also worked on utilizing fact-checked information to verify the truth of social media content generated by users [4,19]. While this previous research can effectively identify false information on Twitter, in practice, the methods either requires a large amount of training data for each false claim or myth being detected, or expect balanced training data.

To mitigate these challenges, we propose a novel reinforcement learning (RL) framework for detecting misinformation on Twitter in a constrained environment, i.e. where data labels are limited and imbalanced. Our approach, DeMis, uses fact-checking articles (FC-articles) as background knowledge. The framework requires a small number of FC-articles related to the target myth theme. Then it weakly labels the unlabeled tweets given the chosen FC-articles. We design the RL mechanism to select high-quality tweets. These weak-labeled tweets are then used to help train the detector. While the joint training of classifier and selector [21] is often used to maximize the model performance, we partially train the classifier before jointly training the classifier and selector. This guides the classifier to gain knowledge about the manually-labeled data prior to learning from the weak and manually labeled data together.

Our contributions are as follows: (1) We propose a novel data-efficient RL framework in which state, action and reward are exclusively designed for misinformation detection. (2) We propose an approach (DeMis) to incorporate FC-articles as expert knowledge as a form of weak supervision. (3) We integrate multiple learning paradigms (reinforcement learning, multi-source joint learning, neural learning) into a framework for identifying misinformation. (4) We compare our model to multiple classic, neural, and reinforcement models and show that our model generally performs better. (5) We demonstrate the effectiveness of our framework when the training data is heavily imbalanced. (6) We release a package for misinformation detection using reinforcement learning, including the code, trained models and data sets.[1]

2 Related Works

Misinformation detection is an active area of research (see [1] for a recent survey). Because fake information can be produced by bots or humans, our work and review focuses on post-level misinformation instead of user-level and reinforcement learning approaches for generating additional training data.

Misinformation Detection: Research on misinformation detection typically falls into two categories based on types of information used to train a classifier [1], content-based and social context-based. Content-based approaches use information extracted from the content of posts such as text, images, and videos. Social context-based approaches use human–content interaction data such as retweets, replies, and likes. While using both types of information achieves slightly better results [26,10,13], because of the additional cost of data collection and the need for timely identification of misinformation,we focus on content-based methods.

Many studies use the lexical and syntactic features extracted from textual data [14,2]. Jin et al. [5] convert the detection problem into a text matching problem. They classify misinformation tweets based on the similarity scores between input tweets and the original verified-false posts. Their best algorithm is BM25 with an accuracy of 0.799. Recently, deep learning models have been shown to be state of the art for misinformation detection [1]. Wang et al. [20] propose EANN, a model that uses convolution neural networks (CNN) to learn latent semantic

[1] https://github.com/GU-DataLab/misinformation-detection-DeMis

text representations and use it along with image data to train a classification layer. Their models are evaluated on Twitter and Weibo data that have both text and images, achieving F1 scores of 0.719 and 0.829, respectively. A CNN model with an attention mechanism has also been proposed [24], improving the state of the art by 9 and 12% on the same data sets. These data sets are balanced and pseudo-labeled using keywords. Hossain et al. [4] introduced *COVIDLIES*, a manually-labeled Twitter data set about COVID-19 misinformation. It consists of 86 myths and 6761 tweets. Their approach has two sub-tasks including related-myth retrieval and stance detection. Using a BERT-based sentence similarity algorithm [25], they achieve the best Hit@k of 60.8 to 96.9 for different k values on the related-myth retrieval task but they obtain an F1 score of only 50.2 on the stance detection task because the data are imbalanced. Recently, Vo et al. [19] proposed a framework to search for fact-checking articles given a tweet, using a large amount of labeled training data (over 10K tweets and 2K FC-articles).

Data-Efficient and Reinforcement Learning Generally, a large amount of labeled data is required to train a reasonably accurate neural network (NN) model. Weak supervision aims to reduce human effort by automatically generating labels given unlabeled data. The quality of labels then heavily relies on the labeling algorithms [23]. An automatic data annotator based on the sources of news articles was proposed in [3]. Each tweet containing at least one URL to a news article was labeled true or false based on trustworthy or untrustworthy sources. Reinforcement learning (RL) techniques [18] have been adopted in many classification tasks to learn a high-quality data selector [23,21]. A model with a RL-based selector in [22] achieves an average F1 score of 0.692 on the Twitter click-bait classification task. Yoon and colleagues [23] propose a RL-based algorithm that quantifies the quality of labeled data. Their experimental results show that removing low-quality data from the training process improves the overall model performance on several classification tasks with accuracy scores ranging from 0.448 to 0.903. Mosallanezhad et al. [11] propose RL-based domain-adaptive learning which learns domain-invariant features and utilizes auxiliary information for fake news detection. Recently, WeFEND [21] was proposed for fake news detection on WeChat. The model trains a weak-labeling annotator using private user reports attached to each news article then selects the high-quality samples using a reinforced selector for training.The model obtains an F1 score of 0.81 on balanced WeChat data. Conceptually, we take a similar approach, building a model using reinforcement learning to identify weak labels. However, our annotator and joint learning paradigm are different.

3 Background and Problem Definition

Misinformation has many definitions. One common feature of these definitions is that misinformation must contain a piece of false information. Kumar et al. [7] define misinformation as false information spread without the intent to deceive, while others [26] define it as any false or inaccurate information regardless of

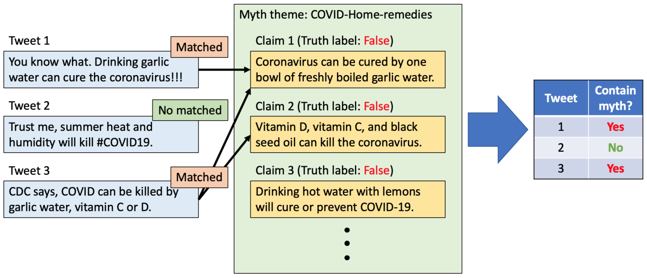

Tweet 1
You know what. Drinking garlic water can cure the coronavirus!!!
Matched
Tweet 2
Trust me, summer heat and humidity will kill #COVID19.
No matched
Tweet 3
CDC says, COVID can be killed by garlic water, vitamin C or D.
Matched
Myth theme: COVID-Home-remedies
Claim 1 (Truth label: False)
Coronavirus can be cured by one bowl of freshly boiled garlic water.
Claim 2 (Truth label: False)
Vitamin D, vitamin C, and black seed oil can kill the coronavirus.
Claim 3 (Truth label: False)
Drinking hot water with lemons will cure or prevent COVID-19.
Tweet
Contain myth?
1 Yes
2 No
3 Yes

Fig. 1: Examples of misinformation tweets and supporting evidence.

intention. In this paper, we follow the later and refer to a misinformation tweet as a tweet containing a piece of myth-related information. A *myth* is a false claim verified by trustworthy fact-checkers. This task is different from fake news detection which focuses on detecting a news article published by a news outlet that is verifiably false, and rumor detection which aims to determine if a story or online post is a rumor or non-rumor regardless of its veracity [8]. Fig. 1 demonstrates how tweets are determined to be misinformation. For example, a claim saying "boiled garlic water could kill the coronavirus" is false. A tweet containing such information (even if it is being refuted) is classified as misinformation conversation, regardless of the user stance. In other words, our goal is to identify that misinformation is being discussed on social media, not the intent or the position of the poster. A tweet that does not is labeled as true information.

Generally, fact checkers provide a set of *FC-article* that each contain a claim, truth label, and fact. A *claim* is a truth-verifiable statement that may be true, false, partially true or have insufficient information to determine whether or not it is true. A *truth label* is the factual state of the related claim at a particular time. It is manually verified by experts in the relevant areas. Different fact checkers have different rating schemes. For example, PolitiFact claims are usually rated using six level of falseness. The *fact* is the supporting information that provides context about the claim and explains details about why a particular truth label is assigned. Different claims of FC-articles may be in the same myth theme as shown in Fig. 1. In this paper, the goal is to predict whether a tweet contains the same piece of misinformation as in the claims of interest. We only use claims and truth labels verified as false since our goal is to detect misinformation discussion.

More formally, the problem we investigate is content-based misinformation. Let $\mathbf{M}$ represent a set of myths and $\mathbf{C}$ represent a set of claims from FC-articles. Suppose we are given a set of target claims $\bar{C}_p$ that are related to the pre-defined myth theme M_p. Our task is to determine a class label y_r for a tweet t_r from Twitter data $\mathbf{T}$ using claim information ($\bar{c}_{pq} \in \bar{C}_p$) related to M_p. If t_r contains misinformation, ($y_r = 1$), otherwise, ($y_r = 0$). We assume that claims across myths in $\mathbf{M}$ are non-overlapping, $\bigcap_{p=1}^{|\mathbf{M}|} \bar{C}_p = \emptyset$. For example, claims $\bar{C}_1$ under the myth theme M_1 about a specific weather condition killing coronavirus, and claims $\bar{C}_2$ under the myth theme M_2 about COVID home-remedies, are not overlapped ($\bar{C}_1 \cap \bar{C}_2 = \emptyset$).

4 Methodology

We propose DeMis, a framework for misinformation detection on Twitter. An overview of the framework is presented in Section 4.1. The main components of the framework are presented in Section 4.2 and 4.3. Section 4.4 presents the integration of all the components.

4.1 Overview of DeMis

The overview of the framework is shown in Fig. 2. We begin by extracting claims $\mathbf{C}$ and target claims $\bar{C}_p$ related to the myth themes of interest from existing FC-articles. Each theme of interest M_p has a small number of manually-labeled tweets. We refer to these tweets as *strong-labeled* tweets. The automatic annotator (Section 4.2) uses a sentence similarity algorithm to calculate similarity scores between all claims $\mathbf{C}$ and unlabeled tweets in $\mathbf{T}$. The scores are used to generate labels for the unlabeled tweets using our proposed labeling function. We refer to tweets with labels generated by the automatic annotator as *weak-labeled* tweets. Once the reinforced selector (Section 4.3) chooses high-quality weak-labeled tweets, they are combined with the strong-labeled tweets for training the misinformation detector. The samples that are selected by the reinforced selector are referred to as *selected* tweets. The reward is computed based on the model performance and used to update the selector for the next iteration. The updated selector selects high-quality weak-labeled tweets to train the detector until the detection classifier converges. The misinformation detector $D_n(\cdot;\theta_n)$ is a transformer-based model with a neural network on top as a classifier layer, where θ_n denotes its parameters. We now present the details.

4.2 Automatic Annotation based on Claims

We propose an unsupervised approach for automatically labeling tweets.[2] There are two main components: sentence similarity ranking and labeling. First, among all claims C, there are claims $\bar{C}_p \subset C$ belonging to the target myth theme M_p that we are interested in. We calculate the similarity scores between each tweet t_r and all claims $c_q \in C$. For each tweet, we obtain a list of all claims L_r ranked by the similarity scores. If at least one of the target claims $\bar{c}_{pq} \in \bar{C}_p$ appears in the top K of the list, then the tweet is labeled as positive (about misinformation). Otherwise, the label is negative (not about misinformation). Any similarity score is reasonable. Given that we are using short texts, we use a sentence transformer [15] in our empirical evaluation. We convert a claim and a tweet into vectors and compute the final similarity score using cosine similarity.

[2] We use the term *unsupervised* because we do not use any labeled data at this stage.

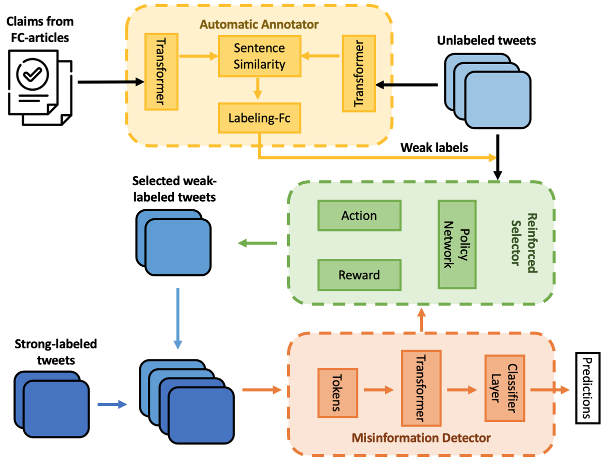


Fig. 2: The architecture of our proposed misinformation detection framework.

4.3 Data Selection via Reinforcement Learning

The goal of the data selection component is to select high-quality weak-labeled samples that improve the detector performance. We propose a performance-driven data selector that uses the policy-gradient reinforcement learning mechanism called the *reinforced selector*. It takes weak-labeled data as input, selects high-quality samples, and then sends them to the classifier to use during training. The reward is computed based on the model performance and used to update the policy network. Because the reward is computed after the data selection process is finished, the policy update is delayed. This is inefficient. To obtain rewards and train the policy network more efficiently, we split the input data $\mathcal{X} = \{x_1, ..., x_n\}$ into N bags $\mathcal{B} = \{B^1, ..., B^N\}$. Each bag B^k contains a sequence of unlabeled samples $\{x_1^k, x_2^k, ..., x_{|B^k|}^k\}$. Each bag is fed into the reinforced selector. For each sample in the bag, the reinforced selector decides on an *action* to retain or remove. The action of the current sample x_i^k is based on the current *state* vector and all the actions of previous samples in the current bag $\{x_1^k, x_2^k, ..., x_{i-1}^k\}$. The *reward* is computed based on the change in performance of the misinformation detector. The remainder of this subsection presents the details of the main components of the RL mechanism: *state*, *action*, *reward* and *optimization*.

State. s_i^k represents the state vector of sample x_i^k. The action a_i^k is decided based on the current and selected samples in the same bag, B^k. The state vector s_i^k consists of two major components, including the representation of the current sample and the average representation of selected samples. We consider quality and diversity for a representation of a sample. For the quality of the sample, we consider a prediction output from the misinformation detector and a small number of elements from the sentence similarity algorithm (Section 4.2). For the current sample, these elements include: (i) the highest similarity score between the current sample and all claims C, (ii) the K-th highest similarity score, (iii) the highest similarity score between the current sample and the target claims $\bar{C}_p$, (iv) the subtraction of (i) and (iii), and (v) the subtraction of (iii) and (ii). For diversity, we calculate the cosine similarity between the current sample and all selected samples in the bag, and then the maximum similarity score is used as the representation of the diversity of the current sample among the selected samples. The weak label of the current sample is also included in the representation vector as a signal for the class distribution. The final current state representation vector contains eight elements: 1) the output probability from the detector, 2) the maximum cosine similarity score between the current sample and the selected samples, 3) the weak label of the current sample, and five elements from the sentence similarity described above. Once we have the current representation vector, we concatenate it with the average of previously selected representation vectors to form the final state vector s_i^k.

Action. An action value of the reinforced selector for any sample is either 1 representing an action to *retain*, or 0 representing an action to *remove* the sample from the training set. We train a policy network $P(\cdot;\theta_s)$ to determine action values, where θ_s indicates its parameters. The policy network is a neural network of two fully-connected layers with the sigmoid (σ) and ReLU activation functions and is defined as $P(s_i^k;\theta_s) = \sigma(W_2 \cdot ReLU(W_1 \cdot s_i^k))$, where W_1 and W_2 are randomly initialized weights. The network outputs the probability of the *retain* action p_i^k for the sample x_i^k given the corresponding state vector s_i^k. Next, the policy $\pi_{\theta_s}(s_i^k, a_i^k)$ determines the action a_i^k by sampling using the output probability p_i^k as follows $\pi_{\theta_s}(s_i^k, a_i^k) = a_i^k p_i^k + (1 - a_i^k)(1 - p_i^k)$.

Reward. As previously mentioned, we use the performance changes of the misinformation detector $D_n(\cdot;\theta_n)$ as the reward function. To determine the initial baseline performance F_{base}, we train the detector on the strong-labeled training set and evaluate it on the validation set. For the k-th bag, the reinforced selector chooses high-quality samples. They are used to re-train the detector, then the performance F_k for the k-th bag is obtained by evaluating the re-trained detector on the validation set. Formally, the reward R_k for the k-th bag is the subtraction of F_{base} and F_k as shown in the equation $R_k = F_{base} - F_k$.

Optimization. The goal is to maximize the expected total reward for each bag B^k. However, the magnitude of reward R_k is undoubtedly small because a performance change ranges from zero to one. Therefore, we use the summation of reward R_k weighted by policy values $\pi_{\theta_s}(s_i^k, a_i^k)$ from every sample in the bag $\{x_i^k\}_{i=1}^{|B^k|}$. Finally, the objective function for the k-th bag is defined as: $J(\theta_s) = \sum_{i=1}^{|B^k|} \pi_{\theta_s}(s_i^k, a_i^k)\, R_k$, and its derivative function is: $\nabla_\theta J(\theta_s) = \mathbb{E}_{\theta_s}[\sum_{i=1}^{|B^k|} R_k \, \nabla_{\theta_s} \log \pi_{\theta_s}(s_i^k, a_i^k)]$.

Since we are using policy-gradient reinforcement learning [18], we update the policy network using the gradient ascend: $\theta_s \leftarrow \theta_s + \alpha \sum_{i=1}^{|B^k|} R_k \, \nabla_{\theta_s} \log \pi_{\theta_s}(s_i^k, a_i^k)$, where α is the learning rate.

```
Algorithm 1: The Overall Training Process of DeMis
  Input  : Misinformation detector $D_n(\cdot;\theta_n)$, policy network $P(\cdot;\theta_s)$ of
           reinforced selector with random weights, strong-labeled data $\mathcal{D}$
  1. Pre-train the detector $D_n(\cdot;\theta_n)$ to predict misinformation using the
     strong-labeled training data $\mathcal{D}_t$.
  2. Pre-train the policy network $P(\cdot;\theta_s)$ by running Algorithm 2 with the
     misinformation detector $D_n(\cdot;\theta_n)$ fixed.
  3. Re-initialize the parameters of the detector $D_n(\cdot;\theta_n)$ with random weights.
  4. Warm up the detector $D_n(\cdot;\theta_n)$ by training for $\mathcal{L}$ epochs.
  5. Jointly train $D_n(\cdot;\theta_n)$ and $P(\cdot;\theta_s)$ using Algorithm 2 until convergence.

  Output: The trained models $D_n(\cdot;\theta_n)$ and $P(\cdot;\theta_s)$.
```

4.4 Model Training

The overall training process is described in Algorithm 1. First, we randomly initialize weights of the misinformation detector and policy network of the reinforced selector. The detection classifier $D_n(\cdot;\theta_n)$ is a neural network model: $p(y|x;\theta_n) = \mathit{Softmax}(W_{L2}(\tanh(W_{L1}x_t + b_1)) + b_2)$, where $p(y|x;\theta_n)$ represents the output probability of being class y given input x from the linear classifier, x represents a contextual representation vector of tweet t from the pre-trained language model (BERTweet) after the dropout layer, W_{Li} is a weight vector at layer i randomly initialized, and b_i is a bias vector at layer i where $i \in \{1, 2\}$. The weights of the classifier are updated using the cross-entropy loss function. We use the softmax function to normalize the values of the output vector from the classifier in order to obtain a probability score for each class.

Second, we get the baseline performance F_{base} by training the detector using the strong-labeled training data $\mathcal{D}_t$ and evaluating it on the validation set $\mathcal{D}_v$. Next, because the joint-training technique can result in a detector over-fitting the small data set, we re-initialize the weights of the detector model and train it for $\mathcal{L}$ epochs instead of training it until convergence (Algorithm 1, step 4-5). This makes the detector under-fit, leaving some room for joint-training. Finally, we jointly train the detector and reinforced selector together until convergence.

Algorithm 2 explains how to train the detector and reinforced selector jointly. The detector provides the mechanism to compute the reward based on its evaluation performance. The selector uses the reward to refine its ability to select high-quality samples that potentially enhance the detector performance. To improve the training stability we update the target policy network slowly: $\theta'_s = \tau\,\theta_s + (1-\tau)\theta'_s$.

5 Experimental Design

Algorithm 2: Learning Algorithm of Reinforced Selector

```
Input : Strong-labeled training data $\mathcal{D}_t$. $N$ bags of weak-labeled
        training data $\mathcal{B} = \{B^1, ..., B^N\}$. A misinformation detector
        $D_n(\cdot; \theta_n)$ and a policy network $P(\cdot; \theta_s)$. Epoch number $L$.
Initialize the target networks as: $\theta'_n \leftarrow \theta_n$ and $\theta'_s \leftarrow \theta_s$
for epoch $\ell \leftarrow 1$ to $L$ do
    Shuffle $\mathcal{B}$ to get a sequence of bags $\{B^1, B^2, ..., B^N\}$ foreach bag $B^k \in \mathcal{B}$ do
        /* We omit superscript $k$ for clarity */
        Sample actions for each data sample in $B$ with $\theta'_s$:
          $A = \{a_1, ..., a_{|B|}\}$, $a_i \sim \pi_{\theta'_s}(s_i, a_i)$
        Train the detector $D_n(\cdot; \theta_n)$ using selected samples based on
          actions $A$ and update weights $\theta_n$
        Compute delayed reward $R_k$
        Update the parameters $\theta_s$ of reinforced selector:
          $\theta_s \leftarrow \theta_s + \alpha \sum_{i=1}^{|B|} R_k \nabla_{\theta_s} \log \pi_{\theta_s}(s_i, a_i)$
    end
    Update the weights of target policy network: $\theta'_s = \tau \theta_s + (1 - \tau)\theta'_s$
    Train the target detector using the selected samples from the target
      selector then update weights $\theta'_n$
    Reset the weights of detector: $\theta_n \leftarrow \theta'_n$
end
Output: The trained models $D_n(\cdot; \theta_n)$ and $P(\cdot; \theta_s)$.
```

5.1 Data Collection

Our empirical evaluation uses one large unlabeled and three manually-labeled Twitter data sets: COVIDLIES [4], COMYTH-W and COMYTH-H. These data sets have different characteristics in terms of myth diversity and training data imbalance. The sizes of positive samples in a training set range from only 40 to 200. In COVIDLIES, misinformation tweets contain claims belonging to multiple myth themes (high-diversity) and have class-imbalances (high-imbalance). In COMYTH-W and H, misinformation tweets contain claims belonging to one myth theme (low-diversity), COVID-weather and COVID-home-remedies, respectively. While COMYTH-W is a balanced data set (low-imbalance), COMYTH-H is not (high-imbalance). Table 1 presents the statistics of these data.

Unlabeled Twitter Data. Our research team collected English tweets related to COVID-19 using hashtags and keywords through the Twitter Streaming API. Between March 2020 and August 2020, we collected over 20 million tweets, not including quotes and retweets. These unlabeled tweets were used to train all models that require unlabeled data.[3]

[3] Our unlabeled tweets do not overlap with any of our labeled data.

COVIDLIES. This data set, shared by Hossain et al. [4], contains 62 claims, along with 6591 tweet-claim pairs. Each tweet has at least one related claim and an annotated stance of the tweet content towards the claims (agree, disagree, no stance). We follow the labeling approach of the original paper [4] and label a tweet as misinformation if and only if the tweet contains a stance. A tweet with no stance is labeled as no misinformation. Among 62 claims, only four claims (of different themes) have more than 100 tweets containing a stance towards them, indicating high diversity. There are 811 annotated tweets, 136 containing misinformation and 675 regular tweets.

COMYTH. To conduct experiments on data sets with specific myth themes, we created a data set of COVID-myth-related tweets and claims from a random sample of tweets. We focus on two myth themes, weather and home-remedies. Our data were labeled using Amazon Mechanical Turk (MTurk). The labeling choices were yes, no, and unsure. Each tweet has three annotations from three different MTurk workers. We compute inter-annotator agreement scores to assess the quality of our labeled data. The task-based and worker-based metrics are recommended by the MTurk official site[4], given their annotating mechanism. All scores range from 85% up to 97%, indicating the high inter-rater reliability for these data sets. The majority voting among three annotators is used to determine a label for each tweet (containing related myths or not). Finally, there are 930 labeled tweets for the weather theme (COMYTH-W), of which 459 tweets contain weather myths. For the home-remedies theme (COMYTH-H), there are 779 labeled tweets, of which 101 tweets contain home-remedies myths. To build a data set of COVID-related claims, we collected claims from PolitiFact, FactCheck.org and Snopes. Our research team manually identified 3 COVID-weather-related claims and 12 COVID-home-remedies-related claims as target claims for our framework.

5.2 Data Preparation

Data sets are split into train, validation and hold out sets with an approximate ratio of 5/2/3. Each tweet is preprocessed by replacing mentions with *@USER* and links with *HTTPURL*. To build weak-labeled data sets, we run our weak annotator as described in Section 4.2 on the unlabeled data set and sample 10K tweets for each class (myth/not-myth).

5.3 Baselines

Our baseline models are categorized into four algorithm groups. The first group contains classic machine learning models, including Naive Bayes (NB), k-Nearest-Neighbor (kNN), Logistic Regression (LR), Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF) and Elastic Net (EN). We adopt the implementations in [6] because their approaches are shown to be highly accurate

[4] Amazon Mechanical Turk - HIT Review Policies

Table 1: Data set details.

Data Set	Myth theme	Split	# Tweets	# Myth	# Not-myth	Myth ratio
COVIDLIES	COVID-mixed	Train	380	64	316	∼17%
		Val	163	27	136	
		Test	268	45	223	
COMYTH-W	COVID-weather	Train	436	213	223	∼50%
		Val	187	96	91	
		Test	307	150	157	
COMYTH-H	COVID-home-remedies	Train	365	48	317	∼13%
		Val	156	26	130	
		Test	258	27	231	

Myth theme indicates whether the data set is for a specific myth or mixed myth themes. Myth ratio indicates the ratio of misinformation.

for detecting low-quality textual content on Twitter. Their feature sets include simple counting properties in a tweet content (Count), Bag-of-Words (BoW) and Term-Frequency-Inverse Document-Frequency (TF-IDF). All the models are trained using different combinations of these features. The second baseline group contains neural network models, including a vanilla neural network (NN) and a convolution neural network (CNN). We follow the setup used in EANN [20]. The third group consists of transformer-based models. We use RoBERTa (RB), BERTweet (BT) and BERTweet-covid (BTC). RoBERTa is an optimized version of BERT. BERTweet is RoBERTa trained on Twitter data, and BERTweet-covid is BERTweet additionally trained on COVID-related tweets. The classification layer is a single layer neural network. The last group contains RL-based models including DVRL [23] and WeFEND [21].

We use DVRL to select high-quality weak-labeled samples. We run the model to estimate the quality of our weak-labeled data. We combine the top v percent of weak-labeled data, sorted by the quality scores with the strong-labeled data, where $v \in \{10, 20, ..., 100\}$ as used in the original paper. In addition, we also use smaller values for $v \in \{0.5, 1, ..., 5\}$ in order to have a more complete stability and sensitivity analysis. Using $v = 100$ means we combine all weak-labeled data with the strong-labeled data for training. For each combination, we train the same misinformation detector as used in our model (Section 4.1) and report the best results based on F1 scores. We implement the WeFEND framework as described in the original paper since the code was not available. Because the original framework uses user reports to generate weak labels but there is no such report publicly available for Twitter, we modify the framework by substituting the weak label annotation part with our weak label annotator to investigate its potential to use public accessible expert knowledge (FC-articles). The rest of the framework remains the same.

Table 2: Experimental results. The best results are bolded.

Type	Algorithm	COVIDLIES (multiple myth themes)				COMYTH-W (one myth theme)				COMYTH-H (one r		
		Acc	Pr	Re	F1	Acc	Pr	Re	F1	Acc	Pr	I
Classic ML	Count [6]	0.7724	0.3095	0.2889	0.2989	0.6645	0.6158	0.8333	0.7082	0.7985	0.2093	0.3
	+BoW [6]	0.8396	0.5200	0.5778	0.5474	0.9414	0.9400	0.9400	0.9400	0.9031	0.5500	0.4
	+TFIDF [6]	0.8545	0.5682	0.5556	0.5618	0.9479	0.9467	0.9467	0.9467	0.9070	0.5600	0.5
DL	NN [6]	0.8408	0.5484	0.2963	0.3845	0.8795	0.7565	0.7862	0.7672	0.9160	0.6149	0.5
	CNN [20]	0.3340	0.1508	0.7183	0.2446	0.7492	0.7275	0.9003	0.7816	0.2313	0.1295	0.5
Trans-former	RB [9]	**0.8756**	**0.6550**	0.5481	0.5964	0.9739	**0.9755**	0.9711	0.9733	0.9328	0.6541	0.7
	BT [12]	0.8557	0.5891	0.5185	0.5450	0.9511	0.9272	0.9778	0.9515	0.9160	0.5936	0.6
	BTC [12]	0.8595	0.5733	0.6370	0.6035	0.9631	0.9531	0.9733	0.9627	0.9367	0.6995	0.7
RL	DVRL [23]	0.8333	0.5204	0.6444	0.5667	0.9577	0.9369	0.9800	0.9578	0.9057	0.5752	0.7
	WeFEND [21]	0.4378	0.1991	0.6765	0.2553	0.9338	0.9323	0.9346	0.9328	0.6460	0.4733	0.7
	DeMis (ours)	0.8483	0.5644	**0.7226**	**0.6210**	**0.9750**	0.9638	**0.9894**	**0.9762**	**0.9406**	**0.7353**	**0.8**
Compare DeMis	vs. best scores	-0.0273	-0.0906	+0.0043	+0.0175	+0.0011	-0117	+0.0094	+0.0029	+0.0039	+0.0358	+0.
	vs. best model	-0.0113	-0.0089	+0.0856	+0.0175	+0.0011	-0117	+0.0183	+0.0029	+0.0078	+0.0812	+0.

5.4 Evaluations and Hyperparameter Tuning

We evaluate all models using accuracy, precision, recall and F1 scores based on positive class (misinformation). We evaluate all models on the test set three times with different random seeds to determine the stability of the results. The average results are reported. For our classic ML models, we conduct a sensitivity analysis using a grid-search on influential parameters. The best parameters varied by classifiers, data sets, and feature sets. For neural network and transformer-based models, we use different learning rates (1e-4, 1e-5, 2e-5, 3e-5, 1e-6). We report the best results based on F1 scores from the parameter tuning step. We present results for the learning rate of 1e-5 for DeMis and use a learning rate for target network τ of 0.001.

6 Results and Analysis

Table 2 shows the experimental results on the test sets, averaged over three runs. The models from four different categories are evaluated on all data sets. The variances of results from different models are not significantly different. Our proposed model outperforms the best baselines F1 scores by ∼2%, ∼1% and ∼8% on COVIDLIES, COMYTH-W and COMYTH-H, respectively. The last two rows of the table show the comparison of DeMis result with the best scores in the same column, and with the second best models based on F1 score.

6.1 Experimental Results

We hypothesize that the most complicated data set is COVIDLIES because of the high diversity of the myth themes and the data imbalance. The baseline models have F1 scores ranging from 0.2446 (CNN) to 0.6035 (BERTweet-covid). Our proposed model outperforms the baselines with an F1 score of 0.6210, slightly better than BERTweet-covid. The difficulty of this data set is two-fold. First, with 136 positive training samples for different myth themes, there are only 10 to 42 samples for each myth theme. This is insufficient for training deep learning models; therefore, the transformer models (RoBERTa and BERTweet-covid) and two of the classic models (Count+Bow and Count+TFIDF) perform better than the deep learning models. The second complexity is the mix of multiple myth themes, each having different contexts, signal words, and writing styles. These signals from different myth themes can mislead the classifiers, resulting in inefficient learning of the positive class. For example, in a batch size of 32, there are likely samples from at least two myth themes. If their characteristics are completely different, then the loss computed using the error from the samples in the batch could be misleading, resulting in under-fitting. While our models perform comparably to the state-of-the-art ones on this high diversity and imbalanced data (COVIDLIES), our model performs better on data sets containing one myth and possible imbalances.

We anticipate that the least complicated data set for this task is COMYTH-W since it contains one myth theme and is balanced data. On this data set, the baseline models perform reasonably with F1 scores ranging from 0.7082 (a classic model with Count features) to 0.9733 (RoBERTa). The notably high F1 score from RoBERTa shows that the data set is uncomplicated for the misinformation detection task and implies marginal room for improvement. Our model performs comparably with RoBERTa, having an F1 score of 0.9762.

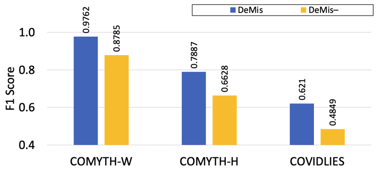


Fig. 3: The model performance of DeMis with and without RL (DeMis–).

We anticipate that the COMYTH-H data set is the second most complicated because it contains one myth theme but has a similar level of imbalance as COVIDLIES (the myth ratios of both data sets are around 10%, see Table 1). The baseline models have F1 scores ranging from 0.1953 (CNN-based model) to 0.7132 (RoBERTa), indicating that this data set is moderately complex for the task. We see that the lowest and highest F1 scores of baseline models on COMYTH-H are much lower than COVIDLIES (0.19/0.71 vs. 0.70/0.97) due to class imbalance and the nature of the myth themes. While there are only three claims related to COVID-weather, there are 12 claims about COVID-home-remedies, leading to a more diverse set of topics about home-remedies, i.e. higher content (vocabulary) diversity. Our model significantly outperforms all baselines with an F1 score of 0.7887 on COMYTH-H, an approximate 8% improvement over RoBERTa (second best).

To better understand the characteristics of the misclassified samples, we look at their distribution. We find that from 20 misclassified samples by RoBERTa and 14 misclassified samples by our model, 12 samples are the same. Our model corrects six false positives and two false negatives that the RoBERTa model misclassifies, but we have two additional false negatives, meaning that our model tends to error on the side of false negative, not false positives.

To investigate the advantage of the reinforced selector, we train our DeMis without RL by substituting it with a random selector (DeMis–). It randomly selects samples instead of selecting only high-quality samples. The results are shown in Fig. 3. On COMYTH-W, the F1 score (yellow) of DeMis without RL is 10% lower than DeMis with RL. Similarly, F1 scores are substantially higher for DeMis with RL on the other two data sets. We observe that recall scores stay the same between DeMis with and without RL because the model without RL still learns good positive examples from the strong-labeled samples. However, the precision scores drop significantly, producing more false positives when low-quality samples are selected. These empirical results suggest that incorporating RL is beneficial for improving the data selection process.

6.2 Robustness of Model

We further investigate the robustness of our model on two imbalanced data sets, COMYTH-H and COVIDLIES. We compare our model with RoBERTa since it is the second-best performer on COMYTH-H and performs comparably to BERTweet-covid on COVIDLIES. A random oversampling algorithm is used to balance the class distribution of these two data sets. We train RoBERTa on these balanced data sets separately and report the results (RoBERTa+). We see that making the data sets more balanced for RoBERTa slightly increases the F1 scores by 0.39% and 2.03% on COMYTH-H and COVIDLIES, respectively. Without any data modification, our model that used imbalanced training data outperforms RoBERTa+ by 7.16% and 0.43%, further highlighting our model's robustness to data imbalances.

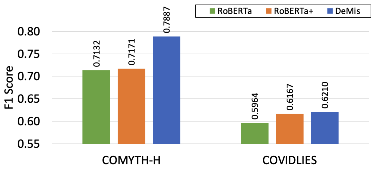
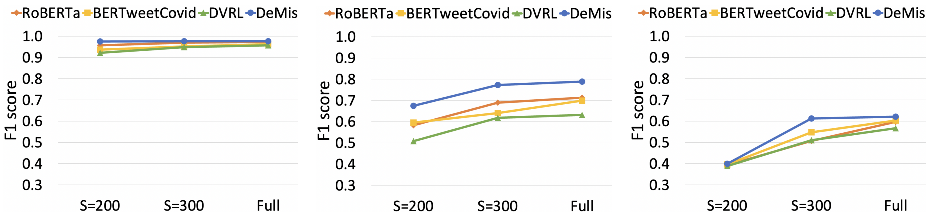


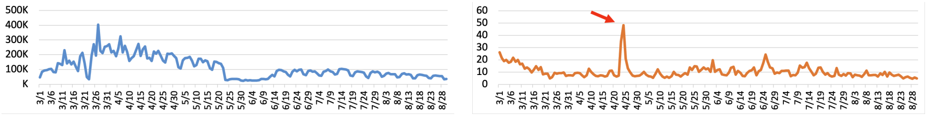
Fig. 4: The model performance of RoBERTa, RoBERTa+, and DeMis.

We also investigate the robustness of our model when smaller sizes of training data are provided. We randomly sample training data of sizes 200 and 300 while keeping the same level of imbalance. Fig. 5 shows the F1 scores of the top performers. Our model outperforms other baselines on smaller sizes of all training data sets. However, we see that smaller sizes of data lead to larger performance deterioration on both imbalanced data sets (COMYTH-H and COVIDLIES) by all the models. In other words, when there are less than 300 training samples, the models underfit the data.



(a) COMYTH-W (b) COMYTH-H (c) COVIDLIES

Fig. 5: The performance of top models on different sizes of training data.



(a) COVID-related tweet count (b) Proportion of misinformation

Fig. 6: Daily tweet counts and proportion of COVID-weather per 10,000 tweets.

6.3 Analysis on Big Data

We conduct a small case study to better understand the prevalence of misinformation on Twitter, we run our model on data containing Covid-related hashtags (Section 5.1) to predict levels of misinformation conversation. We find over 20K misinformation tweets about COVID-weather between March 1 to August 31, 2020. Fig. 6 illustrates the daily number of tweets and the diffusion of misinformation on Twitter related to COVID-weather by DeMis. Misinformation conversation was spreading before March and reached its peak on April 24th (red arrow), the day after the White House promoted new lab results suggesting heat and sunlight slow coronavirus on April 23rd[5]. This small analysis highlights the level of misinformation on a public health related data stream and demonstrates the role prominent leaders play in spreading and/or reinforcing it.

7 Conclusions

This paper proposes DeMis, a novel RL-based framework for misinformation detection that requires only a small amount of labeled training data. We design a novel RL mechanism, inspired by policy-gradient reinforcement learning, that provides high-quality data selection, improving our overall detection performance. We evaluate models on three data sets, and show that they outperforms other baselines by up to 8% (F1 score). Our approach is particularly strong in the presence of class imbalances and comparable to other models when there is high diversity in the myth themes. Finally, we release a resource package to support the community to studying misinformation.

[5] News on Washington Posts

Acknowledgements:

This research was funded by National Science Foundation awards #1934925 and #1934494, and the Massive Data Institute (MDI) and McCourt Institute at Georgetown University. We would like to thank our funders, the MDI staff, and the Georgetown DataLab for their support.

References

1. Guo, B., Ding, Y., Yao, L., Liang, Y., Yu, Z.: The future of false information detection on social media: New perspectives and trends. ACM Computing Surveys **53**(4), 1–36 (2020)
2. Haber, J., Singh, L., Budak, C., Pasek, J., Balan, M., Callahan, R., Churchill, R., Herren, B., Kawintiranon, K.: Lies and presidential debates: How political misinformation spread across media streams during the 2020 election. Harvard Kennedy School Misinformation Review (2021)
3. Helmstetter, S., Paulheim, H.: Weakly supervised learning for fake news detection on twitter. In: ASONAM (2018)
4. Hossain, T., Logan IV, R.L., Ugarte, A., Matsubara, Y., Young, S., Singh, S.: COVIDLies: Detecting COVID-19 misinformation on social media. In: Workshop on NLP for COVID-19 (Part 2) at EMNLP (2020)
5. Jin, Z., Cao, J., Guo, H., Zhang, Y., Wang, Y., Luo, J.: Detection and analysis of 2016 us presidential election related rumors on twitter. In: SBP-BRiMS (2017)
6. Kawintiranon, K., Singh, L., Budak, C.: Traditional and context-specific spam detection in low resource settings. Machine Learning (2022)
7. Kumar, S., Shah, N.: False information on web and social media: A survey. CRC press (2018)
8. Li, Q., Zhang, Q., Si, L., Liu, Y.: Rumor detection on social media: Datasets, methods and opportunities. In: NLP4IF workshop at EMNLP (2019)
9. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: RoBERTa: A robustly optimized bert pretraining approach. arXiv preprint (2019)
10. Min, E., Rong, Y., Bian, Y., Xu, T., Zhao, P., Huang, J., Ananiadou, S.: Divide-and-conquer: Post-user interaction network for fake news detection on social media. In: WWW (2022)
11. Mosallanezhad, A., Karami, M., Shu, K., Mancenido, M.V., Liu, H.: Domain adaptive fake news detection via reinforcement learning. In: WWW (2022)
12. Nguyen, D.Q., Vu, T., Nguyen, A.T.: BERTweet: A pre-trained language model for english tweets. In: EMNLP: System Demonstrations (2020)
13. Nielsen, D.S., McConville, R.: Mumin: A large-scale multilingual multimodal fact-checked misinformation social network dataset. In: SIGIR (2022)
14. Pérez-Rosas, V., Kleinberg, B., Lefevre, A., Mihalcea, R.: Automatic detection of fake news. In: COLING (2018)
15. Reimers, N., Gurevych, I.: Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In: EMNLP (2019)
16. Singh, L., Bansal, S., Bode, L., Budak, C., Chi, G., Kawintiranon, K., Padden, C., Vanarsdall, R., Vraga, E., Wang, Y.: A first look at covid-19 information and misinformation sharing on twitter. arXiv preprint (2020)

17. Singh, L., Bode, L., Budak, C., Kawintiranon, K., Padden, C., Vraga, E.: Understanding high-and low-quality url sharing on covid-19 twitter streams. Journal of computational social science **3**(2), 343–366 (2020)
18. Sutton, R.S., Barto, A.G.: RL: An introduction. MIT press (2018)
19. Vo, N., Lee, K.: Where are the facts? searching for fact-checked information to alleviate the spread of fake news. In: EMNLP (2020)
20. Wang, Y., Ma, F., Jin, Z., Yuan, Y., Xun, G., Jha, K., Su, L., Gao, J.: Event adversarial neural networks for multi-modal fake news detection. In: KDD (2018)
21. Wang, Y., Yang, W., Ma, F., Xu, J., Zhong, B., Deng, Q., Gao, J.: Weak supervision for fake news detection via reinforcement learning. In: AAAI (2020)
22. Wu, J., Li, L., Wang, W.Y.: Reinforced co-training. In: NAACL (2018)
23. Yoon, J., Arik, S., Pfister, T.: Data valuation using reinforcement learning. In: ICML (2020)
24. Yu, F., Liu, Q., Wu, S., Wang, L., Tan, T.: Attention-based convolutional approach for misinformation identification from massive and noisy microblog posts. Computers & Security **83**, 106–121 (2019)
25. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: BERTScore: Evaluating text generation with bert. In: ICLR (2020)
26. Zhou, X., Zafarani, R.: A survey of fake news: Fundamental theories, detection methods, and opportunities. ACM Computing Surveys **53**(5), 1–40 (2020)