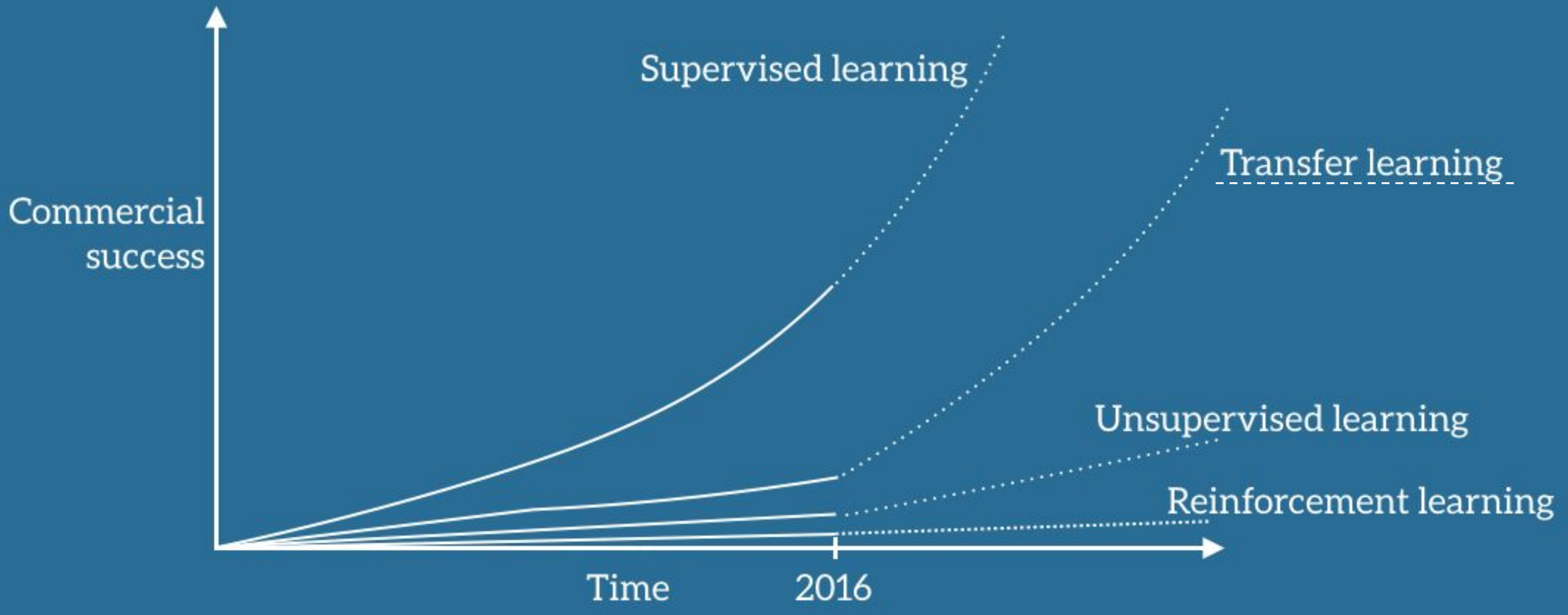
Transfer Learning over Text using ULMFiT

Justin Howard (Fast.ai, USFCA)
Sebastian Ruder (Aylien Ltd., Dublin)

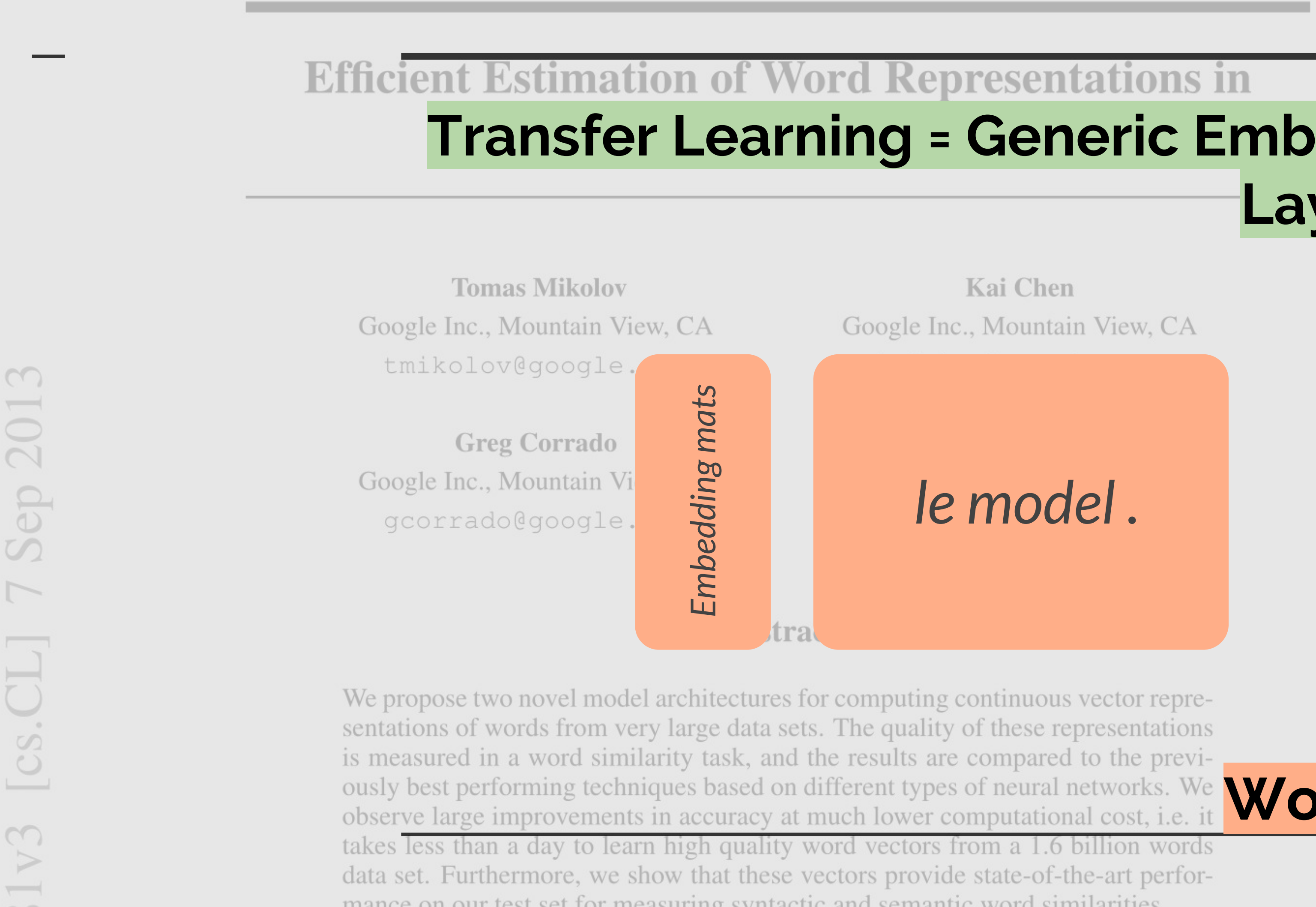
Presented by: Priyansh Trivedi



- Andrew Ng, NIPS 2016 tutorial

Obligatory Celebrity Quote

Transfer Learning = Generic Embedding Layer [7] ?



1v3 [cs.CL] 7 Sep 2013

Efficient Estimation of Word Representations in

Tomas Mikolov
Google Inc., Mountain View, CA
tmikolov@google.

Kai Chen
Google Inc., Mountain View, CA

Greg Corrado
Google Inc., Mountain Vi
gcorrado@google.

tra

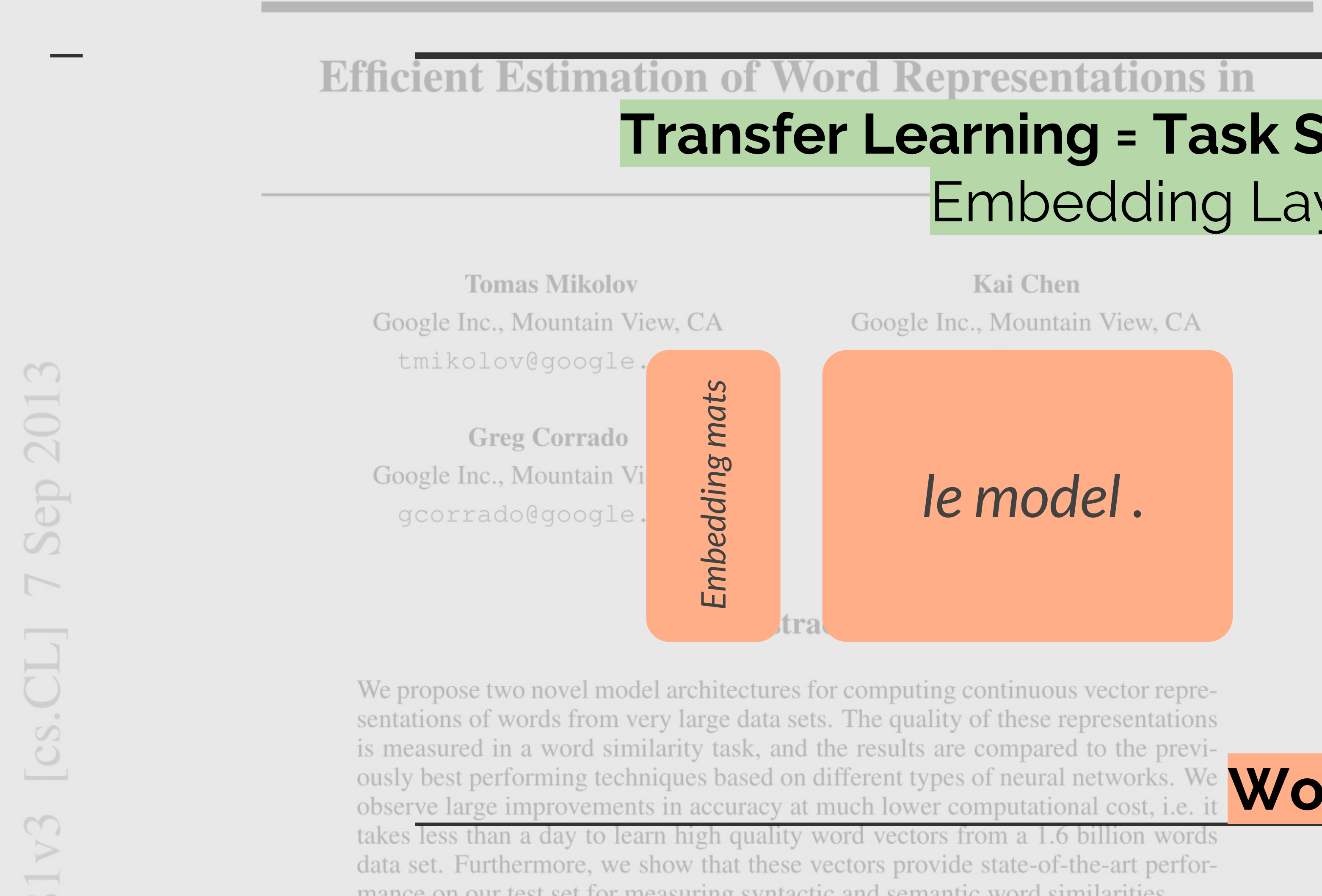
We propose two novel model architectures for computing continuous vector representations of words from very large data sets. The quality of these representations is measured in a word similarity task, and the results are compared to the previously best performing techniques based on different types of neural networks. We observe large improvements in accuracy at much lower computational cost, i.e. it takes less than a day to learn high quality word vectors from a 1.6 billion words data set. Furthermore, we show that these vectors provide state-of-the-art performance on our test set for measuring syntactic and semantic word similarities

Embedding mats

le model .

Word2Vec

Transfer Learning = Task Specific Embedding Layer [2] ?



Efficient Estimation of Word Representations in

Tomas Mikolov
Google Inc., Mountain View, CA
tmikolov@google.

Kai Chen
Google Inc., Mountain View, CA

Greg Corrado
Google Inc., Mountain Vi
gcorrado@google.

tra

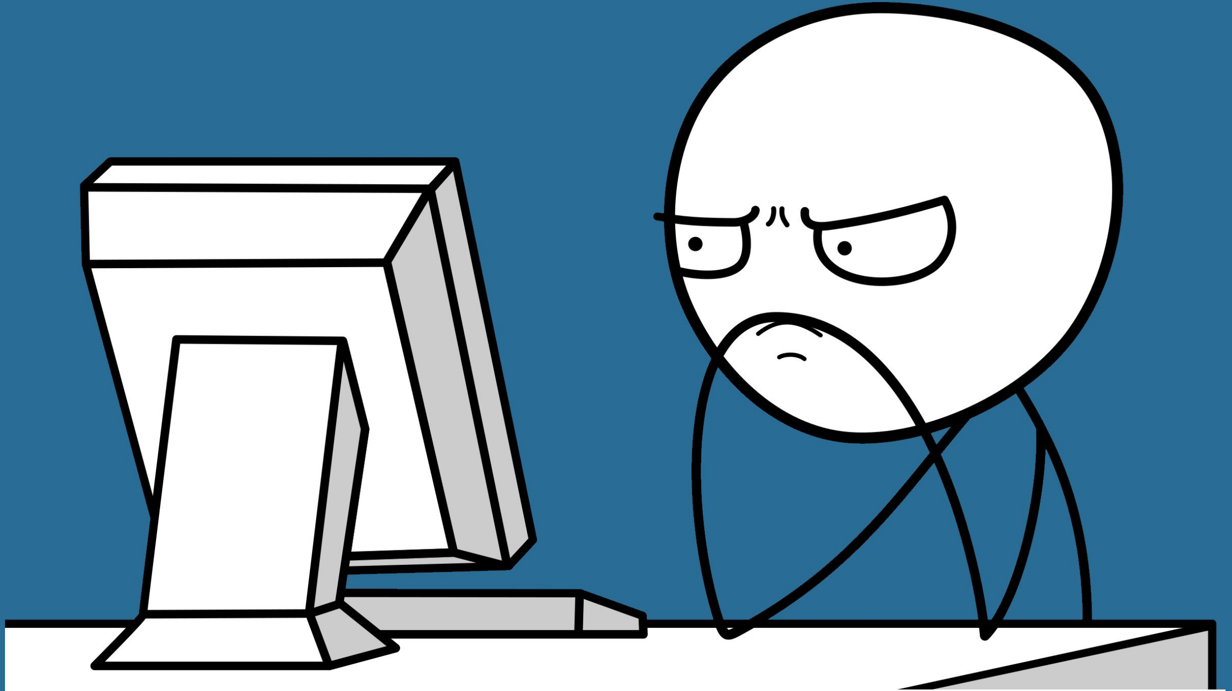
We propose two novel model architectures for computing continuous vector representations of words from very large data sets. The quality of these representations is measured in a word similarity task, and the results are compared to the previously best performing techniques based on different types of neural networks. We observe large improvements in accuracy at much lower computational cost, i.e. it takes less than a day to learn high quality word vectors from a 1.6 billion words data set. Furthermore, we show that these vectors provide state-of-the-art performance on our test set for measuring syntactic and semantic word similarities.

Embedding mats

le model .

Word2Vec

NO



Pre-trained Initializations

Pre-train network on a general dataset (Same task, or not).

Fine tune the network on the task.

Common in CV.

In NLP?

Pre-trained Initializations in NLP?

“depends largely on how semantically similar the tasks are, which is different from the consensus in image processing”

- [3]

NLP Models’ most layers are trained from scratch.

Counterpoint

[4] shows that pretraining LSTMs as Language Models or Autoencoders can *reach SOTA* performance on multiple tasks.

Counterpoint

[4] shows that pretraining LSTMs as Language Models or Autoencoders can *reach SOTA* performance on multiple tasks.

But require millions of in-domain documents to achieve good performance, which severely limits its applicability.

Not the idea of LM fine-tuning but our **lack of knowledge of how to train them effectively** has been hindering wider adoption.

ULMFiT

Universal Language Model Fine-Tuning for Text Classification [1]

1. Method to achieve CV-like transfer learning for “any” task for NLP.

2. Novel techniques to retain previous knowledge and avoid forgetting while fine-tuning.

3. Enables “extremely” sample-efficient transfer learning.

Universal Language Model Fine-Tuning for Text Classification

No frills (attention; shortcuts), regular LSTM Language Model.

General Domain Language Model (*LM*) (Pretraining)

Train on Wikipedia.

Don’t need no labels.

Fine Tune LM on Target Task

1. Decay Learning Rate per Layer.

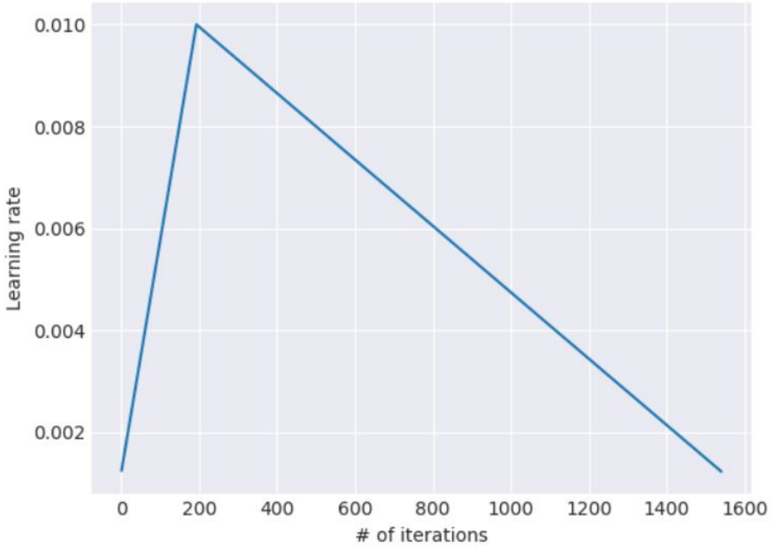
$$\theta_t^l = \theta_{t-1}^l - \eta^l \cdot \nabla_{\theta_t} J(\theta)$$

$$\eta^{l-1} = \frac{\eta^l}{c}$$

Fine Tune LM on Target Task

2. Slanted Triangular Learning Rate per Training Iters

"quickly converge to a suitable region of the parameter space in the beginning of training and then refine its parameters."



Fine Tune LM on Target Task

2. Triangular Learning Rate per Training Iters

$$cut = \lfloor T \cdot cut\_frac \rfloor$$

$$p = \begin{cases} t/cut, & \text{if } t < cut \\ 1 - \frac{t - cut}{cut \cdot (1/cut\_frac - 1)}, & \text{otherwise} \end{cases}$$

$$\eta_t = \eta_{max} \cdot \frac{1 + p \cdot (ratio - 1)}{ratio}$$

cut_frac = 0.1

ratio = 32

η_{max} = 0. 01

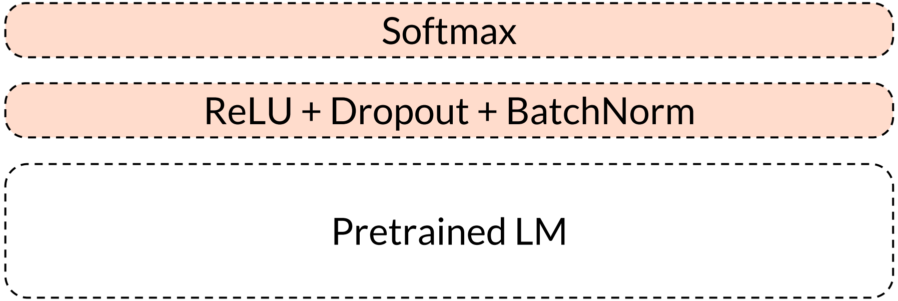
Fine Tune LM on Target Task

So far, we don’t need no labels.

Other schedules have been proposed in [5], [6].

Target task Classifier

Append two layers like so:



Softmax

ReLU + Dropout + BatchNorm

Pretrained LM

Target task Classifier

Input to first layer:

$$\mathbf{h}_c = [\mathbf{h}_T, \texttt{maxpool}(\mathbf{H}), \texttt{meanpool}(\mathbf{H})]$$

Note: the two new layers are the only ones trained from scratch.

Target task Classifier

Gradual Unfreezing to prevent “catastrophic forgetting”:

- Freeze all layers
- Unfreeze last layer for 1 epoch
- Unfreeze $L - i$ layers iteratively as:

    performance converges on validation set.

So far, then

Transfer mechanism:

1. Train LM on general text
2. Train LM on specific text
3. Train Classifier

Techniques Used:

1. Decay LR per layer
2. Decay LR per iter
3. Gradual Unfreezing

Experiments

Tasks and Model

1. Sentiment Analysis
    a. IMDb
    b. Yelp Review
2. Question Classification
    a. TREC-6
3. Topic Classification
    a. AG News
    b. DBpedia

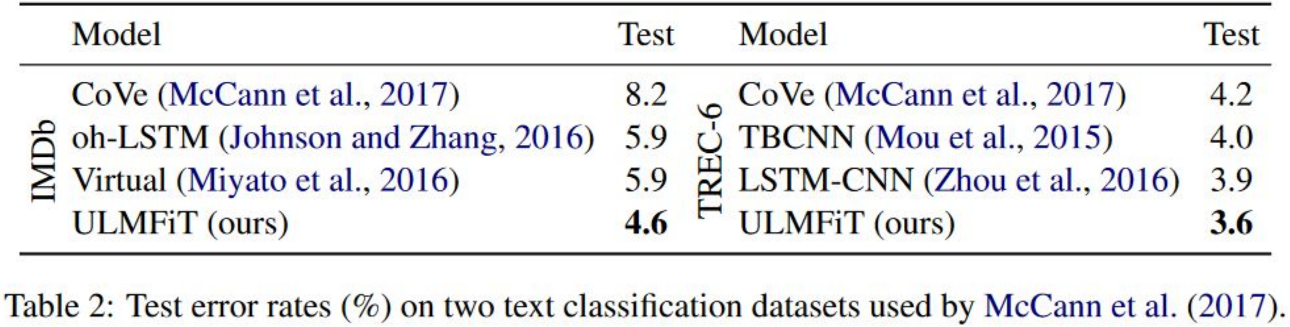
Model:

- LSTM Language Model [8]
- 400d embeddings
- 3 layer
- Uses dropout

Results

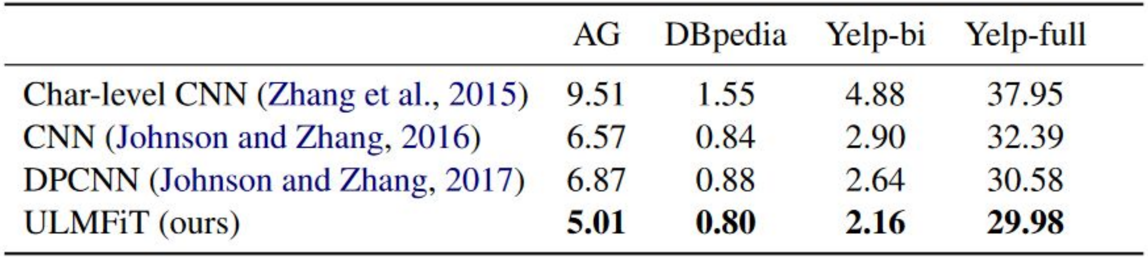
- Beats on all tasks.
- SOTA with a simple model

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 (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**

Low Shot Learning

Ablation: Fine tune only on Labeled Examples (**Supervised**); on all task data that can be used (**Semi-Supervised**) v/s trained from scratch.

Result:
Supervised with 100 examples ~ Train from scratch with 10-20x data

Semi-supervised (50k) + Supervised (100) ~ 100x more data

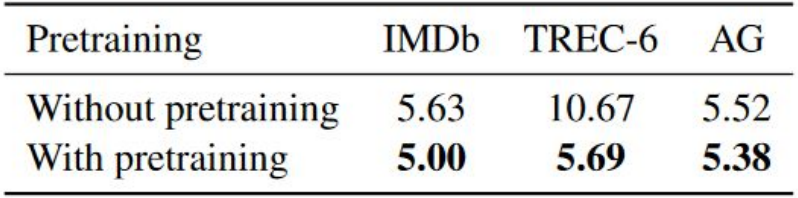
Low Shot Learning

Ablation: Fine tune only on Labeled Examples (**Supervised**); on all task data that can be used (**Semi-Supervised**) v/s trained from scratch.

Implies: General Domain pre-training is a nice idea.

Other Analysis: Pretraining

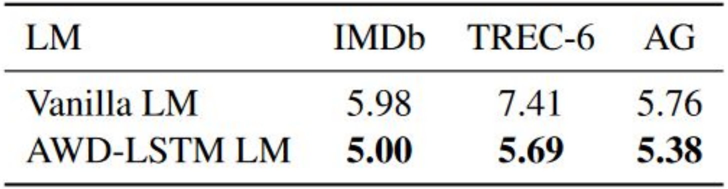
More useful for small datasets.



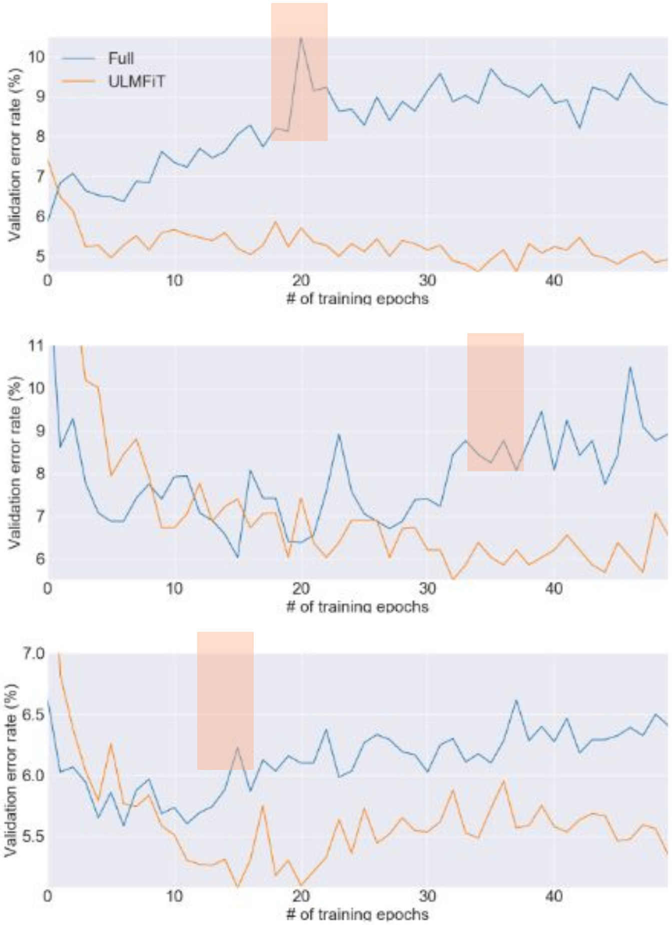
Pretraining	IMDb	TREC-6	AG
Without pretraining	5.63	10.67	5.52
With pretraining	**5.00**	**5.69**	**5.38**

Other Analysis: Language Models

Simple models perform almost as well.



LM	IMDb	TREC-6	AG
Vanilla LM	5.98	7.41	5.76
AWD-LSTM LM	**5.00**	**5.69**	**5.38**



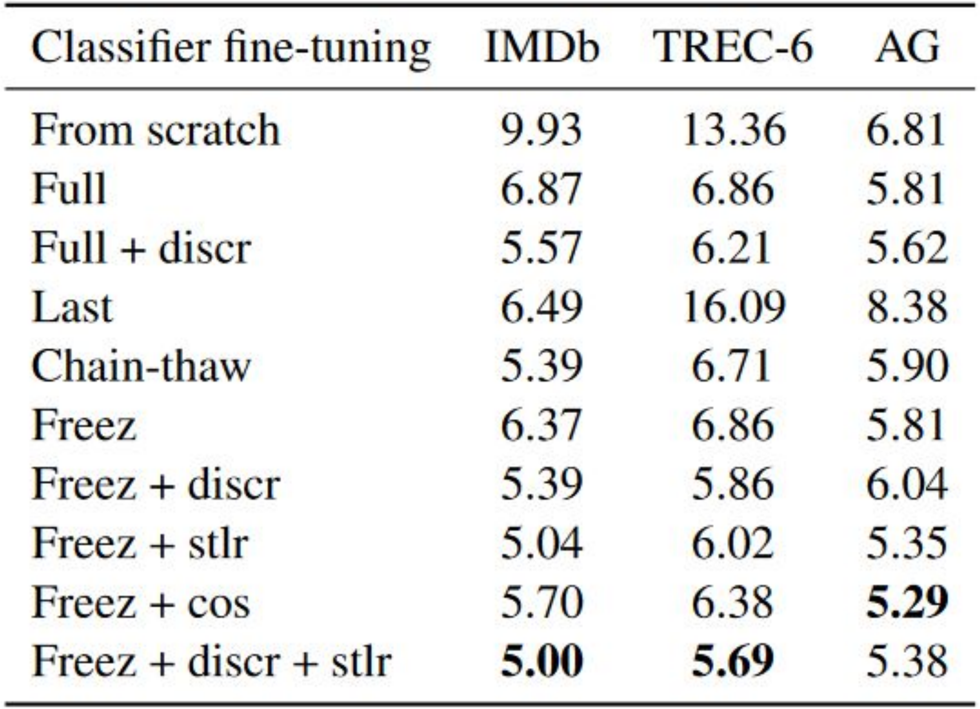
Freezing prevents **catastrophic forgetting** : model deciding to screw pre-trained info and overfit on this dataset.

Other Analysis: **Freezing**

Other Analysis: Fine-Tuning

Assess the impact of:

- Training from scratch
- Fine tuning the full model (implies pre-trained LM)
- Fine tuning the last layer
- Gradual Unfreezing v/s *chain-thawing*
- Discriminative Fine tuning (diff LR per layer)
- Slanted triangular LR v/s *aggressive cosine annealing [6]*



Classifier fine-tuning	IMDb	TREC-6	AG
From scratch	9.93	13.36	6.81
Full	6.87	6.86	5.81
Full + discr	5.57	6.21	5.62
Last	6.49	16.09	8.38
Chain-thaw	5.39	6.71	5.90
Freez	6.37	6.86	5.81
Freez + discr	5.39	5.86	6.04
Freez + stlr	5.04	6.02	5.35
Freez + cos	5.70	6.38	**5.29**
Freez + discr + stlr	**5.00**	**5.69**	5.38

Other Analysis: Fine-Tuning

References

[1] Howard, Jeremy, and Sebastian Ruder. "Fine-tuned Language Models for Text Classification." arXiv preprint arXiv:1801.06146 (2018).

[2] Peters, Matthew E., et al. "Deep contextualized word representations." arXiv preprint arXiv:1802.05365 (2018).

[3] Mou, Lili, et al. "How Transferable are Neural Networks in NLP Applications?." arXiv preprint arXiv:1603.06111 (2016).

[4] Dai, Andrew M., and Quoc V. Le. "Semi-supervised sequence learning." Advances in Neural Information Processing Systems. 2015.

[5] Smith, Leslie N. "Cyclical learning rates for training neural networks." Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on. IEEE, 2017.

[6] Loshchilov, Ilya, and Frank Hutter. "Sgdr: Stochastic gradient descent with warm restarts." (2016).

[7] Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).

[8] Merity, Stephen, Nitish Shirish Keskar, and Richard Socher. "Regularizing and optimizing LSTM language models." arXiv preprint arXiv:1708.02182 (2017).