# Navigating the Kaleidoscope of COVID-19 Misinformation Using Deep Learning

**Yuanzhi Chen**
University of Nebraska-Lincoln
NE 68588, USA
yuanzhi@huskers.unl.edu

**Mohammad Rashedul Hasan**
University of Nebraska-Lincoln
NE 68588, USA
hasan@unl.edu

## Abstract

Irrespective of the success of the deep learning-based mixed-domain transfer learning approach for solving various Natural Language Processing tasks, it does not lend a generalizable solution for detecting misinformation from COVID-19 social media data. Due to the inherent complexity of this type of data, caused by its dynamic (context evolves rapidly), nuanced (misinformation types are often ambiguous), and diverse (skewed, fine-grained, and overlapping categories) nature, it is imperative for an effective model to capture both the local and global context of the target domain. By conducting a systematic investigation, we show that: (i) the deep Transformer-based pre-trained models, utilized via the mixed-domain transfer learning, are only good at capturing the local context, thus exhibits poor generalization, and (ii) a combination of shallow network-based domain-specific models and convolutional neural networks can efficiently extract local as well as global context directly from the target data in a hierarchical fashion, enabling it to offer a more generalizable solution.

## 1 Introduction

Since the start of the Coronavirus or COVID-19 pandemic, online social media (e.g., Twitter) has become a conduit for rapid propagation of misinformation (Johnson et al., 2020). Although misinformation is considered to be created without the intention of causing harm (Lazer et al., 2018), it can wreak havoc on society (Ciampaglia, 2018; Neuman, 2020; Hamilton, 2020) and disrupt democratic institutions (Ciampaglia et al., 2018). Misinformation in general, and COVID-19 misinformation in particular, has become a grave concern for the policymakers due to its fast propagation via online social media. A recent study shows that the majority of the COVID-19 social media data is rife with misinformation (Brennen et al., 2020). The first step towards preventing misinformation is to **detect misinformation** in a timely fashion.

Building automated systems for misinformation detection from social media data is a Natural Language Processing (NLP) task. Various deep learning models have been successfully employed for this type of NLP task of text classification (Kim, 2014; Conneau et al., 2017; Wang et al., 2017; Tai et al., 2015; Zhou et al., 2016). These models learn language representations from a domain, which are then used as numeric features in supervised classification. Due to the prohibitive cost of acquiring labeled data on COVID-19 misinformation, training deep learning models directly using the target data is not a suitable approach.

**Background.** An alternative approach for detecting COVID-19 misinformation from **small labeled data** is transfer learning (Hossain et al., 2020). The dominant paradigm of transfer learning employs a **mixed-domain** strategy in which representations learned from a general domain (source data) by using domain-agnostic models are transferred into a specific domain (target data) (Pan and Yang, 2009). Specifically, it involves creating a pre-trained model (PTM) that learns embedded representations from general-purpose unlabeled data, then adapting the model for a downstream task using the labeled target data (Minaee et al., 2021; Qiu et al., 2020).

Two types of neural networks can be used to create PTMs, i.e., shallow and deep. The shallow models such as Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) learn word embeddings that capture semantic, syntactic, and some global relationships (Levy and Goldberg, 2014; Srinivasan and Ribeiro, 2019) of the words from the source text using their co-occurrence information. However, these PTMs do not capture the context of the text (Qiu et al., 2020). On the other hand, deep PTMs can learn contextual embed-

*Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6000–6017
November 7–11, 2021. ©2021 Association for Computational Linguistics

dings, i.e., language models (Goldberg and Hirst, 2017).

Two main approaches for creating deep PTMs are based on sequential and non-sequential models. The sequential Recurrent Neural Network (Liu et al., 2016) based model such as ELMo (Embeddings from Language Models) (Peters et al., 2018) is equipped with long short-term memory to capture the local context of a word in sequential order. The non-sequential Transformer (Vaswani et al., 2017) based models such as OpenAI GPT (Generative Pre-training) (Radford and Sutskever, 2018), BERT (Bidirectional Encoder Representation from Transformer) (Devlin et al., 2019), and XLNet (Yang et al., 2019) utilize the attention mechanism (Bahdanau et al., 2016) for learning universal language representation from general-purpose very large text corpora such as Wikipedia and BookCorpus (Devlin et al., 2019) as well as from web crawls (Liu et al., 2019). While GPT is an autoregressive model that learns embeddings by predicting words based on previous predictions, BERT utilizes the autoencoding technique based on bidirectional context modeling (Minaee et al., 2021). XLNet leverages the strengths of autoregressive and autoencoding PLMs (Yang et al., 2019).

Unlike the Transformer-based deep PTMs, the shallow Word2Vec and GloVe as well as the deep ELMo PTMs are used only as feature extractors. These features are fed into another model for the downstream task of classification, which needs to be trained from scratch using the target data. The deep PTM based mixed-domain transfer learning has achieved state-of-the-art (SOTA) performance in many NLP tasks including text classification (Minaee et al., 2021).

Irrespective of the success of the mixed-domain SOTA transfer learning approach for text classification, there has been no study to understand how effective this approach is for navigating through the kaleidoscope of COVID-19 misinformation. Unlike the curated static datasets on which this approach is tested (Minaee et al., 2021), the dynamic landscape of the COVID-19 social media data has not been fully explored. Some key properties of the COVID-19 data hitherto identified are: (i) The COVID-19 misinformation spreads faster on social media than any other form of health misinformation (Johnson et al., 2020). As a consequence, the misinformation narrative evolves rapidly (Cui and Lee, 2020). (ii) The COVID-19 misinformation categories are heavily-skewed (Cui and Lee, 2020; Memon and Carley, 2020) and fine-grained (Memon and Carley, 2020). (iii) The COVID-19 social media misinformation types are often ambiguous (e.g., fabricated, reconfigured, satire, parody) (Brennen et al., 2020) and categories may not be mutually exclusive (Memon and Carley, 2020). These properties pose a unique challenge for the mixed-domain SOTA transfer learning approach for creating an effective solution to the COVID-19 misinformation detection problem.

Previously, it has been shown that the transfer learning approach generalizes poorly when the domain of the source dataset is significantly different from that of the target dataset (Peters et al., 2019). On the other hand, **domain-specific models (DSM)**, which learn representations from domains that are similar to the target domain, provide a generalizable solution for the downstream NLP task (Beltagy et al., 2019; Lee et al., 2019; Gu et al., 2021). These models are better at capturing the context of the target domain. However, the efficacy of the DSM-based approach for addressing the COVID-19 misinformation detection problem has not also been investigated.

**In this paper**, we conduct a systematic extensive study to understand the **scope and limitations** of the mixed-domain transfer learning approach as well as the DSM-based approach to detect COVID-19 misinformation on social media. We use both shallow and deep PTMs for the mixed-domain transfer learning experimentations. The deep PTMs include BERT, XLNet, and two variants of BERT, i.e., RoBERTa (Liu et al., 2019) and ALBERT (Lan et al., 2020). While these attention mechanism-based Transformer models are good at learning contextual representations, their ability to learn **global relationships** among the words in the source text is limited (Lu et al., 2020).

The DSMs used in our study are based on shallow architectures. We argue that shallow architectures can be trained efficiently using the limited available domain data. Specifically, we pre-train the DSMs using the small social media data on COVID-19. The shallow DSM-based approach is examined in **two dimensions: graph-based DSM and non-graph DSM**. The graph-based Text GCN (Yao et al., 2019) model can explicitly capture the **global relationships** (from term co-occurrence) by leveraging the graph structure of the text. It creates a heterogeneous word document graph with words and documents as nodes for the whole corpus, and turns document classification problem into a node classification problem. We include another graph-based model in our study, i.e., the VGCN-BERT (Lu et al., 2020). It combines the strength of Text GCN (to capture global relationships) and BERT (to capture local relationships).

The non-graph DSM models such as Word2Vec and GloVe can mainly capture **local relationships** among the words of the source text, represented in the latent space of their word embeddings. For extracting **global relationships** from these embeddings, we utilize a Convolutional Neural Network (CNN). Specifically, we use the word embeddings as input features to a CNN with a one-dimensional kernel (Kim, 2014), which then learns global relationships as high-level features. We **hypothesize** that the local and global relationships should improve the generalization capability of the non-graph DSM+CNN approach.

We evaluate the **generalizability** of the above-mentioned diverse array of NLP techniques via a set of studies that explore **various dimensions of the COVID-19 data**. We focus on the Twitter social media platform because of its highest number of news-focused users (Hughes and Wojcik, 2019). In addition to analyzing the tweet messages, we use online news articles referred to in the tweets. Our study spans along multiple dimensions of the COVID-19 data that include temporal dimension (the context in the dataset evolves), length dimension (short text such as tweets vs. lengthy text such as news articles), size dimension (small dataset vs. large dataset), and classification-level dimension (binary vs. multi-class data).

**Contributions.** We design a novel study for examining the generalizability of a diverse set of deep learning NLP techniques on the multi-dimensional space of COVID-19 online misinformation landscape. Our main contributions are as follows.

- We identify the unique challenges for the deep learning based NLP techniques to detect misinformation from COVID-19 social media data.
- We argue that an effective model for this type of data must capture both the local and the global context of the domain in its latent space of embeddings.
- We show that the mixed-domain deep learning SOTA transfer learning approach is not always effective.
- We find that the shallow CNN classifier initialized with word embeddings learned via the non-graph DSMs is more effective across most of the dimensions of the COVID-19 data space, especially when the labeled target data is small.
- We **explain** why the Transformer-based mixed-domain transfer learning approach is not effective on COVID-19 data as well as why the non-graph DSM+CNN may offer a more generalizable solution.

The rest of the paper is organized as follows. In section 2, we present the diverse NLP techniques, analyze the multi-dimensional datasets, and describe the study design. Results obtained from the experiments are provided in section 3 followed by a detailed analysis. Section 4 presents the conclusion. Appendix provides related work, additional analysis of the datasets, and experiment setting.

## 2 Method

First, we describe how we obtained various PTMs and created DSM embeddings for different models as well as how we fine-tuned/trained the classifiers for the studies. Then, we discuss the datasets and the study design.

### 2.1 Mixed-Domain Transfer Learning

We use the following PTMs: BERT, RoBERTa, ALBERT, XLNet, ELMo, Word2Vec, and GloVe.

**Deep PTMs:** We get the BERT base model (uncased) for sequence classification from the Hugging Face library (Wolf et al., 2020). The embedding vectors are 768-dimensional. This BERT PTM adds a single linear layer on top of the BERT base model. The pretrained weights of all hidden layers of the PTM and the randomly initialized weights of the top classification layer are adapted during fine-tuning using a target dataset. The XLNet is obtained from the Hugging Face library (Wolf et al., 2020) and fine-tuned similar to BERT. Its embedding vectors are 768-dimensional. The RoBERTa (obtained from (Wolf et al., 2020)) and ALBERT (obtained from (Maiya, 2020)) are used by first extracting embeddings from their final layer and then adding linear layers. While the RoBERTa embeddings are 768-dimensional, the ALBERT embeddings are 128-dimensional.

**Shallow PTMs:** We get the ELMo embeddings from TensorFlow-Hub (Abadi et al., 2015). Each embedding vector has a length of 1024. The Word2Vec embeddings are obtained from Google Code (Google Code, 2013). The embedding vectors are 300-dimensional. We get the GloVe pretrained 300-dimensional embeddings from (Pennington et al., 2014).

**CNN:** The ELMo, Word2Vec, and GloVe embeddings are used to train a CNN classifier with a single hidden layer (Kim, 2014). The first layer is the embedding layer. Its dimension varies based on the dimension of pretrained embeddings. The second layer is the one-dimensional convolution layer that consists of 100 filters of dimension 5 x 5 with "same" padding and ReLU activation. The third layer is a one-dimensional global max-pooling layer, and the fourth layer is a dense layer with 100 units along with ReLU activation. The last layer is the classification layer with softmax activation. We use this setting for the CNN architecture as it was found empirically optimal in our experiments. We use cross-entropy as loss function, Adam as the optimizer, and a batch size of 128. The embedding vectors are kept fixed during the training (Kim, 2014).

## 2.2 Domain-Specific Model (DSM) based Learning

We create the DSMs using two approaches: graph-based and non-graph. For the graph-based approach, we use the following models: Text GCN and VGCN-BERT. For training the Text GCN model, we pre-process the data as follows. First, we clean the text by removing stop words and rare words whose frequencies are less than 5. Then, we build training, validation, and test graphs using the cleaned text. Finally, we train the GCN model using training and validation graphs and test the model using a test graph. During the training, early stopping is used.

For training the VGCN-BERT model, first, we clean the data that includes removing spaces, the special symbols as well as URLs. Then, the BERT tokenizer is used to create BERT vocabulary from the cleaned text. The next step is to create training, validation, and the test graphs. The last step is training the VGCN-BERT model. During the training, the model constructs embeddings from word and vocabulary GCN graph.

For the non-graph approach, we create embeddings from the target dataset by using the Word2Vec and GloVe models. First, we preprocess the raw text data by converting the text (i.e., a list of sentences) into a list of lists containing tokenized words. During tokenization, we convert words to lowercase, remove words that are only one character, and lemmatize the words. We add bigrams that appear 10 times or more to our tokenized text. The bigrams allow us to create phrases that could be helpful for the model to learn and produce more meaningful representations. Then, we feed our final version of the tokenized text to the Word2Vec and the GloVe model for creating embeddings. After we obtain the embeddings, we use them to train the CNN classifier described in the previous sub-section, except that the domain-specific word embeddings are adapted during the training.

## 2.3 Dataset

We use two COVID-19 datasets for the study, i.e., CoAID (Cui and Lee, 2020) and CMU-MisCov19 (Memon and Carley, 2020).

The CoAID dataset contains two types of data: true information and misinformation. We use this dataset to investigate the generalizability of the models along three dimensions.

- **Temporal dimension**: Train a model using data from an earlier time, then test its generalizability at different times in the future.
- **Size dimension**: Train models by varying the size of the training dataset.
- **Length dimension**: Train models by varying the length of the samples, e.g., tweet (short-length data) and news articles (lengthy data).

The CMU-MisCov19 dataset is used to analyze a model's performance in fine-grained classification.

### 2.3.1 CoAID: Binary Classification

The CoAID dataset (Cui and Lee, 2020) is used for binary classification since it has only two labels: 0 for misinformation and 1 for true information. This dataset contains two types of data: online news articles on COVID-19 and tweets related to those articles. Datasets of these two categories were collected at four different months in 2020: May, July, September, and November. Thus, the total number of CoAID datasets is 8. The class distribution is **heavily skewed** with significantly more true information samples than misinformation samples. Sample distribution per class (both for the tweets and news articles) is given in the appendix.

#### 2.3.2 CMU-MisCov19: Fine-Grained Classification

The CMU-MisCov19 dataset contains 4,573 annotated tweets (Memon and Carley, 2020). The tweets were collected on three days in 2020: March 29, June 15, and June 24. The categories are fine-grained comprising of 17 classes with **skewed distribution**. This dataset does not have any true information category. Its sample distribution per class is given in the appendix.

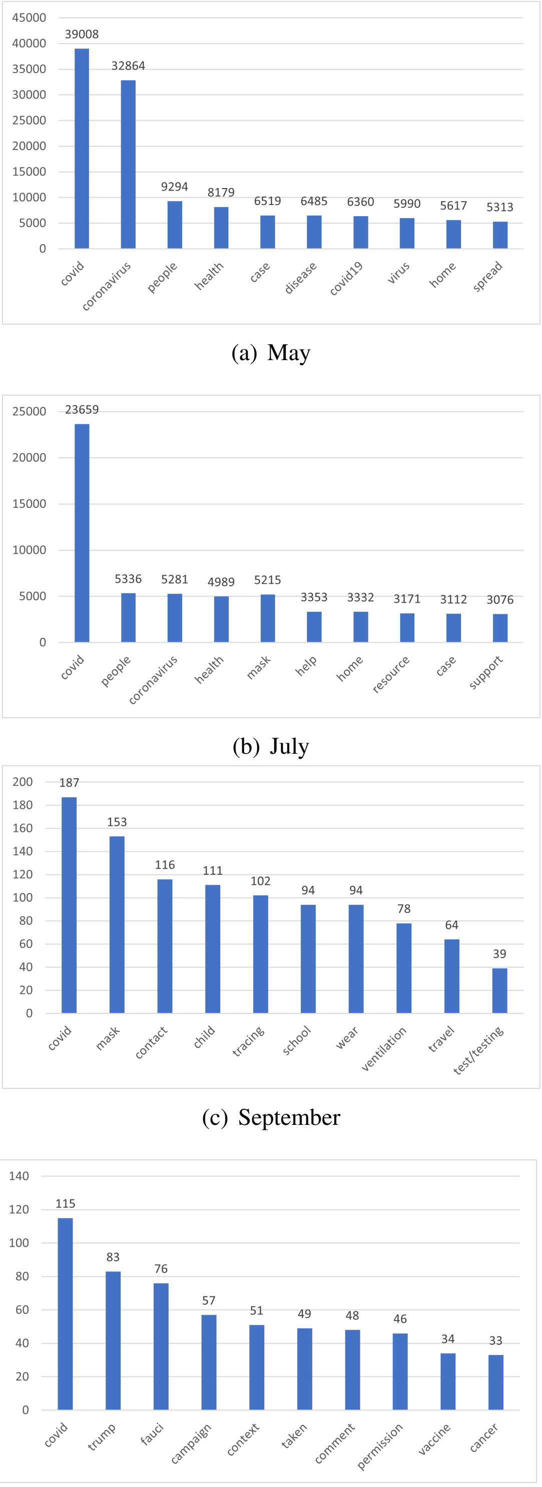
### 2.4 Context Evolution in the COVID-19 Social Media Data

We use the CoAID dataset to understand whether the context of the COVID-19 text evolves. To detect a change in the context over time, we investigate how the distribution of the high-frequency terms evolve for the two categories of the data: tweets and news articles. For each category, we select the top 10 high-frequency words from the 4 non-overlapping datasets belonging to 4 subsequent months, i.e., May, July, September, and November in 2020. Our goal is to determine whether there exists a temporal change in the distribution of high-frequency words.

Figure 1 shows context evolution in the tweets category. We see that during May, the two high-frequency words were covid and coronavirus. The frequent words represent broader concepts such as health, disease, spread, etc. However, over time the context shifted towards more loaded terms. For example, in July two new high-frequency words, such as mask and support, emerged. Then, in September words like contact, school, child, and travel became prominent. Finally, during November, we observe **a sharp change** in the nature of the frequent words. Terms with strong political connotations (e.g., trump, fauci, campaign, and vaccine) started emerging. The evolution in the high-frequency words indicates a temporal shift in the context in the tweets dataset. We observe similar context evolution in the news articles dataset, reported in the Appendix with additional analysis.

### 2.5 Study Design

We describe the design of the studies for comparing the NLP approaches for misinformation detection.



(a) May

(b) July

(c) September

(d) November

Figure 1: CoAID (Tweets): Frequency of top 10 words in four datasets from four subsequent months.

#### 2.5.1 Study 1

Study 1 is designed to explore a model's generalizability in the **temporal dimension** of the data. We fine-tune/train a model using CoAID data collected from May 2020 and test it using data obtained from 3 different months in "future": July, September, and November. The following models are tested in this study: BERT (Mixed-domain Transfer Learning), ELMo (Mixed-domain Transfer Learning), Word2Vec (Mixed-domain Transfer Learning and DSM-based), GloVe (Mixed-domain Transfer Learning and DSM-based), Text-GCN (DSM-based), and VGCN-BERT (DSM-based).

### 2.5.2 Study 2

Study 2 is designed to test the performance of a model along the **length dimension** of the data. We use both short-length data (tweets) and lengthy data (news articles). Specifically, we train a model using the CoAID Twitter dataset to understand a model's performance on the short-length data. Then, we train a model using the CoAID news articles dataset to study a model's performance on the lengthy data. The models used in this study are the same as in Study 1.

### 2.5.3 Study 3

In study 3, we evaluate a model along the **size dimension** of the data. We replicate studies 1 and 2 using a large target dataset, which is created by merging the datasets from May, July, and September. The November dataset is used as the test set. We experiment with two models for this study: BERT (Mixed-Domain Transfer Learning) and Word2Vec (DSM-based).

### 2.5.4 Study 4

To further study the effectiveness of the Transformer-based mixed-domain transfer learning approach, we experiment with two variants of the BERT PTM, i.e., RoBERTa and ALBERT. In addition to this, we study the performance of an autoregressive model XLNet that induces the strength of BERT. For this study, we only use the news articles dataset.

### 2.5.5 Study 5

Study 5 is designed to test a model's performance on the fined-grained CMU-MisCov19 dataset. The models tested are the same as in Study 1.

## 3 Results and Analysis

We evaluate the performance of the models based on the accuracy, precision, recall, and f1 score, with an emphasis on the misinformation class. For each experiment, we average the results for 10 runs. The experiments are done using Scikit-learn (Pedregosa et al., 2011), TensorFlow 2.0 (Abadi et al., 2015), and PyTorch (Paszke et al., 2019) libraries. For creating the Word2Vec embeddings, we used the skip-gram model from the Gensim library (Řehůřek and Sojka, 2010). Finally, the GloVe embeddings are created using the model from (Glove-Python, 2016).

**Results.** Table 1 and Table 2 show the results from studies 1 and 2.

From the results on the CoAID tweets, given in Table 1, we see that for the July tweet test dataset (Table 1), VGCN-BERT has the highest misinformation precision. However, misinformation recall and f1 scores for all models are poor. For September, the Text-GCN has outstanding performance for detecting misinformation, but its performance on true information is extremely poor. Other models perform badly on misinformation. For November, the GloVe-based transfer learning approach achieves excellent performance on both true information and misinformation, where precision, recall, and f1 scores are 1. Text-GCN also has decent scores on misinformation but fails to detect true information. The performance of BERT on both true information and misinformation is also good. However, we notice that no model performs well across three different test datasets. Thus, we see that **mixed-domain transfer learning is not robust when the context of the short-length data (tweets) changes**. This is also true for the DSM-based approach.

Table 2 shows the results of CoAID news articles (lengthy text). For the July test dataset, both Text-GCN and Word2Vec (DSM-based) achieve decent precision, recall, and f1 scores on misinformation. However, Text-GCN has extremely poor performance on true information. On the September data, ELMo exhibits the best misinformation precision, and f1 score, while Word2Vec (DSM-based) gives the best misinformation recall score. Both ELMo and Word2Vec perform well on the true information class as well. As for the November data, both transfer learning and DSM-based Word2Vec obtain optimal misinformation precision score and Word2Vec (DSM-based) obtains the highest f1 score. Besides, VGCN-BERT achieves the highest misinformation recall score. We notice that the DSM-based Word2Vec exhibits comparatively better performance across all test datasets. Thus, the **non-graph DSM+CNN can capture both global and local relationships from lengthy text relatively well**. The performance of the graph-

| **Train: May** | **Test: July** | | | | | | **Test: September** | | | | | | **Test: November** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Model** | **True Information** | | | **Misinformation** | | | **True Information** | | | **Misinformation** | | | **True Information** | | | **Misinformation** | | |
| | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 |
| **Mixed-Domain Transfer Learning** | | | | | | | | | | | | | | | | | | |
| **BERT** | Accuracy = 0.978 | | | | | | Accuracy = 0.942 | | | | | | Accuracy = 0.990 | | | | | |
| | 0.981 | 0.997 | 0.989 | 0.281 | 0.079 | 0.116 | 0.946 | 0.996 | 0.970 | 0.00 | 0.00 | 0.00 | 1.00 | 0.990 | 0.995 | 0.500 | 1.00 | 0.670 |
| **ELMo** | Accuracy = 0.979 | | | | | | Accuracy = 0.941 | | | | | | Accuracy = 0.990 | | | | | |
| | 0.979 | 1.00 | 0.989 | 0.00 | 0.00 | 0.00 | 0.945 | 0.995 | 0.970 | 0.00 | 0.00 | 0.00 | 0.990 | 1.00 | 0.995 | 0.00 | 0.00 | 0.00 |
| **Word2Vec** | Accuracy = 0.979 | | | | | | Accuracy = 0.932 | | | | | | Accuracy = 0.969 | | | | | |
| | 0.979 | 1.00 | 0.989 | 0.00 | 0.00 | 0.00 | 0.949 | 0.980 | 0.964 | 0.20 | 0.09 | 0.12 | 0.99 | 0.979 | 0.984 | 0.00 | 0.00 | 0.00 |
| **GloVe** | Accuracy = 0.979 | | | | | | Accuracy = 0.943 | | | | | | Accuracy = 1.00 | | | | | |
| | 0.979 | 0.999 | 0.989 | 0.31 | 0.02 | 0.03 | 0.946 | 0.998 | 0.971 | 0.00 | 0.00 | 0.00 | 1.00 | 1.00 | 1.00 | **1.00** | **1.00** | **1.00** |
| **DSMs: Graph-based** | | | | | | | | | | | | | | | | | | |
| **Text GCN** | Accuracy = 0.979 | | | | | | Accuracy = 0.946 | | | | | | Accuracy = 0.99 | | | | | |
| | 0.979 | 1.00 | 0.989 | 0.029 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.95 | 1.00 | 0.97 | 0.00 | 0.00 | 0.00 | 0.99 | 1.00 | 0.99 |
| **VGCN-BERT** | Accuracy = 0.978 | | | | | | Accuracy = 0.946 | | | | | | Accuracy = 0.971 | | | | | |
| | 0.979 | 0.999 | 0.989 | **0.495** | 0.019 | 0.035 | 0.947 | 0.999 | 0.972 | **0.287** | 0.029 | 0.052 | 0.994 | 0.977 | 0.984 | 0.21 | 0.389 | 0.246 |
| **DSMs: Non-Graph + CNN** | | | | | | | | | | | | | | | | | | |
| **Word2Vec** | Accuracy = 0.977 | | | | | | Accuracy = 0.946 | | | | | | Accuracy = 0.99 | | | | | |
| | 0.979 | 0.997 | 0.988 | 0.11 | 0.02 | 0.03 | 0.946 | 1.00 | 0.972 | 0.00 | 0.00 | 0.00 | 0.99 | 1.00 | 0.995 | 0.00 | 0.00 | 0.00 |
| **GloVe** | Accuracy = 0.978 | | | | | | Accuracy = 0.946 | | | | | | Accuracy = 0.979 | | | | | |
| | 0.979 | 0.999 | 0.989 | 0.23 | 0.02 | 0.03 | 0.946 | 1.00 | 0.972 | 0.00 | 0.00 | 0.00 | 0.99 | 0.99 | 0.99 | 0.00 | 0.00 | 0.00 |

Table 1: Study 1 & 2: CoAID - Tweet (Temporal & Text Length Dimension). Best results, as well as the optimal models, are highlighted in red. **None of the models generalize well on the tweet data.**

| **Train: May** | **Test: July** | | | | | | **Test: September** | | | | | | **Test: November** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Model** | **True Information** | | | **Misinformation** | | | **True Information** | | | **Misinformation** | | | **True Information** | | | **Misinformation** | | |
| | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 |
| **Mixed-Domain Transfer Learning** | | | | | | | | | | | | | | | | | | |
| **BERT** | Accuracy = 0.814 | | | | | | Accuracy = 0.646 | | | | | | Accuracy = 0.503 | | | | | |
| | 0.814 | 1.00 | 0.898 | 0.00 | 0.00 | 0.00 | 0.929 | 0.676 | 0.779 | 0.018 | 0.135 | 0.036 | 0.962 | 0.511 | 0.656 | 0.009 | 0.144 | 0.009 |
| **ELMo** | Accuracy = 0.559 | | | | | | Accuracy = 0.973 | | | | | | Accuracy = 0.985 | | | | | |
| | 0.777 | 0.642 | 0.703 | 0.11 | 0.19 | 0.14 | 0.979 | 0.993 | 0.986 | **0.83** | 0.64 | **0.72** | 0.986 | 0.99 | 0.992 | 0.88 | 0.37 | 0.52 |
| **Word2Vec** | Accuracy = 0.851 | | | | | | Accuracy = 0.946 | | | | | | Accuracy = 0.98 | | | | | |
| | 0.846 | 0.999 | 0.916 | 0.98 | 0.20 | 0.33 | 0.948 | 0.998 | 0.972 | 0.60 | 0.06 | 0.12 | 0.98 | 1.00 | 0.99 | 1.00 | 0.11 | 0.19 |
| **GloVe** | Accuracy = 0.599 | | | | | | Accuracy = 0.953 | | | | | | Accuracy = 0.984 | | | | | |
| | 0.833 | 0.635 | 0.721 | 0.22 | 0.44 | 0.29 | 0.957 | 0.995 | 0.975 | 0.73 | 0.23 | 0.35 | 0.985 | 0.999 | 0.992 | 0.86 | 0.32 | 0.46 |
| **DSMs: Graph-based** | | | | | | | | | | | | | | | | | | |
| **Text GCN** | Accuracy = 0.814 | | | | | | Accuracy = 0.635 | | | | | | Accuracy = 0.978 | | | | | |
| | 0.00 | 0.00 | 0.00 | 0.81 | 1.00 | 0.90 | 0.97 | 0.633 | 0.766 | 0.095 | 0.66 | 0.165 | 0.978 | 1.00 | 0.989 | 0.00 | 0.00 | 0.00 |
| **VGCN-BERT** | Accuracy = 0.677 | | | | | | Accuracy = 0.64 | | | | | | Accuracy = 0.458 | | | | | |
| | 0.971 | 0.622 | 0.758 | 0.356 | 0.917 | 0.513 | 0.985 | 0.628 | 0.767 | 0.117 | 0.839 | 0.205 | 0.989 | 0.451 | 0.619 | 0.031 | 0.778 | 0.06 |
| **DSMs: Non-Graph + CNN** | | | | | | | | | | | | | | | | | | |
| **Word2Vec** | Accuracy = 0.96 | | | | | | Accuracy = 0.643 | | | | | | Accuracy = 0.99 | | | | | |
| | 0.957 | 0.984 | 0.975 | **0.92** | **0.85** | **0.89** | 0.977 | 0.638 | 0.772 | 0.11 | **0.74** | 0.19 | 0.991 | 0.99 | 0.995 | **0.92** | **0.58** | **0.71** |
| **GloVe** | Accuracy = 0.554 | | | | | | Accuracy = 0.623 | | | | | | Accuracy = 0.452 | | | | | |
| | 0.775 | 0.637 | 0.699 | 0.10 | 0.19 | 0.13 | 0.941 | 0.641 | 0.763 | 0.05 | 0.32 | 0.09 | 0.962 | 0.457 | 0.62 | 0.01 | 0.21 | 0.02 |

Table 2: Study 1 & 2: CoAID - News Articles (Temporal & Text Length Dimension). Best results, as well as the optimal models, are highlighted in red.

| **Train: May + July + September**<br>**Test: November** | **Tweets** | | | | | | **News Articles** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Model** | **True Information** | | | **Misinformation** | | | **True Information** | | | **Misinformation** | | |
| | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 |
| **Mixed-Domain Transfer Learning: BERT** | Accuracy = 0.928 | | | | | | Accuracy = 0.992 | | | | | |
| | 1.00 | 0.927 | 0.962 | 0.12 | **1.00** | 0.22 | 0.994 | 0.998 | 0.996 | **0.883** | **0.719** | **0.787** |
| **DSM (Non-Graph): Word2Vec + CNN** | Accuracy = 0.985 | | | | | | Accuracy = 0.986 | | | | | |
| | 0.99 | 0.995 | 0.992 | **0.71** | 0.63 | **0.67** | 0.992 | 0.994 | 0.993 | 0.71 | 0.63 | 0.67 |

Table 3: Study 3: CoAID Large Dataset (Dataset Size Dimension). Best results are highlighted in red.

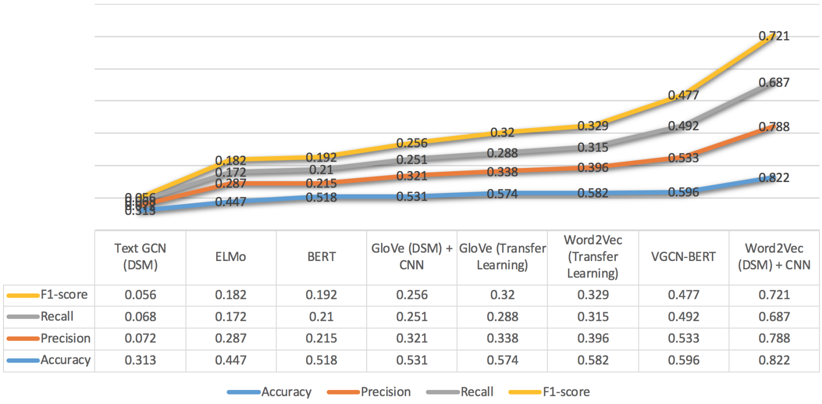
based DSM approach on lengthy text is not as good as on short text. Also, BERT shows unreliable performance as it fails on the misinformation class.

Table 3 shows the results of study 3, i.e., large-dataset-based experiments. The performance of DSM-based Word2Vec is consistent with its performance on the CoAID news articles data (Table 2). Its F1 score on tweets misinformation increases significantly compared to the small-data case (Table 1). **Thus, the non-graph DSM+CNN can capture both global and local relationships if we increase the size of short-length training data (i.e., tweets)**.

Table 4 shows the results obtained from study 4. For the July test dataset misinformation detection, RoBERTa achieves the best performance, while XLNet shows the worst performance. However, for September misinformation, we observe the exact opposite scenario. As for November misinformation, ALBERT achieves the best performance, while XLNet's performance is the worst. **No single Transformer-based model performs well on**

| Train: May | Test: July | | | | | | Test: September | | | | | | Test: November | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | True Information | | | Misinformation | | | True Information | | | Misinformation | | | True Information | | | Misinformation | | |
| | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 |
| Mixed-Domain Transfer Learning | | | | | | | | | | | | | | | | | | |
| ALBERT | Accuracy = 0.611 | | | | | | Accuracy = 0.943 | | | | | | Accuracy = 0.993 | | | | | |
| | 0.858 | 0.625 | 0.723 | 0.25 | 0.547 | 0.343 | 0.96 | 0.981 | 0.97 | 0.483 | 0.298 | 0.368 | 0.996 | 0.996 | 0.996 | **0.842** | **0.842** | **0.842** |
| RoBERTa | Accuracy = 0.936 | | | | | | Accuracy = 0.630 | | | | | | Accuracy = 0.980 | | | | | |
| | 0.937 | 0.987 | 0.962 | **0.925** | **0.711** | **0.804** | 0.952 | 0.641 | 0.766 | 0.068 | 0.447 | 0.118 | 0.991 | 0.989 | 0.990 | 0.55 | 0.579 | 0.564 |
| XLNET | Accuracy = 0.814 | | | | | | Accuracy = 0.97 | | | | | | Accuracy = 0.456 | | | | | |
| | 0.81 | 1.00 | 0.90 | 0.00 | 0.00 | 0.00 | 0.97 | 1.00 | 0.98 | **0.89** | **0.53** | **0.67** | 0.97 | 0.46 | 0.62 | 0.01 | 0.26 | 0.02 |

Table 4: Study 4: CoAID Large Dataset (Various-based Transformer Models). Best results are highlighted in red. **No single model performs well across three datasets**.



| | Text GCN (DSM) | ELMo | BERT | GloVe (DSM) + CNN | GloVe (Transfer Learning) | Word2Vec (Transfer Learning) | VGCN-BERT | Word2Vec (DSM) + CNN |
|---|---|---|---|---|---|---|---|---|
| F1-score | 0.056 | 0.182 | 0.192 | 0.256 | 0.32 | 0.329 | 0.477 | 0.721 |
| Recall | 0.068 | 0.172 | 0.21 | 0.251 | 0.288 | 0.315 | 0.492 | 0.687 |
| Precision | 0.072 | 0.287 | 0.215 | 0.321 | 0.338 | 0.396 | 0.533 | 0.788 |
| Accuracy | 0.313 | 0.447 | 0.518 | 0.531 | 0.574 | 0.582 | 0.596 | 0.822 |

Figure 2: Study 5: CMU-MisCov19 (Fine-grained classification).

**the three datasets.** These results corroborate our previous observation on the mixed-domain transfer learning approach, i.e., it is not robust when the context of the data changes.

Figure 2 shows the results obtained from study 5. We see that the mixed-domain transfer learning approach performs poorly on the fine-grained dataset. The only model that achieves decent performance is the non-graph DSM Word2Vec with CNN.

**Analysis.** Based on the results obtained from the studies, we summarize our observations below. First, we discuss a model's generalizability for binary classification scenarios. Given the length of the text and the size of the dataset, we identify 4 cases.

**Case 1: Length=Short & Size=Small** For case 1, we do not find a single best-performing model. For the tweet dataset, the following models perform slightly better: VGCN-BERT, GloVe (transfer learning and DSM-based), and Text GCN. There are two possible explanations for the poor performance of all models on the short-length tweet data. First, the number of test misinformation samples is significantly smaller. For example, in the 2020 July, September, and November tweet test datasets, the true information samples are larger than the misinformation samples by 46, 17, and 96 times, respectively. Second, the short length of the text and the small size of the training set might have influenced the scope of the context learning by the models.

**Case 2: Length=Long & Size=Small** For the news articles data, the best model is DSM Word2Vec+CNN for the July and November datasets. It achieved the highest precision and recall on the misinformation class. For the September dataset, the ELMo outperforms DSM Word2Vec+CNN.

**Case 3: Length=Short & Size=Large** Both DSM Word2Vec+CNN and BERT-based transfer learning performed well. However, BERT's performance is not consistent. On the tweet dataset (short-length text), the precision of BERT is poor. It indicates that even with larger training data, BERT-based transfer learning does not provide an effective solution for short-length samples. One possible reason is that **although BERT is good at capturing the local relationships (e.g., word order), it does not do equally well on capturing the global relationships from short-length data**.

**Case 4: Length=Long & Size=Large** Both DSM Word2Vec+CNN and BERT-based transfer learning perform well in this case. BERT's performance is slightly better. This indicates that Transformer-based models are suitable when target data is large and texts are lengthy.

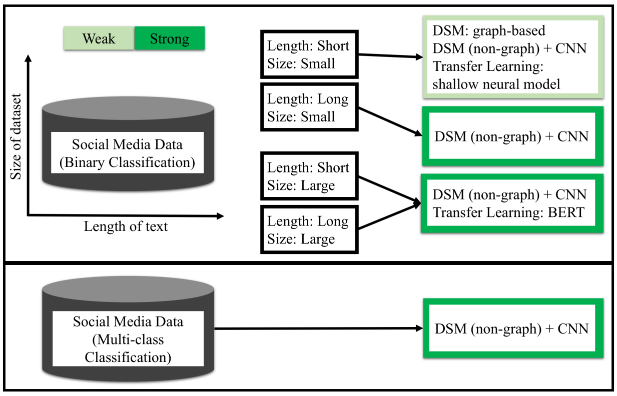


Figure 3: A framework for COVID-19 online misinformation detection.

The results from fine-grained classification show that DSM Word2Vec+CNN outperforms other approaches by a large margin. Apart from the case of binary short length and small size dataset, DSM Word2Vec+CNN is shown to achieve the most effective solution.

One possible reason for the better generalization capability of the non-graph DSM+CNN-based approach is that its hierarchical feature-extraction mechanism is conducive for learning both the **local context** (the non-graph DSM, e.g., Word2Vec captures the local relationships of words in the target text) and the **global context** (the CNN learns global relationships from the word embeddings), which **validates our hypothesis**.

Based on the insights garnered from the above analysis, we draw the following conclusions, summarized in the framework in Figure 3.

- The Transformer-based **mixed-domain transfer learning** approach is effective in limited cases. Also, its performance is not consistent.

- The **graph-based DSM** approach does not yield an effective solution in any of the cases. The VGCN-BERT that combines the benefits of Text GCN with BERT is not effective either.

- The **non-graph DSM + CNN** approach generalizes well across the last three cases.

Our study suffers from **some limitations**. The lack of labeled data narrowed the scope of our investigation. The data scarcity affected our study in two ways. First, due to the small size of the test data, we obtained noisy estimates for the short length and small size data. Second, we could not conduct a multi-dimensional study on the fine-grained classification problem.

## 4 Conclusion

When an unanticipated pandemic like COVID-19 breaks out, various types of misinformation emerge and propagate at warp speed over online social media. For detecting such misinformation, NLP techniques require to capture the context of the discourse from its evolving narrative. We argue that irrespective of the success of the deep learning based mixed-domain transfer learning approach for solving various NLP tasks, it does not yield a generalizable solution. We emphasize the importance of learning the context (both local and global) directly from the target domain via the DSM-based approach. A feasible way to implement a DSM is to utilize shallow neural networks that capture the local relationships in the target data. Representations learned from this type of model can then be used by shallow CNNs to learn global relationships as high-level features. Thus, a combination of non-graph DSM and CNN may lend a more generalizable solution. We perform an extensive study using Twitter-based COVID-19 social media data that includes tweets and news articles referred to in the tweets. Our investigation is performed along the following dimensions of the data: temporal dimension (evolving context), length dimension (varying text length), size dimension (varying size of datasets), and classification-level dimension (binary vs. multi-class data). We show that the mixed-domain transfer learning approach does not always work well. We found the combination of the non-graph DSM (for capturing local relationships) and CNN (for extracting global relationships) to be a promising approach towards creating a generalizable solution for detecting COVID-19 online misinformation.

In the future, we plan to investigate the generalizability of the DSM models created using deep learning architectures such as BERT.

## References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2016. Neural machine translation by jointly learning to align and translate.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Scott Brennen, Felix N Simon, Philip Kleis Howard, and Rasmus undefined Nielsen. 2020. Types, sources, and claims of covid-19 misinformation. *Reuters Institute for the Study of Journalism*.

Giovanni Luca Ciampaglia. 2018. Fighting fake news: a role for computational social science in the fight against digital misinformation. *Journal of Computational Social Science*, 1(1):147–153.

Giovanni Luca Ciampaglia, Alexios Mantzarlis, Gregory Maus, and Filippo Menczer. 2018. Research challenges of digital misinformation: Toward a trustworthy web. *AI Magazine*, 39(1):65–74.

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2017. Very deep convolutional networks for text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1107–1116, Valencia, Spain. Association for Computational Linguistics.

Limeng Cui and Dongwon Lee. 2020. Coaid: COVID-19 healthcare misinformation dataset. *CoRR*, abs/2006.00885.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Glove-Python. 2016. Glove-python. `https://github.com/maciejkula/glove-python`. Accessed: 2021-03-30.

Yoav Goldberg and Graeme Hirst. 2017. *Neural Network Methods in Natural Language Processing*. Morgan and Claypool Publishers.

Google Code. 2013. Google code archive - word2vec. `https://code.google.com/archive/p/word2vec/`. Accessed: 2021-04-05.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing.

Isobel Ashe Hamilton. 2020. 77 cell phone towers have been set on fire so far due to a weird coronavirus 5g conspiracy theory. Business Insider.

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

Tamanna Hossain, Robert L. Logan IV, Arjuna Ugarte, Yoshitomo Matsubara, Sean Young, and Sameer Singh. 2020. COVIDLies: Detecting COVID-19 misinformation on social media. In *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*, Online. Association for Computational Linguistics.

Adam Hughes and Stefan Wojcik. 2019. 10 facts about americans and twitter. Pew Research Center.

N. F. Johnson, N. Velasquez, O. K. Jha, H. Niyazi, R. Leahy, N. Johnson Restrepo, R. Sear, P. Manrique, Y. Lupu, P. Devkota, and S. Wuchty. 2020. Covid-19 infodemic reveals new tipping point epidemiology and a revised $r$ formula.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations*, ICLR 17.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

David M. J. Lazer, Matthew A. Baum, Yochai Benkler, Adam J. Berinsky, Kelly M. Greenhill, Filippo Menczer, Miriam J. Metzger, Brendan Nyhan, Gordon Pennycook, David Rothschild, Michael Schudson, Steven A. Sloman, Cass R. Sunstein, Emily A. Thorson, Duncan J. Watts, and Jonathan L. Zittrain. 2018. The science of fake news. *Science*, 359(6380):1094–1096.

Yann LeCun and Ishan Misra. 2021. Self-supervised learning: The dark matter of intelligence. Facebook.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *NIPS*, pages 2177–2185.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, page 2873?2879. AAAI Press.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. Arxiv:1907.11692.

Zhibin Lu, Pan Du, and Jian-Yun Nie. 2020. Vgcn-bert: Augmenting bert with graph embedding for text classification. In *Advances in Information Retrieval - 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14-17, 2020, Proceedings, Part I*, volume 12035 of *Lecture Notes in Computer Science*, pages 369–382. Springer.

Arun S. Maiya. 2020. ktrain: A low-code library for augmented machine learning. *arXiv preprint arXiv:2004.10703*.

Shahan Ali Memon and Kathleen M. Carley. 2020. Characterizing COVID-19 misinformation communities using a novel twitter dataset. *CoRR*, abs/2008.00791.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. 2021. Deep learning based text classification: A comprehensive review.

Martin Müller, Marcel Salathé, and Per Egil Kummervold. 2020. Covid-twitter-bert: A natural language processing model to analyse COVID-19 content on twitter. *CoRR*, abs/2005.07503.

Scott Neuman. 2020. Man dies, woman hospitalized after taking form of chloroquine to prevent covid-19. *NPR*.

Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew 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, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. 2019. To tune or not to tune? adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 7–14, Florence, Italy. Association for Computational Linguistics.

Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey.

Alec Radford and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. In *arxiv*.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In

*Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA.

M. Schuster and K.K. Paliwal. 1997. Bidirectional recurrent neural networks. *Trans. Sig. Proc.*, 45(11):2673–2681.

Balasubramaniam Srinivasan and Bruno Ribeiro. 2019. On the equivalence between node embeddings and structural graph representations. *CoRR*, abs/1910.00452.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. Cite arxiv:1503.00075Comment: Accepted for publication at ACL 2015.

Atharva Tendle and Mohammad Rashedul Hasan. 2021. A study of the generalizability of self-supervised representations. *Machine Learning with Applications*, 6:100124.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Jin Wang, Zhongyuan Wang, Dawei Zhang, and Jun Yan. 2017. Combining knowledge with deep convolutional neural networks for short text classification. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, pages 2915–2921. AAAI Press.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):7370–7377.

Peng Zhou, Zhenyu Qi, Suncong Zheng, Jiaming Xu, Hongyun Bao, and Bo Xu. 2016. Text classification improved by integrating bidirectional lstm with two-dimensional max pooling. In *COLING*, pages 3485–3495. ACL.

# Appendix

In this section, first, we discuss the related work. Then, we present an analysis of the dataset. Finally, we report the experimental setting and training statistics.

## 5 Related Work

Solving Natural Language Processing (NLP) tasks using Deep Learning (DL) based models is a challenging venture. Unlike computer vision problems in which deep learning supervised model can learn expressive representations directly from raw pixels of the input data while performing a discrimination task, the deep learning based supervised NLP systems cannot use raw text input while solving NLP tasks. The text input data needs to be encoded with latent representations or embeddings. These embeddings are learned by neural models from general-purpose unlabeled data using the self-supervised learning approach (Tendle and Hasan, 2021). The embeddings must capture the multi-dimensional relationships of the text components, which are non-contextual and contextual relationships. The non-contextual relationship includes syntactic relationships and semantic relationships. On the other hand, the contextual relationship includes dynamic representations of words, which requires embeddings to capture the local and global relationships of the words.

Sequence DL models such as Recurrent Neural Network (RNN) have been used to learn the local context of a word in sequential order (Sutskever et al., 2014). RNNs process text as a sequence of words for capturing word dependencies and text structures. However, they suffer from two limitations. First, they are unable to create good representations due to the uni-directional processing (Peters et al., 2018). Second, these models struggle with capturing long-term dependency (Hochreiter and Schmidhuber, 1997). These two issues were partially resolved by introducing the bi-directional LSTM model (Schuster and Paliwal, 1997). This model was combined with two-dimensional max-pooling in (Zhou et al., 2016) for capturing text features. In addition to this type of chain-structured LSTM, tree-structured LSTM such as the Tree-LSTM model was developed for learning rich semantic representations (Tai et al., 2015). Irrespective of the progress harnessed by RNN-based models, they do not perform well in capturing global relationships (i.e., long-term dependencies) among the words of the source text. Also, training this type of model on large data is inefficient.

An efficient approach for some NLP tasks such as text classification is a shallow Convolutional Neural Network (CNN) with a one-dimensional convolutional kernel (Kim, 2014). This model is good at capturing local patterns such as key phrases in the text. However, it does not work effectively if the weights of the input layer are initialized randomly (Kim, 2014). It was shown to be effective only in transfer learning in which, first, word embeddings are created using a self-supervised pre-trained model (PTM) such as Word2Vec (Mikolov et al., 2013), then the CNN uses its single layer of convolution on top of the word embeddings to learn high-level representations.

The use of PTMs for mixed-domain transfer learning ushered in a new era in NLP (Qiu et al., 2020). The PTMs are created from general-purpose unlabeled data by using the self-supervised learning technique. In general, the SSL technique learns representations by predicting a hidden property of the input from the observable properties (LeCun and Misra, 2021). Two types of PTMs are used in NLP: (i) PTMs that are feature extractors, i.e., learn word embeddings (Mikolov et al., 2013; Pennington et al., 2014; Peters et al., 2018), which are used as input to another model for solving a downstream NLP task (Kim, 2014), and (ii) PTMs that learn language models and the same PTM is adapted (fine-tuned) for solving downstream NLP tasks (Devlin et al., 2019). The feature extractor PTMs such as Word2Vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), and ELMo (Peters et al., 2018) are based on both shallow and deep neural network architectures. While shallow Word2Vec and GloVe models learn non-contextual word embeddings from unlabeled source data, the deep ELMo model is good for creating contextual embeddings. Features learned from these PTMs are used as input to another neural network for solving a downstream NLP task using labeled target data.

Both Word2Vec and GloVe learn word embeddings from their co-occurrence information. While Word2Vec leverages co-occurrence within the local context, GloVe utilizes global word-to-word co-occurrence counts from the entire corpus. Word2Vec is a shallow feed-forward neural network-based predictive model that learns embeddings of the words while improving their predictions within the local context. On the other hand, GloVe is a count-based model that applies dimensionality reduction on the co-occurrence count matrix for learning word embeddings. These two models are good at capturing syntactic as well as semantic relationships. Although they can capture some global relationships between words in a text (Levy and Goldberg, 2014; Srinivasan and Ribeiro, 2019), their embeddings are context-independent. Thus, these two models are not good at language modeling. A language model can predict the next word in the sequence given the words that precede it (Goldberg and Hirst, 2017), which requires it to capture the context of the text. The deep architecture feature extractor PTM ELMo (Embeddings from Language Models) (Peters et al., 2018) learns contextualized word embeddings, i.e., it maps a word to different embedding vectors depending on their context. It uses two LSTMs in the forward and backward directions to encode the context of the words. The main limitation of this deep PTM is that it is computationally complex due to its sequential processing of text. Thus, it is prohibitively expensive to train using a very large text corpus. Another limitation of this model, which also applies to feature extractor PTMs in general, is that for solving downstream NLP tasks we need to train the entire model, except for the input embedding layer, from scratch.

The above two limitations of the feature extractor PTMs are addressed by a very deep architecture-based Transformer model (Vaswani et al., 2017). Unlike the feature extractor sequential PTMs, Transformer is a non-sequential model that uses self-attention (Bahdanau et al., 2016) to compute an attention score for capturing the influence of every word on other words in a sentence or document. This process is parallelized, which enables training deep Transformer models efficiently using very large text corpus such as Wikipedia and BookCorpus (Devlin et al., 2019) as well as web crawls (Liu et al., 2019).

There are two main types of Transformer-based deep PTMs: autoregressive and autoencoding. The OpenAI GPT (Generative Pre-training) (Radford and Sutskever, 2018) is an autoregressive model that learns embeddings by predicting words based on previous predictions. Specifically, it is a uni-directional model that predicts words sequentially in a text. On the other hand, BERT (Devlin et al., 2019) utilizes the autoencoding technique based on bi-directional context modeling. Specifically, for training, it uses a masked language modeling (MLM) task. The MLM randomly masks some tokens in a text sequence, then it predicts the masked tokens by learning the encoding vectors.

Variants of BERT such as RoBERTa (Liu et al., 2019) and ALBERT (Lan et al., 2020) were proposed to improve its effectiveness as well as efficiency. RoBERTa (Robustly optimized BERT) improves the effectiveness of BERT by using several strategies that include the following. It trains the model longer using more data, lengthy input, and larger batches. It uses a dynamic masking strategy and removes BERT's Next Sentence Prediction (NSP) task. ALBERT (A Lite BERT) improves the efficiency of BERT by employing fewer parameters, which increases its training speed.

There have been attempts such as in XLNet (Yang et al., 2019) to integrate the strengths of the autoregressive and autoencoding Transformer techniques. XLNet is an autoregressive model that uses a permutation language modeling objective. This allows XLNet to retain the advantages of autoregressive models while leveraging the benefit of the autoencoding models, i.e., to capture the bi-directional context.

Irrespective of the state-of-the-art (SOTA) performance of the deep PTM based mixed-domain transfer learning approach on many NLP tasks (Minaee et al., 2021), this approach is not suitable for detecting misinformation from COVID-19 social media data. It generalizes poorly when the domain of the source dataset used to create the PTMs is significantly different from that of the target dataset (Peters et al., 2019). One solution to this generalizability problem is to create a PTM using data that shares context similar to the target domain, i.e., pre-train a domain-specific model (DSM). This type of model encodes the context of the target domain more effectively to provide a generalizable solution for the downstream task (Beltagy et al., 2019; Lee et al., 2019; Gu et al., 2021). However, pre-training a deep architecture-based DSM (e.g., BERT) for the COVID-19 misinformation detection task in a timely fashion could be infeasible as it requires collecting a large amount of COVID-19 social media data, which must cover the diverse landscape of COVID-19 misinformation. While there was an effort to create such a deep DSM using COVID-19 tweets in (Müller et al., 2020), capturing the dynamic context of the pandemic requires the collection of various types of social media data at a large scale.

Thus, to create DSMs for the COVID-19 domain using quickly collectible small data, shallow architecture based PTMs such as Word2Vec and GloVe are suitable. However, as mentioned earlier, these PTMs are context-independent and are not good at capturing the global relationships well. To compensate for these shortcomings, we used the extracted features from the Word2Vec and GloVe DSMs for training a one-dimensional convolutional kernel-based CNN similar to the shallow architecture given in (Kim, 2014). The CNN learns global relationships by extracting local patterns in a hierarchical fashion by convolving over the word embeddings.

Another type of DSM we used is graph-based that leverages the linguistic-aware graph structure of the text for learning contextual representations, then uses those representations to solve a downstream NLP task. The main intuition driving the graph-based technique is that by modeling the vocabulary graph, it will be possible to encode global relationships in the embeddings. Text GCN (Text Graph Convolutional Network) (Yao et al., 2019) is a graph-based model that explicitly captures the global term-co-occurrence information by leveraging the graph structure of the text. It models the global word co-occurrence by incorporating edges between words as well as edges between a document and a word. Word-word edges are created by using word co-occurrence information and word-document edges are created by using word frequency and word-document frequency. Its input is a one-hot vector representation of every word in the document, which is used to create a heterogeneous text graph that has word nodes and document nodes. These are fed into a two-layer GCN (Graph Convolutional Network) (Kipf and Welling, 2017) that turns document classification into a node classification problem.

Although Text GCN is good at convolving the global information in the graph, it does not take into account local information such as word orders. To address this issue, Text GCN was combined with BERT, which is good at capturing local information. The resulting model is VGCN-BERT (Lu et al., 2020). BERT captures the local context by focusing on local word sequences, but it is not good at capturing global information of a text. It learns representations from a sentence or a document. However, it does not take into account the knowledge of the vocabulary. Thus its language model may be incomplete. On the other hand, Text GCN captures the global vocabulary information. The VGCN-BERT aims to capture both local and global relationships by integrating GCN with BERT. Both the graph embeddings and word embeddings are fed into a self-attention encoder in BERT. When the classifier is trained, these two types of embeddings interact with each other through the self-attention mechanism. As a consequence, the classifier creates representations by fusing global information with local information in a guided fashion.

## 6 Dataset

We describe the sample distribution of both the CoAID (binary) and CMU datasets. Then, we analyze context evolution in the CoAID new articles dataset.

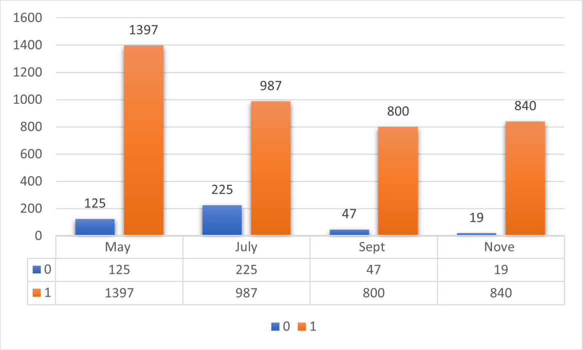


Figure 4: CoAID Tweets: Sample distribution (0: misinformation, 1: true information).

**CoAID Sample Distribution.** Figures 4 and 5 show the distributions of tweets and news articles per category, respectively. We see that the datasets contain significantly more true information than misinformation. Thus, the CoAID data is **heavily skewed**. For the tweets dataset, the sizes of May and July data are larger than that of September and November. Also, the number of misinformation tweets during September and November are negligibly smaller, making it challenging to use these as test datasets.

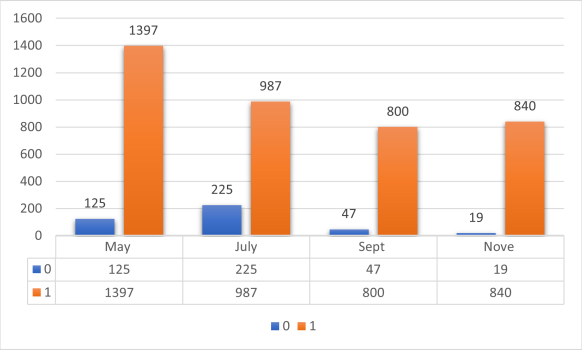


Figure 5: CoAID News Articles: Sample distribution (0: misinformation, 1: true information).

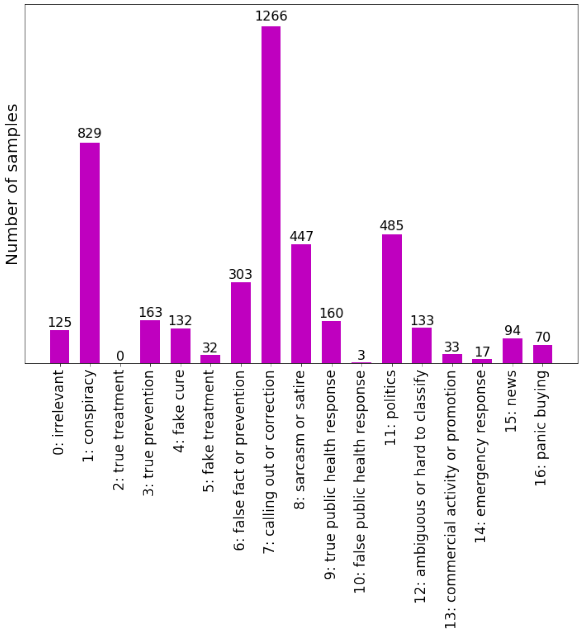
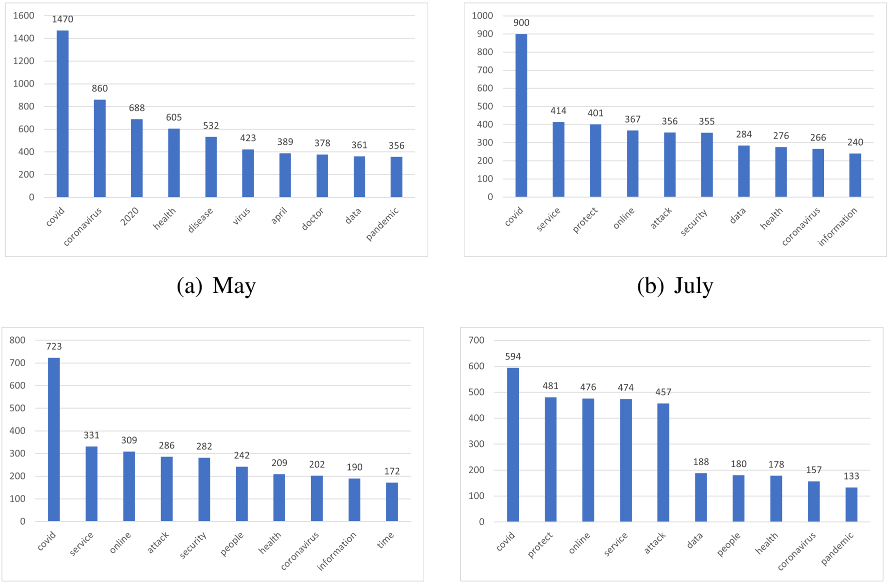


Figure 6: CMU-MisCov19 Dataset: Sample distribution.

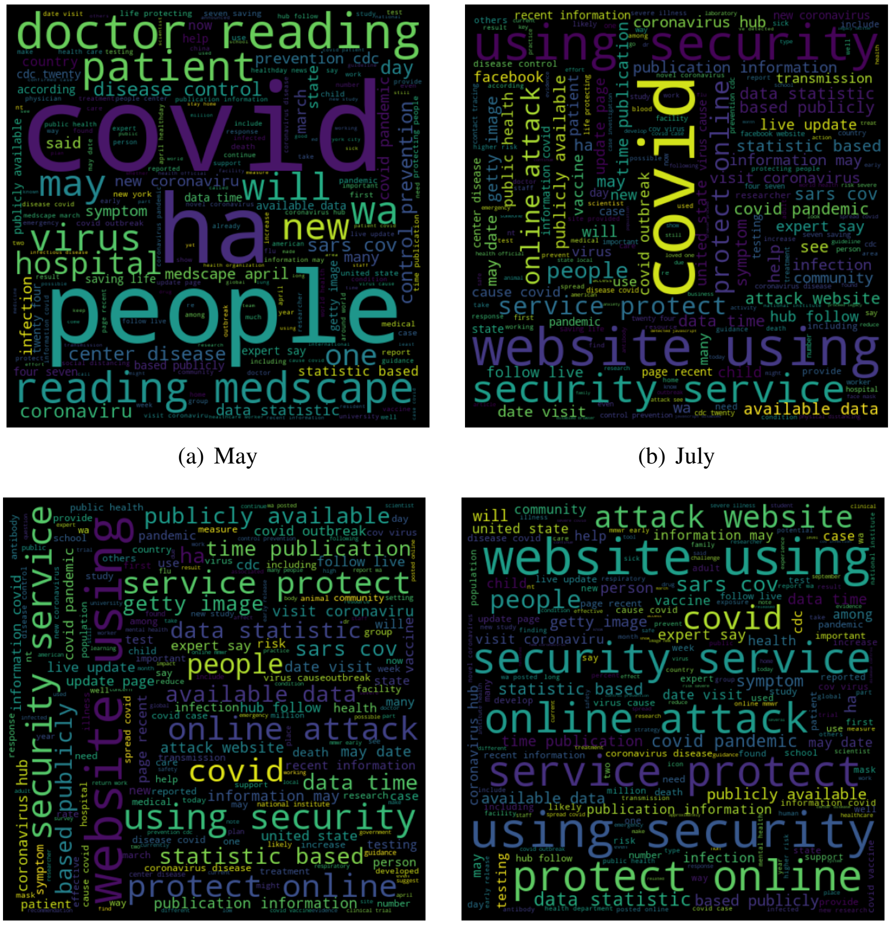
**CMU Sample Distribution.** The CMU-MisCov19 or CMU short-length text dataset misinformation categories are fine-grained comprising of 17 classes. It consists of 4,573 annotated tweets from 3,629 users with an average of 1.24 tweets per user. Figure 6 shows the **heavily skewed** distribution of 4,292 tweets for all categories that were extracted after some pre-processing. Class 7 (calling out or correction) has the most tweets, while class 2 (true treatment) has 0 tweets.



(a) May (b) July

(c) September (d) November

Figure 7: CoAID News Articles: Frequency of top 10 words.



(a) May (b) July

(c) September (d) November

Figure 8: CoAID News Articles: Word Clouds for the four datasets.

|  | Embedding Dimension | # Hidden Layers | Batch Size | Optimizer | Learning Rate |
|---|---|---|---|---|---|
| BERT | 768 | 12 | 32 | AdamW | 2e-5 |
| ELMo | 1024 | 5 | 128 | Adam | 0.001 |
| RoBERTa | 768 | 12 | 16 | AdamW | 2e-6 |
| ALBERT | 128 | 12 | 6 | Adam | 3e-5 |
| XLNet | 768 | 12 | 32 | AdamW | 2e-5 |
| Text GCN | 200 | 2 | N/A | Adam | 0.02 |
| VGCN BERT | N/A | 18 | 16 | Adam | 2e-5 |
| Word2Vec | 300 | 2 | 128 | Adam | 0.001 |
| GloVe | 300 | N/A | N/A | AdaGrad | N/A |

Table 5: Experimental setting.

### 6.0.1 CoAID News Articles: Study of Context Evolution

Figure 7 shows context evolution in the news articles category via the evolution of the distribution of the top ten high-frequency terms. We see, similar to the tweet dataset, context changes over time in the news articles datasets. For example, The May and July datasets have only 4 common high-frequency words: covid, coronavirus, health, data. In the July dataset, we observe the emergence of three new high-frequency words attack, security, and protect, which indicates a change in context. The context in the September dataset seems to be similar to that of the July dataset. These two datasets have eight high-frequency words in common: covid, service, online, attack, security, health, information, coronavirus. The November dataset shares seven common words with the September dataset: covid, service, online, attack, people, health, coronavirus. However, we notice an increase in the frequency in some words such as protect and attack. Also, a new word pandemic is seen to emerge. We gather similar observations about the context evolution from the word clouds in Figure 8.

## 7 Experimental Setting & Training Statistics

We provide the experimental setting for conducting our studies as well as the training statistics.

**Experimental Setting.** Table 5 shows the experimental setting for the studies. We used the default learning rate and batch size for all experiments.

**Training Statistics.** Table 6 shows the training statistics that include the number of parameters for each model, dataset, and the average training time. The inference time is not significant, thus not reported. All experiments were done on a Tesla V100 GPU, except the Text GCN and VGCN BERT based experiments, which were conducted using a CPU. For DSM and CNN based experiments, the CNN was trained for 5 epochs on the CoAID tweet data, 10 epochs on the CoAID news articles data, and 10 epochs on the CMU fine-grained data. The number of epochs was chosen based on the convergence behavior of the models.

| Model | #Parameters | Dataset | Avg. Training Time |
|---|---|---|---|
| **BERT PTM** | 110M | CoAID News Articles (large data) | 1.25 mins |
| **BERT PTM** | 110M | CoAID Tweets (large data) | 1.66 hours |
| **BERT PTM** | 110M | CoAID News Articles (small data) | 30 sec |
| **BERT PTM** | 110M | CoAID Tweets (small data) | 1.06 hours |
| **Word2Vec DSM + CNN** | Word2Vec: 6.7M<br>CNN: 160,301 | CoAID News Articles (large data) | 3.48 mins |
| **Word2Vec DSM + CNN** | Word2Vec: 155.9M<br>CNN: 160,301 | CoAID Tweets (large data) | 1.17 hours |
| **Word2Vec DSM + CNN** | Word2Vec: 5.1M<br>CNN: 160,301 | CoAID News Articles (small data) | 1.3 mins |
| **Word2Vec DSM + CNN** | Word2Vec: 7.8M<br>CNN: 160,301 | CoAID Tweets (small data) | 46 mins |
| **Word2Vec PTM + CNN** | 160,301 | CoAID News Articles (small data) | 2.05 mins |
| **Word2Vec PTM + CNN** | 160,301 | CoAID Tweets (small data) | 2.2 hours |
| **GloVe DSM + CNN** | GloVe: 4.6M<br>CNN: 160,301 | CoAID News Articles (small data) | 1.33 mins |
| **GloVe DSM + CNN** | GloVe: 100.6M<br>CNN: 160,301 | CoAID Tweets (small data) | 45.25 mins |
| **GloVe PTM + CNN** | 160,301 | CoAID News Articles (small data) | 2.08 mins |
| **GloVe PTM + CNN** | 160,301 | CoAID Tweets (small data) | 2.43 hours |
| **ELMo PTM + CNN** | 160,301 | CoAID News Articles (small data) | 8.23 mins |
| **ELMo PTM + CNN** | 160,301 | CoAID Tweets (small data) | 8.63 hours |
| **Text GCN** | N/A | CoAID Tweets (small data) | 8.8 mins |
| **VGCN BERT** | N/A | CoAID Tweets (small data) | 33 hours |
| **Text GCN** | N/A | CoAID News Articles (small data) | 7.32 |
| **VGCN BERT** | N/A | CoAID News Articles (small data) | 15.26 mins |
| **RoBERTa** | 125M | CoAID News Articles (small data) | 5.39 mins |
| **ALBERT** | 11M | CoAID News Articles (small data) | 6.8 hours |
| **XLNet** | 110M | CoAID News Articles (small data) | 35 sec |
| **BERT PTM** | 110M | CMU | 6 mins |
| **Word2Vec DSM + CNN** | Word2Vec: 8.7M<br>CNN: 160,301 | CMU | 3 mins |
| **Word2Vec PTM + CNN** | 160,301 | CMU | 30 sec |
| **GloVe DSM + CNN** | GloVe: 8.6M<br>CNN: 160,301 | CMU | 3.77 mins |
| **GloVe PTM + CNN** | 160,301 | CMU | 9.17 mins |
| **ELMo PTM + CNN** | 160,301 | CMU | 1.05 mins |

Table 6: Training Statistics - DSM: Domain-Specific Model, PTM: Pre-Trained Model