Few-Shot Learning for Clinical Natural Language Processing Using Siamese Neural Networks

David Oniani
University of Pittsburgh
Pittsburgh, PA, United States
davidoniani@pitt.edu

Sonish Sivarajkumar
University of Pittsburgh
Pittsburgh, PA, United States
sos86@pitt.edu

Yanshan Wang
University of Pittsburgh
Pittsburgh, PA, United States
yanshan.wang@pitt.edu

***Abstract*—Clinical Natural Language Processing (NLP) has become an emerging technology in healthcare that leverages a large amount of free-text data in electronic health records (EHRs) to improve patient care, support clinical decisions, and facilitate clinical and translational science research. Recently, deep learning has achieved state-of-the-art performance in many clinical NLP tasks. However, training deep learning models usually require large annotated datasets, which are normally not publicly available and can be time-consuming to build in clinical domains. Working with smaller annotated datasets is typical in clinical NLP and therefore, ensuring that deep learning models perform well is crucial for the models to be used in real-world applications. A widely adopted approach is fine-tuning existing Pre-trained Language Models (PLMs), but these attempts fall short when the training dataset contains only a few annotated samples. Few-Shot Learning (FSL) has recently been investigated to tackle this problem. Siamese Neural Network (SNN) has been widely utilized as an FSL approach in computer vision, but has not been studied well in NLP. Furthermore, the literature on its applications in clinical domains is scarce. In this paper, we propose two SNN-based FSL approaches for clinical NLP, including pre-trained SNN (PT-SNN) and SNN with second-order embeddings (SOE-SNN). We evaluated the proposed approaches on two clinical tasks, namely clinical text classification and clinical named entity recognition. We tested three few-shot settings including 4-shot, 8-shot, and 16-shot learning. Both clinical NLP tasks were benchmarked using three PLMs, including Bidirectional Encoder Representations from Transformers (BERT), Bidirectional Encoder Representations from Transformers for Biomedical Text Mining (BioBERT), and Bio + Clinical BERT (BioClinicalBERT). The experimental results verified the effectiveness of the proposed SNN-based FSL approaches in both clinical NLP tasks.**

***Index Terms*—natural language processing, NLP, Clinical NLP, Few-Shot Learning, Siamese Neural Networks, Deep Learning, Machine Learning, Computational Modeling, Algorithms**

I. Introduction

Deep Neural Networks (DNNs), due to their promising performance [1], currently dominate both Computer Vision (CV) and Natural Language Processing (NLP) literature. However, fully utilizing the capabilities of DNNs requires large training datasets. Reserchers have also tried to reduce the complexity of the DNN models to obtain comparable performance when the size of training dataset is small [2]. The Few-Shot Learning (FSL) paradigm is an alternative attempt to tackle the problem with scarce training instances. The goal of FSL is to efficiently learn from a small number of "shots" (i.e., data samples or instances). The options for the number of samples usually range from 1 to 100 *per class* [6] [7]. There is a growing interest in the Artificial Intelligence (AI) research community in FSL and several different strategies have been developed for FSL, including Bowtie Networks [3], Induction Networks [4], and variations of Prototypical Networks [5].

A Siamese Neural Network (SNN), sometimes called a twin neural network, is an Artificial Neural Network (ANN) that uses two parallel, weight-sharing machine learning models in order to compute comparable embeddings. The SNN architecture has shown good results as a FSL approach in computer vision for similarity detection [11] and duplicate identification [12]. Yet, its usage in NLP has been understudied and, to the best of our knowledge, there have not been any studies of using SNNs for clinical NLP with a few training data instances.

In SNNs, neural networks need to be trained to generate and compute embeddings. In NLP, deep learning has achieved the state-of-the-art performance since it could generate comprehensive embeddings to encode both semantic and syntactic information. The main use of deep learning in NLP is to represent the language in a vector format (i.e., embeddings) so that the vector representation can be used for different NLP tasks, for example, natural language generation, text classification, and semantic textual similarity. Thus, embeddings play a key role in applying deep learning to NLP. Having a robust embedding-generation mechanism is crucial for most NLP tasks. Since the context of words, sentences, and more generally, text is important to learn meaningful embeddings, context-aware embedding-generation models, such as BERT [8], often generate promising results. Furthermore, depending on the domain, the context also varies. For this purpose, engineers and researchers have built domain-specific, specialized models to be used for downstream tasks. Examples of such models include BioBERT trained from biomedical literature texts and BioClinicalBERT trained from clinical texts [10]. Leveraging these contextual embeddings in SNN-based FSL has been rarely investigated in clinical NLP.

FSL is critical for clinical NLP as annotating a large training dataset is time-consuming and expensive. Furthermore, such annotation often requires involving domain experts. It is not uncommon to have a few clinical text samples annotated by physicians. One example could be clinical notes with annotations of a rare disease with the number of samples

naturally limited due to the nature of the disease. Despite such challenges in the clinical domain, the importance of using AI in clinical applications cannot be understated. AI could not only assist physicians in their decision-making and facilitate clinical and translational research, but also significantly reduce the need for manual work. Therefore, in this study, we propose an FSL approach based on SNNs to tackle clinical NLP tasks with only a few annotated training samples. Two SNN-based FSL approaches have been proposed, including pre-trained SNN (PT-SNN) and SNN with second-order embeddings (SOE-SNN). For every approach, three different transformer models - BERT, BioBERT, BioClinicalBERT - have been utilized. We evaluated the proposed approaches on two clinical tasks, namely clinical text classification and clinical named entity recognition. Clinical text classification refers to the classification of clinical texts based on pre-defined classes. Named Entity Recognition (NER) is a subtask of Information Extraction (IE) that seeks to identify entities mentioned in clinical texts. We show that SNN-based approaches outperform the baseline models in few-shot settings for both tasks. Finally, we discuss the limitations and future work.

II. Related Work

There have been studies evaluating the usability of SNNs for image classification. Mahajan et al. used SNNs for the classification of high-dimensional radiomic features extracted from MRI images [13]. Hunt et al. applied SNNs for the classification of electrograms [14]. Zhao et al. have utilized SNNs for hyperspectral image classification [15].

In the context of FSL, SNNs have been used by Torres et al. for one-shot, Convolutional Neural Networks (CNNs) based classification in order to optimize the discovery of novel compounds based on a reduced set of candidate drugs [16]. Droghini et al. employed SNNs for few-shot human fall detection purposes using images [17]. However, none of these studies used SNN-based FSL for NLP.

There is only a recent study by Müller et al [18] where they explored SNNs for FSL in NLP and demonstrated the high performance of pretrained SNNs that embed texts and labels. To the best of our knowledge, none of these studies referenced above are using SNNs to perform FSL in the clinical NLP domain.

III. SNN-Based Models

In this study, we consider two types of SNN-based models for clinical NLP.

A. Model Architecture

The SNN's architecture leverages two parallel weight-sharing ML models ("Fig. 1"). In the forward propagation step, data samples are passed into the models and mapped down to the latent space. The embeddings in the latent space are then compared with each other via a similarity function, as shown in Equ.(1). The similarity function is also a parameter that can be fine-tuned during training and could range from Euclidean distance to Manhattan distance or cosine similarity. Depending on the similarity function, the similarity value can then be mapped onto the $(0, 1)$ interval by applying a Sigmoid function. Finally, a high similarity value means that the input two samples likely belong to the same category, and vice versa.

$$\text{out} = \sigma(\text{distance}(emb_1, emb_2)) \tag{1}$$

During the training process, SNN attempts to have the best approximation for the input embeddings. Such training paradigm is also known as the representation learning. The representation is learned by penalizing the loss if the model yields a high similarity value for inputs from different classes or if the model yields a low similarity value for inputs from the same class.

The SNN architecture naturally allows for data augmentation. For instance, in the case of 8-shot learning, the traditional ML-based training approach would involve passing 8 samples directly into the model. This approach is very limiting with a such small number of samples. SNN takes a different route and instead of passing the samples directly, it considers unique comparisons within the training set. With the training set comprised of 8 unique samples, there are $\frac{8*7}{2} = 28$ unique comparisons in total. Thus, instead of 8 training samples, we get 28, which is 3.5 times more. In case of 16 samples, the improvement is even more significant as the number of unique comparisons is 120 and there is 7.5-fold data augmentation.

More generally, under N-way-K-shot-classification settings, for the dataset D_{train} with N class labels and K labeled samples for each class, the following holds after SNN-style augmentation:

$$D_{\text{train SNN}} = \{(x_i, x_j) \mid x_i, x_j \in D_{\text{train}}, i < j\} \tag{2}$$

$$\text{size}(D_{\text{train SNN}}) = \frac{(NK)^2 - NK}{2} \tag{3}$$

Hence, SNN reformulates the classification problem into a pairwise comparison problem, which has the benefit of obtaining more training data. This becomes increasingly important as the number of training samples decreases.

B. Pre-trained SNN (PT-SNN)

In the first approach, we leverage the pre-trained language models (PLMs) to generate embeddings for the SNN, called pre-trained SNN (PT-SNN). We used three PLMs in this approach, namely BERT, BioBERT, and BioClinicalBERT, to generate embeddings for the input training samples. In the following, we illustrate how to use the PT-SNN in the testing step. Suppose we want to perform binary classification, we are given two classes C_1 and C_2, a training set D_{train}, and a testing set D_{test}. We first compute embeddings for all samples in both D_{train} and D_{test}. For every testing sample, using the generated embeddings, we compute the similarity with respect to every training sample and compute the mean similarity values for classes.[1] For instance, mean similarity value for some sample

[1]We have also tried using maximum similarity values per class instead of mean similarity scores, but since the observed results were similar, we only include results using the mean similarity approach

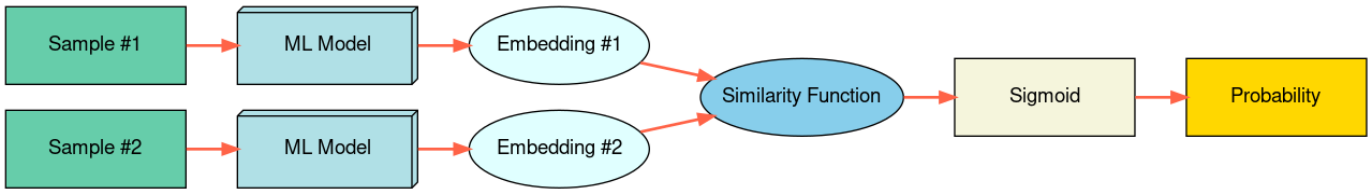

Fig. 1. Siamese Neural Network (SNN) Architecture.

$x \in D_{\text{test}}$ with respect to C_1 and C_2 might be 0.2 and 0.6 respectively. In this case, since 0.6 is greater than 0.2, we classify sample x as being in the class C_2.

Algorithm (1) presents the pseudocode that contains both the classification algorithm as well as evaluation approach. Here, `epochs` refers to the number of averaging epochs for addressing the instability issues. In our case, `epochs` is 5. `d` represents a distance function, which is cosine similarity in our case.

Such a strategy for classification can be slow in cases where the training set is very large. Fortunately, the proposed approach is feasible in the FSL setting where the number of annotated samples is limited. Thus, we do not expect significant performance drawbacks when the number of samples is not very high. Therefore, the PT-SNN approach is suitable for FSL purposes, but may not be as effective for training on very large datasets.

C. SNN With Second-Order Embeddings (SOE-SNN)

The second proposed approach is SNNs with second-order embeddings where we apply an additional Recurrent Neural Network (RNN) layer, such as Long-Short Term Memory (LSTM) or Gated Recurrent Unit (GRU) to the generated embeddings and then train the SNN model in the fashion described in the Model Architecture section ("Fig. 2"). In our experiments, we used bidirectional RNNs for producing second-order embeddings.

Specifically, we first obtain the embeddings for all training samples from the PLMs. All possible unique pairs of samples are then generated and given a label of 0 if the samples in the pair are of the same class or 1 if they come from different classes. During the training process, we use the Binary Cross Entropy (BCE) [22] and AdamW [23] as the loss function and the optimizer, respectively.

Model evaluation is done in the same manner as in pre-trained SNNs, where we compute mean similarity scores and average out the metrics over 5 testing epochs to handle the potential instability issues.

IV. Text Embeddings

In this section, we will describe how to generate embeddings, including sentence-level embeddings and word-level embeddings, using BERT, BioBERT, and BioClinicalBERT.

A. Sentence-Level Embeddings

For contextual, sentence-level embeddings, we used sentence-transformers [24] package. This package provides a set of intuitive and easy-to-use methods for computing dense vector representations of sentences, paragraphs, and images. The models are based on transformer networks such as BERT, RoBERTa, etc., and achieve state-of-the-art performance in various tasks. The generated sentence embeddings are such that similar texts are close in the latent space and can efficiently be found using cosine similarity.

$$\text{cosine similarity(A, B)} = \frac{A \cdot B}{\|\mathbf{A}\| \, \|\mathbf{B}\|} \tag{4}$$

B. Word-Level Embeddings

To generate contextual embeddings at the word level, we used `Transformers` package.

A tricky part of named entity recognition tasks with transformer models is that they rely on word piece tokenization, rather than word tokenization. For example, if we have a word such as `Washington`, it may get tokenized into three separate words `Wash`, `ing`, and `ton`. Then one approach could be to handle this by only train the model on the tag labels for the first word piece token of a word (i.e. only label "Wash"). This is what was done in the original BERT paper.

That being said, such approach does not work well if words that may have prefixes that can also be tokens. One such example can be a word `proto-potato`. In this case, tokenizer may split it into `proto` and `potato` and taking the first subword as the embedding of the entire word would have a semantically incorrect representation. Similarly, if we only took the last subword embedding, some words may have a suffix and we run into the same problem.

In order to solve this issue, we averaged out the embeddings for the subwords and took it as the representation for the word. Thus, we define the word embedding E_{word} with $e_0, e_1, \ldots, e_n$ subwords as follows:

$$E_{\text{word}} = \frac{\sum_{i=0}^{n} e_i}{n} \tag{5}$$

V. FSL Model Evaluation

Systematically evaluating FSL model performance can be tricky since fine-tuning or making predictions on small datasets could potentially suffer from instability [20]. To address this issue, we propose the averaging strategy for model evaluation. For every few-shot experiment (e.g., 4-shot, 8-shot, and 16-shot experiments), using randomized sampling, we sample 4, 8, or 16 samples per class and create a training dataset. We perform this M times and therefore, for every experiment, M randomly generated training sets are evaluated

Algorithm 1: Our Proposed Algorithm for SNN-Style Classification and Evaluation for Few-Shot Learning

Require: E_{test}: Test dataset embeddings
Require: L_{test}: Test dataset labels
Require: D_{train}: Train dataset
Require: d: Distance function
Require: $epochs$: Number of evaluation epochs
Require: $RandSubset$: A function that randomly subsets a dataset with the given seed and generates embeddings
Require: $Mean$: Calculates the mean of a vector
Require: $GetMaxValueKey$: A function that gets the key with the maximum value from the hash table
Require: $ComputeMetrics$: A function for computing evaluation metrics - precision, recall, and F-score

$metrics \leftarrow$ empty hash table;
for $i \leftarrow 0$ **to** $epochs$ **do**
  $E_{train}, L_{train} \leftarrow RandSubset(D_{train}, seed = i)$;
  $predictions \leftarrow$ empty vector;
  foreach $e_{test} \in E_{test}$ **do**
    $similarity \leftarrow$ empty hash table;
    foreach $(e_{train}, l_{train}) \in Zip(E_{train}, L_{train})$ **do**
      $similarity.At(l_{train}).Add(d(e_{train}, e_{test}))$;
    end

    $tmp \leftarrow$ empty hash table;
    foreach $(key,\ val) \in similarity$ **do**
      $tmp.At(key) = Mean(val)$;
    end

    $predictions.Add(GetMaxValueKey(tmp))$;
  end
  $precision, recall, fscore = ComputeMetrics(predictions, L_{test})$;
  $metrics.At(\text{"}precision\text{"}).Add(precision)$;
  $metrics.At(\text{"}recall\text{"}).Add(precision)$;
  $metrics.At(\text{"}fscore\text{"}).Add(precision)$;
end
$precision = Mean(metrics[\text{"}precision\text{"}]))$;
$recall = Mean(metrics[\text{"}recall\text{"}]))$;
$fscore = Mean(metrics[\text{"}fscore\text{"}]))$;

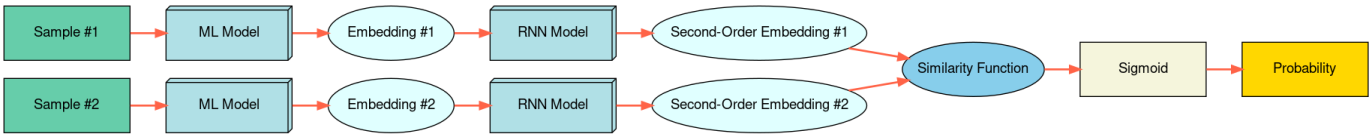

Fig. 2. Siamese Neural Network (SNN) Architecture.

on the test set. Finally, the metrics are averaged out and reported as the final scores.

$$\text{Metric} = \frac{\sum_{i=1}^{M} \text{Metric}_i}{M} \tag{6}$$

In this paper, we choose $M = 3$. Such approach gives a more robust view on the performance of the model in possibly unstable scenarios. We employ this strategy in all reported metrics. As for metrics, we choose Precision, Recall, and F-score:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{7}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{8}$$

$$\text{F-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{9}$$

VI. Baseline Models

We use fine-tuned BERT, BioBERT, and BioClinicalBERT as the baseline models. Instead of approaching the problem

from the SNN perspective, we use the 4, 8, and 16 samples per class directly in order to fine-tune the pre-trained transformer models.

This is done by adding an additional linear layer at the end of the transformer model. We used `Transformers` package [21] in our implementation.

VII. Experiments

A. Classification Tasks and Datasets

We perform both clinical text classification and clinical named entity recognition. For few-shot clinical text classification, we use sentences from the MIMIC-III [25] databases and classify sentences them into 4 different classes classes. As for few-shot clinical named entity recognition, we use the i2b2 (Informatics for Integrating Biology and the Bedside) 2006 de-identitification challenge dataset [19] and do binary classification of one-word entities. The datasets were preprocessed in order to be used in few-shot experiments.

1) MIMIC-III: For sentence classification, the sentences were obtained from the MIMIC-III database. We used the same dataset as in the HealthPrompt paper by Sivarajkumar and Wang [26], but with classes suitable for 4-shot, 8-shot, and 16-shot experiments.

The following were the classes that were left:

- `ADVANCED.LUNG.DISEASE (245 samples)`
- `ADVANCED.HEART.DISEASE (117 samples)`
- `CHRONIC.PAIN.FIBROMYALGIA (48 samples)`
- `ADVANCED.CANCER (34 samples)`

The dataset had two columns: `text` and `label` corresponding to the sentence and label respectively.

In total, we had 444 samples in our dataset. Since we performed 4, 8, and 16 shot experiments, the train size varied and was 16, 32, and 64 samples with the test sizes of 428, 412, and 380 samples respectively.

2) 2006 i2b2 De-Identification Challenge Dataset: The i2b2 (Informatics for Integrating Biology and the Bedside) organized clinical NLP challenge in 2006 with de-identification track focused on identifying Protected Health Information (PHI) from clinical narratives. The i2b2 2006 challenge developed a corpus of de-identified records such that there is one record per patient.

The dataset has both training and testing sets, which are separate from each other. This allows for a more transparent comparison reporting of benchmarks and statistics. One limitation that we also discuss in the "Future Work" section is that we filtered out all multi-word entities and only perform a one-word NER. We also note this in section VIII.

Both training and testing sets are XML files comprised of text, some of which contain `PHI` tags. One example of such text is the following:

```
<PHI TYPE="DATE">11/28</PHI>/03 02:25 PM
```

Here, we have text where `11/28` is a PHI, whose type is `DATE`. The goal is to correctly identify this named entity. While the original dataset does contain classes for `PHI` tags (e.g., `DATE` in this case), we do not use them as separate classes for NER, but only to identify word as a named entity. But similar to sentence classification, we do sample equal number of samples based on class.

We used two files provided by challenge:

- `deid_surrogate_test_all_groundtruth_version2.xml`
- `deid_surrogate_train_all_version2.xml`

We used all of the `PHI`s except for `AGE` since the train file only contained only 13 such tags, which is not enough for conducting 16-shot experiments.

In the end, we had the following 8 classes:

- `ID`
- `HOSPITAL`
- `DATE`
- `PATIENT`
- `LOCATION`
- `DOCTOR`
- `PHONE`
- `NONPHI` (class that represents non-PHI/non-named-entity words)

We have split sentences into words and generate word-level contextual embeddings for every word. In the end, we obtained the dataset that was comprised of the following:

- `Test Set`
  - `Named Entities: 3193 samples`
  - `Other Terms (labeled NONPHI): 16036 samples`
  - `Total: 19229 samples`
- `Train Set`
  - `Named Entities: 9083`
  - `Other Terms (labeled NONPHI): 39319`
  - `Total: 48402 samples`

While the entire test set was used in all experiments, only 32, 64, and 128 samples were taken from the training set for 4-shot, 8-shot, and 16-shot experiments respectively. All experiments were run on four NVIDIA RTX 8000 GPUs.

B. Results

1) Few-Shot Clinical Text Classification: We present the results of 4-shot, 8-shot, and 16-shot experiments for few-shot clinical text classification tasks. We have used BERT, BioBERT, and BioClinicalBERT based models. The results are shown in Table I.

In the 4-shot clinical text classification task, it is clear that the baseline, fine-tuned transformer, approach had the worst performance. PT-SNN and SOE-SNN both significantly outperformed the fine-tuned transformers. PT-SNN marginally outperformed SOE-SNN and their performance was relatively similar. For the 4-shot sentence classification task, the best model was BioBERT-based PT-SNN with the precision of 0.53, recall of 0.45, and f-score of 0.46. In order to measure the statistical significance of the difference between the best fine-tuned transformer approach (BioClinicalBERT-based model) and the best SNN-based approach, we perform a paired t-test. This is done

by creating two vectors $[precision_1, recall_1, fscore_1]$ and $[precision_2, recall_2, fscore_2]$, calculating the difference vector d, and then running the one-sample t-test. The difference was significant with the p-value of 0.0377 (we take 0.05 cutoff).

SOE-SNN outperformed all other approaches. BioClinicalBERT-based SOE-SNN marginally outperformed BioBERT-based variant. PT-SNN came next with fine-tuned transformers having the worst performance. The best baseline model (i.e., fine-tuned transformer) had the precision, recall, and fscore values of 0.35, 0.35, and 0.21 respectively. This is a BioClinicalBERT-based model. BERT-based model had a higher fscore of 0.24, but both precision and recall were lower. The best SNN-based model was based on BioClinicalBERT. After performing a paired t-test in the similar manner as for 4-shot experiment evaluation, we got the p-value of 0.0394, which means that the difference is significant and that our proposed approach has substantial improvement over the baseline model.

As for 16-shot experiments, BioClinicalBERT-based SOE-SNN outperformed all other models. BioClinicalBERT-based PT-SNN came next with the precision, recall, and fscore values of 0.69, 0.54, and 0.58 respectively. The worst performance, again, was shown by the baseline fine-tuned transformer models, with the best score being 0.39 (precision), 0.37 (recall), and (0.27) fscore. After performing a paired t-test, we obtained the p-value of 0.0292. Given the cutoff of 0.05, we got that the difference between the metrics is significant and therefore, the proposed approach significantly outperforms the baseline models. It is noteworthy that the p-value was a lot lower (relatively) than in 8-shot experiments, which can be an interesting area to explore (i.e., what is the threshold for the number of shots at which fine-tuning directly becomes more effective than our proposed SNN-based approaches).

Overall, it is clear that in all cases, SNN-based approaches outperformed the baseline models. The difference in performance for PT-SNN and SOE-SNN was not statistically significant, with the latter marginally outperforming the former. Furthermore, we have measured the significance via paired t-test and discovered that in all three cases (4-shot, 8-shot, and 16-shot experiments), the difference was statistically significant. It should also be noted that we measured statistical significance on the statistics themselves and paired t-test on the model outputs was not necessary.

2) Few-Shot Clinical Named Entity Recognition: We present the results of 4-shot, 8-shot, and 16-shot experiments for few-shot clinical NER tasks. Similar to clinical text classification, we have used BERT, BioBERT, and BioClinicalBERT based models. The results are shown in Table II.

There was not a statistically significant difference between PT-SNN and SOE-SNN. That being said, once again, SNN-based models outperformed the baseline, fine-tuned transformer models in all experiments.

In 4-shot experiments, the best model was BioClinicalBERT-based PT-SNN. The best baseline model was the BERT-based fine-tuned transformer. SNN-based model achieved the precision, recall, and f-score of 0.89, 0.64, and 0.68 respectively, while the fine-tuned transformer had those of 0.70, 0.59, and 0.60 respectively. After performing the paired t-test in the same fashion as in sentence classification tasks, we got the p-value of 0.1290, which is not statistically significant. That being said, in all metrics, SNN-based approach was superior to the fine-tuned transformers.

In 8-shot experiments, similar to the 4-shot scenario, BioClinicalBERT-based PT-SNN was the best model. The best baseline model was the BERT-based fine-tuned transformer. SNN-based model achieved the precision, recall, and f-score of 0.88, 0.55, and 0.59 respectively, while the fine-tuned transformer had those of 0.74, 0.52, and 0.53 respectively. After performing the paired t-test in the same fashion as in sentence classification tasks, we got the p-value of 0.1446, which is not statistically significant. That being said, in all metrics, SNN-based approach was superior to the fine-tuned transformers.

In 16-shot experiments, BioClinicalBERT-based SOE-SNN outperformed all other models. The best baseline model was the BERT-based fine-tuned transformer. SNN-based model achieved the precision, recall, and f-score of 0.90, 0.50, and 0.51 respectively, while the fine-tuned transformer had those of 0.76, 0.54, and 0.55 respectively. After performing the paired t-test in the same fashion as in sentence classification tasks, we got the p-value of 0.7705, which is not statistically significant. That being said, in all metrics, SNN-based approach was superior to the fine-tuned transformers.

It should be noted that metrics recall and F1-score have decreased for the SNN-based approaches when the number of samples increased. This pattern was not observed in SC tasks. This could potentially be explained by the fact that classification was done directly on the word embeddings and some instability may have been introduced during the evaluation process (despite the fact that we averaged out the metrics over 3 different evaluation runs).

VIII. Limitations and Future Work

There are several limitations of the work that can be addressed by further exploring few-shot learning and SNNs. First, it can be interesting comparing results to more traditional baseline models such as SVM, logistic regression, multinomial logistic regression, random forest, etc. Second, other datasets can also be used for evaluating the performance of SNNs for text classification purposes. Third, we had a limitation when doing NER - particularly, we only considered one-word entities. Evaluating the performance of SNNs considering all entities can also be an interesting for the future research. Additionally, since we can do both word-level and sentence-level classification, another interesting direction of research could be document classification, where a document can be a collection of words and sentences.

It is important to note that datasets for FSL and especially, for clinical FSL is difficult to find. Ge et al. [27], in their re-

TABLE I
FEW-SHOT SENTENCE CLASSIFICATION RESULTS

Approach	Model	Shots	Precision	Recall	F-Score
Fine-Tuned Transformer	BERT	4	0.24	0.24	0.14
Fine-Tuned Transformer	BioBERT	4	0.23	0.23	0.16
Fine-Tuned Transformer	BioClinicalBERT	4	0.25	0.31	0.18
Pre-Trained SNN	BERT	4	0.49	0.37	0.37
Pre-Trained SNN	**BioBERT**	**4**	**0.53**	**0.45**	**0.46**
Pre-Trained SNN	BioClinicalBERT	4	0.50	0.42	0.43
SNN With Second-Order Embeddings	BERT	4	0.47	0.51	0.42
SNN With Second-Order Embeddings	BioBERT	4	0.44	0.47	0.44
SNN With Second-Order Embeddings	BioClinicalBERT	4	0.50	0.47	0.42
Fine-Tuned Transformer	BERT	8	0.34	0.30	0.24
Fine-Tuned Transformer	BioBERT	8	0.35	0.32	0.22
Fine-Tuned Transformer	BioClinicalBERT	8	0.35	0.35	0.21
Pre-Trained SNN	BERT	8	0.62	0.45	0.47
Pre-Trained SNN	BioBERT	8	0.61	0.48	0.50
Pre-Trained SNN	BioClinicalBERT	8	0.64	0.44	0.49
SNN With Second-Order Embeddings	BERT	8	0.59	0.47	0.50
SNN With Second-Order Embeddings	**BioBERT**	**8**	**0.62**	**0.52**	**0.55**
SNN With Second-Order Embeddings	**BioClinicalBERT**	**8**	**0.65**	**0.51**	**0.55**
Fine-Tuned Transformer	BERT	16	0.23	0.31	0.14
Fine-Tuned Transformer	BioBERT	16	0.35	0.33	0.18
Fine-Tuned Transformer	BioClinicalBERT	16	0.39	0.37	0.27
Pre-Trained SNN	BERT	16	0.64	0.51	0.52
Pre-Trained SNN	BioBERT	16	0.65	0.55	0.56
Pre-Trained SNN	BioClinicalBERT	16	0.69	0.54	0.58
SNN With Second-Order Embeddings	BERT	16	0.66	0.58	0.58
SNN With Second-Order Embeddings	BioBERT	16	0.68	0.55	0.56
SNN With Second-Order Embeddings	**BioClinicalBERT**	**16**	**0.71**	**0.55**	**0.60**

TABLE II
FEW-SHOT NAMED-ENTITY RECOGNITION RESULTS

Approach	Model	Shots	Precision	Recall	F-Score
Fine-Tuned Transformer	BERT	4	0.70	0.59	0.60
Fine-Tuned Transformer	BioBERT	4	0.66	0.56	0.57
Fine-Tuned Transformer	BioClinicalBERT	4	0.66	0.19	0.15
Pre-Trained SNN	BERT	4	0.87	0.45	0.49
Pre-Trained SNN	BioBERT	4	0.87	0.24	0.18
Pre-Trained SNN	**BioClinicalBERT**	**4**	**0.89**	**0.64**	**0.68**
SNN With Second-Order Embeddings	BERT	4	0.80	0.42	0.41
SNN With Second-Order Embeddings	BioBERT	4	0.83	0.48	0.50
SNN With Second-Order Embeddings	BioClinicalBERT	4	0.76	0.58	0.61
Fine-Tuned Transformer	BERT	8	0.74	0.52	0.53
Fine-Tuned Transformer	BioBERT	8	0.70	0.39	0.38
Fine-Tuned Transformer	BioClinicalBERT	8	0.71	0.30	0.29
Pre-Trained SNN	BERT	8	0.87	0.37	0.37
Pre-Trained SNN	BioBERT	8	0.87	0.25	0.19
Pre-Trained SNN	**BioClinicalBERT**	**8**	**0.88**	**0.55**	**0.59**
SNN With Second-Order Embeddings	BERT	8	0.88	0.35	0.36
SNN With Second-Order Embeddings	BioBERT	8	0.87	0.24	0.20
SNN With Second-Order Embeddings	BioClinicalBERT	8	0.88	0.54	0.60
Fine-Tuned Transformer	BERT	16	0.76	0.54	0.55
Fine-Tuned Transformer	BioBERT	16	0.71	0.40	0.39
Fine-Tuned Transformer	BioClinicalBERT	16	0.74	0.37	0.34
Pre-Trained SNN	BERT	16	0.87	0.33	0.32
Pre-Trained SNN	BioBERT	16	0.87	0.25	0.20
Pre-Trained SNN	BioClinicalBERT	16	0.88	0.47	0.49
SNN With Second-Order Embeddings	BERT	16	0.88	0.34	0.30
SNN With Second-Order Embeddings	BioBERT	16	0.88	0.27	0.22
SNN With Second-Order Embeddings	**BioClinicalBERT**	**16**	**0.90**	**0.50**	**0.51**

view paper, have emphasized that "(68%) studies reconstructed existing datasets to create few-shot scenarios synthetically."

IX. Conclusion

We have conducted few-shot learning experiments evaluating the performance of SNN models on text classification tasks - SC and NER. SNN models were based on transformer models - BERT, BioBERT, and BioClinicalBERT. Fine-tuned versions of BERT, BioBERT, and BioClinicalBERT were also used as the baseline models. Since performance evaluation on small datasets may suffer from instability, a special evaluation strategy was used. We conclude that SNN-based models outperformed the baseline fine-tuned transformer models for sentence classification tasks. A paired t-test was also performed, which showed that for SC tasks SNN models significantly outperform the baseline models. As for NER, SNN models, once again, outperformed the baseline models, yet the performance difference was not statistically significant. Limitations of the work have also been discussed and alongside with potential future directions of exploration.

X. Acknowledgements

The authors would like to acknowledge support from the University of Pittsburgh Momentum Funds, Clinical and Translational Science Institute Exploring Existing Data Resources Pilot Awards, the School of Health and Rehabilitation Sciences Dean's Research and Development Award, and the National Institutes of Health through Grant UL1TR001857.

References

[1] T. J. Sejnowski, "The unreasonable effectiveness of deep learning in artificial intelligence," Proc. Natl. Acad. Sci. U. S. A., vol. 117, no. 48, pp. 30033–30038, 2020.

[2] L. Brigato and L. Iocchi, "A Close Look at Deep Learning with Small Data," 2020 25th International Conference on Pattern Recognition (ICPR), 2021, pp. 2490-2497, doi: 10.1109/ICPR48806.2021.9412492.

[3] Zhipeng Bao and Yu-Xiong Wang and Martial Hebert, "Bowtie Networks: Generative Modeling for Joint Few-Shot Recognition and Novel-View Synthesis," International Conference on Learning Representations (ICLR), 2021.

[4] R. Geng, B. Li, Y. Li, X. Zhu, P. Jian, and J. Sun, "Induction networks for few-shot text classification," 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), 2019, pp. 3904–3913.

[5] X. Liu, P. Liu and L. Zong, "Transductive Prototypical Network For Few-Shot Classification," 2020 IEEE International Conference on Image Processing (ICIP), 2020, pp. 1671-1675, doi: 10.1109/ICIP40778.2020.9191037.

[6] M. Yu et al., "Diverse few-shot text classification with multiple metrics," in Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), 2018, pp. 1206–1215.

[7] E. Manousogiannis, S. Mesbah, A. Bozzon, S. Baez, and R. J. Sips, "Give it a shot: Few-shot learning to normalize ADR mentions in social media posts," in Proceedings of the Fourth Social Media Mining for Health Applications (#SMM4H) Workshop & Shared Task, 2019, pp. 114–116.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional Transformers for language understanding," in Proceedings of the 2019 Conference of the North, 2019, pp. 4171–4186.

[9] J. Lee et al., "BioBERT: a pre-trained biomedical language representation model for biomedical text mining," Bioinformatics, vol. 36, no. 4, pp. 1234–1240, 2020.

[10] E. Alsentzer et al., "Publicly Available Clinical BERT Embeddings," in Proceedings of the 2nd Clinical Natural Language Processing Workshop, 2019, pp. 72–78.

[11] Y. Wu and W. Wang, "Code Similarity Detection Based on Siamese Network," 2021 IEEE International Conference on Information Communication and Software Engineering (ICICSE), 2021, pp. 47-51, doi: 10.1109/ICICSE52190.2021.9404110.

[12] M. Fisichella, "Siamese coding network and pair similarity prediction for near-duplicate image detection," Int. J. Multimed. Inf. Retr., vol. 11, no. 2, pp. 159–170, 2022.

[13] A. Mahajan, J. Dormer, Q. Li, D. Chen, Z. Zhang, and B. Fei, "Siamese neural networks for the classification of high-dimensional radiomic features," Proc. SPIE, vol. 11314, 2020.

[14] B. Hunt, E. Kwan, D. Dosdall, R. S. MacLeod and R. Ranjan, "Siamese Neural Networks for Small Dataset Classification of Electrograms," 2021 Computing in Cardiology (CinC), 2021, pp. 1-4, doi: 10.23919/CinC53138.2021.9662707.

[15] S. Zhao, W. Li, Q. Du and Q. Ran, "Hyperspectral Classification Based on Siamese Neural Network Using Spectral-Spatial Feature," IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium, 2018, pp. 2567-2570, doi: 10.1109/IGARSS.2018.8519286.

[16] L. Torres, N. Monteiro, J. Oliveira, J. Arrais and B. Ribeiro, "Exploring a Siamese Neural Network Architecture for One-Shot Drug Discovery," 2020 IEEE 20th International Conference on Bioinformatics and Bioengineering (BIBE), 2020, pp. 168-175, doi: 10.1109/BIBE50027.2020.00035.

[17] D. Droghini, F. Vesperini, E. Principi, S. Squartini, and F. Piazza, "Few-shot Siamese neural networks employing audio features for human-fall detection," in Proceedings of the International Conference on Pattern Recognition and Artificial Intelligence - PRAI 2018, 2018.

[18] T. Müller, G. Pérez-Torró, and M. Franco-Salvador, "Few-shot learning with Siamese Networks and label tuning," arXiv [cs.CL], pp. 8532–8545, 2022.

[19] O. Uzuner, Y. Luo, and P. Szolovits, "Evaluating the state-of-the-art in automatic de-identification," J. Am. Med. Inform. Assoc., vol. 14, no. 5, pp. 550–563, 2007.

[20] T. Zhang et al., "Revisiting fewsample BERT fine-tuning," in International Conference on Learning Representations (ICLR), 2021.

[21] T. Wolf et al., "Transformers: State-of-the-art natural language processing," in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2020.

[22] I. J. Good, "Rational Decisions," J. R. Stat. Soc., vol. 14, no. 1, pp. 107–114, 1952.

[23] I. Loshchilov et.al., "Decoupled weight decay regularization," in International Conference on Learning Representations (ICLR), 2019.

[24] "Sentence Transformers," https://github.com/UKPLab/sentence-transformers

[25] Johnson, A. E. W. et al. "MIMIC-III, a freely accessible critical care database." Sci. Data 3:160035 doi: 10.1038/sdata.2016.35 (2016).

[26] S. Sivarajkumar and Y. Wang, "HealthPrompt: A Zero-shot learning paradigm for Clinical Natural Language Processing," arXiv.org, 09-Mar-2022. [Online]. Available: https://arxiv.org/abs/2203.05061. [Accessed: 27-Jul-2022].

[27] Y. Ge, Y. Guo, Y.-C. Yang, M. A. Al-Garadi, and A. Sarker, "Few-shot learning for medical text: A systematic review," arXiv [cs.CL], 2022.