Universal Language Model Fine-tuning for Text Classification

Jeremy Howard[1] Sebastian Ruder[2]

[1]University of San Francisco

[2]Insight Centre
NUI Galway, Aylien Ltd.
Dublin

May, 2018
Presented by:
Juan Pinzon, Mar 2019

Outline

1. Introduction
   - Abstract
   - Motivations of the Research

2. What is ULMFiT
   - ULMFiT Contributions
   - How ULMFiT Works
   - Tests & Results

ULMFiT

Abstract

- Authors propose an effective transfer learning method that can be applied to any task in NLP.
- The method significantly outperforms the state-of-the-art on six text classification tasks.
  - Reduction of 18% - 24% error

Motivations of the Research

- The huge impacts that Inductive transfer learning has had in Computer Vision.

Motivations of the Research

- The huge impacts that Inductive transfer learning has had in Computer Vision.
  - CV tasks rarely train from scratch, instead they are fine tuned from pre-trained models such as ImageNet.

Motivations of the Research

- The huge impacts that Inductive transfer learning has had in Computer Vision.
  - CV tasks rarely train from scratch, instead they are fine tuned from pre-trained models such as ImageNet.
- NLP state-of-the-art models are being trained from scratch, requiring large datasets & days to converge.

Motivations of the Research

- The huge impacts that Inductive transfer learning has had in Computer Vision.
  - CV tasks rarely train from scratch, instead they are fine tuned from pre-trained models such as ImageNet.
- NLP state-of-the-art models are being trained from scratch, requiring large datasets & days to converge.
- Word Embeddings only target models 1st layer.

Motivations of the Research

- The huge impacts that Inductive transfer learning has had in Computer Vision.
  - CV tasks rarely train from scratch, instead they are fine tuned from pre-trained models such as ImageNet.
- NLP state-of-the-art models are being trained from scratch, requiring large datasets & days to converge.
- Word Embeddings only target models 1st layer.
- Find a more powerful and easy to implement method for performing Inductive transfer learning for NLP tasks.

ULMFiT Contributions

1. A method that can be used to achieve CV-like transfer learning for any task for NLP.
2. *Discriminative fine-tuning*, *slanted triangular learning rates*, and *gradual unfreezing*, novel techniques to retain previous knowledge and avoid forgetting during fine-tuning.
3. Significantly outperform the state-of-the-art on six representative text classification datasets, with an error reduction of 18-24%.
4. Enables extremely sample-efficient transfer learning.
5. Pre-trained models and code available to enable wider adoption.

How ULMFiT Works

Language Modeling (LM)

"LM is the task of assigning a probability to sentences in a language. [. . .] Besides assigning a probability to each sequence of words, the language models also assigns a probability for the likelihood of a given word (or a sequence of words) to follow a sequence of words." [a]

[a] Page 105, Neural Network Methods in Natural Language Processing, 2017

How ULMFiT Works

- ULMFiT consists of the following steps:
  1. General-domain LM pre-training
  2. Target task LM fine-tuning
  3. Target task classifier fine-tuning

How ULMFiT Works

1. General-domain LM pre-training:
    - An ImageNet-like corpus for language should be large and capture general properties of language
    - The authors pretrained the language model on Wikitext-103, consisting of 28,595 preprocessed Wikipedia articles and 103 million words.
    - This stage is the most expensive, it only needs to be performed once improving performance of downstream models.
2. Target task LM fine-tuning:
    - No matter how diverse the general-domain data used for pretraining is, it is necessary to fine-tune the LM on data of the target task.
    - Authors propose **discriminative fine-tuning** and **slanted triangular learning rates** for fine-tuning the LM.

How ULMFiT Works

Discriminative Fine-Tuning

Instead of using the same learning rate for all layers of the model, discriminative fine-tuning allows for tuning each layer with different learning rates. Regular stochastic gradient descent (SGD) update of a model's parameters θ at time step **t** looks like the following:

$$\theta_t = \theta_{t-1} - \eta \cdot \bigtriangledown_{\theta} J(\theta) \tag{1}$$

The SGD update with **discriminative finetuning** is then the following:

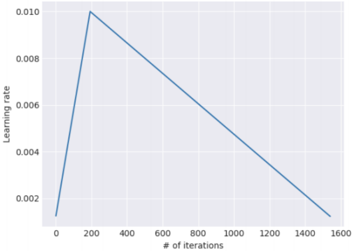
$$\theta_t^{\ell} = \theta_{t-1}^{\ell} - \eta^{\ell} \cdot \bigtriangledown_{\theta^{\ell}} J(\theta) \tag{2}$$

How ULMFiT Works

Slanted Triangular Learning Rates

For adapting parameters to task-specific features, the idea is that the model should quickly converge to a suitable region of the parameter space in the beginning of training and then refine its parameters. Using the same learning rate throughout training is not the best way to achieve this behavior.

STLR, which first linearly increases the learning rate and then linearly decays it according to the following update schedule



How ULMFiT Works

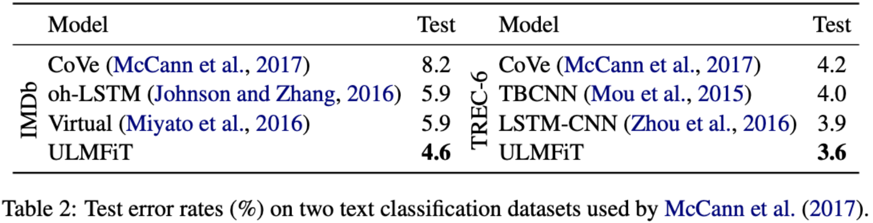
3. Target task Classifier fine-tuning:
   - The pretrained language model is augmented with two additional linear blocks.
   - Following standard practice for CV classifiers, each block uses **batch normalization** and **dropout**, with **ReLU activations** for the intermediate layer and a **softmax activation** that outputs a probability distribution over target classes at the last layer.

Tests & Results

Table: Datasets & Tasks

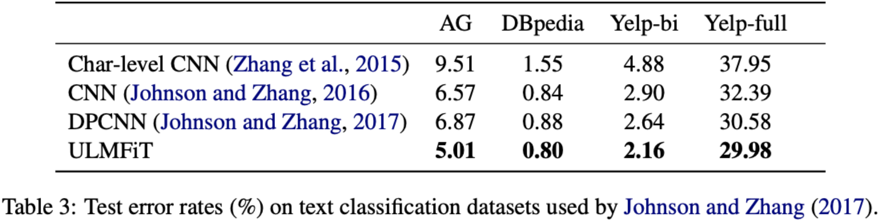
Datasets	Type	# of Classes	# Examples
TREC-6	Question	6	5.5k
IMDb	Sentiment	2	25k
Yelp-bi	Sentiment	2	560k
Yelp-full	Sentiment	5	650k
AG	Topic	4	120k
DBpedia	Topic	14	560k

Tests & Results



	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	**4.6**		ULMFiT	**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	**5.01**	**0.80**	**2.16**	**29.98**

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

Tests & Results

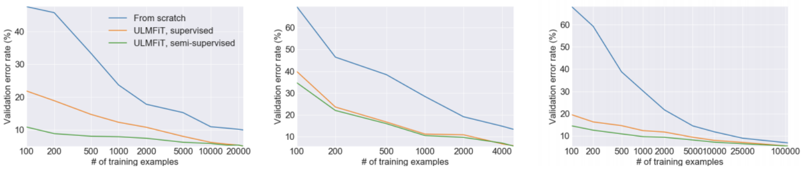


Figure: Validation error rates for supervised and semi-supervised ULMFiT vs. training from scratch with different numbers of training examples on **IMDb, TREC-6**, and **AG** (from left to right).

Table: Validation error rates for ULMFiT with & without pretraining.

Pretraining	IMDb	TREC-6	AG
Without pretraining	5.63	10.67	5.52
With pretraining	**5.00**	**5.69**	**5.38**

Summary

- ULMFiT proves to be an effective and extremely sample-efficient transfer learning method that can be applied to any NLP task.
- several novel fine-tuning techniques where introduced that in conjunction prevent catastrophic forgetting and enable robust learning across a diverse range of tasks.
- This method significantly outperformed existing transfer learning techniques and the state of-the-art on six representative text classification tasks.

Bibliography I

Jeremy Howard and Sebastian Ruder.
Universal Language Model Fine-tuning for Text Classification.
Journal Computing Research Repository (CoRR), 2018.
http://arxiv.org/abs/1801.06146