Final report

Use of ULMFiT model to classify Twitter data

Yikai Zhao
ECEN 765: Machine Learning with Networks
Dr. Xiaoning Qian

Summary

Natural language processing (NLP) has witnessed several breakthroughs in the year of 2018 with new state-of-art models such as ULMFiT, ELMo, and recently BERT. This paper specifically focus on the application performance of ULMFiT and test its performance on text classification in multiple Twitter datasets (the twitter_sentiment dataset, the GOP debate dataset and 20Newsgroup dataset). Additionally, this paper tests the hypothesis that a twitter-trained universal language model would improve the performance of ULMFiT on twitter datasets when comparing with a universal language model trained from Wikipedia texts called Wiki103 (Howard and Ruder 2018, Merity, Keskar et al. 2018). My experimental result shows promising results on twitter datasets (twitter_sentiment and GOP debate) yielding above 80% of accuracy in sentiment analysis (two and three categories) with minimum level of preprocessing and hyper-parameter tweaking. However, the model did not perform well on 20Newsgroup resulting in about 65% of accuracy and F1 score.

Furthermore, my experiment shows the twitter trained universal language model would yield the same level of accuracy as Wiki103, suggesting that given a universal language model trained from a giant corpus of text, a higher focus should be put on the training of later layers rather than the training of new universal language layers.

Use of ULMFiT method to classify Twitter data

Motivation of subject:

In the field of social science, great amount of research are text based. Currently the key methods related to investigate textual data include critical method, framing analysis, sentiment analysis, or broadly speaking, textual content analysis. However, due to the lack of understanding of computational tools, majority of such research is still conducted manually, meaning that researchers have to read through ALL of the texts in order to gain insights. This process is problematic as it is hard to scale up to larger corpus of textual data to analyze. NLP has been one of the forefronts in machine learning and deep learning and many models have achieved state-of-the-art (SOTA) on many NLP tasks. However, these models are all trained from scratch, demanding large amount of data to train and consuming enormous computation power. Hardly would any researcher from the background of social science be capable to incorporate the advancements in NLP to their applied research field. Even though the NLP community has achieved SOTA results in multiple tasks, there is a huge technical gap between the NLP communities and applied social scientist that prevent the transfer of those SOTA models to the social science research.

However, the recent breakthrough in NLP has bridged the gap by adopting the philosophy of inductive transfer learning, which played a large role in computer vision. In NLP, though, transfer learning was mostly limited to the use of pre-trained word embeddings (which, to be fair, improved baselines significantly). Recently, researchers are moving towards transferring entire models from one task to another. One most representative method of the NLP transfer learning is called Universal Language Model Fine-tuning (ULMFiT) (Howard & Ruder, 2018). Their method enables transfer learning for ANY NLP task, akin to fine-tuning ImageNet models. And ULMFiT has achieved SOTAs in multiple tasks and it is still one of the best models in text classification (https://nlpprogress.com/). ULMFiT was built based on 3-layer LSTM architecture called AWD-LSTM which is a multi-layer bi-LSTM network (Merity, Keskar & Socher, 2017).

ULMFiT and research hypotheses

The key mechanism of ULMFiT could be summarized in the following figure borrowed from Howard and Ruder's paper (2018):

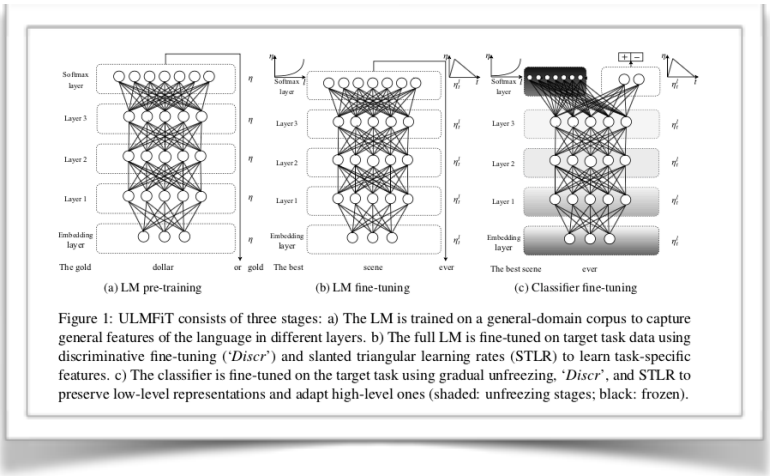
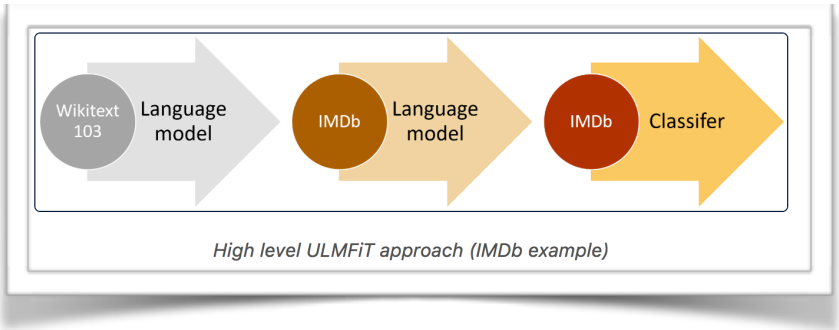


Figure 1: ULMFiT consists of three stages: a) The LM is trained on a general-domain corpus to capture general features of the language in different layers. b) The full LM is fine-tuned on target task data using discriminative fine-tuning ('*Discr*') and slanted triangular learning rates (STLR) to learn task-specific features. c) The classifier is fine-tuned on the target task using gradual unfreezing, '*Discr*', and STLR to preserve low-level representations and adapt high-level ones (shaded: unfreezing stages; black: frozen).

To simply put, a *universal/general-domain* language model (LM) is pre-trained by using enormous amount of textual data from the target language; in the case of this project, I used the language model called Wiki103 that ULMFiT suggested. Wikitext-103 (Merity et al., 2017b) consists of 28,595 preprocessed Wikipedia articles and 103 million words. Secondly, such *universal* LM is applied to the target textual dataset to train a LM on the dataset, in order to gain a contextual understanding of the language. Finally, a supervised classification is conducted on the target textual dataset to learn the classification boundaries. The following chart summarized the above text succinctly.



High level ULMFiT approach (IMDb example)

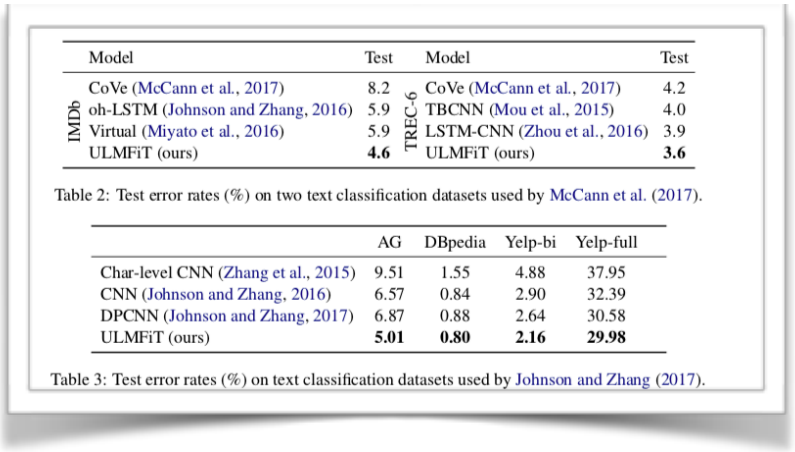
ULMFiT has generated great attention and multiple language models are being developed in different languages (e.g. Rother & Rettberg, 2018). However, little is known about its performance on Twitter data. Although ULMFiT has achieved SOTA on multiple datasets (shown below), it has

not been widely tested on Twitter data. Tweets in itself are special forms of textual data. They are short in length when compared with other textual classification datasets. Also, the language use is more casual mixed with internet slangs and emojis, making it hard to extract meanings automatically from them. However, little attention has been paid to tweets. This study tests the performance of ULMFiT on tweets. Thus this paper raises its first research question:

RQ1: Would ULMFiT perform equally well on tweets when compared to its current performance in other datasets?

Additionally, the intuition of a language model is that it is a model that actually "understand" the language. The model's ability to predict the next word correctly suggests a good understanding of the language itself. If we were to test the performance of the model on twitter datasets, wouldn't be better to train the universal language model in the tongue of languages used on twitter? Thus this research posed its hypothesis:

H1: The universal language model trained from Twitter would achieve higher accuracy in performing other downstream text classification from twitter, than the universal language model trained from other text corpus (in this paper, Wiki103 was used)



	Model	Test		Model	Test
IMDb	CoVe (McCann et al., 2017)	8.2	TREC-6	CoVe (McCann et al., 2017)	4.2
	oh-LSTM (Johnson and Zhang, 2016)	5.9		TBCNN (Mou et al., 2015)	4.0
	Virtual (Miyato et al., 2016)	5.9		LSTM-CNN (Zhou et al., 2016)	3.9
	ULMFiT (ours)	**4.6**		ULMFiT (ours)	**3.6**

Table 2: Test error rates (%) on two text classification datasets used by McCann et al. (2017).

	AG	DBpedia	Yelp-bi	Yelp-full
Char-level CNN (Zhang et al., 2015)	9.51	1.55	4.88	37.95
CNN (Johnson and Zhang, 2016)	6.57	0.84	2.90	32.39
DPCNN (Johnson and Zhang, 2017)	6.87	0.88	2.64	30.58
ULMFiT (ours)	**5.01**	**0.80**	**2.16**	**29.98**

Table 3: Test error rates (%) on text classification datasets used by Johnson and Zhang (2017).

The Experiment and datasets

Because of the huge computation demand, I tried several GPU servers (Terra, AWS and Paperspace) and eventually chose Paperspace due to the convenience of use. I originally used a 8GB RAM GPU but it constantly runs out the RAM due to large amount of hyper-parameters of LSTM models. Eventually I used the an GPU instance featuring 24GB (equivalent to Nvidia K80 GPU) which cuts a single training cycle time from roughly 1 hour to 20 minutes.

Datasets

There are 4 datasets in total to perform the experiment. Three were used in textual classification and one was used in training the twitter universal language model.

To train the twitter universal language model, I used the dataset sentiment 140 (https://www.kaggle.com/kazanova/sentiment140/home). It contains 1,600,000 tweets extracted using the twitter API . The tweets have been annotated (0 = negative, 2 = neutral, 4 = positive) and they can be used to detect sentiment . Since I am only using it to train the language model, the annotated label was not used.

The first dataset to be classified is called “twitter_sentiment” (https://www.kaggle.com/c/twitter-sentiment-analysis2/data) and it featured 100,000 classified tweets with labels “positive” and “negative.”

The second dataset to classify is called “GOP presidential debate” and it includes 139,000 tweets about the early August GOP debate in Ohio with each of them categorized as “positive,” “negative” or neutral.

The third dataset is called “20Newsgroup:” a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. I acknowledge that it is not a twitter dataset. However I experimented on it to test the performance of ULMFiT in a task containing large amount of categories or labels (20 in this case).

Results:

The experiment results can be found in the following documents in the Github repo:

Twitter_sentiment_twitterLM.py : The twitter sentiment classification using the twitter language model

Twitter_sentiment-wiki130.py: The twitter sentiment classification using the Wiki103 language model

gop_debate_Wiki103.py: The classification on the “GOP presidential debate” dataset using Wiki103 language model

gop_debate_TwitterLM.py: The classification on the “GOP presidential debate” dataset using twitter language model

Twitter_language_model_training.py: the scripts for training the twitter language model

To make consistent comparison with Howard and Ruder's paper (2018), most of the results are compiled in the metric of accuracy. Also I have checked the distribution of labels/categorization of different datasets and they all seem to be balanced. Thus this research will only report the accuracy as the metric.

The ULMFiT's performance on tweets (using Wiki103 universal language model) is compiled in the following table:

Dataset	Accuracy
Twitter_sentiment (2 categories)	81.47%
GOP debate (3 categories)	73.77%
20 Newsgroup (20 categories)	65.00%

We can see these results are far from the those in the original ULMFiT paper (Howard and Ruder 2018) that they could achieve the accuracy level as high as 95.4% for example, in the IMDB sentiment classification task. However, I would not consider these results discouraging. Firstly, the testing datasets in the original ULMFiT dataset are pretty developed meaning they have been carefully pre-processed. In contrast, the twitter dataset used in this paper are pretty "raw" in a sense that they have been kept in their original form without any feature engineering; when fitting the model, I did not pre-process them either but directly feed them into the model. This paper believes that with careful preprocessing of the twitter texts, the results will be greatly improved. Secondly, the dataset itself comes platforms such as Crowdflower (https://www.figure-eight.com/platform/text-natural-language-processing-nlp/). Even though such platform is professional in data markup industry, it inevitably possess higher classification bias or mistakes in the annotation, which impacts the performance of the neural network. Finally, to make consistent comparisons with the results trained in the Twitter language model, the models are simply trained at a basic level, meaning no data preprocessing, no extensive training cycles, no fine-tuning of the hyper parameters and the model is only trained in forward direction rather than bi-direction as it is in Howard and Ruder's paper (2008).

To test the hypothesis that the Twitter language model would perform better than Wiki103 language model, I trained a language model out of the dataset called Sentiment 140 featuring 1,600,000 tweets which provide enough texts to train our model to understand the language. The

language training scripts can be found in “Twitter_language_model_training.py” in the Github repository. The comparison is compiled in the following table:

ULMFiT performance comparison by using different universal language models

Dataset	Wiki103 language model	Twitter language model
Twitter_sentiment	81.47%	81.40%
GOP presidential debate	73.77%	71.64%

From the table above we can see that in contrary to our hypothesis that the twitter language model would improve the downstream twitter classification tasks, the results from the twitter language model are actually almost identical to those trained from a general language model (in this paper, Wiki103), if not a bit short. This suggests us that the universal language model may not impact the final performance of the model much so that more focus should be put to other aspects such as data engineering or hyper parameter fine-tuning.

Discussion[1]

This paper extends the evaluation of the performance of ULMFiT to the realm of twitter data. The experiments yield promising results on the text classification tasks on twitter data. However, the twitter language model does not improve the performance as I originally thought.

This project serves as a fundamental step for my future research trajectory. In Communication study research (or generally social science research), there is a significant psychological pattern called framing effect. Framing theory suggests that how something is presented to the audience (called “the frame”) influences the choices people make about how to process that information. Usually such “frames” are comprised of miscellaneous levels: words, sentences, paragraphs and even the relationship between paragraphs. Given human annotated classification as training sets, how well will the ULMFiT perform in terms of correctly classify texts into the predefined categories (“frames”). This area is called automatic thematic analysis or annotated frame analysis in some literature (e.g. Card et al., 2015; Odijk et al, 2013). However, the accuracy is not well-achieved yet. I would like to introduce ULMFiT to the field and investigate its effects. Further, given the current process achieved in NLP, similar transfer learning model has also been

[1] *I have collected almost 40,000 tweets related to the topic of the issue of gene-edit babies and intended to conduct framing analysis on them. However, the annotation process took so long so I did not have enough time to finish it. But I will continue doing it in the next semester.*

promoted such as ELMo and BERT. I would like to compare the performance of ULMFiT, ELMo and BERT, together with traditional NLP algorithms such as Naive Bayes, Latent Dirichlet modeling, in the context of framing analysis.

References:

Card, D., Boydstun, A. E., Gross, J. H., Resnik, P., & Smith, N. A. (2015). The media frames corpus: Annotations of frames across issues. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers) (Vol. 2, pp. 438-444).

Go, A., Bhayani, R. and Huang, L., 2009. Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford, 1(2009), p.12.

Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (Vol. 1, pp. 328-339).

Merity, S., Keskar, N. S., Bradbury, J., & Socher, R. (2018). Scalable Language Modeling: WikiText-103 on a Single GPU in 12 hours.

Merity, S., Keskar, N. S., & Socher, R. (2017). Regularizing and optimizing LSTM language models. arXiv preprint arXiv:1708.02182.

Odijk, D., Burscher, B., Vliegenthart, R., & De Rijke, M. (2013, November). Automatic thematic content analysis: Finding frames in news. In International Conference on Social Informatics (pp. 333-345). Springer, Cham.

Rother, K., Allee, M., & Rettberg, A. (2018). ULMFiT at GermEval-2018: A Deep Neural Language Model for the Classification of Hate Speech in German Tweets. Austrian Academy of Sciences, Vienna September 21, 2018.