Farmers Protest - Stance Detection

Prathamesh Badgujar[1], Aditya Kamble[2], Anuj Kadam[3], Dhruv Shah[4], Anilkumar Kadam[5]

[1,2,3,4]*Student, Department of Computer Engineering, AISSMS College of Engineering, Pune, Maharshtra, India*
[5]*Professor, Department of Computer Engineering, AISSMS College of Engineering, Pune, Maharshtra, India*

---***---

Abstract— *Protests are an important part of the democratic way of administration and are a vital tool for conveying public demands to the ruling government. As voters come to terms with any new rules, there are an increasing range of protests everywhere the world for various reasons. With the advancement of technology, there has additionally been an exponential rise within the use of social media for the exchange data and ideas.*

During this research, data was gathered from the web site "twitter.com", regarding farmers' protest to know the feelings that the public shared on a global level. Due to the repeal of the Farm Laws have been carried out, we aim to use this data to understand whether the government's decision for the appeal was influenced by the public opinion about this topic.

This paper aims to provide a stance prediction deep learning model that has been achieved using the ULMFiT model after fine-tuning, ULMFiT (Universal Language Model Fine-Tuning) model which will be a. Categorizes into For (F), Against (A) and Neutral (N). Proposed model achieved 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)

1. INTRODUCTION

A. Motivation

The Farm Laws that were announced by the Parliament of India in September 2020 were cause for the 2020–2021 Indian farmers' protests. These laws were met with heavy resistance by many farmer unions who deemed it to go against their livelihood, and politicians from the competition who say it might go away farmers at the "mercy of corporates". The union authorities, however, continues that the legal guidelines will make it easy for farmers to promote their produce to large scale buyers and remarked that the protests were just the result of misinformation being spread online. Despite India being in large part self-enough in food grain manufacturing and having welfare schemes, starvation and nutrients stay severe troubles, with India rating as one of the worst nations withinside the international in meals safety parameters.

After the announcement of the acts, farmer unions based around Punjab were a source of the initial protests. After a lot of protests, farmer unions especially from the states of Punjab and Haryana commenced a motion named "Dili-Chalo" (transl. Let's march towards Delhi), where participants numbering in the thousands surrounded Delhi. The authorities ordered the police and regulation enforcement of numerous states to assault the protesters with the use of water cannons and tear gas to prevent the farmer unions from stepping into Haryana first after which Delhi.

The Supreme Court of India ordered stay on the farm Laws in January 2021. Farmer leaders welcomed the live order, which stays in effect. Kerala, Punjab, Chhattisgarh, Rajasthan, Delhi and West Bengal state governments surpassed resolutions in opposition to the farms acts, and 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 micro blogging website "Twitter". Our research mainly aims at analyzing factuality and polarity of twitter data using a deep learning model called ULMFiT.

B. Inductive Transfer Learning

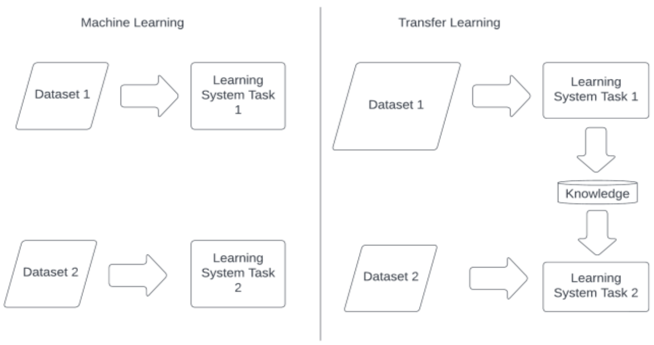

Fig 1. Traditional Machine Learning vs. Transfer Learning

Many NLP state-of-the-art models need to learn from scratch and require big data to attain affordable results, they do now no longer handiest soak up big portions of reminiscence however also are pretty time consuming. In textual data classification, sufficient classified examples are hard to come by, hence we make use of

Inductive transfer learning to solve those challenges. It is the principal idea ULMFiT is primarily based totally upon.

Transfer learning pursuits to imitate the human capacity to collect knowledge at the same time as learning one task and make use of this understanding to remedy a related task. In the conventional method, for instance, two models are trained one at a time without both maintaining and moving knowledge from one to the other. An instance for switch studying alternatively might be to hold knowledge from training a model 1 and to then make use of this information to train some other model. In this case, 1st model might be referred to as the source task and 2nd model, the target task.

C. Overview of ULMFiT Model

1) ***General-Domain Language Model Pretraining:*** In a initial step, the Language Model is pretrained on a huge general collection of texts (WikiText-103 dataset). Now, the model is able to predict the subsequent phrase in a chain with certainty. At this degree the model learns the overall features of the language. Pretraining is mostly useful for datasets containing smaller samples and allows generalization regardless of dataset size. 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 knowledge that is extracted from the initial step is 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 original 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 expecting the subsequent phrase in a chain. Now however, it has additionally discovered assignment-unique functions of the language, consisting of the lifestyles of users in Twitter and their usage of slang language and regional language phrases. Regardless of diversity of the wide-domain information used for pretraining, the information of the goal task will be procured from an entirely different distribution. We therefore fine-tune the Language Model on data of the target task. Given a pretrained 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****:* It is vital a part of the transfer learning approach to fine tune a classifier. Overly competitive fine-tuning can cause huge amounts of forgetting, putting off the gain of knowledge captured via language modeling; gradual convergence can be caused because of overly cautious fine-tuning which results in the data being overfitted. Howard and Ruder[10] proposed gradual unfreezing for fine-tuning the classifier. Gradual unfreezing Howard et al[10] is endorsed to frequently unfreeze the model beginning from the closing layer as this carries the least widespread knowledge rather than fine-tuning all layers at once. Howard et al[10] first unfroze the closing layer and then fine-tuned all 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 not 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 final output is a distribution over the stance labels (For (F) , Against (A) and Neutral (N)).

a) ***Gradual unfreezing:*** Rather than fine-tuning every layer at once, which may cause 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-tune all unfrozen layers for one epoch. We then unfreeze the subsequent decrease frozen layer and repeat, till we fine-tune 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.

Steps in BPT3C:

- The record is split into constant duration batches.
- At the start of every batch, the model is initiated with the very last state of the preceding batch with the aid of using maintaining tune of the hidden states for mean and max-pooling.

- The gradients are back-propagated to the batches whose hidden states contributed to the very last prediction.

2. RELATED WORK

For many years, operations comparable to stemming or lemmatization, still as shallow models, such as SVMs, were popular in NLP. Young et al. [2] claim that word embedding models like word2vec and GloVe, ultimately led the path for the success of deep learning in NLP. One among the most criticisms relating to pretrained word embeddings is that they solely push antecedently learned data to the first layer of a NN, whereas the remaining layers of it still must be trained from scratch. Neelakantan et al. [6] experimented with coaching each individual vector for every word. These approaches overcome the problem of missing topic. But, they have to train the actual model from scratch.

In their seek for better approaches, several researchers searched for strategies that had antecedently proven thriving in Computer Vision. Ruder [1] claims that language modelling is especially suited to capturing sides of language that are vital for target tasks. The OpenAI Transformer is similar to ELMo however it needs some minor changes within the model design for transfer [12]. Both are proven to provide superb empirical results.

Except for achieving progressive leads to varied tasks, ULMFiT includes many techniques for fine tuning model that could boost performance for alternative strategies, for example the OpenAI Transformer.

3. DATASET USED

D. For Language Model Training:

We data-set has been acquired from 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.

E. For Stance Detection Classification:

For stance detection, we used a small subset of the tweets dataset previously mentioned. 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. 2 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 as artificial super-sampling. In this technique, we translated each tweet classified as “A” (against) to some other language of choice, 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.

4. SYSTEM ARCHITECTURE

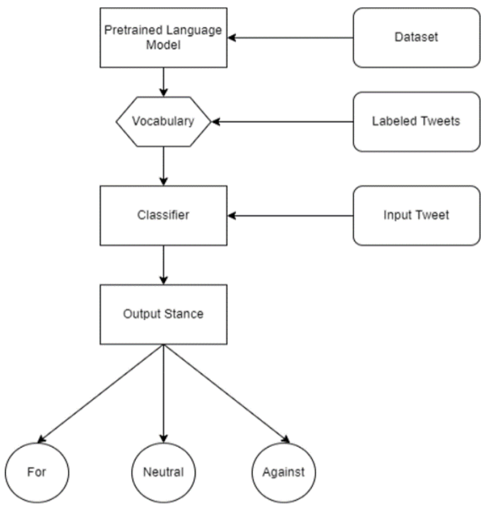

Fig. 3 System architecture of the proposed system

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 volcabulary 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

5. METHODOLOGY

F. 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 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 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 good time. They seem to be violence thirsty goons. #FarmersProtest

G. Performing exploratory data analysis

We further perform some basic data analysis on this data by making data-set operations such as joining the tweets and users datasets etc., 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 most followed people who tweeted about this topic, amount of interactions carried out their tweets, etc.

C. Training the ULMFiT language model

AS mentioned in previous section, the only column needed for language model is the actual text column. Hence we dropped all other columns from the tweets dataset except for "renderedContent". All these tweets are supplied to the ULMFiT model that has been pre-trained on the Wikipedia dataset containing huge corpus. 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 optimal learning rate for training of our language model, which came out to be 0.00209.

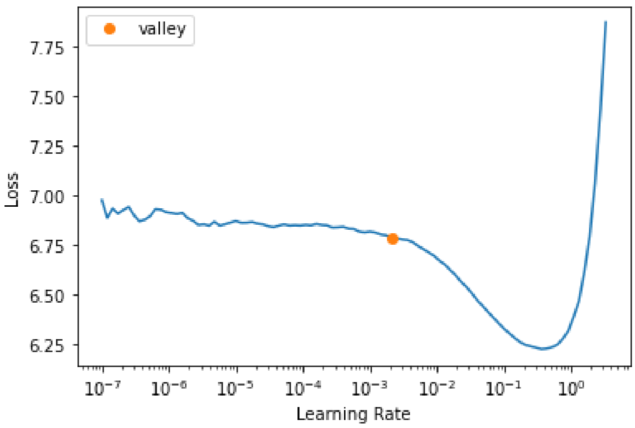

Fig. 4 Finding optimal learning rate for the language model

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 text classifier.

H. Training the stance prediction model (text classification model)

As a final step, we make use of the manually labelled dataset mentioned in section III that contained about 7300 manually labelled tweets. We used the vocabulary from 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 was able to achieve 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.

I. 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 tweet as input from user, pre-processed it fed it to the model generated in previous step, to give the output as either F, N or A.

6. RESULTS

J. 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. 5 Language model performance

K. Text Classification

We made use of the language model as vocabulary or feature vectors for our main task, that was the stance detection of tweets. We call it an extended model. The performance metrics that were used were quite accurate initially. But accuracy tends to favour 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 would be the 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 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. 6 Classification model trained on base ULMFiT language model

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, 7 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%

7. 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 topic relevancy of the data helps the model 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. ULMFiT model can be used for any type of text classification, not only stance.

8. 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, arXiv:1310.4546v1 [cs.CL], Oct. 16, 2013

[4] Jeffrey Pennington, Richard Socher and Christopher D. Manning, GloVe: Global Vectors for Word Representation, Computer Science Department, Stanford University, 2014

[5] 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

[6] 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

[7] Bryan McCann, James Bradbury, Caiming Xiong and Richard Socher, Learned in Translation: Contextualized Word Vectors, arXiv:1708.00107v2 [cs.CL], Jun. 20, 2018

[8] Kaiming He, Georgia Gkioxari, Piotr Dollár and Ross Girshick, Mask R-CNN, arXiv:1703.06870v3 [cs.CV], Jan. 24, 2018

[9] Joao Carreira, Andrew Zisserman, Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset, arXiv:1705.07750v3 [cs.CV], Feb. 12, 2018

[10] Jeremy Howard, Sebastian Ruder, Universal Language Model Fine-tuning for Text Classification, arXiv:1801.06146, Jan. 18, 2018

[11] 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

[12] Alec Radford, Karthik Narasimhan, Tim Salimans and Ilya Sutskever, Improving Language Understanding by Generative Pre-Training, OpenAI Blog, Jun. 11, 2018

[13] Sinno Jialin Pan, Qiang Yang, A Survey on Transfer Learning, IEEE Transactions on Knowledge and Data Engineering, vol. 22, issue 10, Oct. 2010

[14] Dipanjan Sarkar, A Comprehensive Hands-on Guide to Transfer Learning with Real-World Applications in Deep Learning, Towards Data Science, Nov. 14, 2018 [Blog]

[15] 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]

[16] Ben Krause, Emmanuel Kahembwe, Iain Murray and Steve Renals, Dynamic Evaluation of Neural Sequence Models, arXiv:1709.07432, Sep. 21, 2017

[17] 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

[18] Jason Brownlee, What Are Word Embeddings for Text?, Machine Learning Mastery, posted on Oct. 11, 2017, https://machinelearningmastery.com/what-are-word-embeddings/

[19] Stephen Merity, Nitish Shirish Keskar, Richard Socher, Regularizing and Optimizing LSTM Language Models, arXiv:1708.02182v1 [cs.CL], Aug. 8, 2017

[20] Diederik P. Kingma, Jimmy Ba, Adam: A Method for Stochastic Optimization, arXiv:1412.6980v9 [cs.LG], Jan. 20, 2017

[21] 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

[22] Michael Nielsen, Improving the way neural networks learn, Neural Networks and Deep Learning, posted in Oct. 2018, http://neuralnetworksanddeeplearning.com/chap3.html

[23] 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

[24] Leslie N. Smith, Cyclical Learning Rates for Training Neural Networks, arXiv:1506.01186v6 [cs.CV], Apr. 4, 2017

[25] 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

[26] Sebastian Ruder, An overview of gradient descent optimization algorithms, arXiv:1609.04747v2 [cs.LG], Jun. 15, 2017

[27] 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

[28] 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

[29] 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

[30] Roopal Garg, Deep Learning for Natural Language Processing: Word Embeddings, datascience.com, posted on Apr. 26, 2018 [Blog]