Tuning Personal Health Mention Detection BERT Model within Social Media Data with a Genetic Algorithm

VICTORIA ARMSTRONG, Queen's University, Canada
EMILY MEDEMA, Queen's University, Canada
DEBBIE WANG, Queen's University, Canada

The detection of personal health mentions (PHM) in tweets is a prevalent and often discussed problem within computer science. There are many different approaches to accurately classifying the tweets and dealing with the inherent issues present in tweet data such as low word count and figurative language. One particularly interesting approach is utilizing a transformer model, such as BERT, to reduce the issues with low word count and a relatively low number of tweets meeting the criteria. The main issue inherent with BERT is the tuning of the model requiring expertise and containing a large amount of variance. However, if one were to represent tuning as a pre-defined search space, a Genetic Algorithm (GA) could be used to tune BERT, resulting in more fit and robust solutions and a less intensive tuning step. On the whole, we propose that introducing GAs as a tuning algorithm for the BERT model will result in less variance and better classifications of PHM. As a result of our experiments, we have found that even short training times have improved the PHM classifications of a zero-shot BERT model classification task.

CCS Concepts: • **General and reference** → Performance; • **Information systems** → *Social networks*; • **Computing methodologies** → **Genetic algorithms**; *Bio-inspired approaches*; *Natural language processing*.

Additional Key Words and Phrases: datasets, BERT, natural language processing, tuning, personal health mention detection, tweets

ACM Reference Format:
Victoria Armstrong, Emily Medema, and Debbie Wang. 2023. Tuning Personal Health Mention Detection BERT Model within Social Media Data with a Genetic Algorithm. 1, 1 (June 2023), 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Project code can be found at the following Github Repository: PHM Evolutionary Optimization

1 PROBLEM DESCRIPTION

Due to the increasing usage of social media and the corresponding growth of data, there has been more and more research into monitoring and analyzing social media data for public health information and insight [7]. The use of this data has the potential to greatly improve public health but there are numerous potential issues, such as biases in the data, the amplifications of those biases through particular models, issues with parsing the language, and slang used [2, 7]. Despite these issues, social media data has already been used in many public health applications from tracking the spread of influenza to monitoring the public's reactions to vaccinations [7].

Authors' addresses: Victoria Armstrong, victoria.armstrong@queensu.ca, Queen's University, 21-25 Union St, Kingston, Ontario, Canada, K7L 3N5; Emily Medema, emily.medema@queensu.ca, Queen's University, 21-25 Union St, Kingston, Ontario, Canada, K7L 3N5; Debbie Wang, debbie.wang@queensu.ca, Queen's University, 21-25 Union St, Kingston, Ontario, Canada, K7L 3N5.

There is a big push to use social media data for public health surveillance over more traditional methods as it is more scalable and has the potential for near-real-time responsiveness [7]. For optimal use of this data, it is very important that we can classify whether a post is a personal health mention or not. Any misclassification regarding personal health mentions can impact and skew any of the conclusions drawn. Through our algorithm, we want to accurately identify posts that contain a mention of a specific disease or condition and a person who is affected, and exclude posts that are expressing general concern or are spreading awareness about a disease or condition [7]. For example, in Figure 1, we want to be able to discern between the tweet where the person is mentioning having a stroke and the tweet that the user is quote tweeting, which we define as an awareness tweet.

Fig. 1. A Self-mention Tweet Quote Tweeting an Awareness Tweet

There are a few challenges in using social media posts that we must consider. First, there is a relatively low amount of training data with relatively high parsing complexity, especially for tweets (low word counts and common use of memes, jokes, and sarcasm). To counteract this, [7] used WESPAD which "learns to partition the word embeddings space to more effectively generalize from few training examples, and to distort the embeddings space to more effectively separate examples of true health mentions from the rest." However, for our project, we are taking inspiration from Madukwe et al. and using a GA to tune a Bidirectional Encoder Representations from Transformer (BERT) model on the PHM tweet data [4, 12]. In particular, we will be tuning which layers we are updating from BERT, single or dual fully connected layers, dropout rates, internal activation type, the number of inner layers, and final activation function. We believe that evolving the selection of these parameters will increase classification accuracy.

2 LITERATURE REVIEW

The paper "A GA-Based Approach to Fine-Tuning BERT for Hate Speech Detection" published in IEEE in 2020 by Madukwe et al. utilized a combination of genetic algorithms (GAs) and the BERT language model to improve the

classification of hate speech detection within tweets. We will be taking a similar approach to the detection of personal health mentions (PHM) in tweets. Using a GA approach to tuning the BERT language mode can help reduce the large amount of inherent variance in the tuning of BERT and other contextual word embedding models [12]. This variance is due to there being several factors that could be changed to optimize the model and as tuning is typically done manually and requires expertise and a lot of computing power, it tends to have high variance in the results [12]. Therefore, one potential way to reduce the intensiveness of the task is to utilize a GA to automatically find a near-optimal fine-tuned architecture for the BERT model we will be using for PHM detection [12].

As mentioned before, PHM detection has been done with other models. Karasani et al. developed their model, WESPAD, to perform PHM detection [7]. WESPAD combines lexical, syntactic, word embedding-based, and context-based features and through this setup can train on a smaller amount of data than normal and performs well in comparison to baseline and state-of-the-art models [7]. In particular, WESPAD outperforms CNN, LTSM, and FasText models on this data [7]. However, other people have used the BERT model - a model not tested against in the Karasani paper - to perform PHM detection and related analysis on the data. For instance, in the paper by Aduragba et al., a BERT model is used on the PHM data from the Karasani paper and other tweet datasets to incorporate emotions into PHM detection [1]. Similarly, as seen in the article written by Naseem et al., BERT has been used in conjunction with a Bi-LTSM to detect figurative language and use that detection to more accurately classify PHM in tweets [13]. There has been continual work in regards to personal health mention detection from work with Figurative Language detection in conjunction with CNNs, to solely CNN detection, to integrating Permutation Based Word Representation Learning [6, 8, 11]. We will be focusing on transformer models (BERT in particular), as transformers have been shown to result in state-of-the-art performance and deal well with less data [6]. In addition to focusing solely on BERT, we will be using the aforementioned method of tuning utilizing a GA. As far as we are aware, this has not been done in terms of personal health mention detection and could be incredibly useful in this field of research.

Contextualized embeddings such as CoVe, ELMo, OpenAI GPT, and BERT [4] are currently one of the major NLP paradigms and models [12]. In particular, the BERT model is a language representation model developed by Google. BERT was designed for pre-training bidirectional representations from unlabeled text which result in a model that can be fine-tuned with just one additional output layer, in this case, fine-tuning on the PHM data [4]. Specifically, BERT utilizes the pre-training done on the Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) tasks as a starting point for the downstream or fine-tuning on the domain-specific data [12]. It is important to note that different encoder layers in the BERT architecture capture different semantic and syntactic information. This is especially important in the fine-tuning we will be doing with the GA as the lower layers will most likely contain generic information that may not be as useful [12].

Genetic Algorithms - a population-based biologically inspired optimization method from the family of evolutionary algorithms (EAs) - can be used to automatically search a predefined search space [12, 14]. As tuning a model can be represented as a predefined search space, we can therefore use a GA to reduce the labour required for tuning by utilizing a GA to do the searching and increase the chances of finding a global optimum [12]. The overall approach to GAs is to use genetics-inspired operators such as mutation, crossover, and selection to explore and then exploit possible solutions to get to the optimum [12]. GAs have historically been used to tune fuzzy network controllers as well as more general controller tuning and even for agent-based models [3, 5, 10]. It seems to be a natural extension to use an algorithm designed to find the optimal solution in a predefined search space for tuning a model.

3 DATA

For the data, we chose two datasets to work with, the PHM2017 dataset [7] and the FLU2013 dataset [9]. We will be following the PHM2017 dataset's classification method, with 0 being not related, 1 being awareness, 2 being others, and 3 being self. The classification not related means that the tweet is unrelated to the disease being described. As an example, the tweet: "HA @ twitter catching the bird flu" is labelled as an Influenza mention, but has nothing to do with Influenza in actuality. The Awareness classification means that the tweet is raising awareness for the disease being mentioned. The classification of Others means someone other than the author of the tweet was affected by the disease being mentioned. Self-classification is the opposite of the prior, where the author is the one being affected by the disease.

While these classifications were already defined for PHM2017, for the FLU2013 dataset the labelling had to be adjusted to match what we defined. Initially, the FLU2013 was separated into three different datasets referring to the classifications of not related versus related, infection versus awareness, and others versus self (with a label of either 0 or 1 for each classification). These datasets were further subsetted for each year the data was collected, specifically in 2009 and 2012. The initial intuition was to combine all the tweets for a given year. However, after doing so with the 2009 flu data, we recognized that there were less than 40 such tweets that fit this condition. Following this, we decided to combine the data on our own. In the not related vs related datasets, we removed tweets that were identified as being related, with the intuition being that if they were related, then they would appear in the latter categories regardless, and here we were able to keep the label as 0. In the infection versus awareness category, infection-classified tweets were removed as this would be covered by the latter category of self versus others. Here the awareness label of 1 was able to be kept as well. In the others versus self dataset, all tweets were kept, but the others classification was changed from 0 to 2, and the self-label was changed from 1 to 3.

In order to work with both datasets, we combined the FLU2013 and PHM2017 datasets into one, duplicates were removed, and only the first instance of the tweet was kept.

There are a few challenges in terms of using these datasets. First of all, from the given datasets, PHM2017 has a fairly low PHM count, thus leading to an unbalanced dataset. While adding the FLU2013 does inflate these numbers by a bit, the non-related and awareness classifications still have much more than the self and others health mentions, as shown in Figure 2.

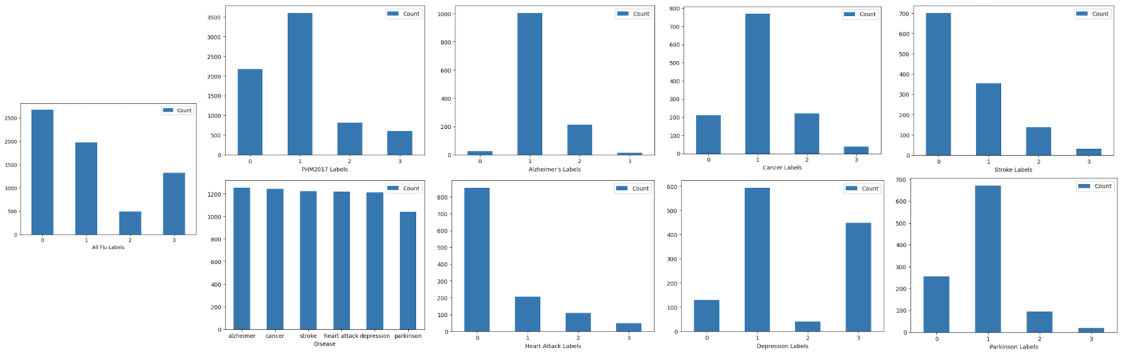

Fig. 2. Data distribution charts from FLU2013 and PHM2017.

In addition to there being an imbalance in the original dataset, many tweets were unavailable to us. This could be because the original tweet is no longer available (i.e. likely deleted) or because the tweets are protected, due to privacy settings. This further decreased the number of tweets we were able to use. Another challenge faced in this task is just

how these different disease names can be used in the common vernacular. For example, stroke can be used to mention someone having a stroke or stroking a pet or someone's hair. While this makes the task of classification harder, it also makes the task at hand interesting and a worthwhile challenge to take on as if we can accurately classify an unbalanced dataset with this approach it could be useful in other areas of study.

4 EVOLUTIONARY ALGORITHM DESIGN

4.1 Algorithm Description

In terms of our evolutionary computation technique, we decided to use a GA. While GAs may be slower than other evolutionary computation techniques, given our representation method and the fact that we are trying to optimize a discrete function optimization problem, GAs allow us to search and define a predefined space that we feel would serve our purposes. Additionally, [12] has shown this to be a possible and effective method of tuning a BERT model for hate-speech detection (a similar problem to personal health mention detection). Following Table IV [12] in their paper and our knowledge of Simple GA's (SGA) we created the technical sketch seen in Table 1.

Technical Aspect	**Choice**
Representation	Binary string
Mutation	Bit flip
Recombination	Uniform crossover
Parent selection	tournament (size = 10)
Survivor selection	$\mu + \lambda$
Fitness	F1 score
Population size	20
No. of generations	20

Table 1. Technical Sketch of EA

The representation used for the chromosome in our GA is a 28-bit long binary string to symbolize the different choices we can make in BERT.

Mutation is a method of introducing diversity into a population with random changes in an individual. For the mutation method, bit-flip is standard in GA. Knowing that it would work with our representation of choice, we decided to keep the mutation method.

Recombination, or crossover, is the method by which how parents mix or combine chosen parents to produce offspring. Although the original Simple Genetic Algorithm (SGA) has a recombination method of one-point crossover, there is positional bias in any sort of n-point crossover. To avoid this, we instead chose to implement uniform-crossover. While in uniform-crossover there is distributional bias introduced - meaning one is less able to tell which child is more similar to a certain parent - we decided that for our purposes distributional bias rather than positional was better for our problem.

In the SGA, the parent selection method is the roulette wheel method. However, [12] specified using a tournament method with a size of 10, so for now that is what we have implemented. We have also included in our parent selection file more methods such as multi-pointer selection (MPS), which is a form of the roulette wheel method, as well as random uniform in case we would like to experiment more in the future with different methods.

Survivor selection is the method by which the next generation is chosen. In the SGA technical sketch, survivor selection is described as "generational", and [12] used "elitism" instead. Knowing that we would like to keep the best possible solutions, we opted for the "elitism" option instead and implemented $\mu + \lambda$ survivor selection.

A fitness value provides a numeric way of interpreting how good a given individual is. The functions for calculating fitness values are problem specific, and for our purposes, we decided to use F1-scores rather than accuracy. This was

Feature to Tune	Label	Bitstring Position	Possible Values
BERT Layer	a	0 - 3	integer between 0 and 13
Fine-Tuning Architecture	b	4	0 or 1
Dropout	c	5 - 11	decimal between 0 and 1
Intermediate Layer Size	d	12 - 25	integer between 0 and 16383
Internal Activation Neuron	e	26	0 or 1
Final Activation Neuron	f	27	0 or 1

Table 2. The features to tune, the label corresponding to Figure 3, positions in the bitstring (0 indexed, inclusive) that encode this feature, and the possible architecture values these bitstrings encode.

mainly because, as mentioned earlier, the data we will be using is unbalanced. With unbalanced data it is easy to get a false high accuracy, it is harder to do so with the F1-score and as such we chose that as our fitness.

The termination criteria for our model is hitting the threshold of the maximum number of generations that we set. The population size and the number of generations were both from [12], and seem to be chosen arbitrarily, so this is certainly something else that we could experiment more with as well.

4.2 Fine-Tuning with Bert

To serve as our evaluation function (assessing the fitness of the model) for our genetic algorithm, we will fine-tune a BERT model. As referenced earlier in the literature review section, we use the BERT model as an encoder transformer. We selected a number of hyperparameters to fine-tune the pre-trained LLM and then evaluate its success on our labelled dataset, which we then convert into a fitness value.

4.2.1 BERT Implementation. While many flavours of the BERT encoder exist, we chose to use `bert-based-uncased`. In this case, the BERT model does not make distinctions between capitals, for example, 'english' and 'English' would be considered the same word embedding. We selected this model because tweets are not typically constructed following traditional sentence structure, so removing the sensitivity to sentence casing was desirable. We implemented the model fine-tuning in Python 3 using the Pytorch library. Before training, the data was cleaned and separated into the appropriate input and label columns and stored in a Pandas DataFrame that was exported to a pickle file. It is important that this step was completed prior to any GA training, as we did not want the time overhead that stems from file I/O being repeated for every candidate solution in the population. We loaded the pre-trained BERT model and used Pytorch to add on the classifier head based on the specification that comes from our GA candidate solutions. Figure 3 shows the break up of the bitstring and the partition labels correspond to the labels shown in Table 2 with the parameters being tuned, the bitstring cutoffs and the possible values. For the BERT layer representation, 4 bits represent values above the allowable range, so anything above 13 was set to a default value of 13. Dropout values must exist between 0 and 1, so they were scaled by the max allowable value to restrict the value to the correct bounds.

Fig. 3. Example of the candidate solution where different portions of the bitstring encode different architecture values.

4.2.2 Evaluating Model Fitness. As our primary objective is to determine the best hyperparameters for our fine-tuned model, we needed to determine a way to measure the fitness of our model. We eliminated accuracy as a fitness measure

as we don't have a completely balanced dataset and it's possible to get a decent accuracy with a null-class prediction with unbalanced data sets. Precision and recall alone were also eliminated for similar reasoning. We settled on using the F1-score as it is the harmonic mean of the precision and recall, allowing us to focus on maximizing both instead of needing to select one. Values for F1-score exist in [0,1] where 1 is a perfect precision and recall. Because we have a multi-class problem, we use the weighted F1-score that's included in Scikitlearn's metrics. Algorithm 1 shows how we train our model to produce our fitness metric.

Algorithm 1: Finetuned BERT Fitness

```
Result: fitness value for an individual in the population
parse bitstring into torch parameters;
load pretrained BERT model;
load and tokenize data;
freeze desired number of layers;
while current-epoch ≤ num-epochs do
    train();
    evaluate();
end
predictions = model.predict(data);
f1 = f1-score(predictions);
return f1;
```

5 EVOLUTIONARY ALGORITHM RESULTS

To assess our approach, we trained our genetic algorithm for 20 generations and then assessed its ability to classify tweets into 4 categories: unrelated, awareness, other health mention, and personal health mention. We evaluate the model on a set of unseen testing data. We repeated our training procedure 5 times to get aggregate results, as GAs are non-deterministic. We also construct a basic zero-shot classification model using BERT to compare as a baseline.

5.1 Experiments

We trained our genetic algorithm on a high-performance Linux server running Ubuntu, with 2 ADM Epyc 2.6/3.3GHz CPU cores, and a single A40 (48G) GPU accelerator. We conducted baseline zero-shot-classification using a MacBook Air 2019 running Ventura 13.0 with a 2.6 GHz 6-Core Intel Core i7. Similarly to [12], we train our GA for 20 generations, each including 10 epochs of training for every candidate solution. We split our data into a 70% training set, and 15% for each validation and testing set. This split occurs randomly for each candidate solution. In our zero-shot classification task, we select 15% of the data randomly as our testing set, to match the testing set size used in our GA.

5.2 Results

We ran our GA 5 times and found that on average, the best results were obtained from using the parameters listed in Table 3. The mean best fitness (MBF), i.e. F1-score, was 0.784. The number of solutions (SR) with F1-scores above the threshold 0.5 was 100%. The zero-shot classification task achieved an accuracy of 24.04% and a weighted F1-score of 0.228. In all 5 instances, BERT Layer 11 was selected, additionally ReLU and Softmax were always selected as the

internal and final activation neurons. A single layer fully connected network was selected 3 times, where a dual layer fully connected network was selected twice. Dropout percentages and intermediate layer sizes varied across the 5 runs. It should be noted that if a single fully connected layer architecture is selected, the intermediate layer size information is discarded as it is not needed in the architecture implementation.

Architecture Feature	Bitstring Value	Architecture Value
BERT Layer	1011	BERT Layer 11
Fine-Tuning Architecture	0	Single Layer FC
Dropout	0100011	0.976
Intermediate Layer Size	010000000000000	8192
Internal Activation Neuron	0	ReLU
Final Activation Neuron	0	Softmax

Table 3. Genetic algorithm selected hyperparameter values with their corresponding torch architecture value (i.e. human interpretation of the string).

6 COMPARISON

Comparing our result to an off-the-shelf version of BERT without fine-tuning, we see an almost 250% increase in F1-Score. It's evident that by completing fine-tuning, we can achieve better overall performance. Drawing a direct comparison to previous models such as WESPAD [7], is not entirely accurate as their work took considerably more time with greater computational resources, nonetheless, both models find success in classifying personal health mentions. Additionally, despite not having similar training times or computational resources, we do see improved performance in terms of F1-score from our fine-tuned GA when compared to their models. In comparison to Madukwe et al's methodology, we tune different features using the same bitstring representation, while they used 4 bits to encode a 1-1 correspondance between integer values and decimal drop out values [12]. We chose to represent a larger number of dropout values by selecting 7 bits and dividing by the maximum value represented by these bits. We cannot directly compare our results to Madukwe et al.'s due to differing architectures [12], however, we have further reinforced their argument that GAs are capable of performing a robust hyperparameter search.

7 DISCUSSION

Overall, our results demonstrate that a genetic algorithm can provide a robust parameter search method. Given the time constraints, this approach is a better fit for design problems, rather than repetitive planning. That is, in practice, we would use this GA to fine-tune a classifier once offline and then deploy it, rather than continually fine-tuning online, as this is infeasible with the length of time it takes to train. Our work shows the feasibility of using a GA for hyperparameter tuning in the case of personal health mention classification, however, further investigation into misclassifications and how we can better fine-tune our model to reduce these should be completed.

7.1 Limitations

Computational resources restricted our use of the genetic algorithm. Even running on a high-performance machine, a single GA run took hours to complete. Our results would be strengthened by either greater computational power or a longer experimentation period where we could perform a greater number of runs. One aspect that we did not consider tuning with our genetic algorithm is the amount of training data. In general, LLMs can be fine-tuned with smaller amounts of training data than what is normally required to train a model from scratch. Further experimentation could be conducted

using smaller subsets of the data to see if similar accuracy could be achieved. This would reduce the computational overhead and thus the time is taken for each run, allowing for a more exhaustive search.

Other limitations pertained to issues with the data itself. As mentioned previously, we had to change the labels in the FLU2013 datasets for them to match the PHM2017 labels. However, the intuition and methods we used for changing the labels did not bear the results we had wanted. As an example, when there is any mention of swine flu/H1N1 the label should be 0 as it is not related to influenza. While they were correctly classified as non-related in one of the flu datasets, they were incorrectly classified as related in others while caused them to be counted as related in our dataset. To fix this, we would likely have to manually look at the dataset, and due to time constraints, this was unachievable. Furthermore, the decision to keep the first instance of the tweet when removing duplicates was arbitrary. With more time we would have liked to further experiment with other methods of removing duplicates, and observe if there are significant differences in results.

REFERENCES

[1] Olanrewaju Tahir Aduragba, Jialin Yu, and Alexandra I. Cristea. 2022. Incorporating Emotions into Health Mention Classification Task on Social Media. arXiv:2212.05039 [cs.CL]

[2] Christine Basta, Marta R. Costa-jussà, and Noe Casas. 2019. Evaluating the Underlying Gender Bias in Contextualized Word Embeddings. https://doi.org/10.48550/ARXIV.1904.08783

[3] Benoît Calvez and Guillaume Hutzler. 2006. Automatic Tuning of Agent-Based Models Using Genetic Algorithms. In *Multi-Agent-Based Simulation VI*, Jaime S. Sichman and Luis Antunes (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 41–57.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv* abs/1810.04805 (2019).

[5] F. Herrera, M. Lozano, and J.L. Verdegay. 1995. Tuning fuzzy logic controllers by genetic algorithms. *International Journal of Approximate Reasoning* 12, 3 (1995), 299–315. https://doi.org/10.1016/0888-613X(94)00033-Y

[6] Adith Iyer, Aditya Joshi, Sarvnaz Karimi, Ross Sparks, and Cecile Paris. 2019. Figurative Usage Detection of Symptom Words to Improve Personal Health Mention Detection. arXiv:1906.05466 [cs.CL]

[7] Payam Karisani and Eugene Agichtein. 2018. Did You Really Just Have a Heart Attack? Towards Robust Detection of Personal Health Mentions in Social Media. https://doi.org/10.48550/ARXIV.1802.09130

[8] Pervaiz Iqbal Khan, Imran Razzak, Andreas Dengel, and Sheraz Ahmed. 2020. Improving Personal Health Mention Detection on Twitter Using Permutation Based Word Representation Learning. In *Neural Information Processing*, Haiqin Yang, Kitsuchart Pasupa, Andrew Chi-Sing Leung, James T. Kwok, Jonathan H. Chan, and Irwin King (Eds.). Springer International Publishing, Cham, 776–785.

[9] Alex Lamb, Michael J. Paul, and Mark Dredze. 2013. 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*. Association for Computational Linguistics, Atlanta, Georgia, 789–795. https://aclanthology.org/N13-1097

[10] F.H.F. Leung, H.K. Lam, S.H. Ling, and P.K.S. Tam. 2003. Tuning of the structure and parameters of a neural network using an improved genetic algorithm. *IEEE Transactions on Neural Networks* 14, 1 (2003), 79–88. https://doi.org/10.1109/TNN.2002.804317

[11] Linkai Luo, Yue Wang, and Hai Liu. 2022. COVID-19 personal health mention detection from tweets using dual convolutional neural network. *Expert Systems with Applications* 200 (2022), 117139. https://doi.org/10.1016/j.eswa.2022.117139

[12] Kosisochukwu Judith Madukwe, Xiaoying Gao, and Bing Xue. 2020. A GA-Based Approach to Fine-Tuning BERT for Hate Speech Detection. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. 2821–2828. https://doi.org/10.1109/SSCI47803.2020.9308419

[13] Usman Naseem, Jinman Kim, Matloob Khushi, and Adam G. Dunn. 2023. Robust Identification of Figurative Language in Personal Health Mentions on Twitter. *IEEE Transactions on Artificial Intelligence* 4, 2 (2023), 362–372. https://doi.org/10.1109/TAI.2022.3175469

[14] Colin Reeves and Jonathan E Rowe. 2002. *Genetic Algorithms: Principles and Perspectives: A Guide to GA Theory*. Vol. 20. Springer Science & Business Media.