iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** VIII **Month of publication:** August 2021

DOI: https://doi.org/10.22214/ijraset.2021.37349

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VIII Aug 2021- Available at www.ijraset.com

Transfer Learning for Spam Text Classification

Pratiksha Bongale[1], Abdul Razak M S[2], Dr. Nirmala C R[3], Dr. Roopa G M[4]
[1, 2, 3, 4]*Department of Computer Science and Engineering, Bapuji Institute of Engineering and Technology, Davanagere, India*

Abstract: ***Today's world is mostly data-driven. To deal with the humongous amount of data, Machine Learning and Data Mining strategies are put into usage. Traditional ML approaches presume that the model is tested on a dataset extracted from the same domain from where the training data has been taken from. Nevertheless, some real-world situations require machines to provide good results with very little domain-specific training data. This creates room for the development of machines that are capable of predicting accurately by being trained on easily found data. Transfer Learning is the key to it. It is the scientific art of applying the knowledge gained while learning a task to another task that is similar to the previous one in some or another way. This article focuses on building a model that is capable of differentiating text data into binary classes; one roofing the text data that is spam and the other not containing spam using BERT's pre-trained model (bert-base-uncased). This pre-trained model has been trained on Wikipedia and Book Corpus data and the goal of this paper is to highlight the pre-trained model's capabilities to transfer the knowledge that it has learned from its training (Wiki and Book Corpus) to classifying spam texts from the rest.***
Keywords: Domain adaptation, Transfer Learning, Text Classification, Language Model.

I. INTRODUCTION

Today's world is majorly dependent on communication networks. Communication has been an essential part of all our lives for decades now. It is what connects people across the globe and aids many useful processes going on including space researches, businesses, industrial work, telephone communication, etc. People have been using text messaging and e-mails as a medium of communication for the past decades.

They have been popular for being very easily accessible, very fast, convenient, and very low cost. The imperfections in the messaging protocols along with the rapidly growing traffic of electronic business and transactions contribute to great IT threats. E-mail or in general, Spam messages are one of the most sensitive issues of today's communication system.

They could lead to individuals getting trapped into fraudulent transactions, fake news, businesses falling into wrong deals, so and so forth. Spam messages enter user's inboxes without the user's knowledge and overfill their inboxes. This not only can possess great threat but also increases the network traffic enormously leading to server bottlenecks and also reduced performance in some scenarios.

Spam classification becomes a crucial problem as filtering messages might be difficult if the countermeasures are overly specific that leaves a chance of the deletion of legitimate messages also. Many users have been targeted over these spam messages and the issue persists. Many researchers have presented Machine Learning approaches to solve this problem using various algorithms over the recent years.

This article focuses on utilizing the Transfer Learning approach for classifying spam messages using a pre-trained Language Model. Transfer Learning technique has been overly popular for providing proven results in rendering attractive performances while forecasting results in the target domain utilizing the knowledge that it has gained from a similar source domain dataset. In this context, the source domain dataset is the one that can be easily found/collected and usually is highly labeled, for instance, restaurant reviews; while the target domain dataset is the one that is generally hard to find as they are poorly labeled, for instance, financial datasets.

Usually, the source dataset for training is chosen in such a way that it comprises the target domain data also. However, in many real-life scenarios, this becomes a difficult task to gather good source domain data for the model to be trained on.

Since the model is trained using a high-resource source dataset and is expected to adapt to another (target) domain of low-resource, the methodology is understood as domain adaptation. This paper showcases the usage of BERT's pre-trained model *bert-base-uncased* that is trained on the entire Wikipedia which is 2,500 million words and the Book Corpus which is 800 million words on classifying Spam messages by the virtue of fine-tuning.

II. LITERATURE REVIEW

Transfer Learning has been a popular interest of many researchers worldwide. There have been numerous attempts in solving several daily-life problems using Transfer Learning. There is no rigid definition of Transfer Learning [1], however, this paper follows the notion of transfer learning being a method to be able to perform a job on a target domain dataset with the help of some knowledge gained from a similar source domain dataset. This concept has been implemented in many applications including finance [2] and speech recognition [3].

Transfer Learning can take up many forms: *pretraining*, in this scenario, a model is trained on a source dataset and then the learned parameters are used to actuation of the target job; or can be used for *feature extraction*, a famous NLP method [7, 8]; another form can be *parameter sharing*, that involves transferring of specific parameters only that are found to be overlapping between the source and target domains [8, 9, 10, 11, 12]; it can also be used by *fine-tuning*, where a pre-trained model is taken and some or all of its weights are frozen and new layers are added to the downstream task so that the trained weights are left unaltered while fine-tuning, this method will be followed in this article; and Transfer Learning can also be used for direct *prediction*, in which case, a pre-trained model is directly tested using test data.

The victory of Language Models over the last few years has been prominent making room for advanced applications of language models in diverse areas of interest. Several researchers have adopted the Transfer Learning methodology for various purposes and have been able to achieve good performances. Image Recognition applications have seen a huge surge in usage of Transfer Learning approaches [4, 5, 6]. In these researches, a subset of the hyperparameters learned from the source domain are used to trigger the matching hyperparameters of the Artificial Neural Networks (ANNs) for the target domain data.

There have been many useful kinds of research for recognizing images using CNNs over years, but very few works have highlighted the use of Transfer Learning in Natural Language Processing (NLP). The authors in [13] have made use of the parameter sharing transfer learning approach to apply dynamic transfer learning for solving a very popular Natural Language Processing task, Named Entity Recognition (NER). This paper concentrates on the fine-tuning method of applying Transfer Learning by freezing the entire BERT's architecture followed by appending a dense layer and a Softmax layer at the end to form the output layer of the altered model.

III. DATASET

The dataset used to fine-tune the pre-trained bert-base-uncased model is a Comma Separated Values (CSV) file consisting of only 2 columns, one *label*, holding binary values, 0 for not spam and 1 for spam; and the other *text*, holding all the messages (spam and not spam). The dataset consists of over 5.5k rows of labeled text messages.

label	text
0	Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...
0	Ok lar... Joking wif u oni...
1	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's
0	U dun say so early hor... U c already then say...
0	Nah I don't think he goes to usf, he lives around here though
1	FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some fun you up for it still? Tb ok! XxX std chgs to send, Ã¥Â£1.50 to rcv
0	Even my brother is not like to speak with me. They treat me like aids patent.
0	As per your request 'Melle Melle (Oru Minnaminunginte Nurungu Vettam)' has been set as your callertune for all Callers. Press *9 to copy your friends Callertune
1	WINNER!! As a valued network customer you have been selected to receivea Ã¥Â£900 prize reward! To claim call 09061701461. Claim code KL341. Valid 12 hours only.
1	Had your mobile 11 months or more? U R entitled to Update to the latest colour mobiles with camera for Free! Call The Mobile Update Co FREE on 08002986030

Fig. 1. The dataset.

The pre-trained model bert-base-uncased which has already been trained on millions of words from the Wikipedia and Book Corpus data is now fine-tuned for classification on these 5.5k sentences.

IV. METHODOLOGY AND RESULTS

Since the task of handling a huge neural network having millions of hyperparameters requires a large processing unit, the code execution was done on Google Colab's GPU instance.

Pre-trained Model	Hidden Layers	Attention Heads	Hidden size	Parameters (in millions)
bert-base-uncased	12	12	768	110

Table 2. Hyperparameter configuration of bert-base-uncased.

To carry out the training and classification process, the transformers library came in really handy for importing the required pre-trained model. Along with installing the transformers library, many other necessary packages such as numpy, pandas, PyTorch were also imported. The dataset consisting of over 5.5k, to be precise, 5,572 text messages were loaded into Google Colab's runtime to be used for the fine-tuning purpose. The dataset was then split into 70% training data and 30% combined validation and testing data. In other words, the whole dataset is split into 70% training data, 15% validation data, and the rest (15%) as testing data. The data is split using a *random_state* of 2018 for the initialization of the internal random number generators for the data split to happen and also to reproduce the same data split and results over multiple executions of the code. The data splitting is also stratified with respect to the *label* column to keep the same distribution of the labels in all the splits. For instance, if the data set consists of 30% of 0s and 70% of 1s then all the splits must have the same distribution to rely on the prediction and not have biased results.

After the data splitting process, the pre-trained model and the tokenizer were loaded to convert all the textual data into tokens, i.e., every word is converted into tokens and stored in the bag-of-words. The model then assigns IDs to every token of every data split (training, validation, and testing) for the model to process as no machine can understand natural languages used by human beings. Before tokenization, the maximum sequence length was checked and set to 25 to not populate sequences with a lesser number of tokens or more padding.

The model is not passed sequences of tokens but tensors of tokens, hence, the integer series of all the data splits are converted to tensors using the torch module. The tensors are loaded into the pre-trained models using DataLoaders with *batch_size* 32. The data is first converted into tokens, then masked with 0s and 1s for padding and attention respectively, and finally coupled with the respective labels for loading. This is then sampled using the RandomSampler and ultimately loaded into the model for fine-tuning.

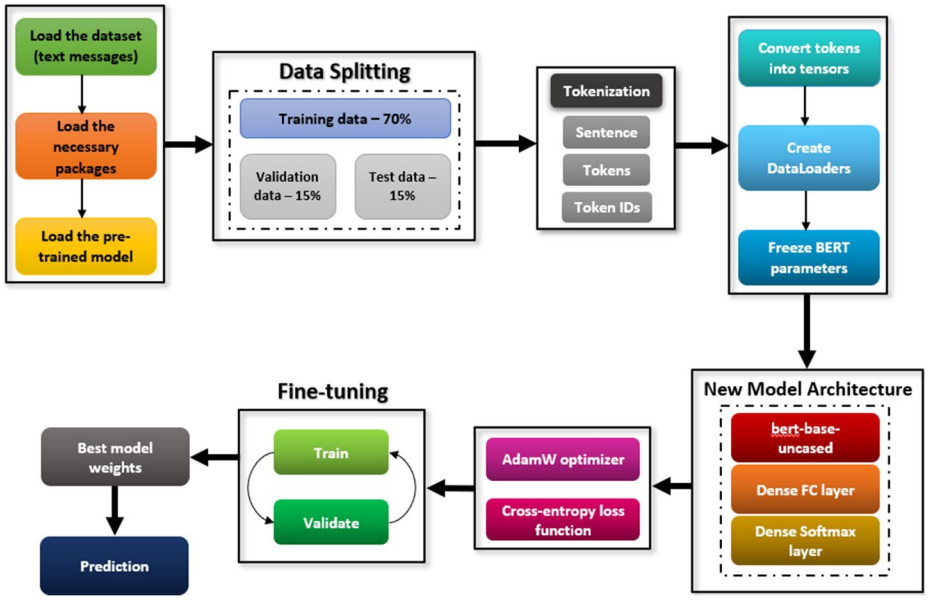


Fig. 2. The flow diagram.

Now since the approach being followed is to fine-tune bert-base-uncased, the entire architecture is frozen. Doing this prevents the parameters from being altered and helps keep the learned weights intact for further usage. The new layers are now added to the existing architecture. The existing model is subjected to Dropout at a rate of 0.1. Following this, a ReLU activation function is applied. A dense layer was then added, post which the output layer was added which is just a Softmax layer to get binary predictions. The optimizer chosen was AdamW with a learning rate of 1e-3. There was a class imbalance that was adjusted and as far as the loss function is concerned, *cross_entropy* was employed to facilitate binary classification. The model was trained for a whole of 10 epochs through the entire dataset. The model's weights were saved after every epoch from which the one with the least validation loss value was chosen as the final model. The final model gave a precision (the true positive values over total predicted positive values, true +ve, and false +ve) of 99% for the sentences labeled 0 (not spam) and 90% for the sentences labeled 1 (spam). It gave a recall score (the true positive values over total actual positive values, true +ve, and false -ve) of 98% and 92% for the not spam and spam classes respectively. The f1-score measured 98% and 91% for the not spam and spam classes respectively.

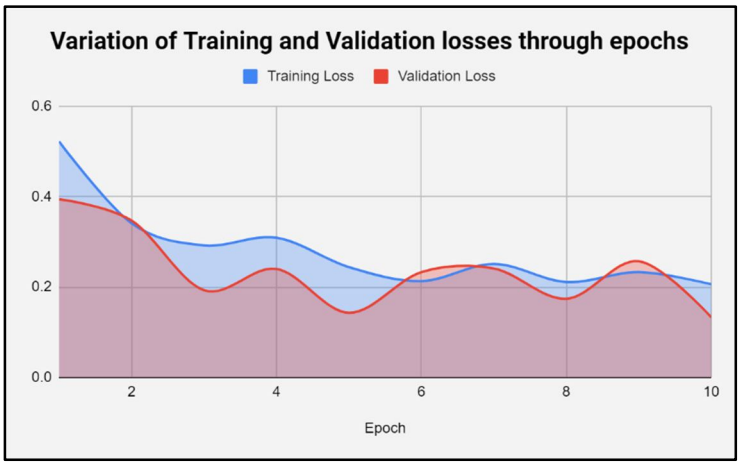


Fig. 3. Variation of Training and Validation losses through epochs.

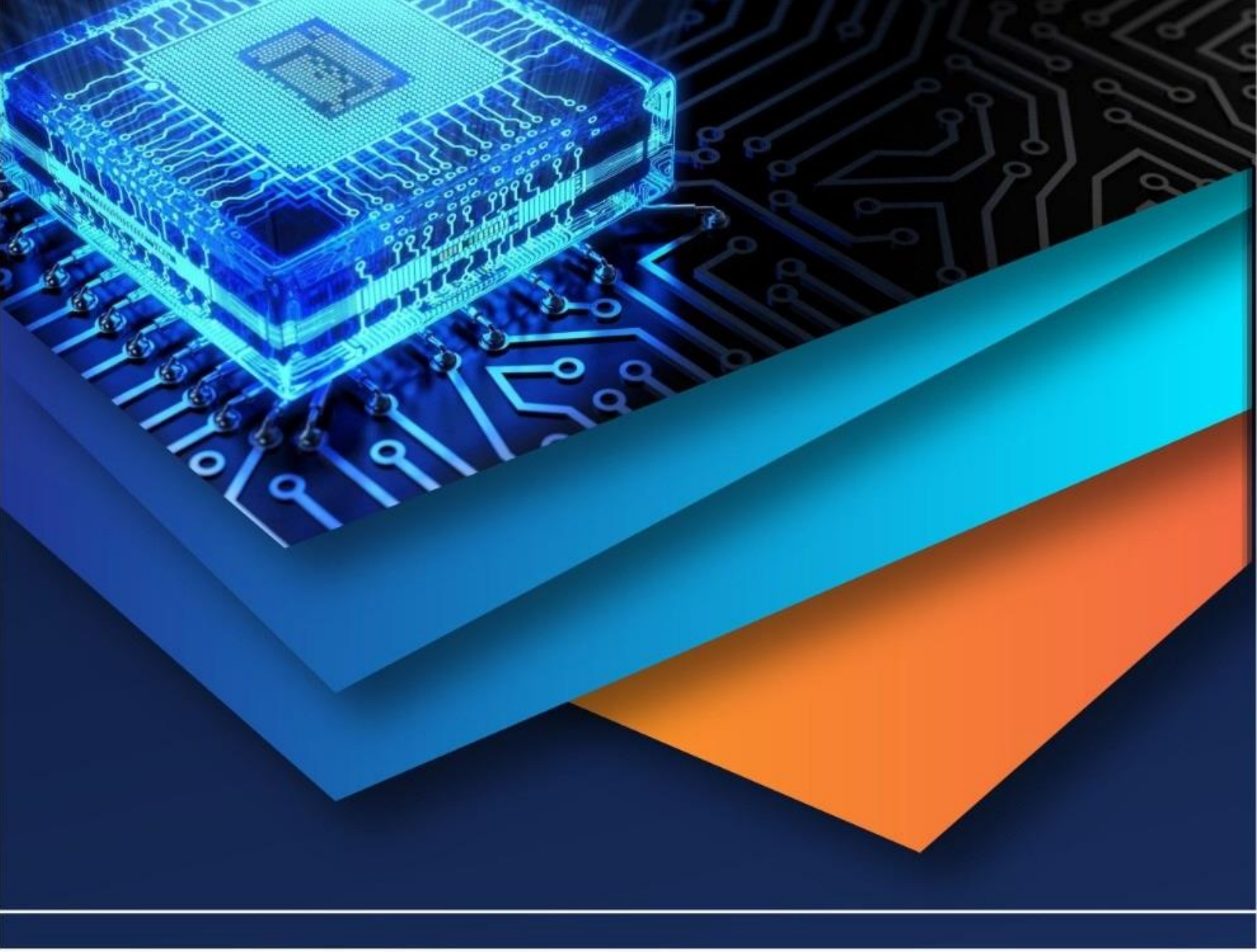
It is evident from figure 3 that the model converges through each epoch bit by bit and that both the losses are nearing 0.1 during the epoch. The model can be subjected to training for a larger number of epochs to get achieve even better results.

V. CONCLUSION

This paper showcases the use of Transfer Learning for classifying the textual sentences into binary classes, *spam*, and *not spam*. The methodology leverages the pre-trained model *bert-base-uncased* version of BERT. The novelty proposed is to determine how well can the pre-trained model utilize its previous knowledge onto classifying slightly different sentences into binary classes. The existing model layers are entirely frozen and two new layers were added to it, one fully connected (FC) layer and the other a Softmax layer to produce the output. With this approach, the already learned weights are unaffected and only the new ones are updated with every epoch. The model turns out to perform well giving an f1-score of 98% for the *not-spam* class and a 91% for the *spam* class. This model can potentially be modified further and experimented with different Transfer Learning approaches for better performances in the future.

REFERENCES

[1] Li, Q. (2012). Literature survey: domain adaptation algorithms for natural language processing. Department of Computer Science, The Graduate Center, The City University of New York, pages 8–10.
[2] Wang, D. and Zheng, T. F. (2015). Transfer learning for speech and language processing. In Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2015 Asia-Pacific, pages 1225–1237. IEEE.
[3] Stamate, C., Magoulas, G. D., and Thomas, M. S. (2015). Transfer learning approach for financial applications. arXiv preprint arXiv:1509.02807.
[4] Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks? In Advances in neural information processing systems, pages 3320–3328.
[5] Oquab, M., Bottou, L., Laptev, I., and Sivic, J. (2014). Learning and transferring mid-level image representations using convolutional neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1717–1724.
[6] Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In European conference on computer vision, pages 818–833. Springer.
[7] Bhatia, P.; Guthrie, R.; and Eisenstein, J. 2016. Morphological priors for probabilistic neural word embeddings. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 490–500.
[8] Peng, N., and Dredze, M. 2017. Multi-task domain adaptation for sequence tagging. In Proceedings of the 2nd Workshop on Representation Learning for NLP, 91–100.
[9] Yang, Z.; Salakhutdinov, R.; and Cohen, W. 2016. Multi-task cross-lingual sequence tagging from scratch. arXiv preprint arXiv:1603.06270.
[10] Fan, X.; Monti, E.; Mathias, L.; and Dreyer, M. 2017. Transfer learning for neural semantic parsing. arXiv preprint arXiv:1706.04326.
[11] Guo, H.; Pasunuru, R.; and Bansal, M. 2018b. Soft layer-specific multi-task summarization with entailment and question generation. arXiv preprint arXiv:1805.11004.
[12] Wang, Z.; Qu, Y.; Chen, L.; Shen, J.; Zhang, W.; Zhang, S.; Gao, Y.; Gu, G.; Chen, K.; and Yu, Y. 2018. Label-aware double transfer learning for cross-specialty medical named entity recognition. arXiv preprint arXiv:1804.09021.
[13] Parminder B.; Kristjan A.; and Busra C. 2020. Dynamic Transfer Learning for Named Entity Recognition. arXiv:1812.05288v4 [cs.LG] 20 Jan 2020.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY