Evaluation of various text classification methods in disease type detection based on symptoms

Yaning Wang(yanwa579)
732A81-text mining
yanwa579@student.liu.se

Abstract

This project focuses on classifying text to identify disease types based on symptoms, employing various techniques such as eXtreme Gradient Boosting (XGB), Random Forest (RF), Multinomial Naive Bayes (MNB), Artificial Neural Network (ANN), and Convolutional Neural Network (CNN) for text classification. The Symptom2Disease dataset comprises 1200 data records with 'label' and 'text' components, is vectorized using a tf_idf vectorizer. Results show that the F1-scores range from 0.88 to 0.98, with the CNN model outperforming the others. Despite that CNN demonstrates superior performance, it has limitations, including higher computational cost, model building complexity, and intricate hyperparameter tuning. Future works are encouraged to utilize alternative models, e.g., Bidirectional Encoder Representations from Transformers, Recurrent neural network, etc.

1 Introduction

Text classification is a frequently employed machine learning technique in the field of natural language processing (NLP). It involves categorizing documents into specific labels or classes based on the dataset. Text classification has been widely used in sentiment analysis, spam email classifying, and more. One important application is in disease detection. Early disease detection is vital. It offers remote diagnosis and treatment recommendations which can assist individuals in securing necessary care without the need for frequent hospital visits. Additionally, it can aid healthcare professionals in making more precise decisions and increasing diagnostic accuracy. Further, it enables effective management of medical resources within hospitals.

Numerous studies have been conducted in the field of disease detection ([1],[2],[3],[4]). Among them, Rinkal Keniya [1] proposed eleven machine learning models, including several different K-Nearest Neighbors (KNN), Naive Bayes, and trees, to predict diseases based on patient symptoms, age, and gender. Their research showed that the Weighted KNN model achieved the highest accuracy of 93.5%. Shubham Bind [2] presented a comprehensive overview of commonly employed machine learning techniques, including ANN, KNN, Support Vector Machines (SVM), MNB, and RF, summarizing their applications in predicting Parkinson's disease.

In this study, text classification is used to predict diseases based on the symptoms. In addition to the methods mentioned above, we further incorporated XGB and CNN methods. We utilize the MNB classifier as a baseline to compare its performance with other models including XGB, RF, ANN, and CNN classifiers.

2 Method

2.1 Datasets

The dataset 'Symptom2Disease' [5] comprises 1200 data records written in English, consisting of two columns: 'label' and 'text'. The 'label' column encompasses 24 different diseases, while the 'text' column comprises symptom descriptions. It is crucial that the data is balanced for attaining precise and accurate results because an unbalanced dataset can produce bias towards the majority label and lead to bad predictions for the minority label. The dataset 'Symptom2Disease is balanced, with an equal number of data records (around 50) for each label, thus eliminating the need for oversampling (by replicating instances from the minority label to increase their representation) or undersampling (by reducing the instances from the majority label) techniques to balance the dataset during experimentation. The Symptom2Disease dataset is split into two subsets: training and test data. The training data constitutes 80% of the dataset, while the remaining 20% is the test data. The training

data is utilized for model training, while the test data validates the model's accuracy in disease identification. Table 1 presents the examples of the dataset employed in this experiment.

Labels	Texts (example for one record)
Psoriasis	The skin on my palms and soles is thickened ...
Varicose Veins	The veins on my legs cause a lot of discomforts…
Typhoid	There is a distinct pain in my abdominal part…
Chicken pox	I've lost my appetite and can't seem to eat anything…
Impetigo	I initially had rashes on my face and near my nose…
Dengue	I have been vomiting frequently and have lost my…
Fungal infection	"I have raised lumps, a rash that looks red and…"
Common Cold	"I've been sneezing nonstop and I can't seem to shake …"
Pneumonia	"I have a lot of difficulty breathing. I don't feel well…"
Dimorphic Hemorrhoids	"I've been experiencing a lot of bowel movement…"
Arthritis	"My muscles have been feeling really weak."
Acne	Lately I've been experiencing a skin rash...
Bronchial Asthma	doctor, i have been having high fever since past few..."
Hypertension	"My symptoms include a headache, chest pain..."
Migraine	"I've been grumpy and gloomy lately…
Cervical spondylosis	"I've had back pain, a persistent cough, and weakness..."
Jaundice	"I've been losing weight, feeling really fatigued, and .."
Malaria	"I've had a high fever, chills, and intense itching…"
urinary tract infection	I get a burning sensation when I pee…
allergy	"I have trouble breathing and get short of breath..."
gastroesophageal reflux	"I occasionally have trouble swallowing food..."
drug reaction	I have severe nausea and chest discomfort....
peptic ulcer disease	Constipated and diarrheal bowel motions have been...
diabetes	I have slow healing of wounds and cuts…

Table 1: The list of 24 disease labels with one example record for each label (totally 50 data records).

2.2 Data preprocessing

In the field of NLP, the pre-processing of a dataset is a critical step in model training. This process involves reducing noise in the data and enhancing model performance. To achieve this, we convert the label column of the dataset into a numeric representation instead of characters. Additionally, the text should be converted to lowercase, and all punctuations, specified symbols, and stop words should be removed. Finally, the text is vectorized using a tf_idf vectorizer, also known as the Term Frequency-Inverse Document Frequency, to transform it into numerical vectors. This ensures that the text is in a suitable format to be fed into the MNB, XGB, RF, NN, and CNN models.

2.3 The machine learning algorithms

In the current study, the dataset comprises textual data and corresponding labels, thereby we apply supervised learning techniques.

2.3.1 Multinomial Naive Bayes

As part of our experiment in text classification, we utilized MNB as a baseline. This method is a form of supervised learning, which relies on making independent assumptions about probabilities. While this assumption may not always true in practical applications, it does require fewer hyperparameters for classification tasks. Given that our dataset contained 24 categories, we opted to use the MultinomialNB model from the sklearn library to train our model. No additional hyperparameters were required for this step.

2.3.2 eXtreme Gradient Boosting Classifier

The XGB Classifier is a machine learning model that utilizes the gradient-boosted decision trees algorithm to classify data. It is an ensemble learning method [6]. However, it requires more hyperparameters to be set than the MNB classifier. In our experiment, we utilized the XGBClassifier directly from the sklearn library with the following hyperparameters setup: n_estimators=2000, learning_rate=0.25, max_depth=4, colsample_bytree=0.3, n_jobs=-1, random_state=40.

2.3.3 Random Forest

The RF is an ensemble learning method that involves employing multiple tree-structured classifiers to predict the most popular class. This approach effectively mitigates overfitting by combining the results of various trees [7]. Compared to other classification techniques, it requires fewer hyperparameters, thus streamlining the training process. Our model will be trained using the RF model from the sklearn library, without additional hyperparameter configurations.

2.3.4 Neural network

Text classification tasks often leverage NN, also known as the artificial neural network (ANN), composing an input layer, a hidden layer, and an output layer. The number of nodes in each layer is task dependent. The input layer should match the input features, while the output layer should match the output classes, or labels [8]. In our experiment, we constructed a five-layer network (Fig. 1). The first layer, referred to as the embedding layer, transforms integers into dense vectors of fixed size. The second layer utilizes a pooling operation, while the following two dense layers are built with 16 and 32 neurons, using the 'relu' activation function. This function helps introduce nonlinearity to the network and overcome the vanishing problem. The last layer is a dense layer with 24 neurons and a 'softmax' activation function, as we have multiple classes to consider. Fig. 1 shows the ANN model structure used in this study.

2.3.5 Convolutional Neural network

In deep learning, CNN is a widely utilized method with diverse applications, such as signal identifica-

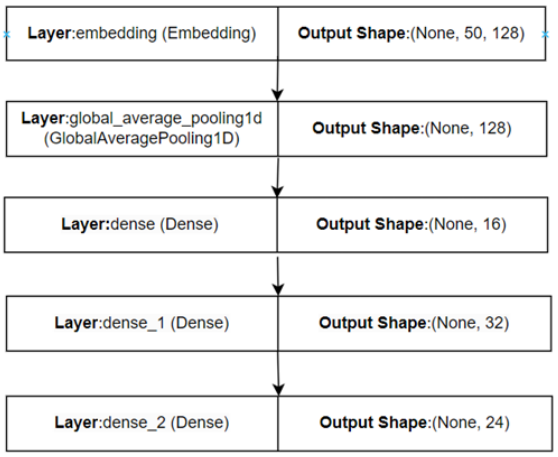

Figure 1: The model structure of the ANN with 5 layers.

tion, image classification, and human action recognition. CNNs aims to extract features from data with specific structures, resulting in a reduction of hyperparameters compared to general neural networks, thanks to their weight-sharing property. When constructing the convolution aspect, four key components should be considered: the number of filters in the convolution, determining the output space's dimension; the kernel size, specifying the convolution window's size; the stride, adjusting the density of convolving; and the padding, employed to control the addition of extra values (usually zeros) around the borders [9]. Ultimately, a model can be built using both the CNN layer and dense layers. Fig. 2 shows the CNN model structure used in this study.

2.4 Model evaluation

Various evaluation metrics, including accuracy, F1-score, mean square error, precision, and recall can be employed to assess the performance of models. The F1-score, calculated as the harmonic average of precision and recall values, provides a comprehensive assessment of the model's performance. Precision is calculated by dividing the number of True Positives (TP) by the sum of True Positives (TP) and False Positives (FP), while recall is calculated by dividing the number of True Positives (TP) by the sum of True Positives (TP) and False Negatives (FN). In this study, we employ F1-score and accuracy to assess model performance on the balanced dataset.

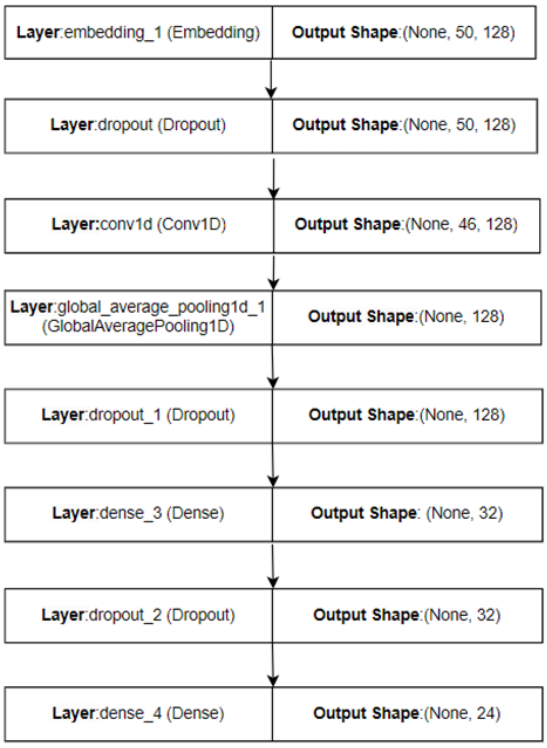

Figure 2: The model structure of the CNN with 8 layers.

3 Results

For each model, we trained the model with varied hyperparameter settings. The hyperparameters that yielded optimal results were chosen. The F1-scores obtained with these selected hyperparameters for the five distinct algorithms are presented in Table 2. Specifically, the F1-scores for XGB, RF, and MNB were computed using the classification report function from the sklearn library. For ANN and CNN, the F1-scores were obtained by employing the evaluate function of each respective model.

The results reveal varying performance among XGB, RF, and MNB models, with the XGB model exhibiting a lower F1-score of 0.88, contrasting with the RF model's higher performance with an F1-score of 0.95. Notably, the baseline method, MNB, achieved an intermediate F1-score of 0.93. These comparative results emphasize the nuanced distinctions in model performance.

For the ANN and CNN models, the accuracy in training and testing of both ANN and CNN models demonstrates an increasing trend with a rising number of epochs, as depicted in Fig. 3 and Fig. 4. Nevertheless, beyond approximately epoch 10, the rate of accuracy improvement diminishes with further increases in epochs. Notably, the ANN

Classifier	F1_score
XGB	0.88
RF	0.95
MNB (baseline)	0.93
ANN	0.90
CNN	0.98

Table 2: The F1-score of different methods.

model exhibits a slight overfitting, implying superior performance on training data but diminished performance on test data. Despite varying hyperparameter tuning throughout the experiment, no improvement in the model's performance was observed.

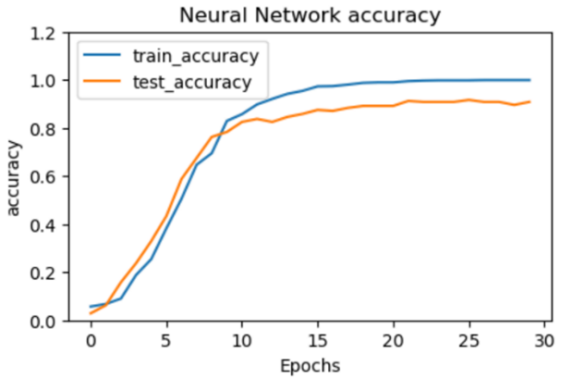

Figure 3: The variation of accuracy with epochs for the ANN model.

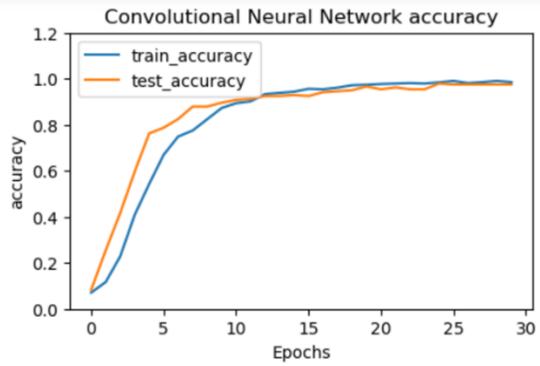

Figure 4: The variation of accuracy with epochs for the CNN model.

In summary, it can be concluded that the CNN model outperformed the other classifiers. However, it is essential to highlight that certain classifiers (XGB and ANN) failed to surpass the baseline performance. Despite these variations, it is noteworthy that the majority of models achieved remarkably high accuracy levels in comparison to the baseline.

4 Discussions

Although all models give relatively high F1-scores (>=0.88), there are limitations in the current methods. One such limitation occurred during the vectorization step, where we utilized the tf_idf vectorizer. While this method is effective, it does not consider the semantic similarities between words. Despite this limitation, we found that using the tf_idf vectorizer on the Symptom2Disease dataset revealed no degradation in the F1-scores of our classifiers prompting us to maintain its use. However, it's important to point out that the tf_idf approach may not be universally suitable. For scenarios where semantic relationships are crucial, an alternative called Word2Vec word embedding can be applied. This approach captures semantic relationships by preserving contextual information and transforming words into a dense vector space.

It is important to point out the computational demands associated with training ANN and CNN, particularly when dealing with large amounts of datasets or large-scale models with numerous layers and hyperparameters. To mitigate this challenge, we strategically employed the Symptom2Disease dataset, which comprises only 1200 data records, and utilized models with a limited number of layers (specifically 5 and 8 layers). This approach allowed us to train the models more efficiently while achieving satisfactory performance. Throughout the experiment, we used an 11th Gen Intel(R) Core (TM) i7 processor with 4 cores and 8 logical processors.

Moreover, it is imperative to emphasize that building ANN and CNN models and adjusting the hyperparameters can take a significant amount of time. In CNN, the hyperparameters that need to be optimized include the size and number of kernels, the length of strides, the pooling size, the dropout rate, and the number of layers and nodes. These hyperparameters are crucial in minimizing the loss function, which ultimately determines the performance of the model.

5 Conclusions

This project aimed to classify texts based on NLP to assist in disease detection. To this end, the Symptom2Disease dataset was used to access five models including XGB, RF, MNB, ANN, and CNN. The F1-score of these model results ranges from 0.88 to 0.98, with the CNN model demonstrating better performance than other models. Notably, XGB

(0.88) and ANN (0.90) have lower F1-scores than the baseline MNB model (0.93). These models can be applied for similar classification tasks after necessary changes, including setting and tuning hyperparameters, constructing neural network architecture, and preprocessing and vectorizing text into numeric vectors. It should be noted that the CNN and ANN require higher computational cost, increased model building complexity, and more intricate hyperparameter tuning compared to XGB, RF and MNB. Further, for scenarios where semantic relationships are crucial, an alternative called Word2Vec word embedding can be applied instead of the tf_idf approach. Additionally, widely used models such as Bidirectional Encoder Representations from Transformers (BERT) and Recurrent neural network (RNN) are also found to be applied in text classification tasks. Further research in this area is essential for continued advancements.

6 Code availability

The entirety of the code, dataset, and associated files utilized in the experiment are accessible within the following public repository: https://github.com/Yaning2022/text-mining-project.git

7 References

[1] Keniya, R., Khakharia, A., Shah, V., Gada, V., Manjalkar, R., Thaker, T., ... & Mehendale, N. (2020). Disease prediction from various symptoms using machine learning. Available at SSRN 3661426.

[2] Bind, S., Tiwari, A. K., Sahani, A. K., Koulibaly, P., Nobili, F., Pagani, M., ... & Tatsch, K. (2015). A survey of machine learning based approaches for Parkinson disease prediction. Int. J. Comput. Sci. Inf. Technol, 6(2), 1648-1655.

[3] Al-Garadi, M. A., Yang, Y. C., Lakamana, S., & Sarker, A. (2020). A text classification approach for the automatic detection of twitter posts containing self-reported covid-19 symptoms.

[4] Ahsan, M. M., Luna, S. A., & Siddique, Z. (2022, March). Machine-learning-based disease diagnosis: A comprehensive review. In Healthcare (Vol. 10, No. 3, p. 541). MDPI.

[5] Symptom2Disease,accessed:2023-1-13,https://www.kaggle.com/datasets/niyarrbarman/symptom2disease/data

[6] Sharma, R. (2021). DETECTING PARKINSON'S DISEASE USING MACHINE LEARNING. International Journal of Innovations in Engineering Research and Technology, 8(07), 267-269.

[7] Breiman, L. (2001). Random forests. Machine learning, 45, 5-32.

[8] Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., & Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. Heliyon, 4(11).

[9] Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A survey of convolutional neural networks: analysis, applications, and prospects. IEEE transactions on neural networks and learning systems.