Relation Extraction with Few-shot Transfer Learning

Alexander Taylor
Department of Computer Science
UCLA
ataylor2@cs.ucla.edu

Brendon Ng
Department of Computer Science
UCLA
brendon.ng@cs.ucla.edu

Ayushi Agarwal
Department of Computer Science
UCLA
ayushi15@cs.ucla.edu

Minal Mishra
Department of Computer Science
UCLA
minamishra6@cs.ucla.edu

1 Introduction

Vast amounts of unstructured text data related to the COVID-19 pandemic are openly available on the internet in the form of social media posts, online knowledge bases, etc. Efficient methods for extracting structured information from this data would allow for more effective mobilization of resources for building strategies for mitigation of the pandemic. However, structuring and labeling documents in order to extract information is costly, time inefficient, and typically requires heavy manual annotation, such as the case in Riedel et al. [2010]. This necessitates approaches for dataset construction that are able to alleviate the manual annotation workload while attaining comparable quality to gold-standard datasets constructed entirely through manual annotation. Such approaches would enable the construction of COVID-19 data from a variety of unstructured sources, which would potentially serve as a foundation for developing better quality prediction models and lead to more valuable future predictions related to the spread and mitigation of COVID-19.

The task of relation extraction (RE) is of particular importance in downstream applications, for instance, knowledge graph construction from relation triples as done in Wang et al. [2021]. A difficulty inherent to the task of constructing a structured relation extract dataset from raw text data is the identification of existing and novel relations present in the data. To address this challenge, we formulate this problem as a distant supervision (DS) RE problem and leverage existing knowledge bases, such as Riedel et al. [2010], to learn the semantic and syntactic structures that define relationships in text and apply this to the aforementioned unstructured data sources.

Yet another challenge inherent to our task is the collection of high-quality examples from social media and news sources, which are known to have quality issues as stated in Salvatore et al. [2021]. To address this issue, we supplemented our DSRE approach with manual annotation to verify the quality of our constructed dataset to accept only high quality examples and to extract additional relations from the data. Given the need to be able to capture semantic structure with limited prior knowledge, we formulated our dataset as a few-shot dataset (using the structure in Gao et al. [2019]) and compared the performance of few-shot meta-learning methods on our new dataset against performance on the test set of FewRel 1.0, to verify its quality and difficulty.

In this paper we discuss the problem of dataset creation from unstructured, raw data and performing sentence-level RE from these constructed datasets in a few-shot setting. Our main contributions are as follows:

1. Building a pipeline for the construction of FewRel COVID-19 Twitter-News, a few-shot RE dataset constructed from the COVID-19 Twitter Data Stream and the Aylien Free Coronavirus News Dataset.

2. We apply meta-learning to the sentence-level RE task and formulate the problem as metric-learning-based few-shot learning.

3. We present an empirical evaluation of the performance of various existing approaches for few-shot meta-learning applied to sentence-level RE and compare these results against the models' performances on the test set of the benchmark dataset used for training, FewRel 1.0. Our results indicate that our dataset provides a high-quality, challenging benchmark for few-shot RE tasks on COVID-19 data.

2 Problem statement

The mobilization of efforts to mitigate the spread of COVID-19 has resulted in vast amounts of data being made available, much of which is unstructured and thus is time and effort intensive to make use of. The aim of this project is two-fold: we want to address the the issue of high-quality dataset creation and annotation from unstructured sources, and show that it is possible to leverage these datasets for relation extraction in a limited data setting.

The first aim is addressed by applying a Generative Adversarial Network-based (GAN) data filtering approach to the NYT Corpus Knowledge Base (NYT Corpus KB) from Riedel et al. [2010] to prune false positive labeled sentences, which improves the performance of the Piece-wise Convolutional Neural Network (PCNN) RE approach as shown in Qin et al. [2018] and our own ablation testing. After performing Named Entity Recognition (NER) on the COVID-19 Twitter and the Aylien Free Coronavirus News Dataset, our PCNN DSRE model was then used to extract relations R_i given entities e_j, e_k at the sentence level, resulting in triples $(e_j/e_k/R_i)$. We manually filtered these extracted triples $(e_j/e_k/R_i)$ and their respective sentences and structured them according to the few-shot format of FewRel 1.0 as given in Gao et al. [2019], yielding the new dataset: FewRel COVID-19 Twitter-News (FRCTN).

As we previously mentioned, few-shot learning can make full use of the limited training instances, which could potentially improve performance in RE. Therefore, we address the second aim by formulating RE on FRCTN as a few-shot meta-learning task in this work. We applied several popular benchmark few-shot meta-learning methods (GNN, Protonet, Siamese, SNAIL) to FRCTN (all methods were pre-trained in the RE task on the FewRel 1.0 dataset prior to their application to FRCTN). This yields a performance benchmark for future methods to compare against as well as an indication of the effectiveness of applying few-shot meta-learning to novel relation extraction.

3 Approach and implementation

3.1 Dataset and Data Preparation

3.1.1 Data Collection

As previously mentioned, our novel dataset FRCTN was constructed with data from the COVID-19 Twitter Data Stream and the Aylien Free Coronavirus News Dataset. The COVID-19 Twitter Data Stream was accessed through a Twitter-provided API api. Due to API-call limitations, we were only able to collect ∼48,000 tweets, which were additionally constrained by being limited to [] words. In order to supplement the data collected from the COVID-19 Twitter Data Stream, we extracted relations from ∼1,000 documents from the Aylien Free Coronavirus News Dataset.

3.1.2 Data Cleaning

To clean the data collected from Twitter, we removed all special characters such as emojis, hyperlinks, and user mentions (ex. @Tom_Cruise). To formulate our data as a sentence-level RE problem, we converted each corpus (both documents and tweets) into individual sentences, removed all punctuation, and stripped the sentences of excess whitespace.

3.1.3 Named Entity Recognition

We made use of two python modules from Honnibal and Montani [2017] to perform NER on each sentence in our cleaned data to provide the arguments necessary for the RE task. The first module focused on extracting entities similar to those in the NYT Corpus KB, which were from the following categories: person, location, organization, geo-political entity. The second module focused on entities from the biomedical domain to improve the performance of our DRE approach on tweets and news snippets referring to COVID-19 directly or attributes of the virus. An example of NER focusing on entities from the biomedical domain is given in Figure 1.

1.

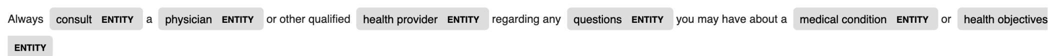


Figure 1: Named Entitiy Recognition using modules from Honnibal and Montani [2017]

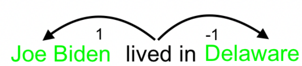
3.1.4 DSGAN

As stated above, distant supervision approaches require leveraging knowledge bases as sources of 'ground truths' to train the system that can then be applied to subsequent datasets. From Qin et al. [2018] and Zeng et al. [2015], we know that directly training on examples from knowledge bases such as the NYT Corpus is weakened by the noise present in the data.

Qin et al. [2018] developed a GAN-based approach to address the issue of false positive results and saw statistically significant performance improvements in DSRE tasks on the NYT Corpus KB. For each sentence in a given corpus, DSGAN generates the likelihood of belonging to the class of positive labels and of belonging to the class of negative labels. Through the adversarial training process, this approach effectively learns to distinguish true and false positive labels, and is able to prune the training corpus and improves the accuracy of PCNN approaches as shown in Qin et al. [2018] (See Appendix: Table 1).

3.1.5 PCNN

Zeng et al. [2015] uses the Skip-gram model from Mikolov et al. [2013] to train its word embeddings into a vector of dimension $d_w \in \mathbb{R}$, and uses the method defined by Zeng et al. [2014] to encode position features (PF). PF is defined as the relative distance between the given word$_i$ and the two entities, entity$_1$ and entity$_2$.



In the example above, the relative distances from *lived* to entity$_1$ (**Joe Biden**) is 1 and entity$_2$ (**Delaware**) is -1. Zeng et al. [2014] randomly defines two PF embedding matrices and defines the two PF representations as the lookup of each of these relative differences, which is a vector v of dimension $d_p \in \mathbb{R}$. Thus, our word embeddings are of the form

$$v = [W, PF_1, PF_2]$$

To construct a matrix representation of a given sentence, we append the position embedding vectors to the word embedding vectors, which yields our sentence embedding $\mathbf{S} \in \mathbb{R}^{s \times (d_w + 2 \cdot d_p)}$, where s is the length of the sentence.

We then use convolutional layers with n filters $W = \{w_1, ..., w_n\}$ to distill features (as shown in Albawi et al. [2017]) into a matrix $C \in \mathbb{R}^{n \times (s+w-1)}$, where w is the size of each filter and s is the number of tokens in the sentence.

However, where this approach differs from traditional convolution is its use of max pooling; single max pooling convolution is too coarse for relation extraction as it reduces the size of the hidden layers quickly as explained in Zeng et al. [2015]. Thus, the authors of Zeng et al. [2015] introduce a piece-wise max pooling procedure that involves dividing the output of each hidden layers into 3 segments, the divisions of with depending on the placement of the two entities, and performing max pooling over each segment. The resulting vectors are then concatenated and have a non-linear function applied to them.

3.1.6 Manual labeling

To guarantee that the relation triples and corresponding sentences extracted from the NYT Corpus KB and Aylien Free Coronavirus News Dataset by the DSRE PCNN+ATT approach was correct and of high quality, we filtered the results manually. DSRE approaches are highly biased to the distribution of relations from the knowledge base used for training as shown in Zeng et al. [2015], which prevents them from identifying novel relations in data they are applied to. To align with the typical framework for a few-shot meta-learning task, we chose 5 out of the 53 relations extracted (as shown below) to retain from the application of DSRE PCNN+ATT.

1. "location/location/contains"
2. "person/business/works_for"
3. "person/title/has_title"
4. "people/location/lives_in"
5. "person/person/relation"

In addition, we also introduced 2 novel relations as shown below:

1. "people/virus/infected"
2. "person/symptom/has_symptom"

We labeled these relations manually because we did not have pre-trained model to label these relations.

An example with the relations count of instances in each relation and example of each is given in table 1.

Relation	Example	Number of instances
location/location/contains	newest round of covid19 flareup in **china** stringent coronavirus lockdown in **dalian**	59
business/person/company	When there is no business activity paying these interests is also a big challenge said **Charanjiv Singh** chairman of **the Chandigarh Beopar Mandal**"	58
person/title/has_title	According to **Cancer Council chief executive Sanchia Aranda** non urgent surgeries have already been delayed across the country and hospitals have made changes to the ways treatments are being delivered	48
people/location/lives_in	newest Save the Children **UK** chief executive **Kevin Watkins** said if we act now and act decisively we can prevent and contain the pandemic threat facing the poorest countries	51
person/person/relation	**Katharine McPhee** and her husband **David Foster** are putting on concerts every evening from their home which can be seen on Instagram	40
people/virus/infected	The NSW government on Sunday urged **young people** to take the **COVID19** pandemic seriously revealing more than a quarter of the states current coronavirus cases are in people aged under 29	55
person/symptom/has_symptom	I did not have many symptoms while other positive **patients** had **cough and fever**	59

Table 1: Relations, their examples and instances in each relation

3.1.7 FewRel 1.0 Format

As previously mentioned, FewRel 1.0 is a common benchmark dataset for RE tasks based on the Wikipedia corpus and the Wikidata Knowledge Base. We structured our data according to the format of FewRel 1.0 to allow direct application of few-shot meta-learning approaches pre-trained on FewRel 1.0 to FRCTN.

3.2 Few-Shot Meta-Learning

Given that RE tasks on raw, unannotated data are often in low-data settings, we formulated our work as a few-shot meta-learning problem. FewRel 1.0 is a meta-learning dataset that can be used for various tasks, including RE, which makes it a popular benchmark for few-shot meta-learning approaches. Thus, as previously mentioned, we pre-train the following models on the FewRel 1.0 dataset and apply the models to the RE task on FRCTN: GNN, Protonet, Siamese, and SNAIL.

We selected these models because all four are diverse, popular benchmarks that are commonly used to measure the performance of novel few-shot meta-learning models, and are thus openly available for use. In the following sections, we describe each model and motivate its selection.

3.2.1 GNN

The Graph Neural Network (GNN) few-shot model from Garcia and Bruna [2017] is frames the problem as a supervised message-passing task trained end-to-end on graph convolutional neural networks. The input signal to the model $F \in R^{V \times d}$ where each $F_i = \sum_{j \sim i} w_{i,j} F_j$ where $i \sim j$ implies $(i, j) \in E$ and each $w_{i,j}$ is a learned parameter. This input signal is then used to weight each node representation to produce the node representation of the next layer.

$$\mathbf{x}_l^{(k+1)} = Leaky\_ReLU(\sum_{F_i \in F} F_i \mathbf{x}_l^{(k)} \theta_{F_i,l}^{(k)})$$

The authors also utilize a multilayer perceptron function to learn edge features.

$$\Phi(x_i^{(k)}, x_j^{(k)}) = \mathbf{MLP}(abs(x_i^{(k)}, x_j^{(k)})$$

A softmax function is then applied to each of the node representations in the current layer's adjacency measure and used to update the residual. The general framework of the approach is shown below in Figure 2.

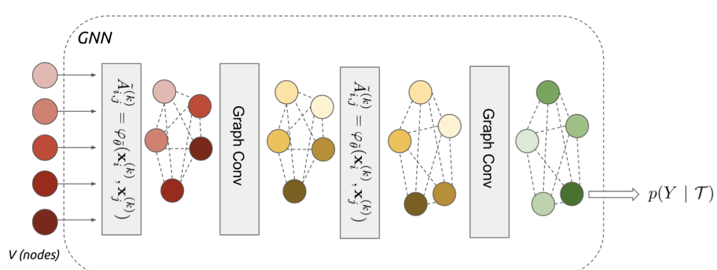


Figure 2: Illustration of the GNN few-shot meta-learning approach taken from Garcia and Bruna [2017]

The contribution of this approach is its ability to perform few-shot meta-learning tasks with fewer parameters than the other popular benchmark methods, Siamese and Protonet.

In this work, GNN was trained using 20000 iterations on the FewRel dataset on two different settings: 5-way 1-shot and 5-way 5-shot few-shot learning with a batch size of 4. The results were tested on FRCTN in 5-way 1-shot, 5-way 5-shot and 5-way 10-shot settings.

3.2.2 Protonet

Protonet, an implementation of Prototypical Networks, falls under the class of metrics-based meta-learning methods, meaning that each model learns the metric space of the training examples and creates a representative data-point, or prototype, for each class, and new examples are assigned to the class of the nearest prototype based on distance, similar to K-Nearest Neighbors (KNN) as stated in Snell et al. [2017]. The Protonet implementation we make use of in this work depends on the euclidean distance measure.

We included this approach on our analysis because it is a very common benchmark for novel few-shot meta-learning approaches applied to the FewRel 1.0 dataset.

In this work, Protonet was trained using 20000 iterations on the FewRel dataset under two different settings: 5-way 1-shot and 5-way 5-shot few-shot learning with a batch size of 4. pre-trained model was then tested on FRCTN in 5-way 1-shot, 5-way 5-shot and 5-way 10-shot settings.

3.2.3 Siamese

Similar to Protonet, Siamese networks are a type of Prototypical Networks. However, instead of learning a metric space based on a pre-defined distance heuristic, the 'distance' between two given data points is computed by an 'energy' function. Siamese networks consist of two identical convolutional neural networks conjoined in their output layer at a single node which contains the aforementioned energy function. The figure below outlines the best architecture found in Koch et al. [2015].

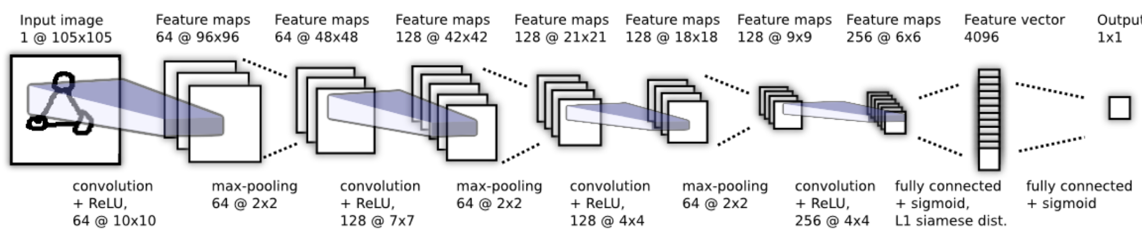


Figure 3: Best convolutional architecture selected for few-shot task. Only one twin is depicted, but joins with the other twin immediately after the 4096 unit fully-connected layer where the L1 component-wise distance between vectors is computed. Taken from Koch et al. [2015]

Similar to Protonet, we included this approach on our analysis because it is a very common benchmark for novel few-shot meta-learning approaches applied to the FewRel 1.0 dataset.

In this work, Siamese was trained using 20000 iterations on the FewRel dataset in the 5-way 1-shot and 5-way 5-shot few-shot learning settings with a batch size of 4. The pre-trained model was then tested on FRCTN in 5-way 1-shot, 5-way 5-shot and 5-way 10-shot settings.

3.2.4 SNAIL

Simple Neural Attentive Learner (SNAIL) was proposed in Mishra et al. [2017] as a simple neural network-based meta-learning approach using 1-D convolutional layers over the temporal dimension in concert with soft attention layers, which allows the model to attend to important features from past and current iterations. Shown below is an example of how the temporal convolutional layers and soft attention layers are structured.

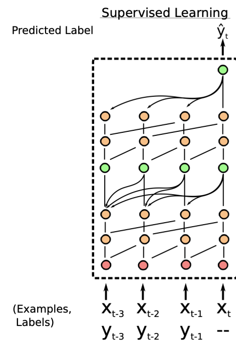


Figure 4: Overview of SNAIL. in this example,two TC layers (in orange) are interleaved with two soft attention layers (in green). Taken from Mishra et al. [2017]

We included SNAIL in our work because it outperforms the baselines on the Omniglot and MiniImageNet few-shot meta-learning datasets as shown in Mishra et al. [2017].

In our work, SNAIL was trained using 20000 iterations on the FewRel dataset in the 5-way 1-shot and 5-way 5-shot few-shot learning settings with a batch size of 4. The pre-trained model was then tested on FRCTN under 5-way 1-shot and 5-way 5-shot settings. We were unable to test SNAIL under the 5-way 10-shot setting because the size of SNAIL's temporal convolutional layers is determined by the training settings, and we were unable to train SNAIL under the 5-way 10-shot setting due to the size of FRCTN.

4 Evaluation

4.1 DSGAN Pre-filtering and PCNN

We applied DSGAN to our work on both the NYT Corpus KB and the manually labeled Twitter data and were able to replicate the findings in Qin et al. [2018] that showed a statistically significant difference (according to McNemar's Test) in RE performance in the Piecewise-Convolutional Neural Network with Attention (PCNN+ATT) on the RE task.

Model	AUC	**+DSGAN**	p-value
PCNN+ATT	0.269	**0.294**	1.193e-178

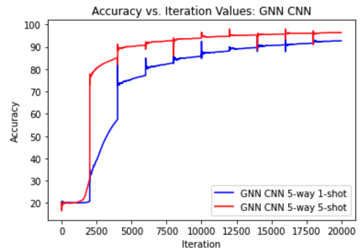
Table 2: Performance of PCNN+ATT without DSGAN pre-filtering and PCNN+ATT with DSGAN pre-filtering on the NYT Corpus KB using AUC

We were able to remove 16,000 false positive sentences from the NYT Corpus KB, as well as add back 3,000 false negative sentences prior to training our DSRE approach on the filtered NYT Corpus KB. We also applied post-filtering of the data produced the DSRE approach after manual filtering, which removed 0 false positive labels. This indicates that we provided high quality annotations for FRCTN.

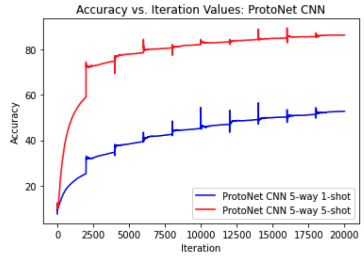
4.2 Few-Shot Meta-Learning Approaches

As previously mentioned, 4 baseline methods, GNN, Protonet, SNAIL, and Siamese, were all trained on the FewRel 1.0 benchmark dataset. Figure 5 shows the accuracy of training with respect to iterations of each model during training on the FewRel 1.0 dataset. As shown in each graph, we see consistent performance across learning settings for the Siamese networks. SNAIL and GNN experience similar accuracy dropoffs between the 5-way 5-shot task and the 5-way 1-shot task. Protonet shows the largest dropoff in performance across the two tasks.

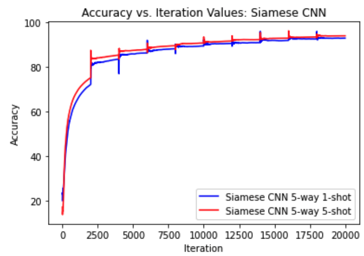
The aforementioned models were then tested on our novel dataset, FRCTN, to evaluate the quality and difficulty of our data, and thus the efficacy of our pipeline. The testing on FRCTN was done under



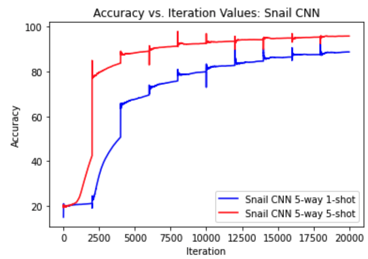
(a)



(b)



(c)



(d)

Figure 5: The accuracy versus iteration curves for GNN, Protonet, Siamese, SNAIL trained on FewRel dataset

three learning settings: 5-way 1-shot, 5-way 5-shot, and 5-way 10-shot. All testing was performed using 100 iterations. The metrics F1 score, Precision, Recall, and Support were calculated for all relations (See Appendix Sections SNAIL models, Protonet Models, Siamese Models, GNN Models). The final F1 scores for each model's performance on FRCTN are given in Table 3. Since, as stated in Section 3.2.4, the SNAIL model can only be tested in the conditions it was trained under, we are only able to show two results for the SNAIL model for the two training settings.

Models	5 way 1 shot			5 way 5 shot		
	5 way 1 shot	5 way 5 shot	5 way 10 shot	5 way 1 shot	5 way 5 shot	5 way 10 shot
Protonet	0.52	0.66	0.71	**0.55**	**0.72**	**0.78**
SNAIL	0.46	-	-	-	0.62	-
Siamese	**0.53**	**0.68**	**0.73**	0.53	0.69	0.75
GNN	0.48	0.61	0.65	0.50	0.70	0.72

Table 3: Performance (F1 scores) of various models on 5 way 1 shot training and 5 way 5 shot training

We see in the table above that the F1 scores of Protonet and Siamese are fairly similar under similar settings, and significantly outperform the other two benchmarks. We see decent model performance across all benchmarks, however, which indicates that existing few-shot meta-learning frameworks are able to identify relations effectively, indicating the quality and difficulty of our data.

We include results of different models when tested on FewRel 1.0 test dataset given in Table 4 to give context to the relative performances of GNN, Protonet, Siamese, and SNAIL. These results are taken from Han et al. [2018]. Through these results, we can see that the accuracy (which is directly comparable to F1 score) is much higher for the FewRel dataset than the FRCTN dataset, which implies that our dataset is more difficult to learn from, thus leaving room for future improvements.

5 Discussion and future work

In this paper, we discussed our novel pipeline for constructing a RE dataset framed as a few-shot meta-learning problem. We found that our application of DSGAN to PCNN+ATT added statistically

Models	5 way 1 shot	5 way 5 shot	10 way 1 shot	10 way 5 shot
Protonet	74.01	**89.46**	61.30	**81.66**
SNAIL	72.69	84.22	58.15	68.36
Siamese	**75.76**	85.80	**64.58**	77.42
GNN	71.18	85.71	56.01	74.33

Table 4: Performance (accuracy) of various models on different settings of few shot learning on FewRel 1.0 test dataset. The values are taken from Lin et al. [2021].

significant improvements over vanilla PCNN+ATT. After manual filtering of the relations extracted from COVID-19 Twitter Data Stream and the Aylien Free Coronavirus News Dataset, we structured the data into the format of FewRel 1.0 to create our novel dataset, FRCTN. We then pre-trained four benchmark models (GNN, Protonet, Siamese, and SNAIL) on the benchmark FewRel 1.0 dataset and applied them to FRCTN. We found that Siamese outperformed the other models in the 5-way 1-shot RE task and that Protonet outperformed the other models in the 5-way 5-shot RE task, with comparable performances from both models. When compared with model performances on the FewRel 1.0 Dataset, we found that the results across both datasets were consistent in relative model performance, with FRCTN being more difficult. This leaves room for future models to improve upon the baseline set for the RE task under the aforementioned learning settings on FRCTN. Future improvements we would like to make to the pipeline used to create FRCTN are as follows:

1. We would like to incorporate data filtering approaches tailored to Twitter data as shown in Hettiarachchi and Ranasinghe [2020] andSharma et al. [2020]
2. We would also like to increase the size of FRCTN to increase the number of few-shot learning settings it can support
3. Lastly, we would like to automate the incorporation of more complete data annotation to allow it to support more IE tasks

Future work in this direction would lead to improvements in dataset creation from raw, unstructured data, which would lead to more full utilization of the rich data widely available in such forms.

6 Roles of team members

Alexander: Wrote Project Paper; Performed background research for PCNN, DSGAN, Distant Supervision for Relation extraction, GNN-based FS meta-learning approach, SNAIL, Protonet, and Siamese Networks. Implemented pre-training of DSGAN as well as the meta-learning networks: Siamese, SNAIL, and GNN. Performed Mcnemar's test on DSGAN+PCNN and Vanilla PCNN results with help from Brendon. Led project meetings and discussions.

Ayushi: Wrote Project Paper; The role was to research different sources of data collection. Then collected data from two different sources applied NER using different libraries to get entities that can be leveraged for covid data. Cleaning the dataset as per required by visualizing the results of NER. Extracting relations using PCNN. Research FewRel dataset so that it can be determined how it can be leveraged to train our meta learning models. Transformed data to FewRel format and did analysis on different few shot meta learning models. Trained Protonet. Tested the created dataset on all the models and collected evaluation metrics. Led project meetings and discussions.

Brendon: Reformatting data to match required format of model to train with, create accuracy visualizations, testing models, sections of the in-class project presentation, and running the ablation tests for evaluation. Wrote preprocessing scripts to prepare data for various model runs, trained PCNN + attention model, and helped with PCNN comparison between Vanilla PCNN and DSGAN filtered PCNN.

Minal: Researched about the few-shot FewRel dataset to utilize it in the project. Researched about the twitter api's for dataset preparation. Performed manual labelling of the dataset for 7 different relations containing 50 instances each to train and test the datasets on the few-shot learning model. Contributed to the project presentation and worked on the Introduction and Problem Statement of the Project Paper.

References

Twitter api. URL `https://documenter.getpostman.com/view/9956214/T1LMiT5U#acd41db7-05bc-4359-be63-dd5249dfef32`.

Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6, 2017. doi: 10.1109/ICEngTechnol.2017.8308186.

Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. Fewrel 2.0: Towards more challenging few-shot relation classification. *arXiv preprint arXiv:1910.07124*, 2019.

Victor Garcia and Joan Bruna. Few-shot learning with graph neural networks. *arXiv preprint arXiv:1711.04043*, 2017.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. *arXiv preprint arXiv:1810.10147*, 2018.

Hansi Hettiarachchi and Tharindu Ranasinghe. Infominer at WNUT-2020 task 2: Transformer-based covid-19 informative tweet extraction. *CoRR*, abs/2010.05327, 2020. URL `https://arxiv.org/abs/2010.05327`.

Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.

Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015.

Qing Lin, Yongbin Liu, Wen Wen, and Zhihua Tao. Ensemble making few-shot learning stronger. *arXiv preprint arXiv:2105.11904*, 2021.

Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *ICLR*, 2013.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. *arXiv preprint arXiv:1707.03141*, 2017.

Pengda Qin, Weiran Xu, and William Yang Wang. DSGAN: Generative adversarial training for distant supervision relation extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 496–505, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1046. URL `https://aclanthology.org/P18-1046`.

Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In *ECML/PKDD*, 2010.

Camilla Salvatore, Silvia Biffignandi, and Annamaria Bianchi. Social media and twitter data quality for new social indicators. *Social Indicators Research*, 156(2):601–630, 2021. doi: 10.1007/s11205-020-02296-w. URL `https://doi.org/10.1007/s11205-020-02296-w`.

Karishma Sharma, Sungyong Seo, Chuizheng Meng, Sirisha Rambhatla, and Yan Liu. Covid-19 on social media: Analyzing misinformation in twitter conversations, 2020.

Jake Snell, Kevin Swersky, and Richard S Zemel. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*, 2017.

Wenguang Wang, Yonglin Xu, Chunhui Du, Yunwen Chen, Yijie Wang, and Hui Wen. Data Set and Evaluation of Automated Construction of Financial Knowledge Graph. *Data Intelligence*, 3(3): 418–443, 09 2021. ISSN 2641-435X. doi: 10.1162/dint_a_00108. URL `https://doi.org/10.1162/dint_a_00108`.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics. URL `https://aclanthology.org/C14-1220`.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1753–1762, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1203. URL `https://aclanthology.org/D15-1203`.

7 Appendix

Model	AUC	**DSGAN**	p-value
CNN+ONE	0.177	**0.189**	4.37e-04
CNN+ATT	0.219	**0.226**	8.36e-03
PCNN+ONE	0.206	**0.221**	2.89e-06
PCNN+ATT	0.253	**0.264**	2.34e-03

Table 5: Performance of PCNN and CNN methods without on NYT corpus using AUC compared with PCNN and CNN methods with DSGAN pre-filtering. The values are taken from the original paper

7.1 SNAIL Models

7.1.1 SNAIL trained in a 5-way 1-shot setting

Relation	Precision	Recall	F1-score	Support
noun/symptom/covid	0.34	0.33	0.34	320
location/location/contains	0.65	0.61	0.63	365
business/person/company	0.35	0.29	0.31	375
people/people/title	0.44	0.39	0.41	365
people/location/lives_in	0.41	0.39	0.40	400
people/virus/affected	0.38	0.43	0.41	325
person/person/relation	0.60	0.80	0.68	350

Table 6: 5-way 1-shot training setting SNAIL Performance on the generated twitter-news dataset in 5-way 1-shot classification setting

7.1.2 SNAIL trained in a 5-way 5-shot setting

Relation	Precision	Recall	F1-score	Support
noun/symptom/covid	0.53	0.53	0.53	350
location/location/contains	0.68	0.76	0.72	380
business/person/company	0.57	0.43	0.49	345
people/people/title	0.64	0.67	0.65	335
people/location/lives_in	0.61	0.46	0.53	390
people/virus/affected	0.50	0.59	0.54	345
person/person/relation	0.77	0.89	0.83	355

Table 7: 5-way 1-shot training setting SNAIL performance on the generated twitter-news dataset in 5-way 5-shot classification setting

7.2 Protonet Models

7.2.1 Protonet Models trained in a 5-way 1-shot setting

Relation	Precision	Recall	F1-score	Support
noun/symptom/covid	0.44	0.44	0.44	370
location/location/contains	0.60	0.67	0.64	370
business/person/company	0.50	0.35	0.41	325
people/people/title	0.52	0.38	0.44	340
people/location/lives_in	0.48	0.42	0.45	375
people/virus/affected	0.42	0.57	0.48	355
person/person/relation	0.71	0.82	0.76	365

Table 8: 5-way 1-shot training setting Protonet performance on the generated twitter-news dataset in 5-way 1-shot classification setting

Relation	Precision	Recall	F1-score	Support
noun/symptom/covid	0.62	0.61	0.61	355
location/location/contains	0.73	0.79	0.76	380
business/person/company	0.62	0.57	0.59	375
people/people/title	0.74	0.63	0.68	360
people/location/lives_in	0.57	0.56	0.57	320
people/virus/affected	0.56	0.57	0.57	320
person/person/relation	0.77	0.90	0.83	325

Table 9: 5-way 1-shot training setting Protonet performance on the generated twitter-news dataset in 5-way 5-shot classification setting

Relation	Precision	Recall	F1-score	Support
noun/symptom/covid	0.70	0.64	0.67	370
location/location/contains	0.76	0.86	0.81	380
business/person/company	0.68	0.63	0.65	310
people/people/title	0.75	0.68	0.71	370
people/location/lives_in	0.61	0.63	0.62	345
people/virus/affected	0.60	0.59	0.60	355
person/person/relation	0.83	0.91	0.86	370

Table 10: 5-way 1-shot training setting Protonet performance on the generated twitter-news dataset in 5-way 10-shot classification setting

7.2.2 Protonet Models trained in a 5-way 5-shot setting

Relation	Precision	Recall	F1-score	Support
noun/symptom/covid	0.53	0.49	0.51	355
location/location/contains	0.60	0.71	0.65	315
business/person/company	0.49	0.38	0.43	365
people/people/title	0.60	0.47	0.52	395
people/location/lives_in	0.45	0.43	0.44	320
people/virus/affected	0.47	0.56	0.52	370
person/person/relation	0.66	0.78	0.72	380

Table 11: 5-way 5-shot training setting Protonet performance on the generated twitter-news dataset in 5-way 1-shot classification setting

Relation	Precision	Recall	F1-score	Support
noun/symptom/covid	0.71	0.68	0.69	335
location/location/contains	0.81	0.87	0.84	350
business/person/company	0.64	0.62	0.63	405
people/people/title	0.77	0.66	0.72	385
people/location/lives_in	0.63	0.66	0.64	355
people/virus/affected	0.67	0.70	0.69	325
person/person/relation	0.84	0.90	0.87	345

Table 12: 5-way 5-shot training setting Protonet performance on the generated twitter-news dataset in 5-way 5-shot classification setting

Relation	Precision	Recall	F1-score	Support
noun/symptom/covid	0.71	0.74	0.73	305
location/location/contains	0.87	0.88	0.87	385
business/person/company	0.71	0.71	0.71	350
people/people/title	0.88	0.74	0.80	375
people/location/lives_in	0.66	0.70	0.68	315
people/virus/affected	0.72	0.71	0.72	375
person/person/relation	0.85	0.91	0.88	395

Table 13: 5-way 5-shot training setting Protonet performance on the generated twitter-news dataset in 5-way 10-shot classification setting

7.3 Siamese Models

7.3.1 Siamese trained in a 5-way 1-shot setting

Relation	Precision	Recall	F1-score	Support
noun/symptom/covid	0.53	0.46	0.53	345
location/location/contains	0.62	0.76	0.68	365
business/person/company	0.47	0.37	0.41	395
people/people/title	0.54	0.54	0.54	345
people/location/lives_in	0.45	0.29	0.35	340
people/virus/affected	0.45	0.48	0.46	370
person/person/relation	0.61	0.74	0.67	340

Table 14: Siamese Performance on the generated twitter-news dataset in 5-way 1-shot classification setting

Relation	Precision	Recall	F1-score	Support
noun/symptom/covid	0.61	0.67	0.64	365
location/location/contains	0.78	0.84	0.81	355
business/person/company	0.69	0.55	0.61	375
people/people/title	0.67	0.73	0.70	335
people/location/lives_in	0.63	0.47	0.54	375
people/virus/affected	0.59	0.65	0.62	330
person/person/relation	0.77	0.86	0.82	365

Table 15: Siamese Performance on the generated twitter-news dataset in 5-way 5-shot classification setting

Relation	Precision	Recall	F1-score	Support
noun/symptom/covid	0.63	0.74	0.71	355
location/location/contains	0.79	0.86	0.82	320
business/person/company	0.69	0.61	0.64	325
people/people/title	0.73	0.78	0.76	380
people/location/lives_in	0.71	0.59	0.64	405
people/virus/affected	0.69	0.69	0.69	355
person/person/relation	0.80	0.87	0.84	360

Table 16: Siamese Performance on the generated twitter-news dataset in 5-way 10-shot classification setting

7.3.2 Siamese trained in a 5-way 5-shot setting

Relation	Precision	Recall	F1-score	Support
noun/symptom/covid	0.49	0.59	0.54	350
location/location/contains	0.64	0.65	0.65	365
business/person/company	0.46	0.42	0.44	375
people/people/title	0.48	0.52	0.50	405
people/location/lives_in	0.49	0.39	0.43	355
people/virus/affected	0.46	0.46	0.46	310
person/person/relation	0.71	0.71	0.71	340

Table 17: Siamese Performance on the generated twitter-news dataset in 5-way 1-shot classification setting

Relation	Precision	Recall	F1-score	Support
noun/symptom/covid	0.66	0.72	0.69	370
location/location/contains	0.80	0.73	0.76	375
business/person/company	0.66	0.61	0.63	375
people/people/title	0.67	0.78	0.72	335
people/location/lives_in	0.67	0.53	0.59	365
people/virus/affected	0.61	0.68	0.64	355
person/person/relation	0.78	0.83	0.80	325

Table 18: Siamese Performance on the generated twitter-news dataset in 5-way 5-shot classification setting

Relation	Precision	Recall	F1-score	Support
noun/symptom/covid	0.68	0.76	0.72	385
location/location/contains	0.91	0.88	0.90	405
business/person/company	0.79	0.65	0.71	340
people/people/title	0.69	0.79	0.74	350
people/location/lives_in	0.74	0.62	0.67	345
people/virus/affected	0.64	0.66	0.65	330
person/person/relation	0.81	0.85	0.84	345

Table 19: Siamese Performance on the generated twitter-news dataset in 5-way 10-shot classification setting

7.4 GNN Models

7.4.1 GNN trained in a 5-way 1-shot setting

Relation	Precision	Recall	F1-score	Support
noun/symptom/covid	0.37	0.50	0.43	340
location/location/contains	0.59	0.58	0.58	330
business/person/company	0.43	0.34	0.38	395
people/people/title	0.48	0.46	0.47	370
people/location/lives_in	0.39	0.30	0.34	365
people/virus/affected	0.40	0.43	0.41	355
person/person/relation	0.68	0.76	0.72	345

Table 20: GNN Performance on the generated twitter-news dataset in 5-way 1-shot classification setting

Relation	Precision	Recall	F1-score	Support
noun/symptom/covid	0.44	0.66	0.53	345
location/location/contains	0.65	0.71	0.68	390
business/person/company	0.68	0.44	0.53	340
people/people/title	0.73	0.58	0.65	330
people/location/lives_in	0.56	0.41	0.48	345
people/virus/affected	0.49	0.50	0.49	390
person/person/relation	0.79	0.92	0.85	360

Table 21: GNN Performance on the generated twitter-news dataset in 5-way 5-shot classification setting

Relation	Precision	Recall	F1-score	Support
noun/symptom/covid	0.50	0.67	0.57	360
location/location/contains	0.69	0.83	0.75	350
business/person/company	0.74	0.42	0.54	335
people/people/title	0.72	0.61	0.66	340
people/location/lives_in	0.69	0.48	0.57	355
people/virus/affected	0.51	0.59	0.55	375
person/person/relation	0.81	0.92	0.86	385

Table 22: GNN Performance on the generated twitter-news dataset in 5-way 10-shot classification setting

7.4.2 GNN trained in a 5-way 5-shot setting

Relation	Precision	Recall	F1-score	Support
noun/symptom/covid	0.41	0.49	0.44	330
location/location/contains	0.62	0.61	0.61	380
business/person/company	0.48	0.40	0.44	340
people/people/title	0.52	0.51	0.52	385
people/location/lives_in	0.42	0.38	0.40	365
people/virus/affected	0.44	0.45	0.45	350
person/person/relation	0.68	0.71	0.70	350

Table 23: GNN Performance on the generated twitter-news dataset in 5-way 1-shot classification setting

Relation	Precision	Recall	F1-score	Support
noun/symptom/covid	0.57	0.69	0.62	325
location/location/contains	0.80	0.79	0.79	380
business/person/company	0.67	0.60	0.63	345
people/people/title	0.75	0.73	0.74	375
people/location/lives_in	0.61	0.56	0.58	360
people/virus/affected	0.62	0.62	0.62	365
person/person/relation	0.84	0.88	0.86	350

Table 24: GNN Performance on the generated twitter-news dataset in 5-way 5-shot classification setting

Relation	Precision	Recall	F1-score	Support
noun/symptom/covid	0.59	0.75	0.66	345
location/location/contains	0.79	0.82	0.80	370
business/person/company	0.74	0.61	0.67	395
people/people/title	0.74	0.73	0.74	295
people/location/lives_in	0.67	0.63	0.65	375
people/virus/affected	0.66	0.59	0.62	350
person/person/relation	0.84	0.90	0.87	370

Table 25: GNN Performance on the generated twitter-news dataset in 5-way 10-shot classification setting