Islamic University of Technology (IUT)

Enhancing Performance and Efficiency in Personal Health Mention Detection

Authors

Nuzhat Nower, 180041116

Fida Kamal, 180041208

Alvi Aveen Khan, 180041229

Supervisor

Tareque Mohmud Chowdhury

Assistant Professor

Dept. of CSE, IUT

Co-Supervisor

Tasnim Ahmed

Lecturer

Dept. of CSE, IUT

A thesis submitted in partial fulfilment of the requirements
for the degree of B. Sc. Engineering in Computer Science and Engineering

Academic Year: 2021-2022

Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT)
A Subsidiary Organ of the Organization of Islamic Cooperation (OIC)
Dhaka, Bangladesh

December 26, 2022

Declaration of Authorship

This is to certify that the work presented in this thesis is the outcome of the analysis and experiments carried out by under the supervision of Tareque Mohmud Chowdhury, Assistant Professor of the Department of Computer Science and Engineering (CSE), Islamic University of Technology (IUT), Dhaka, Bangladesh. It is also declared that neither of this thesis nor any part of this thesis has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Authors:

Nuzhat Nower

Student ID - 180041116

Fida Kamal

Student ID - 180041208

Alvi Aveen Khan

Student ID - 180041229

Approved By:

Supervisor:

Tareque Mohmud Chowdhury
Assistant Professor
Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT), OIC

Co-Supervisor:

Tasnim Ahmed
Lecturer
Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT), OIC

Acknowledgement

We would like to thank our mentor and advisor, **Tareque Mohmud Chowdhury**, Assistant Professor, Department of Computer Science & Engineering, IUT. The inspiration, advice and knowledge he provided were invaluable to this study, and this research would not have been possible without it. He guided us through each phase of the thesis, from the initial conceptualization, through the implementation details and the final reviews, for which we are grateful.

We are also grateful to **Tasnim Ahmed**, Lecturer, Department of Computer Science & Engineering, IUT for his insights, suggestions and constant support. His enthusiasm for this research was on par with ours and had a huge contribution towards its successful completion.

Abstract

A major task that must be accomplished by organisations which monitor public health is the analysis of personal health related posts made on online social media platforms, known as Personal Health Mention (PHM) Detection. PHM detection is essential to quickly detecting epidemics, which allows health organisations to prepare themselves and warn the general public to take precautionary steps. Unfortunately, there are multiple issues that make this a difficult task. The informal nature of the language used in social media makes it difficult to understand the context of health mentions, and the architectures that are capable of discerning context are computationally expensive to use. Such architectures are also generally mistrusted in the medical community due to their black-box nature, which makes it difficult to explain their decisions.

In this thesis, we address each of these issues separately. We propose four transformer-based ensemble architectures that have not been previously explored in the PHM domain, computationally efficient training mechanisms to reduce the resource usage of such architectures and methods to explain the outputs produced by the architectures in order to address the concerns related to explainability. As an initial step towards this work, we also propose and compare a few transformer-based models trained on an imbalanced PHM dataset produced by collecting a large number of public posts from Twitter. The empirical results show that we have achieved state-of-the-art performance on the dataset, with an average F1 score of 94.5% with the RoBERTa-based classifier.

Contents

1 Introduction

1.1 Overview

One of the vital services that many government health organizations provide is looking out for developing threats to public health. This is done by the Centers for Diseases Control and Prevention in the USA and the Care Quality Commission in the UK to name just a few. Traditionally, the organizations did this by manually collecting health reports, a lengthy process which did not allow them to keep up with health risks that spread quickly [46]. As the use of social media platforms becomes commonplace, these organizations have turned to online health monitoring as an alternative to such issues [20].

1.2 Problem Statement

The first step in online health monitoring is correctly identifying the posts that are relevant to the task at hand, Personal Health Mention (PHM) detection. It is required to collect posts that mention health issues that the author of the post or someone they know is facing [30]. A post with the text 'I have a headache', for example, is a personal health mention but one which shares an article link with possible cures to headaches is not. The majority of the recent work done to solve this problem makes use of deep-learning models, which are computationally expensive to use. However, it is important to remember that the end product will be used by health organizations with limited resources. As such, limiting resource usage is vital.

1.3 Research Challenges

There are several issues that are faced when trying to address the problem stated above. The challenges are varied and occur both with the data that must be used and with the models being created. Addressing these challenges will make the process of creating and training the models easier and also allow the models to attain improved performance. Some of the challenges are discussed below:

1. Noisy Text: Text on social media sites are uniquely different from general text deu to the nature of the noise they contain. Social media presents an informal setting which allows for the relaxation of common rules that are usually followed in other forms of textual communication. It is generally accepted that users will disregard minor spelling mistakes or use short forms of common words. In addition to this, the majority of the data in the PHM domain is collected from the social media site Twitter, which limits posts to 280 characters. As a result, users are forced to invent new and unique ways of shortening their posts so as to fully communicate their thoughts within that limit. Such noise makes it difficult to train natural language processing architectures [13].

2. Context Recognition: Informal communication also has a tendency to encourage the use of sarcastic, figurative or hyperbolic language. This can go so far as to become confusing even for other humans. Understanding the context of a post is essential to being able to correctly identify whether it is a PHM or not, and such language makes the task all the more difficult.

3. Resource Usage: Currently, one of the most reliable methods of understanding context is to use models based on the transformer architecture. However, these models are very large and require significantly powerful resources to run. This presents a problem for health organizations which have limited resources that would be better spent on more direct ways of supporting public health. The computational expense of the architecture thus prevents practical applications from using them.

1.4 Thesis Objectives

The objective of this thesis is to identify areas in the PHM domain which can be improved upon by examining existing work. This includes the use of the correct architecture, dataset and training process so as to ensure a positive outcome.

In addition, it is also important to be mindful of the computational expense of the proposed architecture, and to optimize the resource usage to minimize this expense.

1.5 Thesis Contribution

In this thesis, we explore the use of transformer-based architectures on a relatively new dataset. The performance of such architectures on this dataset has not previously been studied, and we fill this gap in knowledge. By using the proposed architecture, we are able to achieve state-of-the-art results, with an F1-Score of 94.5%. We also explore the use of optimization techniques that can be used to reduce the computational expense of the architectures.

1.6 Organization of the Thesis

The rest of this thesis is organized as follows: in Chapter 2, a review of existing literature in the PHM domain is presented, along with discussions about their contributions and limitations. In Chapter 3, the datasets that exist in the domain are discussed. Chapter 4 provides proposals for possible points of improvement in existing work, while Chapter 5 presents the experimental findings of our work. Chapter 6 concludes the thesis and discusses possible directions for future work.

2 Literature Review

There are various approaches that have been explored in existing literature. These approaches are being discussed in this chapter.

2.1 Traditional Machine Learning Approaches

The traditional machine learning approaches that have been used are comparatively straightforward and do not provide information about possible approaches that would be useful anymore. However, they are still being included for reference, but are being combined under one section.

These approaches almost invariably consist of two sections, a feature extraction section and a classification section. To extract features, various methods have been used including using user mentions and emotion keywords [17], n-gram feature extraction [5], and topic modelling [9, 45]. To actually classify the text as a positive or negative personal health mention, the extracted features were passed to a classifier. Popular classifiers in the domain include Support Vector Machines (SVMs) [5, 44], K-Nearest Neighbour (KNN) networks [17] and Decision Trees.

2.2 Deep Learning Approaches

The reliance on handcrafted features was a huge issue for traditional machine learning models, since the process of identifying which features to extract required medical expertise and was cumbersome and time-consuming. Deep-learning models bypassed this need by using word embeddings, which are vectors representing linguistic information about the words in a piece of text. In the domain of PHM, the importance of word embeddings was proven by Iyer et al. [15]. Their research adds to the justification behind the widespread use of word embeddings in the domain. Jiang et al. [18] for example, used Word2Vec [37] word embeddings to train a Long Short-Term Memory (LSTM) network, while Wang et al. [53] used GloVe-based word embeddings [47] along with a bi-directional LSTM (Bi-LSTM) network.

2.3 WESPAD Method

One of the most prominent works in recent years that made use of deep learning algorithms was by Karisani and Agichtein [23], who proposed the Word Embedding Space Partitioning and Distortion (WESPAD) method. This combines lexical, syntactic, word embedding-based, and context-based features. It partitions the word embedding space to generalise from a small amount of training data and distorts the embedding space to effectively detect true health mentions.

2.4 Contextual Word Representations

Biddle et al. [6] was the first to use contextual word representations in the PHM domain. These became popular in the domain of NLP in general due to the introduction of the BERT architecture [12], which is a transformer-based architecture. Initially proposed by Vaswani et al. [52], transform-based architectures use a concept called self-attention which allows the models to learn the relationship between different parts of a sentence. This essentially means that the models are extremely effective in understanding context, which has consequently led to their use in a variety of language-related tasks [2, 16, 32]. Several architectures have since been developed based on the transformer architecture, with varying degrees of changes.

Transformer-based architectures have shown such a significant improvement over previous architectures in the domain of NLP, that simply fine-tuning these pre-trained models to a dataset is often enough to achieve exceptionally good results. In the domain of PHM detection, Khan et al. [26] was one of the first to use a pre-trained transformer-based model. They used the XLNet architecture [54] and fine-tuned it to the HMC2019 dataset. Their work showed that the permutation-based word representations used by the XLNet architecture were more effective than the bi-directional word representations used by the BERT architecture. The authors later also provided a comprehensive comparison of the performance of various pre-trained transformer-based models on the HMC2019 dataset [28], where they concluded that the RoBERTa architecture [33] achieved the best performance.

A similar approach was taken by Naseem et al. [43], who proposed the PHS-BERT pre-trained language model. This model was initialised with the weights from the BERT architecture and then further pre-trained on social media texts collected from Twitter so as to ensure improved performance on downstream tasks that used data from social media. Their model was evaluated on seven Personal Health Surveillance tasks, one of which was PHM detection.

2.5 Adversarial Training

To improve performance further, Khan et al. [29] proposed a new approach to the fine-tuning process. They used Adversarial Training (AT) to generate perturbed examples, which they used to fine-tune the BERT and RoBERTa architectures. Along with this, they used the Barlow Twins (BT) contrastive loss function, which pushed the clean and perturbed examples closer together, thus allowing the models to learn noise-invariant feature representations. In this manner, they improved the robustness of the models against noise. Another work [27] seems to have used the same approach, with no apparent differences.

2.6 CT-BERT + Bi-LSTM

Naseem et al. [42] proposed another architecture which combined several features, including contextual representations, parts-of-speech tags and sentiment distributions. They initialised the BERT architecture with the weights of CT-BERT [40], which was pre-trained on Twitter posts related to the COVID-19 pandemic for a variety of natural language processing tasks, including classification. Due to the training process, the CT-BERT architecture is able to obtain superior performance compared to similar architectures on health-related social media data specifically. The authors further concatenated the context embeddings from this architecture with parts-of-speech tags generated using an English parts-of-speech tagger [7]. This module was included to capture syntactic dependencies in text, which they showed contributed to improved performance. Next, the authors used a Bi-LSTM classifier, the output of which they concatenated with attention scores generated by a symptom-based attention mechanism [55] as well as Valence, Arousal, and Dominance (VAD) sentiment distributions [38]. The symptom-based attention mechanism helped the model learn to pay special attention to symptom keywords, and the VAD sentiment distributions, previously used by Biddle et al. [6], was already known to contribute towards improved performance.

3 Datasets

This chapter examines the various datasets that exist in the PHM domain, as well as the shortcomings of each.

3.1 PHM2017

The PHM2017 dataset was introduced by Karisani and Agichtein [23] and has since been widely used in the PHM domain. The datasets that existed before this all had significant shortcomings, either because they used the names of medicines instead of disease keywords [18], consisted of just a single disease class [30] or had very few samples [25]. The PHM2017 dataset covers six different disease classes with a total of 7,192 samples, making it a significantly better dataset.

Table 1: Disease-Wise Class Distribution of PHM2017 Dataset

Topic	**Tweet Count**	**Self Mention**	**Other Mention**	**Awareness**	**Non Health**
Alzheimer's	1256	1%	17%	80%	2%
Heart Attack	1219	4%	9%	17%	70%
Parkinson's	1040	2%	9%	65%	24%
Cancer	1242	3%	18%	62%	17%
Depression	1213	37%	3%	49%	11%
Stroke	1222	3%	11%	29%	57%

The samples were collected from Twitter using keyword searches of the colloquial names of diseases and were manually annotated into four classes, self-mention, other mention, awareness and non-health. The disease-wise class distribution for the dataset is shown in Table 1. Self-mention and other-mention refer to posts that discuss a disease that the author or someone the other knows is facing, meaning both of these categories are positive classifications. Awareness and non-health either discuss the disease in generic terms without specifying that someone is suffering from the disease or are entirely unrelated to the disease. As such, both of these categories are negative classifications.

3.2 HMC2019

Despite its widespread use, the PHM2017 dataset had one major issue. Posts collected from social media texts frequently involved sarcasm or figurative language. The dataset did not explicitly take these use cases into account, and as a result, models that trained on the dataset, struggled with figurative language [19]. To address this issue, Biddle et al. [6] expanded the PHM2017 dataset and introduced a new dataset, the HMC2019 dataset. This dataset has a separate class for figurative health mentions, which allows models that are trained using this dataset to learn to identify figurative language.

The authors of the HMC2019 dataset combined the self-mention and other-mention classes from the PHM2017 dataset into a single class, the positive class. Similarly, they combined the awareness and non-health classes into a single negative class. Additionally, they introduced a new class label, figurative, with the aim of creating a dataset that focused on figurative use cases of disease keywords. All of the samples were manually annotated to divide them into one of these three classes. A lot of work that has made use of the HMC2019 dataset has further combined the figurative and negative classes into a single negative class, thus creating a binary classification dataset. Samples of text from each class are shown in Table 2.

Twitter policy[1] does not allow tweets to be stored as plain text in public datasets, so the PHM2017 dataset only contained the tweet IDs. The authors of the HMC2019 dataset had to re-scrape the data using the IDs, which resulted in the loss of a large number of samples due to deleted tweets. At the time of download, only 5,497 samples could be collected. The six disease keywords from the PHM2017 dataset, along with four new ones, were then used to collect an additional 14,061 tweets.

[1] https://developer.twitter.com/en/developer-terms/policy

Table 2: Sample Documents from the HMC2019 Dataset

Examples Tweet	**Label**
i think this is migraine territory the scent of my unscented lotion is bother me	Figurative Mention
watch these hand picked migraine videos to help you learn coping skills and prepare ahead	Non-Health Mention
i had to work through a migraine today but i get to come in and leave earlier tomorrow so nice	Health Mention

3.3 Illness Dataset

The Illness Dataset was introduced by Karisani et al. [24] and contains 22,660 tweets mentioning four diseases: Alzheimer's, Parkinson's, Cancer and Diabetes. The data was collected from public posts made on the social media platform Twitter between 2018 and 2019. It is an imbalanced dataset. The class distribution for the dataset is shown in Fig. 1.

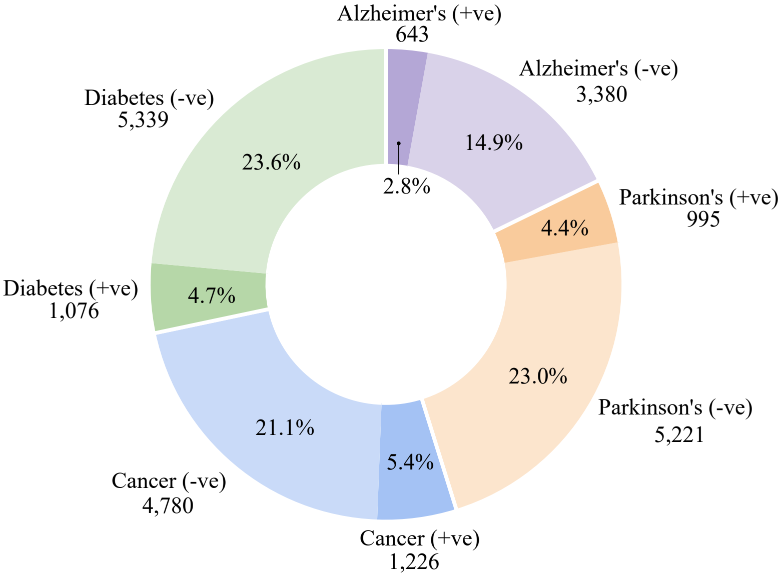


Figure 1: Class Distribution of the Illness Dataset

3.4 RHMD

The RHMD dataset was created by Naseem et al. [41] with the intention of contributing comparatively longer samples of text to the PHM domain. To this end, the dataset was created by collecting posts from Reddit using disease and symptom keyword searches. In total, the dataset consists of 10,015 posts across fifteen diseases. An additional benefit of the posts being collected from Reddit is that the text can be made publicly available, so none of the samples were lost. The RHMD dataset originally consisted of four class labels, positive, negative, figurative, and hyperbolic. Samples of text from each class are shown in Table 3.

Table 3: Sample Documents from the RHMD Dataset

Examples of Posts	**Label**
I have a book shopping addiction and I don't even have the attention span to read books.	Hyperbolic Health Mention
Grandpa is having a heart attack, we have to act fast!!!	Personal Health Mention
My aunt is marrying a sherrif. Their invitation nearly gave everyone a heart attack.	Figurative Health Mention
New Jersey Assembly passes bill approving marijuana for PTSD treatment.	Non-Personal Health Mention

4 Proposed Pipeline

There are several shortcomings to existing literature that can be improved upon. Proposals for possible improvements are discussed in this section.

4.1 Ensemble Techniques

Ensemble based approaches have repeatedly been proven to improve performance on a variety of tasks [2, 8]. As such, it can be expected that using these techniques in the PHM domain will also lead to significant improvements. In this section, we propose four possible ensemble techniques that we hope to explore in our future work.

4.1.1 Feature Averaging

For feature averaging, we simultaneously fine-tune multiple pre-trained models. Each of the models gives an output vector, and a new vector is created by taking the average of the output vectors of all the models. This new vector is then passed to the classifier network.

4.1.2 Concatenation

The process of implementing concatenation should be similar to that of feature averaging. However, instead of taking the average of the output vectors of all the models, we concatenate the vectors to create a larger one. This concatenated vector is then passed to the classifier network.

4.1.3 Max-Voting

For max-voting, we examine the results of different models on the same dataset. In cases where the models disagree on the correct output, the output chosen by the majority of the models is assumed to be the correct one.

4.1.4 Probability Averaging

Similar to max-voting, probability averaging also works with the fine-tuned models. In this case however, we work with the probability values assigned to the different possible classes by each of the models. The average of the probability values is calculated and the classification is performed based on these values.

4.2 Explainability

By nature, deep-learning algorithms such as those used in this paper are difficult to understand. The mechanisms via which the models adjust themselves to the provided input and learn to accurately make predictions are hidden inside a theoretical black box. This makes it difficult to integrate systems built on top of these algorithms into real-life applications. Sensitive use cases, such as those involving healthcare, require the systems to meet strict guidelines. Regulations such as those by the European Union[2] also ensure the right of users to demand a clear explanation behind the decisions made by such systems.

Explainable Artificial Intelligence (XAI) aims to provide these explanations by analysing the models and the factors that lead to the decisions they take. One possible approach to doing this is to use Shapley Additive Explanations (SHAP) [36]. The Shapley value is a concept from game theory that assigns a positive or negative value to the contributors involved in the final outcome of a solution. In terms of machine learning, this means assigning a Shapely value to each of the features from a sample that either contributed towards or against the final outcome. To calculate the Shapley value for a specific feature, we take every possible combination of features that involves the feature we are interested in and compare the difference between the output with the feature present and absent for each combination. The mean of the results is the Shapley value for the feature.

4.3 Resource Optimization

The most commonly used architecture in the PHM domain is the transformer architecture. Unfortunately, the models developed using this architecture tend to be quite large and computationally expensive to run. This makes it essential that we explore techniques that attempt to reduce this computational expense. However, no such techniques have been explored in existing literature. As such, there is a gap in research here that demands further work.

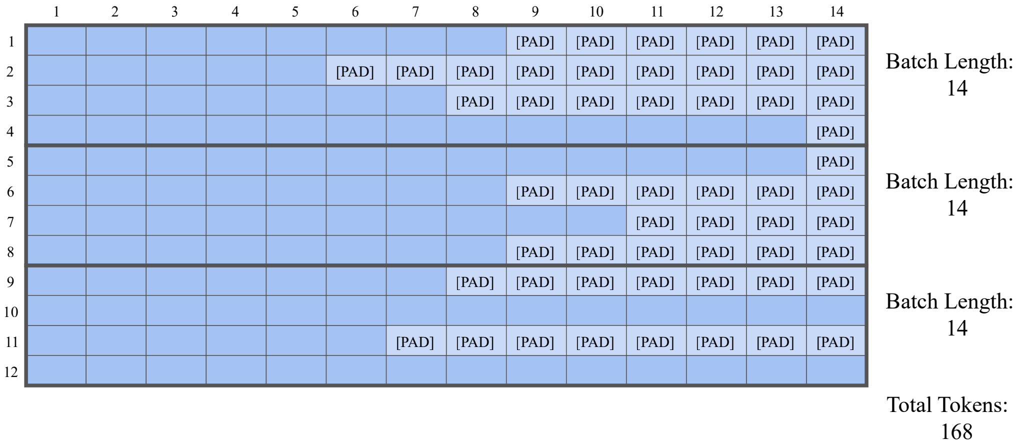
Here, we propose two possible approaches that we have studied for resource optimization which we hope to implement in our future work.

[2] https://gdpr.eu/

4.3.1 Dynamic Padding

One requirement of transformer models is that every array of tokens provided to it has to be the same length. This is due to the fact that transformer models perform matrix multiplications internally, and matrices need to have rows of the same length. However, real-life sentences vary in length, which in turn means that the array of tokens generated for the sentences will also vary in length. To be able to work with transformers, we are then forced to add padding to the ends of shorter sentences in order to make the array of tokens match the length of the largest one. This is as simple as adding zeros to the end of the original array of tokens.

The most naive method of adding padding is to simply take the longest sentence in a dataset and pad every other sentence to match that length, as shown in Fig. 2. This method adds a lot of overhead. Even though all the extra padding does not provide the model with any extra information, the model is still forced to process it. The HMC2019 dataset for example, has an average token length of just 34.30, but the maximum token length is 341, an increase of 10 times.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1									[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]
2						[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]
3								[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]
4														[PAD]
5														[PAD]
6									[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]
7											[PAD]	[PAD]	[PAD]	[PAD]
8									[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]
9								[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]
10														
11							[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]
12														

Batch Length: 14

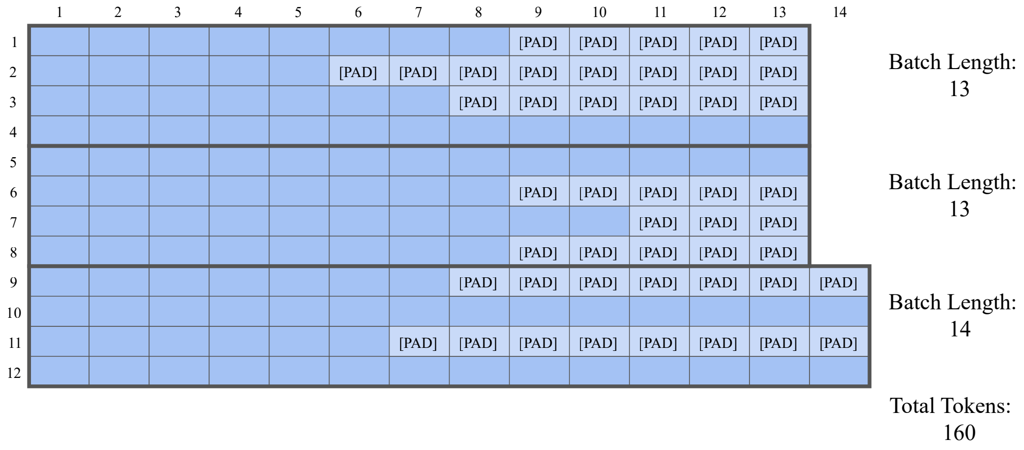
Batch Length: 14

Batch Length: 14

Total Tokens: 168

Figure 2: Token Lengths Under Normal Padding

The sentences we feed to the transformer-based models are more often than not fed in batches. The ideal scenario would be to give the model the entire dataset at once, but that is not practically possible due to memory limitations. Because of this, fixed-sized groups of data are provided to the model one by one. This fixed size is known as the batch size. Dynamic Padding involves reducing the amount of padding required by utilising the fact that the data is being provided in batches. The transformer models do not actually require that every sentence in the entire dataset be the same length. They only require that the sentences they are being given simultaneously be of the same length. This means that we do not need to pad sentences to the length of the longest sentence in the entire dataset, just to the length of the longest sentence in the batch being processed. This reduces the amount of padding the model must process, which in turn improves its training and inference times. Fig. 3 shows the change in the total number of tokens when dynamic padding is used.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1									[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	
2						[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	
3								[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	
4														
5														
6									[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	
7											[PAD]	[PAD]	[PAD]	
8									[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	
9								[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]
10														
11							[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]
12														

Batch Length: 13

Batch Length: 13

Batch Length: 14

Total Tokens: 160

Figure 3: Token Lengths Under Dynamic Padding

Along with dynamic padding, it is also recommended to use uniform-length batches. This involves sorting the tokens by size before dividing them into batches and adding padding. This results in sentences of similar lengths being grouped together, ensuring that we do not end up in a scenario where a large amount of padding must be added to every sentence in a batch just because there is a single long sentence in that batch. The amount of padding applied is the least it could possibly be in this scenario. Fig. 4 shows the change in the total number of tokens with dynamic padding and uniform-length batches implemented.

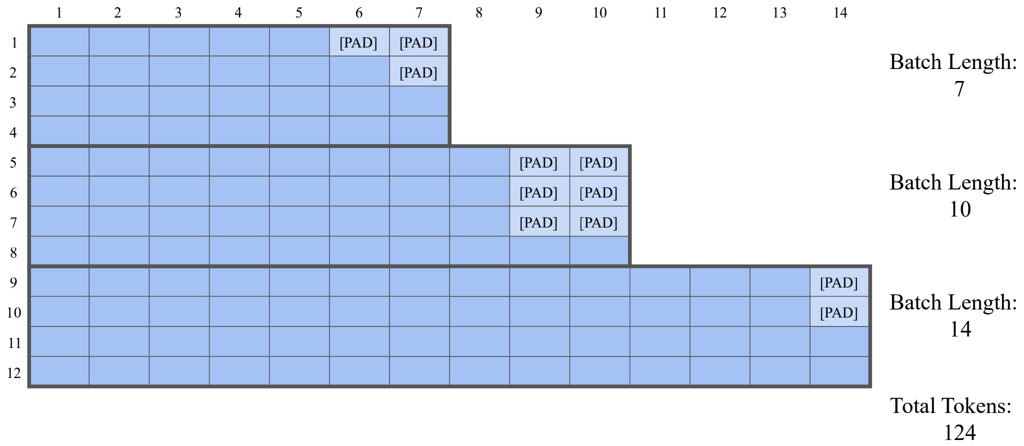


Figure 4: Token Lengths Under Dyanamic Padding with Uniform-Length Batches

In addition to the reduced training and inference times, using dynamic padding also allows us to use the largest size supported by the pre-trained models as the maximum token length. The high computational cost and memory requirements of processing the padded token arrays frequently force researchers to limit the maximum token length to a much smaller value. Previous work on the HMC2019 dataset, for example, used a maximum token size of 128 [27, 29]. Limiting the maximum token length in this manner results in sentences being truncated, which in turn results in the models being unable to utilise the full context provided by the sentence. This shortcoming affects the RHMD dataset in particular since the dataset contains a significant number of longer sentences. Since dynamic padding reduces the computational cost, the maximum token length can be set to a larger value, thus preventing most of the sentences from being truncated.

4.3.2 Gradient Checkpointing

Very large models with a lot of layers require a huge amount of memory to train. Some models are created by large corporations that are able to obtain commercial-grade processors with large amounts of memory to train such huge models on. However, the size of these models makes it impossible for the average researcher to even evaluate the models, let alone train or fine-tune them. Gradient checkpointing

[10] is a mechanism that attempts to reduce the amount of memory required by such large models, allowing us to use them on processors with limited memory. For a network with n layers, the memory consumption is reduced from $O(n)$ to $O(\sqrt{n})$ using this mechanism.

During the training process of a neural network, on every epoch, the model goes through a forward pass and then a backward pass. During the forward pass, the model calculates the output for each neuron based on the input to that neuron. The output from the final layer of neurons is compared to the target values using a loss function. The output from the loss function is then used to perform a backward pass through the model, where the model goes over each layer of neurons in the reverse direction, calculating the gradient values. These values are then used to update the weights of the neurons.

The most memory-intensive part of the whole process is the gradient calculation. The model is storing the output values of every neuron during the forward pass and then using those values again during the backward pass to calculate the gradient. This uses up a huge amount of memory for models with a large number of layers. Instead of storing the outputs during the forward pass, gradient checkpointing suggests that we discard them and re-calculate the values during the backward pass. This reduces the memory required to store the outputs during the forward pass.

There are various other considerations that must be made when using gradient checkpointing. It is not feasible to discard the output of every single layer, since doing so would result in an unacceptable increase in the time required to recalculate all the values. Instead, specific layers, called 'checkpoints', have their outputs stored while the others have their outputs discarded and recalculated. Additionally, some layers, such as dropout layers, behave differently each time a forward pass is performed on them. If we were to discard the output of a dropout layer, the value that was calculated during the forward pass, which is used to calculate the loss, would be different from the value that is calculated when we run the forward pass again during gradient calculation. This is because dropout layers randomly drop a certain percentage of the previous layer's weights. The specific weights that are dropped will most likely be different on each of the forward passes. Because of issues like this, care must be taken to checkpoint any layers or modules that contain layers that behave in a non-idempotent manner.

The one unavoidable drawback to using gradient checkpointing is that there is some additional computation required in order to recalculate the forward pass values during gradient calculation. The use of dynamic padding in combination with gradient checkpointing, however, reduces this computational overhead to a negligible amount.

5 Experimental Findings

As a starting point for our research, the effectiveness of a few renowned transformer-based natural language processing models on the Illness Dataset was explored. The use of transformer-based models has not previously been studied on this dataset, so our findings present important new information. In this chapter, the specifics of how the models were assessed are presented.

5.1 Feature Extractors

As feature extractors, three models, BERT [11], RoBERTa [33], and XLNet [54], were used. All of these models are based on the original transformer model proposed by Vaswani et al. [52], which was a Sequence-to-Sequence architecture. This architecture has an encoder section, which extracts features from input sequences, and a decoder section, which predicts the output from the features.

The models we are working with fall into two categories, Autoregressive (XLNet) and Autoencoding (BERT, RoBERTa). Autoregressive models are pre-trained to predict a new sequence from the current one. Autoencoding models are pre-trained by masking parts of the input sequence and making the model reconstruct it. Training in this way allows the models to gain an understanding of the language.

This in turn means that the models require a minimal amount of fine-tuning when they are used in downstream tasks.

5.2 Tokenization

Each of the feature extractors requires that the inputs be of the same shape. Input vectors for the input texts must be created in order to accomplish this. The corresponding tokenizers for each of the feature extractors were used to handle this vector generation, giving us the input tokens.

Each input token represents a word or a word fragment. If a token corresponds to a word fragment, meaning a complete word has multiple tokens, then related words will have some common tokens. If a token corresponds to a complete word, then related words will have token values that are very close.

[PAD] tokens are added to the end of shorter token sequences to make the sequences the same shape. Due to computational limitations, a fixed sequence length of 128 was used for our experiments. It was observed that for sample text from the dataset, token sequences are unlikely to be longer than this. Fig. 5 shows a histogram of the token lengths of the tokens generated by the RoBERTa model's tokenizer.

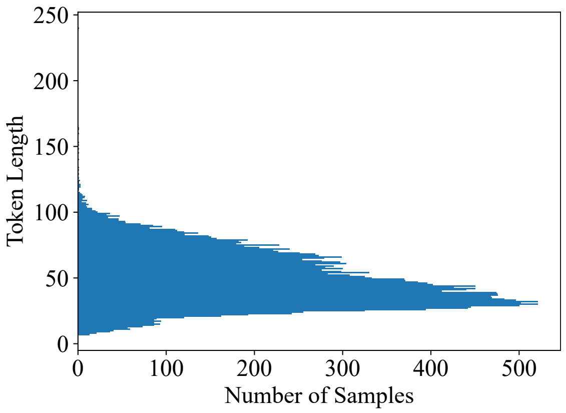


Figure 5: Token Lengths of the Illness Dataset

There are also some special tokens used by the tokenizer, such as the [SEP] token, which denotes the point where one input ends and another one begins, and the [UNK] token, which is used to mark a word the tokenizer has not seen before.

5.3 Classifier Network

The models being used were all pre-trained on a large corpus of text. This makes them effective feature extractors since they have a good grasp of the basic linguistic features. However, the text they were pre-trained on was generic, not specific to the task at hand. If we use a classifier network before using the models for a downstream task, the performance will be better than if we used the models as is [21]. Using a classifier network makes it possible to further adjust the pre-trained weights to the particular data being used. The architecture of the classifier network used in the experiments is shown in Fig. 6. This section was trained from scratch.

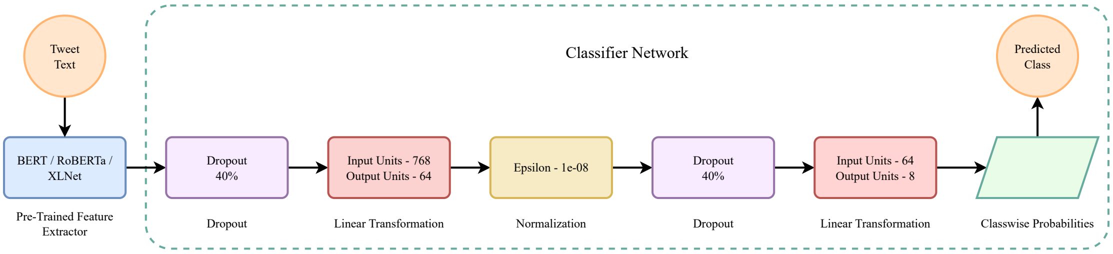


Figure 6: Classifier Network Architecture

Both dropout layers exist to prevent overfitting. The dropout rate being set to 40% means that a random 40% of the output units from the previous layer will be set to 0. This increases the overall robustness of the model. The dropout rate was determined by trial and error.

The normalisation layer normalises the activation values of the previous layer, bringing them to a similar scale. Gradient descent subsequently becomes more stable, cutting down on the time needed to train the model. The epsilon value used does not affect the model itself, but rather provides numerical stability for the internal calculations to avoid divisions by zero. As such, the exact value chosen is insignificant.

The linear transformation layers change the number of dimensions the model is dealing with. The first linear transformation layer reduces the dimensions with the purpose of reducing computational complexity. It brings down the number of dimensions from 768, which is the output size for all of the pre-trained models used, to 64. The second linear transformation layer exists to bring the 64 dimensions down to the 8 dimensions required as the final output. Each of the values from the 8 dimensions represents a different class. To be able to determine the probability that each of these classes is the correct one, a probability value must be assigned to them. The softmax layer translates the values from the 8 dimensions into probability values. From this, the class that has the highest probability is chosen as the prediction.

5.4 Experiment Setup and Hyperparameters

The experiments were carried out in Kaggle[3] with CUDA 11.4. We used an Intel Xeon CPU with 13 GB of RAM and an NVidia Tesla P100 GPU with 15.9 GB of Video RAM.

The entire dataset was first divided into the training, validation, and test sets in a 60:20:20 ratio. After each epoch of training on the training set, the model's performance was assessed using the validation set, and its weights were saved if performance improved. Each classifier underwent 15 epochs of training. Training beyond this was found to result in insignificant performance improvements. The test set was used for the final evaluation and remained unseen to the model during the training phase.

Due to the increased memory utilisation brought on by using larger batch sizes, the batch size had to be limited. Based on the hardware configuration available, it was determined that a batch size of 16 had an acceptable memory usage.

We used a learning rate of $1e-5$. A lower learning rate means the model takes smaller steps during the training phase, which allows it to better adjust the error

[3] https://www.kaggle.com/

values from the loss function. The loss function used in this study was categorical cross-entropy. However, such a low learning rate would normally mean that the model would take a larger number of epochs to converge to an acceptable extent. We were able to avoid this issue by using pre-trained models.

The first 20% of the training steps were used as warmup steps, meaning the learning rate was gradually increased from 0 to $1e - 5$ during this time. The reduced learning rate during the initial stages makes the learning process less volatile since the model is less likely to become misled [14].

At the end of each step of batch training, the weight values were optimised. This was accomplished by utilising the AdamW optimiser [34] with a weight decay of 0.01.

5.5 Result Analysis

We analysed the results obtained by applying each of the chosen feature extractors on the Illness Dataset. The takeaway from this analysis is discussed in this section.

5.5.1 Evaluation Metrics

The metrics chosen to evaluate the models were accuracy, precision, recall, F1 score, AUC-ROC and MCC.

Accuracy was chosen since it is a widely used evaluation metric. Unfortunately, for imbalanced datasets, such as the one used in our experiments, this metric does not give an accurate representation of the performance [3]. The precision and recall metrics can help deal with this and are popular evaluation metrics in their own right [1]. However, even these metrics have their share of issues, since they do not penalise incorrect outcomes. The F1 score metric is capable of giving an accurate representation of how well the models performed, taking into account all of these issues [2].

The Area Under the Receiver Operating Characteristic Curve (AUC-ROC) metric effectively measures how well a model is able to differentiate between classes [1, 39].

The Matthews Correlation Coefficient (MCC) metric measures how correlated the predicted and target values are. Both of these metrics evaluate the models from a completely different perspective than the other metrics, which is why they are included in the evaluation process [3].

5.5.2 Performance Comparison

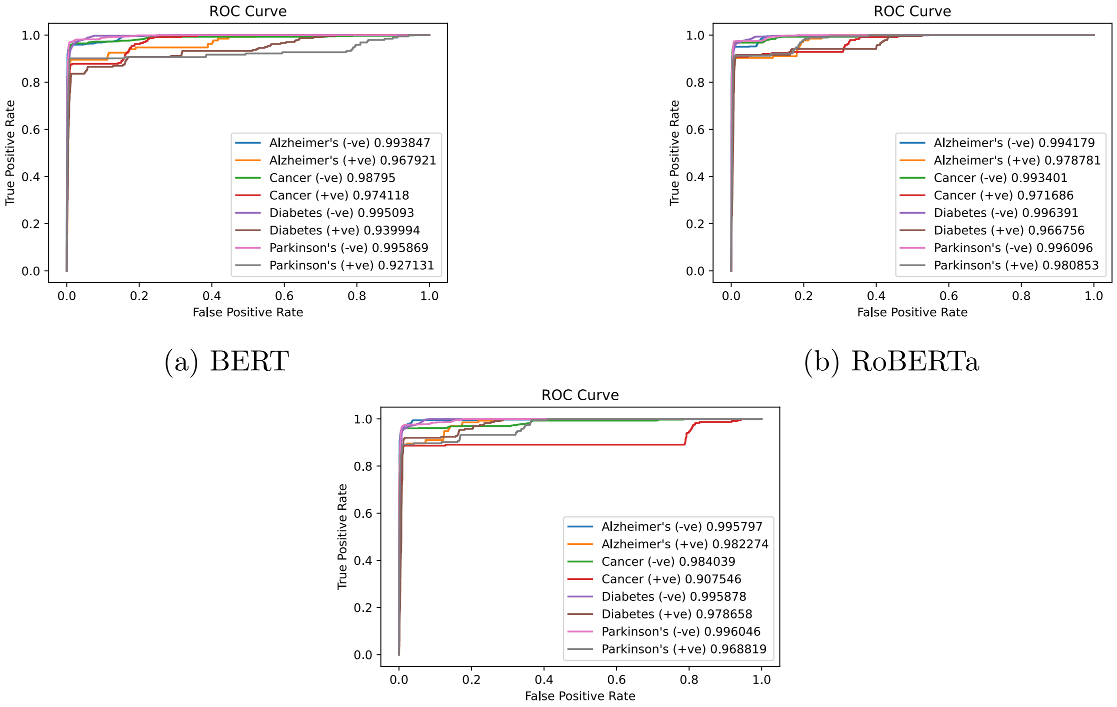
Table 4 shows a comparison of the results obtained by our classifier when working with each of the three feature extractors, alongside baseline classification models. The performance of the best results obtained in each metric is highlighted in bold. The baseline models considered in this comparison come from the two previous works that analysed their performance against the Illness Dataset. Karisani et al. [24], used the COCOBA model, a multi-view active learning model, while Karisani [22] used the CEPC model, which is a multiple-source unsupervised model.

The results make it clear that the RoBERTa-based classifier outperforms all the others. It is closely followed by the XLNet-based classifier, with the difference in performance between the two being minimal. The possible reasons behind this are further analysed in subsection 5.5.3.

Table 4: Experimental Results

Paper	**Model**	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**ROC-AUC**	**MCC**
Karisani et al. [24]	COCOBA	-	-	-	0.809	-	-
Karisani [22]	CEPC	-	0.786	0.841	0.811	-	-
Ours	BERT	0.937	0.938	0.937	0.938	0.986	0.923
	RoBERTa	**0.944**	**0.947**	**0.944**	**0.945**	**0.990**	**0.931**
	XLNet	0.941	0.943	0.941	0.942	0.986	0.928

The ROC curves for the BERT, RoBERTA and XLNet models are shown in Fig. 7a, 7b and 7c respectively. For all three models, we can see that the positive class for each disease has worse performance than the negative class. This phenomenon is further analysed in subsection 5.5.4.



(a) BERT

(b) RoBERTa

(c) XLNet

Figure 7: ROC Curves Produced by Fine-Tuned Classifier Networks

5.5.3 Performance Analysis

The results of this study support those of Yang et al. [54] and Liu et al. [33]. Yang et al. [54] confirmed that XLNet outperforms state-of-the-art models in a variety of natural language processing experiments, while Liu et al. [33] demonstrated that RoBERTa outperforms both BERT and XLNet on the GLUE benchmark. The improved pre-training procedure for these models is what accounts for the better performance. XLNet outperforms models like BERT because it is pre-trained on a larger amount of data and creates word permutations during the pre-training phase. On the other hand, RoBERTa was a development over BERT and was pre-trained on longer sequences, used larger batch sizes, and adopted dynamic masking, where the masked word changed dynamically during pre-training.

5.5.4 Error Analysis

The three models all performed better on the negative classes than they did on the corresponding positive classes, as shown in Fig. 7. The significant difference in the number of samples in the classes, as depicted in Fig. 1, explains this disparity. The extent of the difference between the sample sizes directly relates to the degree of difference in ROC-AUC scores.

To enhance performance on imbalanced datasets, previous studies have tested various strategies. This problem has been successfully addressed by randomly oversampling the smaller class [35, 48] or undersampling the larger class [4, 51]. The latter solution, however, may lead to overfitting or bias. Recently, few-shot learning has also been effectively used [49].

Table 5 provides samples of erroneous predictions made by the RoBERTa model. Analysis of the errors reveals that the model usually makes mistakes that involve mislabelled samples, multiple disease mentions and humorous posts using a first-person perspective. For example, the post 'They told you animal proteins are good for you but it causes cancer, diabetes, inflammation, kidney stones, etc.' is correctly predicted as a negative health mention, but the label and prediction disagree on the disease name since multiple diseases are mentioned.

Table 5: Samples of Erroneous Predictions

Label	**Prediction**	**Sample Text**
Parkinson's (Positive)	Parkinson's (Negative)	Steps to Better Walking Even With #Parkinson's #Disease [url]
Cancer (Negative)	Diabetes (Negative)	They told you animal proteins are good for you but it causes cancer, diabetes, inflammation, kidney stones, etc.
Alzheimer's (Negative)	Alzheimer's (Positive)	Old McDonald had alzheimer's Have you any wool....

6 Conclusion and Future Works

In this thesis, we have explored the use of transformer-based architectures on one of the datasets in the PHM domain and shown that their use leads to improved results. Comparisons with existing work show that the results we obtained were state-of-the art. Additionally, we have also discussed possible approaches that we plan to undertake in our future work, including the use of ensemble architectures, explainability of the outputs of the existing architectures and resource optimization techniques.

One additional possibility for future work is the development of a domain-specific model. It would be useful to take a lightweight transformer-based architecture, such as ALBERT [31] or DistilBERT [50] and train the model from scratch on data related to social media and health specifically. Theoretically, this should result in significantly better performance than all of the existing work. Combined with the optimization techniques proposed in this thesis, the domain-specific model should be a significant contribution. However, the time and resources required to train even lightweight architectures from scratch presents a challenge that is difficult to overcome, making this a daunting task.

References

[1] Sabbir Ahmed, Md Bakhtiar Hasan, Tasnim Ahmed, Md Redwan Karim Sony, and Md Hasanul Kabir. Less is more: Lighter and faster deep neural architecture for tomato leaf disease classification. *IEEE Access*, 2022.

[2] Tasnim Ahmed, Mohsinul Kabir, Shahriar Ivan, Hasan Mahmud, and Kamrul Hasan. Am i being bullied on social media? an ensemble approach to categorize cyberbullying. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 2442–2453. IEEE, 2021. doi: 10.1109/BigData52589.2021.9671594.

[3] Tasnim Ahmed, Shahriar Ivan, Mohsinul Kabir, Hasan Mahmud, and Kamrul Hasan. Performance analysis of transformer-based architectures and their ensembles to detect trait-based cyberbullying. *Social Network Analysis and Mining*, 12(1):1–17, 2022. doi: 10.1007/s13278-022-00934-4.

[4] Ashish Anand, Ganesan Pugalenthi, Gary B Fogel, and PN Suganthan. An approach for classification of highly imbalanced data using weighting and undersampling. *Amino acids*, 39(5):1385–1391, 2010.

[5] Eiji Aramaki, Sachiko Maskawa, and Mizuki Morita. Twitter catches the flu: detecting influenza epidemics using twitter. In *Proceedings of the 2011 Conference on empirical methods in natural language processing*, pages 1568–1576, 2011.

[6] Rhys Biddle, Aditya Joshi, Shaowu Liu, Cecile Paris, and Guandong Xu. Leveraging sentiment distributions to distinguish figurative from literal health reports on twitter. In *Proceedings of The Web Conference 2020*, pages 1217–1227, 2020. doi: 10.1145/3366423.3380198.

[7] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.", 2009.

[8] Pete Burnap and Matthew L Williams. Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. *Policy & internet*, 7(2):223–242, 2015.

[9] Liangzhe Chen, KSM Tozammel Hossain, Patrick Butler, Naren Ramakrishnan, and B Aditya Prakash. Syndromic surveillance of flu on twitter using weakly supervised temporal topic models. *Data mining and knowledge discovery*, 30(3):681–710, 2016. doi: 10.1007/s10618-015-0434-x.

[10] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016. doi: 10.48550/arXiv.1604.06174.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, 2019. doi: 10.18653/v1/N19-1423.

[13] Samah Jamal Fodeh, Mohammed Al-Garadi, Osama Elsankary, Jeanmarie Perrone, William Becker, and Abeed Sarker. Utilizing a multi-class classification approach to detect therapeutic and recreational misuse of opioids on twitter. *Computers in biology and medicine*, 129:104132, 2021. doi: 10.1016/j.compbiomed.2020.104132.

[14] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. doi: 10.48550/arXiv.1706.02677.

[15] Adith Iyer, Aditya Joshi, Sarvnaz Karimi, Ross Sparks, and Cecile Paris. Figurative usage detection of symptom words to improve personal health mention detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1142–1147, 2019. doi: 10.18653/v1/P19-1108.

[16] Shaoxiong Ji, Matti Hölttä, and Pekka Marttinen. Does the magic of bert apply to medical code assignment? a quantitative study. *Computers in Biology and Medicine*, 139:104998, 2021. doi: 10.1016/j.compbiomed.2021.104998.

[17] Keyuan Jiang, Ricardo Calix, and Matrika Gupta. Construction of a personal experience tweet corpus for health surveillance. In *Proceedings of the 15th workshop on biomedical natural language processing*, pages 128–135, 2016. doi: 10.18653/v1/W16-2917.

[18] Keyuan Jiang, Shichao Feng, Qunhao Song, Ricardo A Calix, Matrika Gupta, and Gordon R Bernard. Identifying tweets of personal health experience through word embedding and lstm neural network. *BMC bioinformatics*, 19 (8):67–74, 2018. doi: 10.1186/s12859-018-2198-y.

[19] Aditya Joshi, Sarvnaz Karimi, Ross Sparks, Cécile Paris, and C Raina Macintyre. Survey of text-based epidemic intelligence: A computational linguistics perspective. *ACM Computing Surveys (CSUR)*, 52(6):1–19, 2019. doi: 10.1145/3361141.

[20] Aditya Joshi, Ross Sparks, Sarvnaz Karimi, Sheng-Lun Jason Yan, Abrar Ahmad Chughtai, Cecile Paris, and C Raina MacIntyre. Automated monitoring of tweets for early detection of the 2014 ebola epidemic. *PloS one*, 15(3): e0230322, 2020. doi: 10.1371/journal.pone.0230322.

[21] Mohsinul Kabir, Tasnim Ahmed, Md Bakhtiar Hasan, Md Tahmid Rahman Laskar, Tarun Kumar Joarder, Hasan Mahmud, and Kamrul Hasan. Deptweet: A typology for social media texts to detect depression severities.

Computers in Human Behavior, 139:107503, 2023. doi: 10.1016/j.chb.2022.107503.

[22] Payam Karisani. Multiple-source domain adaptation via coordinated domain encoders and paired classifiers. *arXiv e-prints*, pages arXiv–2201, 2022.

[23] Payam Karisani and Eugene Agichtein. Did you really just have a heart attack? towards robust detection of personal health mentions in social media. In *Proceedings of the 2018 World Wide Web Conference*, pages 137–146, 2018. doi: 10.1145/3178876.3186055.

[24] Payam Karisani, Negin Karisani, and Li Xiong. Contextual multi-view query learning for short text classification in user-generated data. *arXiv preprint arXiv:2112.02611*, 2021.

[25] Makoto P Kato, Kazuaki Kishida, Noriko Kando, Tetsuya Saka, and Mark Sanderson. Report on ntcir-12: The twelfth round of nii testbeds and community for information access research. In *ACM SIGIR Forum*, volume 50, pages 18–27. ACM New York, NY, USA, 2017. doi: 10.1145/3053408.3053413.

[26] Pervaiz Iqbal Khan, Imran Razzak, Andreas Dengel, and Sheraz Ahmed. Improving personal health mention detection on twitter using permutation based word representation learning. In *International Conference on Neural Information Processing*, pages 776–785. Springer, 2020. doi: 10.1007/978-3-030-63830-6_65.

[27] Pervaiz Iqbal Khan, Imran Razzak, Andreas Dengel, and Sheraz Ahmed. A novel approach to train diverse types of language models for health mention classification of tweets. In *Artificial Neural Networks and Machine Learning – ICANN 2022*, pages 136–147, 2022. doi: 10.1007/978-3-031-15931-2_12.

[28] Pervaiz Iqbal Khan, Imran Razzak, Andreas Dengel, and Sheraz Ahmed. Performance comparison of transformer-based models on twitter health mention classification. *IEEE Transactions on Computational Social Systems*, pages 1–10, 2022. doi: 10.1109/TCSS.2022.3143768.

[29] Pervaiz Iqbal Khan, Shoaib Ahmed Siddiqui, Imran Razzak, Andreas Dengel, and Sheraz Ahmed. Improving health mention classification of social media content using contrastive adversarial training. *IEEE Access*, 10:87900–87910, 2022. doi: 10.1109/ACCESS.2022.3200159.

[30] Alex Lamb, Michael Paul, and Mark Dredze. Separating fact from fear: Tracking flu infections on twitter. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 789–795, 2013.

[31] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[32] Md Tahmid Rahman Laskar, Xiangji Huang, and Enamul Hoque. Contextualized embeddings based transformer encoder for sentence similarity modeling in answer selection task. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 5505–5514, 2020.

[33] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. doi: 10.48550/arXiv.1907.11692.

[34] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[35] Jiaqi Lun, Jia Zhu, Yong Tang, and Min Yang. Multiple data augmentation strategies for improving performance on automatic short answer scoring. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13389–13396, 2020.

[36] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30:4765–4774, 2017.

[37] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. doi: 10.48550/arXiv.1301.3781.

[38] Saif Mohammad. Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 english words. In *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 174–184, 2018. doi: 10.18653/v1/P18-1017.

[39] Md. Samin Morshed, Sabbir Ahmed, Tasnim Ahmed, Muhammad Usama Islam, and A. B. M. Ashikur Rahman. Fruit quality assessment with densely connected convolutional neural network, 2022. URL https://arxiv.org/abs/2212.04255.

[40] Martin Müller, Marcel Salathé, and Per E Kummervold. Covid-twitter-bert: A natural language processing model to analyse covid-19 content on twitter. *arXiv preprint arXiv:2005.07503*, 2020. doi: 10.48550/arXiv.2005.07503.

[41] Usman Naseem, Jinman Kim, Matloob Khushi, and Adam G Dunn. Identification of disease or symptom terms in reddit to improve health mention classification. In *Proceedings of the ACM Web Conference 2022*, pages 2573–2581, 2022. doi: 10.1145/3485447.3512129.

[42] Usman Naseem, Jinman Kim, Matloob Khushi, and Adam G Dunn. Robust identification of figurative language in personal health mentions on twitter. *IEEE Transactions on Artificial Intelligence*, pages 1–1, 2022. doi: 10.1109/TAI.2022.3175469.

[43] Usman Naseem, Byoung Chan Lee, Matloob Khushi, Jinman Kim, and Adam Dunn. Benchmarking for public health surveillance tasks on social media with a domain-specific pretrained language model. In *Proceedings of NLP Power! The First Workshop on Efficient Benchmarking in NLP*, pages 22–31, 2022. doi: 10.18653/v1/2022.nlppower-1.3.

[44] Robert T Olszewski. Bayesian classification of triage diagnoses for the early detection of epidemics. In *Flairs conference*, pages 412–416, 2003.

[45] Michael Paul and Mark Dredze. You are what you tweet: Analyzing twitter for public health. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 5, pages 265–272, 2011. doi: 10.1609/icwsm.v5i1.14137.

[46] Michael J Paul and Mark Dredze. Social monitoring for public health. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 9(5):1–183, 2017. doi: 10.1007/978-3-031-02311-8.

[47] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. doi: 10.3115/v1/D14-1162.

[48] Siyuan Qiu, Binxia Xu, Jie Zhang, Yafang Wang, Xiaoyu Shen, Gerard De Melo, Chong Long, and Xiaolong Li. Easyaug: An automatic textual data augmentation platform for classification tasks. In *Companion Proceedings of the Web Conference 2020*, pages 249–252, 2020.

[49] Anthony Rios and Ramakanth Kavuluru. Few-shot and zero-shot multi-label learning for structured label spaces. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2018, page 3132. NIH Public Access, 2018.

[50] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[51] Parinaz Sobhani, Herna Viktor, and Stan Matwin. Learning from imbalanced data using ensemble methods and cluster-based undersampling. In *Interna-*

tional Workshop on New Frontiers in Mining Complex Patterns, pages 69–83. Springer, 2014.

[52] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30:5998–6008, 2017.

[53] Chen-Kai Wang, Onkar Singh, Zhao-Li Tang, and Hong-Jie Dai. Using a recurrent neural network model for classification of tweets conveyed influenza-related information. In *Proceedings of the International Workshop on Digital Disease Detection Using Social Media 2017 (DDDSM-2017)*, pages 33–38, 2017.

[54] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32:5753–5763, 2019.

[55] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489, 2016. doi: 10.18653/v1/N16-1174.