iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** V **Month of publication:** May 2022

DOI: https://doi.org/10.22214/ijraset.2022.43070

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: 7.538
Volume 10 Issue V May 2022- Available at www.ijraset.com

Stance Prediction of Tweets on Farmers Protests in India

Aditya Kamble[1], Prathamesh Badgujar[2], Anuj Kadam[3], Dhruv Shah[4], A.J. Kadam[5]
[1, 2, 3, 4, 5]*Department of Computer Engineering, AISSMS College of Engineering*, Pune, Maharshtra, India

Abstract: ***Protests are an integral part of democracy and are a vital tool for the general public to convey their demands and/or discontentment to the ruling government. As voters return to term with any new rules, there are an increasing range of protests everywhere in the world for numerous socio-political reasons.***
With the advancement of technology, there has additionally been an exponential rise within the use of social media for the exchange of data and ideas.
During this research, knowledge was gathered from the web site "twitter.com", regarding farmers' protest to know the feelings that the public shared on a global level.
Sadly now since the Farm Laws are repealed, we have a tendency to aim to use this knowledge to know the general public stance on these laws, and whether or not it affected the government's decision.
This paper proposes a stance prediction deep learning model achieved after fine tuning the well known ULMFiT (Universal Language Model Fine-tuning) model by Howard and Ruder. Categories to be classified into are For (F), Against (A) and Neutral (N). Proposed model achieved an F1 score of 0.67 on our training and test data, which is essentially a labeled subset of the actual data.
Keywords: Dataset, ULMFiT, deep learning, text classification, Language Model (LM)

I. INTRODUCTION

A. Motivation

The 2020–2021 Indian farmers' protest changed into a protest in opposition to 3 farm acts that have been passed through the Parliament of India in September 2020.
The acts, frequently known as the Farm Bills, were termed as "anti-farmer legal guidelines" through many farmer unions, and politicians from the competition who say it might go away farmers at the "mercy of corporates". The union authorities, however, continue that the legal guidelines will make it easy for farmers to promote their produce immediately to large buyers, and said that the protests are primarily based totally on misinformation. Despite India being in large part self-enough in foodgrain manufacturing and having welfare schemes, starvation and nutrients stay in severe troubles, with India rating as one of the worst nations withinside the international in meals safety parameters.
Soon after the acts were introduced, unions commenced protecting nearby protests, normally in Punjab. After months of protests, farmer unions—specially from Punjab and Haryana—commenced a motion named "Dilli Chalo" (translate. Let's go to Delhi), wherein tens of lots of farming union participants marched closer to the nation's capital. The Indian authorities ordered the police and regulation enforcement of numerous states to assault the protesters using water cannons, batons, and tear gas to save the farmer unions from stepping into Haryana first after which Delhi.
November 2020 noticed a national well known strike in aid of the farmers and lots converging at numerous border factors at the manner to Delhi.
The Supreme Court of India ordered a stay on the implementation of the farm legal guidelines in January 2021. Farmer leaders welcomed the live order, which stays in effect. Six country governments (Kerala, Punjab, Chhattisgarh, Rajasthan, Delhi and West Bengal) passed resolutions in opposition to the farm acts, and 3 states (Punjab, Chhattisgarh and Rajasthan) have tabled counter-regulation of their respective country assemblies.
The main objective of this research is to understand the stance of the public on farmers' protest shared on the microblogging website "Twitter". Our research mainly aims at analyzing factuality and polarity of twitter data using a deep learning model called ULMFiT.

©IJRASET: All Rights are Reserved

B. Inductive Transfer Learning

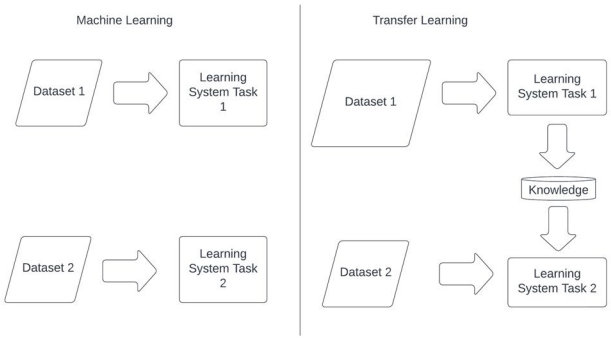


Fig 1. Traditional Machine Learning vs Transfer Learning

Many state-of-the-art models in NLP need to learn from scratch and require big datasets to attain affordable results, they are no longer handiest soak up big portions of reminiscence however also are pretty time consuming. Specifically in textual content classification, there may not also be sufficient categorized examples to start with. Inductive transfer learning tackles precisely those challenges [13]. It is likewise the principal idea ULMFiT is primarily based totally on.

Transfer learning pursuits to imitate the human capacity to collect knowledge at the same time as learning one task and to then make use of this understanding to remedy some related task. The conceptual distinction to conventional machine learning is displayed in Figure 1. In the conventional approach, for instance, two models are trained one at a time without both maintaining or moving knowledge from one to the other. An instance for switch studying alternatively might be to hold knowledge (e.g. weights or features) from training a model 1 and to then make use of this knowledge to train some other model. In this case, model 1 might be referred to as the source task and model 2 the target task [14].

C. Overview of ULMFiT Model



Fig. 2 Overview of ULMFiT Model

1) *General-Domain Language Model Pretraining:* In a primary step, a Language Model is pretrained on a huge general-area corpus (in our case the WikiText-103 dataset). Now, the model is capable of predicting the subsequent phrase in a chain with a positive diploma of certainty. At this degree the model learns the overall features of the language. Pre Training is most useful for small datasets and allows generalization regardless of size of the dataset. Although this step is expensive, it only needs to be carried out once and improves overall performance and convergence of downstream models.
2) *Target Task Language Model Fine-Tuning:* Following the transfer learning approach, the know-how received withinside the first step need to be applied for the goal assignment. However, the target dataset (i.e. the Farmers Protest Tweets dataset) is probable from a exclusive distribution than the source dataset. To deal with this issue, the Language Model is therefore fine-tuned at the records of the target dataset. Just as after the primary step, the model is at this factor capable of expect the subsequent phrase in a chain. Now however, it has additionally discovered assignment-unique functions of the language, consisting of the lifestyles of handles in Twitter or using slang and some indian phrases. Regardless of how diverse the general-domain information used for pre-training is, the information of the goal task will probably come from a different distribution. We therefore fine-tune the Language Model on data of the target task. Given a pre-trained general-area Language Model, this stage converges quicker because it simplest desires to evolve to the idiosyncrasies of the target data, and it permits us to educate a robust Language Model even for small datasets.

3) *Target Task Classifier:* Fine-tuning the target classifier is the maximum vital part of the transfer learning approach. Overly competitive fine-tuning will cause catastrophic forgetting, putting off the gain of the information captured via language modeling; too cautious fine-tuning will cause gradual convergence and result in overfitting the data. Besides discriminative finetuning and triangular studying rates, Howard and Ruder[10] proposed the gradual unfreezing for fine-tuning of the classifier. Rather than fine-tuning all layers at once, which risks catastrophic forgetting, Howard et al[10] endorsed to regularly unfreeze the model beginning from the closing layer as this carries the least widespread knowledge. Howard et al[10] first unfroze the closing layer and fine-tracked all of the unfrozen layers for one epoch. They then unfroze the subsequent lower frozen layer and repeated, till they fine tuned all layers till convergence on the closing iteration. Since ultimately, in our case, we do now no longer need our model to predict the subsequent phrase in a chain however to offer a stance classification, in a 3rd step the pretrained Language Model is extended through linear blocks in order that the very last output is a possibility distribution over the stance labels (i.e. For (F) , Against (A) and Neutral (N)).

a) *Gradual unfreezing:* Rather than one time fine-tuning of all layers, which risks in the creation catastrophic forgetting, we propose to step by step unfreeze the version beginning from the last layer as this consists of the least trendy knowledge We first unfreeze the remaining layer and fine-music all unfrozen layers for one epoch. We then unfreeze the subsequent decreased frozen layer and repeat, till we finetune all layers till convergence on the remaining iteration.

b) *Backpropagation Through Time (BPTT) for Text Classification (BPT3C):* Since the model architecture for training and fine-tuning is that of an LSTM, the paper[10] implements the backpropagation through time(BPTT) approach to be able propagate gradients without them exploding or disappearing. In order to make fine-tuning a classifier for big documents feasible,Howard et al[10] proposed BPTT for Text Classification (BPT3C): The document gets divided into fixed length batches. At the start of every batch, the version is initialized with the very last state of the preceding batch; a track of the hidden states for mean and max-pooling is kept; gradients are back-propagated to the batches whose hidden states contributed to the very last prediction. In practice, variable duration backpropagation sequences are used.

D. Steps in BPT3C

1) The record is split into constant duration batches.
2) At the start of every batch, the model is initiated with the very last state of the preceding batch with the aid of maintaining the tune of the hidden states for mean and max-pooling.
3) The gradients are back-propagated to the batches whose hidden states contributed to the very last prediction.

II. RELATED WORK

For many years, reduction operations, comparable to stemming or lemmatization, still as shallow models, such as SVMs, had dominated NLP [2]. Young et al. [2] claim that introduction of word embeddings, with its most outstanding pre-trained examples word2vec [3] and GloVe [4], ultimately led the approach for the success of deep learning in NLP.

One among the most criticisms relating to pre-trained word embeddings, however, is that they solely transfer antecedently learned data to the primary layer on a neural network, whereas the remainder of it still must be trained from scratch [1].

McCann et al. [7] used an associate degree encoder of a supervised neural AI to contextualize word embeddings and finally concatenated these vectors with the pre trained word embeddings. Neelakantan et al. [6] experimented with coaching individual vectors for every word sense. These approaches tackle the problem of missing context. However, they do have to train the actual task model from scratch.

In their seek for novel approaches, several NLP researchers looked to strategies that had antecedently proven thriving in Computer Vision (CV). Ruder [1] claims that language modeling is especially suited to capturing sides of language that are vital for target tasks. A lot of outstanding models that support this approach are Embeddings from Language Models (ELMo), the OpenAI Transformer and Universal Language Model Fine-tuning (ULMFiT) [10]. The OpenAI Transformer is similar to ELMo however it needs some minor changes within the model design for transfer [12]. Both (ELMo and the OpenAI Transformer) are proven to provide superb empirical results.

Except for achieving progressive leads to varied tasks, ULMFiT incorporates many fine-tuning techniques that are loosely applicable and could boost performance for alternative strategies as well, for example the OpenAI Transformer.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue V May 2022- Available at www.ijraset.com

III. DATASET USED

A. For Language Model Training

We mainly used data from a well known website Kaggle.com.

The name of the dataset is "Farmers Protest Tweets Dataset", which contains 2 files, first one is the one containing actual tweets extracted from twitter.com having hashtag "#FarmersProtest" and the second one containing data about the users who made those tweets. Data for tweets is collected using the Twitter API through the snscrape Python library. The first (Tweets) dataset has around 855850 rows and 14 columns and the second dataset has around 169000 rows and 19 columns. We used only the tweets dataset for training the language model. We only kept the actual tweets column called "renderedContent" and discarded all other columns since they were useless for our task.

B. For Stance Detection Classification

For stance detection, we used a subset of the aforementioned tweets dataset. We manually labeled 12000 tweets from the tweets dataset as For (F), Against (A) or Neutral (N). The distribution of tweets found is shown below –

```
In [30]: len(df[df['stance']=='A'])
Out[30]: 727
In [31]: len(df[df['stance']=='F'])
Out[31]: 8802
In [32]: len(df[df['stance']=='N'])
Out[32]: 2349
```

Fig. 3 Overview of the dataset used for training classification model

As indicated by the above figure, there were very few tweets which were against the Protest as compared to Supporting and Neutral ones. This would lead to an imbalanced dataset and consequently a biased model towards positive and neutral stances.

To tackle this problem, we used a technique called artificial supersampling. In this technique, we translated each tweet classified as "A" to some other language like French, German etc. and then translated it back to English, till the number of samples classified as "A" were equal to those classified as "F" and "N".

Finally, we chose 2500 random samples from each category to train the stance detection (Text Classification) model and put 15% from them for testing.

IV. SYSTEM ARCHITECTURE

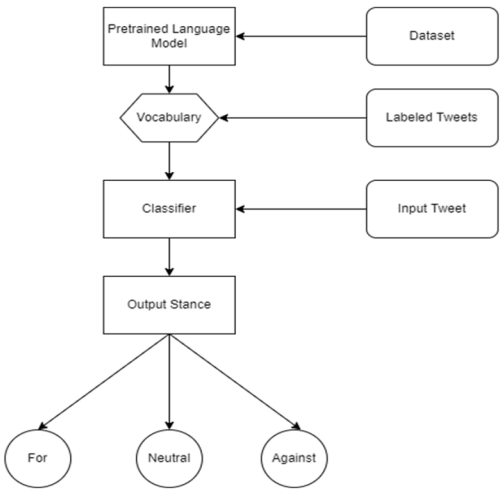


Fig. 4 System architecture of the proposed system

©IJRASET: All Rights are Reserved

As shown in Fig. 4, proposed system mainly consists of 4 steps –

1) Processing the input data and feeding it to the ULMFiT model.
2) Training the model on our data and generating vocabulary for text classification.
3) Training the classification model using the generated vocabulary and labeled data.
4) Performing multi-class classification using BPT3C model trained in previous step

V. METHODOLOGY

A. Preprocessing the data

As mentioned in section III, we used a dataset from Twitter having 920000 tweets. However, some of these tweets were duplicates. So as an initial step, we dropped these duplicates, leaving us with about 855850 unique tweets.

As the next step, we dropped all unnecessary columns. The main tweets column called "renderedContent" was used for language model training, so we cleaned the tweets as our next step. As a preprocessing step for actual text data, we removed all the links to websites and other stuff because they would not add any value to our model. Next, we removed all unnecessary punctuations and whitespaces in between as well as at the end of the tweets. We however decided to keep the hashtags as well as emojis in the tweets because they contribute to actual knowledge of our model as well. Please refer to the example given below

Raw Tweet	They can't be farmers. Looks like Gundas are having a good time. They seem to be violence thirsty goons. #FarmersProtest twitter.com/IndiaToday/sta…
Cleaned Tweet	They can't be farmers. Looks like Gundas are having a good time. They seem to be violence thirsty goons. #FarmersProtest

B. Performing exploratory data analysis

We further performed some basic exploratory data analysis on this data by joining the tweets and users datasets, thus providing us with the information of tweets as well as the users who made these tweets. Through our analysis, we were able to answer some questions like how was the trend of tweets across time frame of Nov 2020 till Jul 2021, at what time of the day were there most number of tweets, who were the top 10 famous people who tweeted about this topic, number of replies received to their tweets, etc.

C. Training the ULMFiT language model

AS mentioned in the previous section, the only column needed for the language model is the actual text column. Hence we dropped all other columns from the tweets dataset except for "renderedContent". All these tweets were supplied to the ULMFiT model pre-trained on the Wikipedia dataset. We divided the dataset into 2 parts of 90% and 10% for training and validation respectively. The fastai library converts the text into its own format for better processing and understanding of the data. As the next step, we found the optimal learning rate for training of our language model, which came out to be 0.00209.

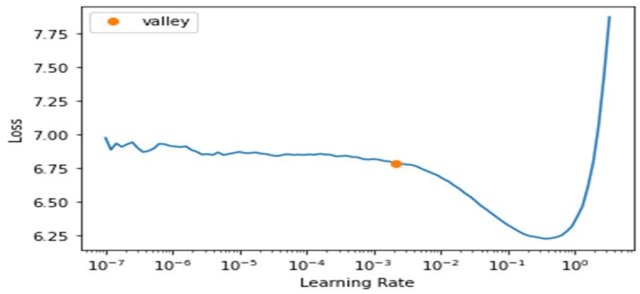


Fig. 5 Finding optimal learning rate for the language model

After that, we used this learning rate to train our language model using ULMFiT recommended method of training one layer while others are frozen. Finally, our language model achieved accuracy of 0.45, which is the accuracy of predicting next words, given the current word sequence. We saved this model for use in a text classifier.

D. Training the Stance Prediction model (Text Classification Model)

As a final step, we used the manually labeled dataset mentioned in section III, which were about 7300 labeled tweets. We used the vocabulary from the language model as features in our text classification model to predict 3 labels namely F (For), A (Against) and N (Neutral) using 80% and 20% data for training and validation respectively. The classifier model achieved a respectable F1 score of 0.67 and accuracy of about 0.7. We exported this model as a .pkl file and used it for prediction.

E. Deploying the Model

We used flask, a famous lightweight web framework for deploying our model on the web. We created a simple HTML form to take a tweet as input from the user, preprocess it, feed it to the model generated in the previous step, to give the output as either F, N or A.

VI. RESULTS

A. Transfer Learning (Language Model Training)

We trained the original ULMFiT model by providing our dataset for better understanding of the language as well as the topic in consideration. The tweets made by Indians are not entirely in English and there are certain words/phrases written in tweets for better impact/understanding of the tweet. The performance of the model is measured by how accurately it can predict the next set of words by looking at the current set of words. The metric used for this task was accuracy. Our model achieved nearly 45% accuracy in the task, which is considered very respectable.

epoch	train_loss	valid_loss	accuracy	time
0	3.485758	3.418579	0.384397	2:03:51
1	3.364655	3.255469	0.409624	2:03:36
2	3.219054	3.158205	0.424908	2:03:31
3	3.105550	3.104273	0.434011	2:03:31
4	3.089722	3.066870	0.440533	2:03:30
5	3.027117	3.041426	0.444992	2:03:29
6	2.953732	3.025707	0.447964	2:03:32
7	2.952369	3.014702	0.450352	2:03:34

Fig. 6 Language model performance

B. Text Classification

We used the language model as vocabulary or feature vectors for our main task, that was stance detection of tweets. We call it an extended model The performance metrics used were accuracy initially. But accuracy tends to favor the most dominant class in the dataset generally, because it only considers how many predictions were correct out of the entire predictions made. Hence we decided to add another performance metric for comparison, that was F1 score.

The extended model achieved an accuracy of 67.76% and F1 score of 68.18%. In order to determine if the model actually learned anything from our dataset, we compared our new model with the base ULMFiT language model, which was not trained on our dataset i.e. which never saw the vocabulary of the tweets, only the vocabulary it learned from its training on Wikipedia 103 dataset. The text classification model trained on a base ULMFiT model achieved an accuracy of 64.7% and an F1 score of 64.3% respectively.

epoch	train_loss	valid_loss	accuracy	f1_score	time
0	0.674457	0.778153	0.635807	0.638622	00:20
1	0.644133	0.760406	0.651464	0.647435	00:20
2	0.588668	0.766110	0.646018	0.647746	00:20
3	0.505813	0.801894	0.646698	0.643432	00:20

Fig. 7 Classification model trained on base ULMFiT language model

©IJRASET: All Rights are Reserved

epoch	train_loss	valid_loss	accuracy	f1_score	time
0	0.575358	0.711336	0.672109	0.675506	00:23
1	0.552132	0.715280	0.685034	0.689217	00:23
2	0.474403	0.738644	0.680952	0.686000	00:23
3	0.391327	0.800225	0.683673	0.685779	00:23
4	0.313231	0.893396	0.677551	0.681188	00:23

Fig, 8 Classification model trained on extended ULMFiT language model

So we can say that the model actually learned the vocabulary from the dataset and that helped it achieve better performance, albeit very slight. Overall, 3% better performance achieved by just ingesting the topic relevant data and training on it with some moderate hardware seems to be worth the effort.

	Metrics	
	Accuracy	F1 score
Extended Model	67.76%	68.18%
Base Model	64.7%	64.3%

VII. CONCLUSION AND FUTURE SCOPE

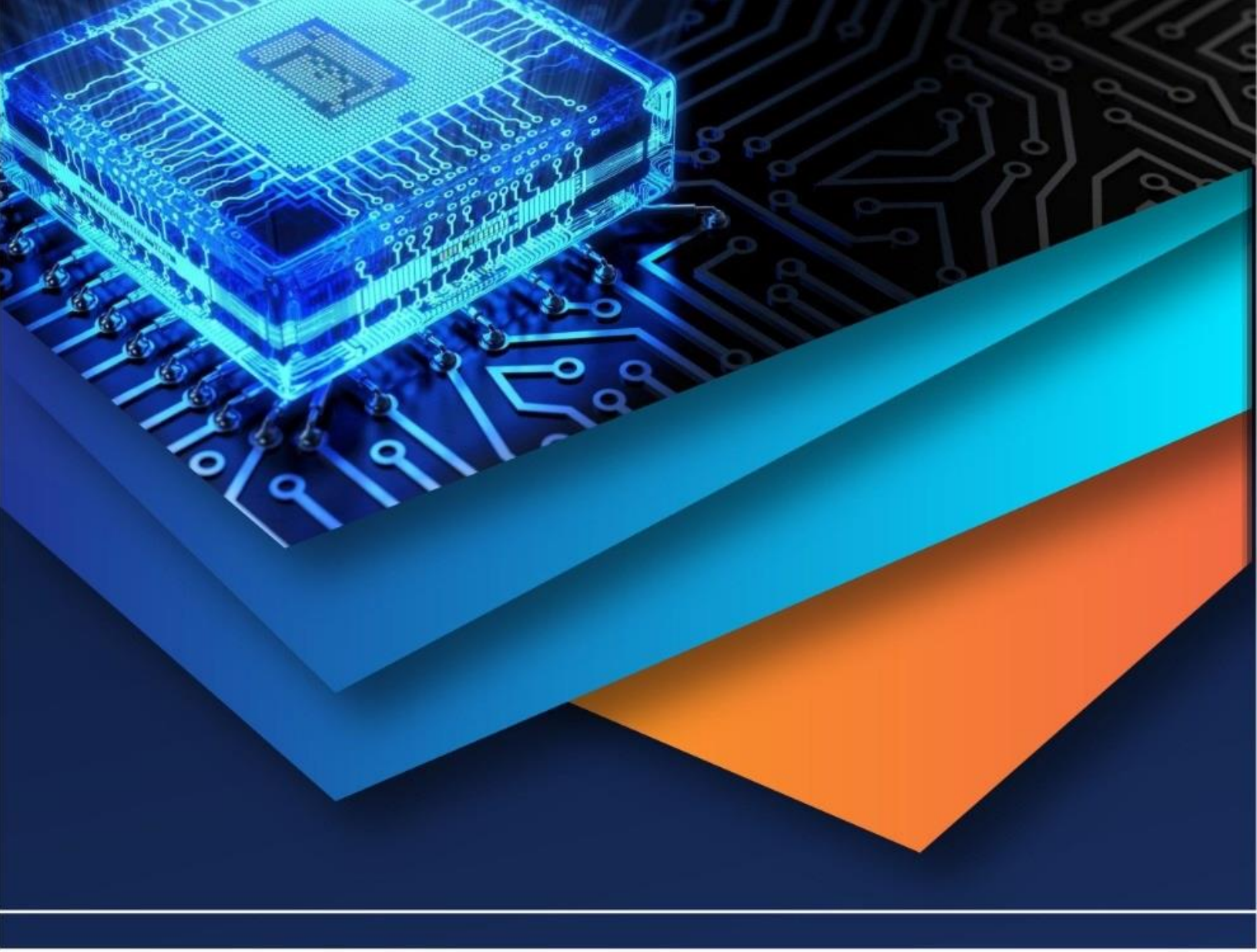
In this research, we deduced an extension to an existing state of the art model and tried to compare it to the original model, which evidently showed to have somewhat better performance. The relevant data helps models to understand the topic better.
Further, this model can also be used to classify tweets written in some other languages like Marathi, Punjabi etc. by providing respective language data. The ULMFiT model can be used for any type of text classification, not only stance.

REFERENCES

[1] Sebastian Ruder, NLP's ImageNet moment has arrived, The Gradient, Jul. 8, 2018, https://thegradient.pub/nlp-imagenet/ [Blog]
[2] Tom Young, Devamanyu Hazarika, Soujanya Poria and Erik Cambria, Recent Trends in Deep Learning Based Natural Language Processing, IEEE Computational Intelligence Magazine, vol. 13, issue 3, pp. 55-75, Aug. 2018
[3] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado and Jeffrey Dean, Distributed Representations of Words and Phrases and their Compositionality, arX
[4] iv:1310.4546v1 [cs.CL], Oct. 16, 2013
[5] Jeffrey Pennington, Richard Socher and Christopher D. Manning, GloVe: Global Vectors for Word Representation, Computer Science Department, Stanford University, 2014
[6] Matthew E. Peters, Waleed Ammar, Chandra Bhagavatula and Russell Power, Semi-supervised sequence tagging with bidirectional language models, arXiv:1705.00108v1 [cs.CL], Apr. 29, 2017
[7] Arvind Neelakantan, Jeevan Shankar, Alexandre Passos and Andrew McCallum, Efficient Non-parametric Estimation of Multiple Embeddings per Word in Vector Space, arXiv:1504.06654v1 [cs.CL], Apr. 24, 2015
[8] Bryan McCann, James Bradbury, Caiming Xiong and Richard Socher, Learned in Translation: Contextualized Word Vectors, arXiv:1708.00107v2 [cs.CL], Jun. 20, 2018
[9] Kaiming He, Georgia Gkioxari, Piotr Dollár and Ross Girshick, Mask R-CNN, arXiv:1703.06870v3 [cs.CV], Jan. 24, 2018
[10] Joao Carreira, Andrew Zisserman, Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset, arXiv:1705.07750v3 [cs.CV], Feb. 12, 2018
[11] Jeremy Howard, Sebastian Ruder, Universal Language Model Fine-tuning for Text Classification, arXiv:1801.06146, Jan. 18, 2018
[12] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee and Luke Zettlemoyer, Deep contextualized word representations, arXiv:1802.05365v2 [cs.CL], Mar. 22, 2018
[13] Alec Radford, Karthik Narasimhan, Tim Salimans and Ilya Sutskever, Improving Language Understanding by Generative Pre-Training, OpenAI Blog, Jun. 11, 2018
[14] Sinno Jialin Pan, Qiang Yang, A Survey on Transfer Learning, IEEE Transactions on Knowledge and Data Engineering, vol. 22, issue 10, Oct. 2010
[15] Dipanjan Sarkar, A Comprehensive Hands-on Guide to Transfer Learning with Real-World Applications in Deep Learning, Towards Data Science, Nov. 14, 2018 [Blog]
[16] Jeremy Howard, Lesson 10: Deep Learning Part 2 2018 - NLP Classification and Translation, https://www.youtube.com/watch?v=h5Tz7gZT9Fo&t=4191s%5D, May 7, 2018 [Video]
[17] Ben Krause, Emmanuel Kahembwe, Iain Murray and Steve Renals, Dynamic Evaluation of Neural Sequence Models, arXiv:1709.07432, Sep. 21, 2017
[18] Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov and William W. Cohen, Breaking the Softmax Bottleneck: A High-Rank RNN Language Model, arXiv:1711.03953, Nov. 10, 2017

[19] Jason Brownlee, What Are Word Embeddings for Text?, Machine Learning Mastery, posted on Oct. 11, 2017, https://machinelearningmastery.com/what-are-word-embeddings/
[20] Stephen Merity, Nitish Shirish Keskar, Richard Socher, Regularizing and Optimizing LSTM Language Models, arXiv:1708.02182v1 [cs.CL], Aug. 8, 2017
[21] Diederik P. Kingma, Jimmy Ba, Adam: A Method for Stochastic Optimization, arXiv:1412.6980v9 [cs.LG], Jan. 20, 2017
[22] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever and Ruslan Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, The Journal of Machine Learning Research, vol. 15, issue 1, pp. 1929-1958 Jan. 2014
[23] Michael Nielsen, Improving the way neural networks learn, Neural Networks and Deep Learning, posted in Oct. 2018, http://neuralnetworksanddeeplearning.com/chap3.html
[24] Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan and Sune Lehmann, Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm, arXiv:1708.00524v2 [stat.ML], Oct. 7, 2017
[25] Leslie N. Smith, Cyclical Learning Rates for Training Neural Networks, arXiv:1506.01186v6 [cs.CV], Apr. 4, 2017
[26] Jason Yosinski, Jeff Clune, Yoshua Bengio and Hod Lipson, How transferable are features in deep neural networks?, arXiv:1411.1792v1 [cs.LG], Nov. 6, 2014
[27] Sebastian Ruder, An overview of gradient descent optimization algorithms, arXiv:1609.04747v2 [cs.LG], Jun. 15, 2017
[28] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler und Sepp Hochreiter, GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium, arXiv:1706.08500v6 [cs.LG], Jan. 12, 2018
[29] Xiang Jiang, Mohammad Havaei, Gabriel Chartrand, Hassan Chouaib, Thomas Vincent, Andrew Jesson, Nicolas Chapados and Stan Matwin, Attentive Task-Agnostic Meta-Learning for Few-Shot Text Classification, ICLR 2019 Conference Blind Submission, Sep. 28, 2018
[30] Arun Rajendran, Chiyu Zhang and Muhammad Abdul-Mageed, Happy Together: Learning and Understanding Appraisal From Natural Language, Natural Language Processing Lab, The University of British Columbia, 2019
[31] Roopal Garg, Deep Learning for Natural Language Processing: Word Embeddings, datascience.com, posted on Apr. 26, 2018 [Blog]

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)