Machine Learning Project Report: Medical Text Classification

Tia Manoukian, Jad Alaouie, Joe Naimeh

Dr. Rida Assaf

Abstract—This project addresses medical classification, aiming to accurately classify medical abstracts into five different health conditions. To achieve this, we employed various machine learning and deep learning techniques; namely Logistic Regression, Support Vector Machines, XGBoost, BERT, and Recurrent Neural Networks (RNNs). We preprocessed the data tailoring to each of these models, but we generally used tokenization, TF-IDF vectorization, and sequence padding. Then, we conducted Hyperparameter tuning and fine-tuning of the pre-trained models, in order to optimize their performance. All in all, we have come across promising accuracies across all models, especially with BERT (Bidirectional Encoder Representations from Transformers) and RNNs (Recurrent Neural Networks) using LSTM (Long Short Term Memory) and Support Vector Machines (SVMs). These models achieved the highest accuracies of 87% on the test dataset. This study highlights the effectiveness of machine learning and deep learning approaches in medical classification tasks, offering valuable insights for healthcare applications.

Contents

1. Introduction

This project focuses on the task of medical classification, where the goal is to classify medical abstracts into five different categories of health conditions based on their textual content: Nervous System Disease, General Pathological Conditions, Neoplasms, Digestive System Diseases, and Cardiovascular Diseases. The complexity of medical data, including variations in language, terminology, and context, poses significant challenges for traditional classification methods. However, advancements in machine learning and deep learning techniques offer promising solutions to address these challenges, and accurately categorize.

The findings of this study reveal promising results across all models, with varying levels of accuracy achieved on the test dataset. Notably, SVMs, BERT and Bidirectional LSTM emerge as the top-performing models, achieving the highest accuracies. These results demonstrate the efficacy of leveraging advanced machine learning and deep learning techniques for medical classification tasks. In addition, the comparative analysis of model performances provides valuable insights into the strengths and limitations of each approach, guiding future research and applications in healthcare informatics.

Through this project, we aim to contribute to the ongoing efforts in optimizing the use of artificial intelligence in order to improve medical document classification, leading to eventual improved healthcare outcomes.

2. Dataset

2.1. Description of the Dataset

The dataset given for this project consists of medical abstracts paired with corresponding condition labels, providing a comprehensive overview of various medical conditions. The dataset contains information on a variety of medical conditions, categorized into five distinct classes: **Nervous System Disease, General Pathological Conditions, Neoplasms, Digestive System Diseases, and Cardiovascular Diseases**. The dataset offers valuable insights into different aspects of these conditions. However, we observed a skewedness in the data distribution, indicating a potential class imbalance across medical conditions. To address this issue and streamline our workflow for efficiency, we engineered a new dataset named 'new_data.csv'. This dataset underwent pre-processing, including the removal of stop words, punctuation, and extra spaces, as well as lemmatization and tokenization. Additionally, we performed data augmentation by incorporating synonyms of words to enhance the model's learning capacity, as well as both downsampling and upsampling techniques to balance the distribution of the minority and majority classes, respectively. By leveraging this refined dataset, we optimized the training process of our models, ensuring robust performance without the need for repeated pre-processing steps.

2.2. Data Preprocessing

Before training the machine learning and deep learning models, several pre-processing steps were undertaken to prepare the textual data for analysis. We then saved these changes and enhancements into a file called `new_data.csv`, which we will use from this point on to train all our models. These steps included:

- The dataset was **split** into training (60%), validation (20%), and test (20%) sets to evaluate model performance effectively.
- **Tokenization** of the text data using the NLTK library, which breaks down the abstracts into individual words or tokens. Stop words, punctuation, and extra spaces were removed to clean the text and reduce noise. In addition to stemming, we applied lemmatization techniques to normalize the text, reducing words to their base or dictionary form, ensuring consistency in word forms.
- **TF-IDF vectorization** and **word embeddings** converted the text data into numerical representations. TF-IDF vectorization assigns weights to words based on their frequency in the document and rarity across the whole text. Word embeddings capture semantic meaning and relationships between words by representing them as dense vectors in a high-dimensional space. Note that this was especially useful for the *recurrent neural networks* model.
- We also used **Data Augmentation** techniques, such as synonym replacement, to enhance the diversity of the training data, improving the models' learning. By replacing words in the medical abstracts with their synonyms, we exposed the models to different variations of the same concepts.
- The dataset underwent both **upsampling of the minority class** and **downsampling of the majority class** to address class imbalance issues. Upsampling involves randomly duplicating instances from the minority class to balance the class distribution,

while downsampling involves randomly removing instances from the majority class. This helps prevent the models from being biased towards the majority class and improves their ability to learn from the minority class, addressing the skewedness of the data.
- For *BERT models*, the data was tokenized using the **BERT BioMedBERT tokenizer**, enabling the representation of text inputs in a format suitable for BERT's architecture.
- Additionally, for models such as *recurrent neural networks (RNNs)* and *transformers using BERT (Bidirectional Encoder Representations from Transformers)*, the text sequences were **padded to a fixed length** to facilitate batch processing.

Overall, these pre-processing steps ensured that the dataset was appropriately formatted and ready for training these diverse machine learning models to more accurately classify medical conditions based on abstracts.

2.3. Data before and after

Data Before Pre-Processing and Augmentation:

Original Medical Abstract: "Early gastric cancer. Twenty-eight-year experience. A retrospective study of early gastric cancer (60 patients) was performed to evaluate its diagnosis and treatment. Ninety-five per cent of patients presented with nonspecific gastrointestinal symptoms and 53.3% had been treated for presumed benign disease for up to 48 months before diagnosis. Fiberoptic endoscopy detected these lesions more accurately than radiologic examination. The disease-free 5-year survival rate after resection was 76.4%. Survival showed no significant correlation with sex, tumor site, macroscopic appearance, extent of gastric resection, or histopathologic type. Tumors larger than 1.5 cm in diameter, invasion of submucosa, or lymph node metastasis resulted in significantly lower survival rates. Three of eight patients with nodal metastasis survived 5 or more years, including one who had second-echelon deposits. A high index of suspicion may permit more frequent detection. Extended lymphadenectomy (R2) is recommended to achieve the highest possible cure rate".

Data After Pre-Processing and Augmentation:

Pre-Processed Medical Abstact 1: "early gastric cancer twenty eight year experience angstrom retrospective study early gastric cancer sixty patient equal perform measure informationtechnology diagnosis treatment ninety fivespot per penny affectedrole stage nonspecific gastrointestinal symptom fiftythree three treated presume benign disease fortyeight calendarmonth earlier diagnosis character optic endoscopy detect lesion accurately radiologic examination disease free five year survival rate resection seventysix four survival picture significant correlation sex tumor website macroscopic appearance extent gastric resection operatingroom histopathologic type tumor large one five curium indium diameter invasion submuc osa operatingroom lymph node metastasis result significantly lower survival rat three eight patient nodal metastasis survive five oregon longtime admit one worldhealthorganization second ech el deposit deoxyadenosinemonophosphate high index intuition may let frequent detection strain lymphadenectomy r2 recommend achieve high possible cure pace"

Pre-Processed Medical Abstract 2: "early gastric cancer twenty eight year know retrospective study earlyon gastric cancer sixty patient perform measure informationtechnology diagnosis treatment ninety fivespot per cent patient show nonspecific gastrointestinal symptom fiftythree three experience tempered presume benign disease fortyeight month earlier diagnosis roughage optic endoscopy detect wound accurately radiologic interrogation disease free five year survival pace resection seventysix four survival show significant correlationcoefficient sex tumor site macroscopic appearance extent gastric resection operatingroom histopathologic type tumor large one five curium indiana diameter invasion submuc osa oregon lymph node metastasis result significantly lower survival rate three eightspot affectedrole nodal metastasis outlive five operatingroom year include one worldhealthorganization second ech el deposit vitamina high index suspicion whitethorn allow frequent signaldetection strain lymphadenectomy r2 commend achieve gamey potential cure rate"

Pre-Processed Medical Abstract 3: "early gastric cancer twenty eight year know deoxyadenosinemonophosphate retrospective study early gastric cancer sixty patient washington evaluate informationtechnology diagnosis treatment ninety five per cent affectedrole give nonspecific gastrointestinal symptom fiftythree three induce constitute treated presume benign disease fortyeight month earlier diagnosis roughage optic endoscopy detected lesion accurately radiologic examination disease free five year survival pace subsequently resection seventysix four survival indicate nobelium significant correlation sex tumor website macroscopic appearance extent gastric resection oregon histopathologic type tumor large one five centimeter diameter invasion submuc osa oregon lymph node metastasis result importantly abject survival rat trey eightspot affectedrole nodal metastasis survive five operatingroom class include one worldhealthorganization moment ech el deposit deoxyadenosinemonophosphate high exponent suspicion may license frequent detection prolong lymphadenectomy r2 recommend achieve high possible cure rate"

3. Models

In this project, we explored various machine learning models to tackle the medical text classification problem. The models we considered include Logistic Regression, Support Vector Machines (SVM), XGBoost, Transformers using BERT, and Recurrent Neural Networks (RNNs). We will explain below how each model offers distinct advantages and is suitable for different aspects of the classification task.

3.1. Logistic Regression

It's always best to start simple! Logistic Regression is a simple statistical model used for binary classification tasks, where the outcome variable takes on two possible values. By analyzing patient data and input features, logistic regression can estimate the probability of a patient belonging to a specific medical condition class, aiding healthcare professionals in diagnosis and treatment decision-making. Its simplicity, interpretability, and ability to handle binary outcomes make logistic regression particularly useful in medical settings where predicting the presence or absence of a condition is paramount.

3.1.1. Training, Tuning, and Validation

- We trained the Logistic Regression model using the TF-IDF vectorization on the new data.
- Evaluated the model's generalization ability by computing the accuracy score on the validation data, resulting in a score of 84%.
- Validation accuracy was evaluated to assess model performance using a hold-out validation set. Additionally, learning curves were plotted to analyze the model's performance with varying training sizes.

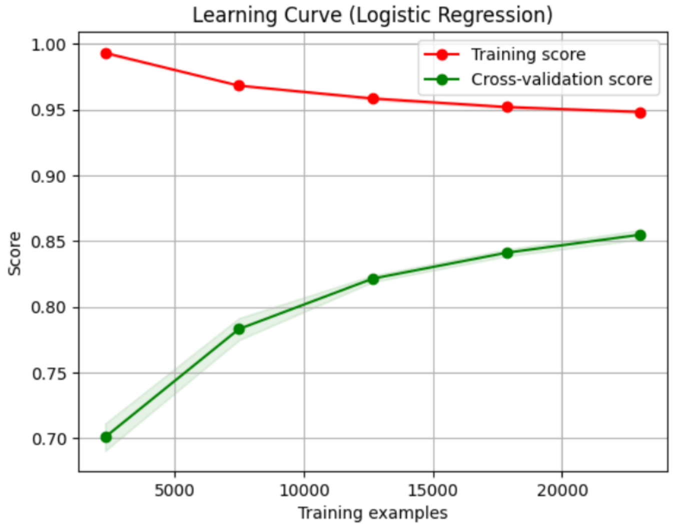

Figure 1. Logistic Regression Learning Curve

3.1.2. Rationale

Logistic Regression stands out as a foundational model renowned for its simplicity and ease of interpretation. It serves as an excellent initial step in text classification endeavors, providing a solid baseline against which to compare more complex algorithms. Its straightforward nature makes it an accessible choice for understanding the fundamental dynamics of the data and establishing a benchmark for performance evaluation.

3.2. Support Vector Machines (SVM)

SVM is a powerful supervised learning algorithm used for classification tasks. SVM works by mapping data to a high-dimensional feature space so that data points can be categorized, even when the data are not otherwise linearly separable. They offer a robust method for discerning patterns within patient data to categorize individuals into different medical conditions. SVMs excel in finding optimal decision boundaries, effectively separating patients into distinct classes based on their clinical attributes or biomarkers. By maximizing the margin between different classes, SVMs strive to achieve accurate predictions, making them a valuable asset in medical diagnosis and prognosis.

3.2.1. Training, Tuning, and Validation

- We trained the SVM model using the TF-IDF transformed new pre-processed text data.
- Hyperparameters such as kernel type, regularization parameter (C), and class weight were tuned using grid search with cross-validation.
- Model performance was evaluated using accuracy metrics on the validation set, and learning curves were plotted to assess model convergence and generalization. SVM performed pretty well on the validation set with an accuracy of 87.44%.

3.2.2. Rationale

We opted for Support Vector Machines (SVMs) due to their proficiency in managing high-dimensional datasets. Renowned for their capability to discern intricate decision boundaries, SVMs excel in accommodating non-linear data through techniques like the kernel trick, rendering them well-suited for text classification endeavors.

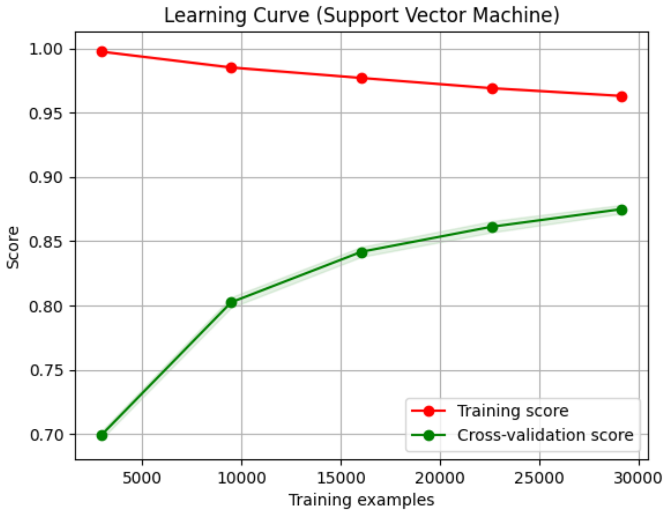

Figure 2. SVM Learning Curve

3.3. XGBoost

XGBoost stands out as a powerful ensemble learning technique that excels in discerning complex patterns within patient data to predict various medical conditions. It is an implementation of gradient-boosting decision trees. By iteratively refining weak learners, XGBoost constructs a strong predictive model that optimally separates patients into different classes based on their clinical attributes or biomarkers. Its ability to handle non-linear relationships and incorporate feature importance makes XGBoost particularly effective for medical diagnosis and prognosis, providing clinicians with valuable insights for personalized patient care.

3.3.1. Training, Tuning, and Validation

- Our initial XGBoost model was configured with parameters including a maximum depth of 5, minimum child weight of 1, and a gamma value of 0, among others.
- To further optimize the model's performance, a grid search was conducted over a range of hyperparameters including max depth, min child weight, gamma, subsample, colsample by tree, and eta
- Model performance was evaluated using accuracy metrics on the validation set, and learning curves were plotted to assess convergence and potential overfitting. But this model actually performed the least accurately with an 84% accuracy.

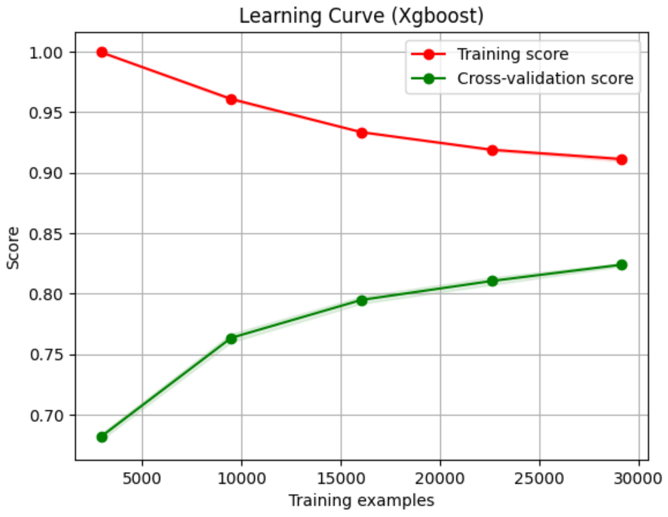

Figure 3. XGBOOST Learning Curve

3.3.2. Rationale

XGBoost was the method of choice for its impressive track record in achieving high accuracy rates, its efficient processing speed, and its ability to effortlessly handle vast amounts of data. This robust algorithm has been widely acclaimed for its capability to effectively handle both structured and unstructured data, making it particularly well-suited for the complexities of text classification tasks.

3.4. Transformers using BERT

BERT (Bidirectional Encoder Representations from Transfomers) is one of the models that are based on **Transformers Architecture** and had a gigantic impact in the world of AI when they were first published. It is a **Language Model** that has the has the capacity to learn specific tasks when it comes to language (NLP) and it can do this because it understands language. It has the understanding of how the words relate to each other. By fine-tuning BERT we can specifically make it perform well on different types of language tasks. This type of learning is referred to as **Transfer Learning** where the model is already trained on extremely large datasets, and our role is to fine-tune it to make it perform well on our own data.

Figure 4. BERT

3.4.1. Architecture and Implementation

- **Tokenization:** The BioMedBERT (A variation of BERT which is pre-trained particularly on data related to the Bio-Medical Field which relates to our data) Tokenizer converts the text into tokens (numbers that have meaningful values) that can be processed by the model.

- **Model Architecture:** Essentialy, BERT is a multi-layer Bidirectional Transformer Encoder. For Natural Language Processing or Sequence Classification tasks, the output of the transformer is taken from the first token of the sequence denoted as [CLS] this token is named **Classifier Token**.

- Model performance was evaluated using accuracy metrics on the validation set. However, due to computational constraints, fine-tuning was limited.

- **Fine-Tuning:** BioMedBERT is a pre-trained model with large corpus related to the BioMedical Field which is better than using the regular BERT model which corpus is related to general task. The model needs to be fine-tuned in order to reveal its power. However, due to time constraints and our limited resources, we weren't able to fine-tune it enough to get the best out of it.

- **Training Procedure:** The model comes with its initial weights, therefore training the model in our project and on our data will adjust the weights of the entire network including the final classification layers, in order to minimize the loss function, typically categorical cross-entropy for our case. The latter happens using **Back-Propagation** and **Adam** as an optimization algorithm.

- **Accuracy:** Model performance was evaluated using accuracy metrics on the validation set, in addition to the classification report which showed some good results in both Precision and Recall. BioMedBERT performed well on the validation set with an accuracy of 87.3472%. However, further fine-tuning can reach even higher accuracy on validation data, but due to limited resources and the long time it takes for training we decided to stop at 87.3472% accuracy.

3.4.2. Rationale

BERT transformer models have shown state-of-the-art performance on various natural language processing tasks, including text classification, by capturing contextual information and semantic meaning effectively.

3.5. Recurrent Neural Networks (RNNs)

RNNs are a class of neural networks designed for sequential data processing.

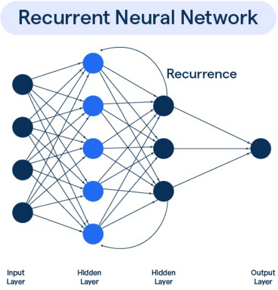

Figure 5. Simple Recurrent Neural Network Architecture

3.5.1. Architecture and Implementation

- Our Recurrent Neural Network model is built using **TensorFlow** and **Keras**.

- **Embedding Layer:** The Embedding Layer's job is to convert the input text tokens into dense vectors of a fixed length (200 in our case. A little bit higher than the average sentence length to make sure that we cove a greater majority and truncate a few only). The embedding matrix has a shape of [vocab size, 200]. *vocab size = 19968*

- **Spatial Dropout1D:**Its role is to regularize the model and to avoid and reduce overfitting.

- **Bidirectional LSTM Layers:** We wrapped the LSTM units in Bidirectional wrapper to allow the network to get the information from both the past and the future. It is essential in order to understand the context of the medical abstract that could potentially be missed when only looking in a single direction.

- **Dropout Layers:** To avoid overfitting, we also used Dropout Layers that sets a some of the input units to 0 at each update during training.

- **Dense and Batch Normalization Layers:** The final step in our model is passing the data into dense layers of ReLU activation followed by a Batch Normalization layer that normalizes the output of the previous layer (ReLU). Which will improve stability and performance.

- **Output Layer:** The output layer consists of a Softmax activation function that outputs the probability distribution over the predefined classes.

- **Hyperparameters:** Embedding dimension, LSTM units, Dropout Rates, and Regularization were optimized manually everytime. Since RNN didn't take as much time as BioMedBERT we were able to spend a lot of time on hyper-parameter tuning. Unfortunately, we couldn't improve further than 86-87% accuracy on validation data!

- Model performance was evaluated based on accuracy metrics on the validation set, and convergence was assessed using learning curves

3.5.2. Rationale

RNNs were chosen for their ability to model sequential data, capturing dependencies between words in text sequences. They are a suitable model for tasks where context matters, such as text classification.

4. Results

- We find that it is essential to discuss the difficulties that arise during the course of the project. When we first started, we faced a lot of difficulties because our preprocessing techniques were quite basic, which led to very poor accuracy levels. Later, when we investigated Recurrent Neural Networks (RNNs), we discovered additional difficulties since at first, these networks had trouble learning and had accuracy rates as low as 20%. Even with significant efforts to maximize performance through hyperparameter tuning, using random techniques for RNNs as well as grid search, the procedure remained laborious, especially when grid search required a lot of processing power. Out of the five models that were examined, BERT stood out as a promising candidate with the potential for better performance. However, due to practical limitations like scarce resources and lengthy training times, such as the three epochs that it takes to complete, we were unable to fully utilize BERT's potential, which resulted in an accuracy plateau between 86 and 87%. Furthermore, the underlying complexity and lack of structure in our dataset highlighted the limitations posed by its unstructured nature, as our experiments with Decision Trees and Random Forests demonstrated limited utility.
- Finally, after examining each model's accuracy on the test data, we came up with this bar chart that shows different models' performance on our Test Data:
  - **Support Vector Machines (SVMs):** 87.44%
  - **BioMedBERT:** 86.88%
  - **Logistic Regression:** 86.87%
  - **Recurrent Neural Networks:** 86.17%
  - **XGBOOST:** 84.00%

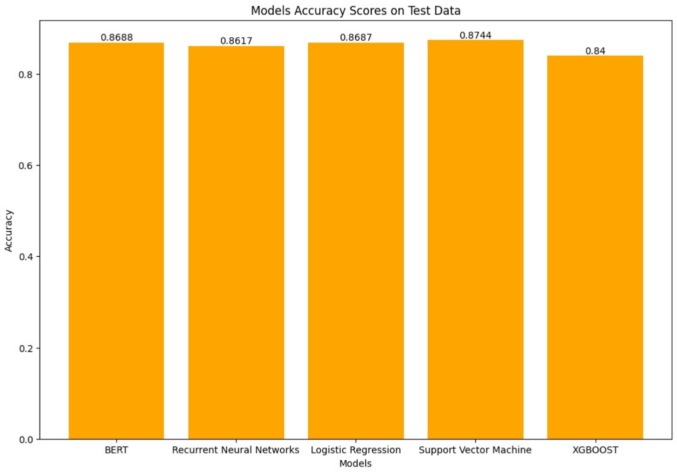

Figure 6. Comparing Accuracy Results of all 5 models on Test Data

5. Conclusion

In conclusion, this report includes a complete investigation of the use of different Machine Learning Algorithms for Medical Text Classification. By using Logistic Regression, SVM, XGBOOST, BioMedBERT, and a Recurrent Neural Network model. Data augmentation was required to avoid overfitting, especially in the RNN model, so our solution was to use synonyms for each word in every medical abstract. Our tests on these different models showed how these learning algorithms could learn and handle complicated data (unstructured text data). SVMs and BERT mainly outperformed the rest of the models providing the highest accuracies and best precision and recall. Future work should focus on the ability to improve BERT because we strongly believe that it can do much better but due to time constraints limited resources and not enough powerful GPUs that can handle such tasks, further optimizing BioMedBERT is essential through intensive hyperparameter tuning and maybe by adding and collecting more data from different resources.

6. References

- Sequence Prediction Using Recurrent Neural Networks: https://dilnaz-n.medium.com/sequence-prediction-using-recurrent-neural-networks-be94d2aab58b

- Training of Recurrent Neural Networks (RNN) in TensorFlow: https://www.geeksforgeeks.org/training-of-recurrent-neural-networks-rnn-in-tensorflow/

- Building a Text Classifier using RNN: https://medium.com/nerd-for-tech/building-a-text-classifier-using-rnn-57b546d3d35a

- Getting Started with AI: Building an RNN from scratch and practicing resilience: https://medium.com/@adachoudhry26/getting-started-with-ai-building-an-rnn-from-scratch-and-practicing-resilience-ba3c10be6a22

- Natural Language Processing: From Basics to using RNN and LSTM: https://medium.com/analytics-vidhya/natural-language-processing-from-basics-to-using-rnn-and-lstm-ef6779e4ae66

- Multi-label Text Classification with BERT and PyTorch Lightning: https://curiousily.com/posts/multi-label-text-classification-with-bert-and-pytorch-lightning/

- Fine-tuning BERT for Text Classification: A Step-by-Step Guide: https://medium.com/@coderhack.com/fine-tuning-bert-for-text-classification-a-step-by-step-guide-1a1c5f8e8ae1

- Publicly Available Clinical BERT Embeddings: https://arxiv.org/abs/1904.03323

- Mastering Text Classification with BERT: A Comprehensive Guide: https://medium.com/@ayikfurkan1/mastering-text-classification-with-bert-a-comprehensive-guide-194ddb2aa2e5

- Improving Bert-Based Model for Medical Text Classification with an Optimization Algorithm: Gasmi, K. (2022). Improving Bert-Based Model for Medical Text Classification with an Optimization Algorithm. In: Bădică, C., Treur, J., Benslimane, D., Hnatkowska, B., Krótkiewicz, M. (eds) Advances in Computational Collective Intelligence. ICCCI 2022. Communications in Computer and Information Science, vol 1653. Springer, Cham. https://doi.org/10.1007/978-3-031-16210-7_8
- med-bert: https://huggingface.co/praneethvasarla/med-bert

- GitHub Repo: rnn-text-classification: https://github.com/gdimitriou/rnn-text-classification

- GitHub Repo: rnn-text-classification-tf: https://github.com/roomylee/rnn-text-classification-tf

- rnn-text-classification-tf: https://github.com/roomylee/rnn-text-classification-tf