Transfer Learning with Augmented Vocabulary for Tweet Classification

Vineet Kumar[1], Karthikeya Racharla[1], and Debapriyo Majumdar[2]

[1] IIT Kharagpur, [2] ISI Kolkata

6th IEEE International Conference on Multimedia Big Data

26th September 2020

Introduction

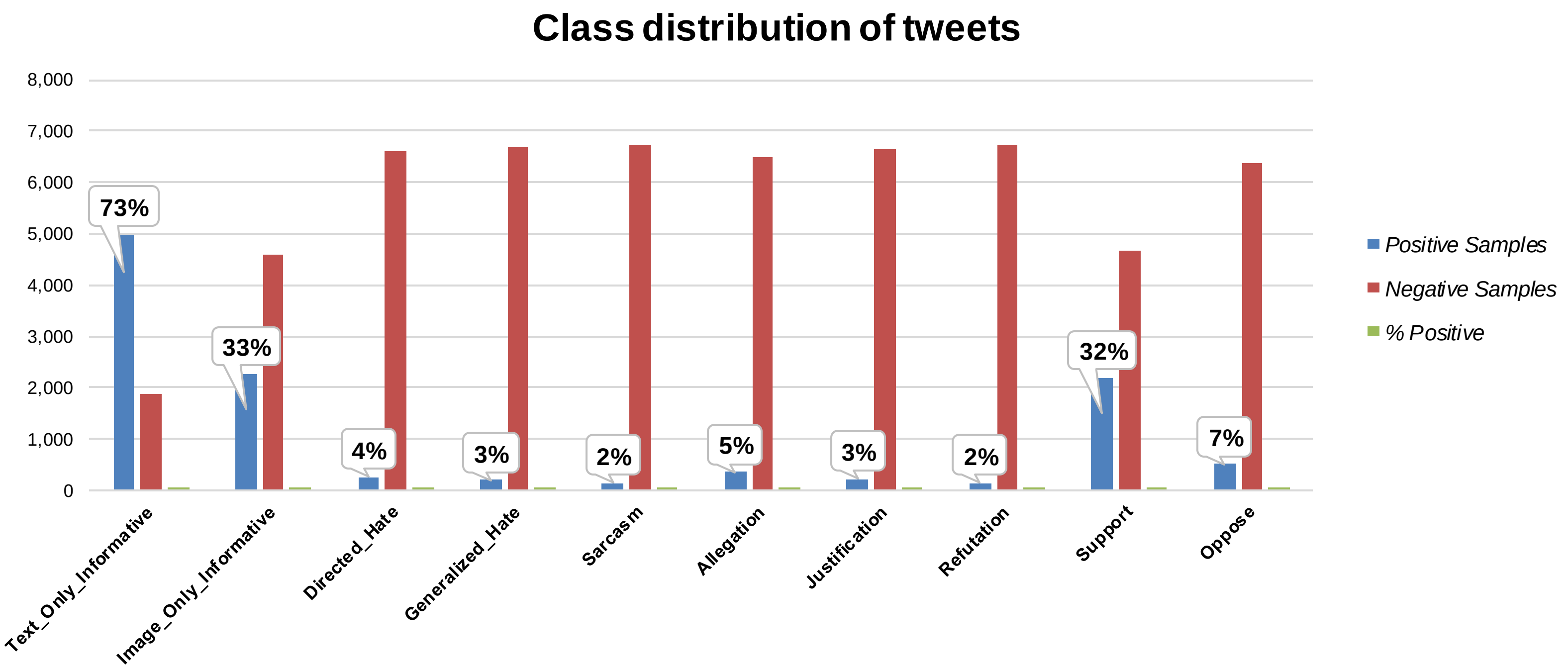
- What is our Paper all about?
  - Our approach and methodology of our participation in IEEE **BigMM** Grand Challenge (BMGC), 2020
  - The challenge was aimed at research towards deeper understanding of multiple facets involved in *#MeToo* movement
- **Problem Context:** Classify a set of tweets pertaining to the *#MeToo* movement based out of five linguistic aspects
- Our best performing approach (team name: entropy) has **ranked first** on the leader-board when the grand challenge finished, achieving the AUC of **0.56365**

Dataset Description

- Kaggle-hosted competition
- Due to Twitter legal requirements, the dataset for the challenge only had tweet-ids and their corresponding labels
- Participants had to download tweets directly, using tweet-ids
  - Training set - close to **8,000** tweet IDs and **10** labels
  - Test set - close to **2,000** tweet IDs *(released at the later part of challenge)*
- ***Tweet Hydration:*** Extracted corresponding tweets associated with their tweet-IDs using **Hydrator API**
- We were able to hydrate only -
  - 6,900+ tweets from Train set *(rest 14% tweets were deleted)*
  - 1,700+ tweets from Test set *(rest 13% tweets were deleted)*

Dataset Description

- 10 data labels on *Relevance, Hate Speech, Dialogue Acts, Stance, Sarcasm*
- *Evaluation metric:* Mean column-wise AUC (**A**rea **U**nder the ROC **C**urve)
- Data suffers with severe class-imbalance



Feature Extraction from Text data

- Basic tweet pre-processing with analysis restricted to the textual part of tweets
- Removal of stop-words found to hamper classification performance
- **Vectorization**: Process of transforming the collection of texts in a corpus to a numerical representation of feature vectors
  - We made use of term-weighting based vectorization approaches
  - ***TfidfVectorizer()*** to convert the raw tweets into representable TF-IDF matrix forms
- **Tokenization:** Indexed dictionary of all the tokens in the tweet corpus, then vectorized each token by turning it into sequences of integers using indices of the token dictionary
  - The coefficients corresponding to each token were extracted using these approaches, followed by zero-padding (for uniform length)
  - ***Tokenizer()*** class provided by *keras* was used

Our Approaches

- Baseline ML Models
  - Logistic Regression
  - GaussianNB
  - Support Vector Machines
- Tree-based ML Classifiers
  - RandomForest
  - XGBoost
  - LightGBM
- Deep-Learning Frameworks
  - Bi-LSTM (Bidirectional LSTM) with Glove Embeddings
- Transfer Learning Architectures
  - BERT (Transformer-based Approach)
  - ULMFiT (Domain specific language-model with Fine-tuning)

Methodology: ML Classifiers

- *Logistic Regression, MNB* and *SVM* didn't produce encouraging results in comparison to Tree-based classification approaches
- *XGBoost* has slightly improved the AUC score – able to address class imbalance while training
  - Overall, *XGBoost* has produced 0.5182 mean-AUC on the Leaderboard
- We chose final model for each class based on AUC score obtained on our validation set
- For these 5 labels, *XGBoost* and *RandomForest* were found to be better

Classifiers	Class label	AUC on Validation set
XGBoost	*Image_Only_Informative*	0.539
	Sarcasm	0.535
	Justification	0.505
	Refutation	0.578
RandomForest	*Allegation*	0.515

Methodology: BiLSTM

- Bidirectional LSTMs generally work better with sequential classification tasks
- Methodology
  - Used pretrained glove-twitter embeddings (trained on 2B uncased tweets)
  - Implemented BiLSTM using Bidirectional layer wrapper (Keras)
  - Introduced dropout for regularization & Conv1D filter for context capturing
- Results were not much encouraging

Labels / Model	BiLSTM
Support	0.50
Oppose	0.49
Directed Hate	0.52
Generalized Hate	0.51

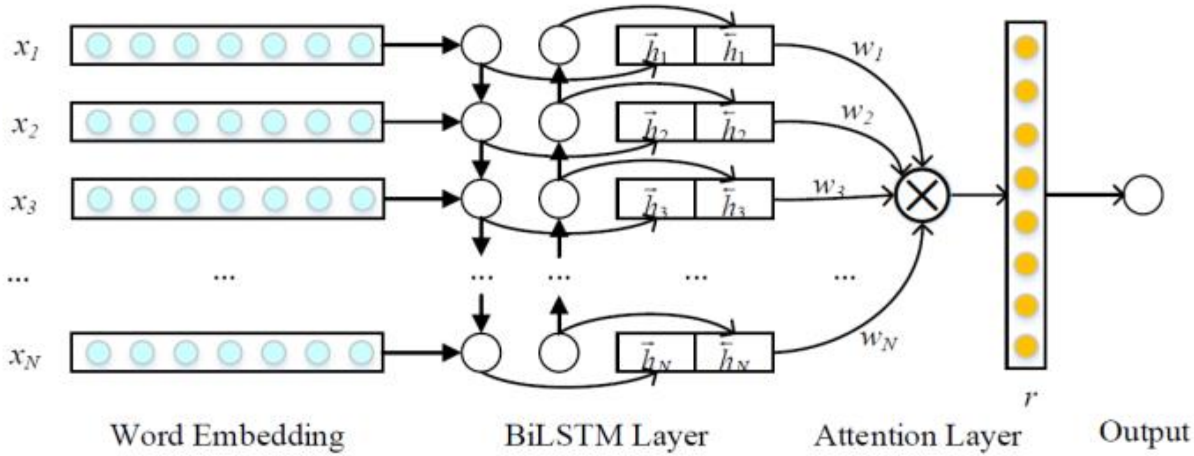


Image credits: https://www.aclweb.org/anthology/S18-1040.pdf

Methodology: BERT

- Getting a lot of exactly curated training data is always a challenge
- To perform well, deep learning models need to be trained with a lot of data
- BERT outperformed many state-of-the-art NLP / NMT tasks
- Our methodology
  - Used `BertForSequenceClassification` class
  - Fine tuned with our dataset using AdamW optim
  - Used 2×10^{-5} as learning rate & 10^{-8} as **eps** value
- Results: Achieved 67% overall accuracy

Labels / Models	BERT	BERT + ROS
Directed Hate	0.51	0.51
Generalized Hate	0.50	0.50

- Most of our vocabulary was unrecognised by BERT

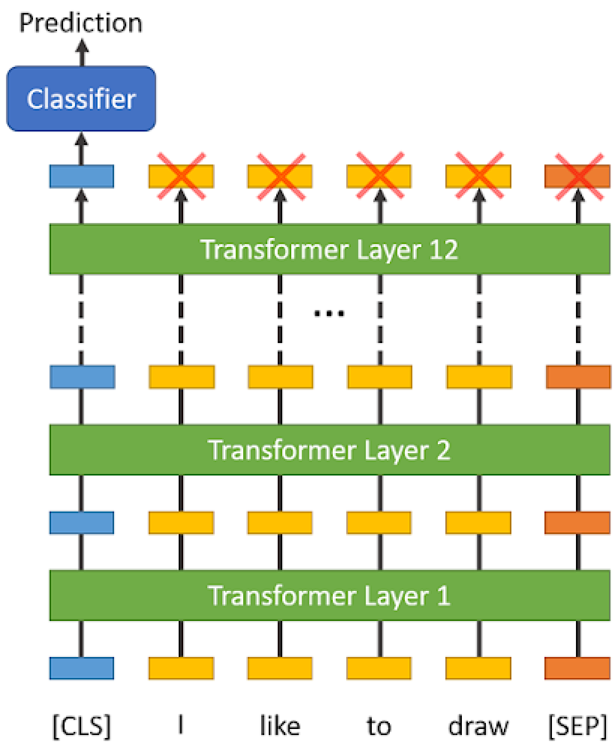
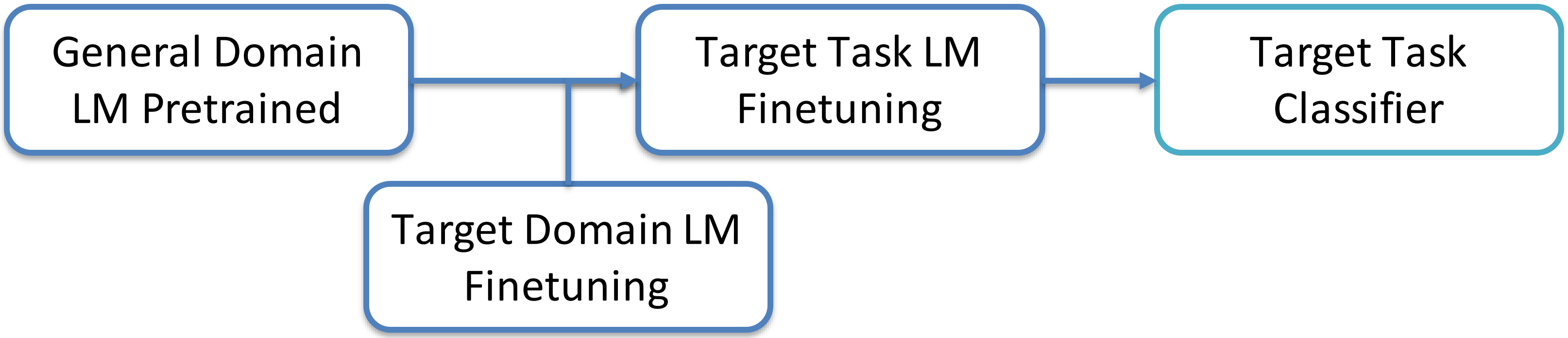


Image credits: https://mccormickml.com/2019/07/22/BERT-fine-tuning/

Methodology: ULMFiT

- Fundamental idea behind ULMFiT is tackle the problem of **insufficient data**



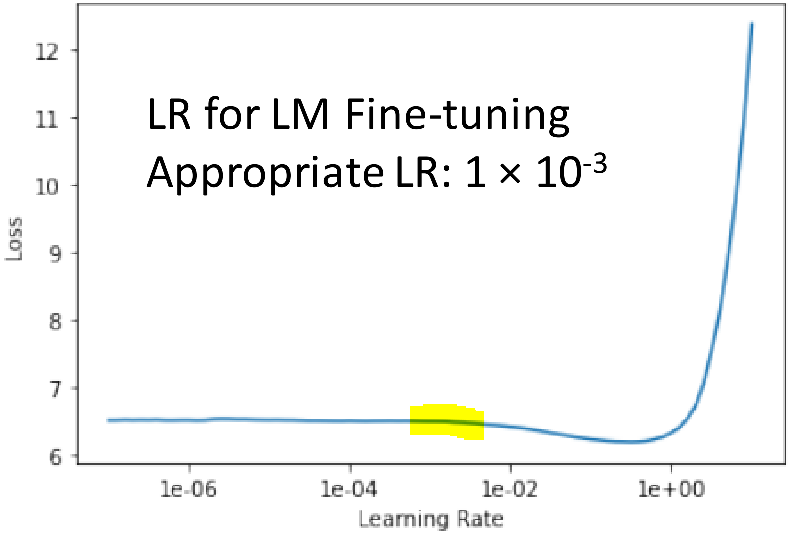
- ULMFiT language model is pre-trained on the *wikitext-103* corpus
- Augmented *Kaggle Twitter Sentiment 140* and *SEM Eval Stance* Datasets
- Used *fastai.text* tokenization technique for appropriate batch sub-division
- Fed the dataset into **LM** (language model), which is essentially AWD-LSTM

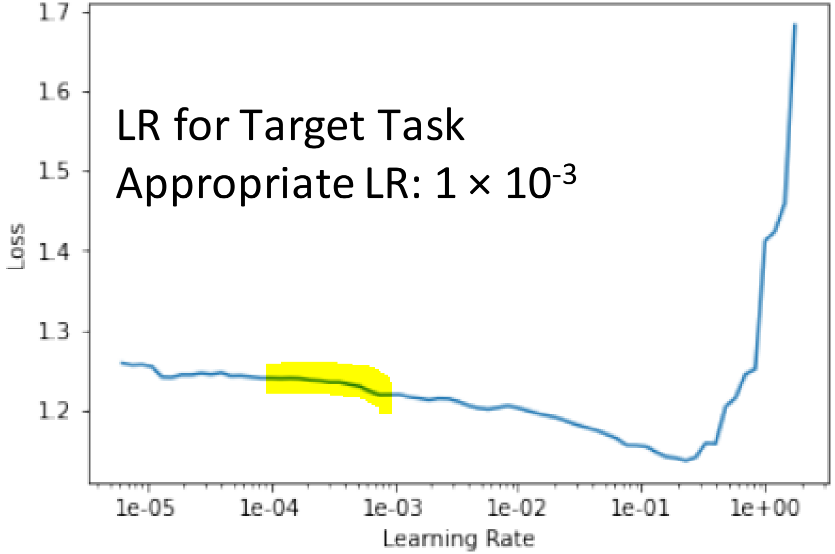
Methodology: ULMFiT

- Stance Rule –

Support	⋀	Oppose	Stance
0		0	None
1		0	Favour
0		1	Against

- Learning rate should be order of magnitude below the point at which the loss starts to diverge





- Gradual Unfreezing of layers helps in avoiding “catastrophic forgetting”

Results

Labels/Models	MNB	SVM	RF	LGBM	XGB
Text Only Informative	0.506	0.517	0.505	0.500	0.510
Image Only Informative	0.499	0.509	0.501	0.500	**0.539**
Directed Hate	0.500	0.510	0.503	0.491	0.503
Generalized Hate	0.500	0.524	0.508	0.513	0.524
Sarcasm	0.500	0.496	0.499	0.485	**0.535**
Allegation	0.500	0.515	**0.515**	0.491	0.472
Justification	0.500	0.504	0.508	0.495	**0.505**
Refutation	0.500	0.546	0.565	0.510	**0.578**
Support	0.496	0.496	0.499	0.504	0.511
Oppose	0.500	0.514	0.511	0.498	0.523

Labels/Models	BiLSTM + Glove	BERT	BERT + ROS	ULMFiT	ULMFiT + AV
Support	0.50	0.49	0.48	0.52	**0.58**
Oppose	0.49	0.49	0.51	0.51	**0.73**
Directed Hate	**0.52**	0.51	0.51	0.50	0.50
Generalized Hate	**0.51**	0.51	0.50	0.51	0.49

References

In addition to those mentioned in our paper, here are the references used for our presentation

- J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," arXiv preprint arXiv:1801.06146, 2018
- Z. Huang, W. Xu, and K. Yu, "Bidirectional lstmcrf models for sequence tagging," arXiv preprint arXiv:1508.01991, 2015
- Chris McCormick , "BERT Fine-Tuning Tutorial with PyTorch", https://mccormickml.com/2019/07/22/BERT-fine-tuning/
- Sandra Faltl et al., "Universal Language Model Fine-Tuning (ULMFiT)", https://humboldt-wi.github.io/blog/research/information_systems_1819/group4_ulmfit/

Codes for reproducing our results are hosted at
https://github.com/vntkumar8/transfer-learning-metoo