

# MultiFiT: Efficient Multi-lingual Language Model Fine-tuning

Julian Eisenschlos<sup>1†\*</sup>

Marcin Kardas<sup>4\*</sup>

<sup>1</sup>ASAPP, Inc.

<sup>4</sup>n-waves, Wrocław

Sebastian Ruder<sup>2,3‡\*</sup>

Sylvain Gugger<sup>5</sup>

<sup>2</sup>National University of Ireland

<sup>5</sup>fast.ai

Piotr Czapla<sup>4\*</sup>

Jeremy Howard<sup>5,6</sup>

<sup>3</sup>Aylien Ltd., Dublin

<sup>6</sup>University of San Francisco

## Abstract

Pretrained language models are promising particularly for low-resource languages as they only require unlabelled data. However, training existing models requires huge amounts of compute, while pretrained cross-lingual models often underperform on low-resource languages. We propose Multi-lingual language model Fine-Tuning (MultiFiT) to enable practitioners to train and fine-tune language models efficiently in their own language. In addition, we propose a zero-shot method using an existing pretrained cross-lingual model. We evaluate our methods on two widely used cross-lingual classification datasets where they outperform models pretrained on orders of magnitude more data and compute. We release all models and code<sup>1</sup>.

## 1 Introduction

Pretrained language models (LMs) have shown striking improvements on a range of natural language processing (NLP) tasks (Peters et al., 2018a; Howard and Ruder, 2018; Devlin et al., 2018). These models only require unlabelled data for training and are thus particularly useful in scenarios where labelled data is scarce. As much of NLP research has focused on the English language, the larger promise of these models is to bridge the digital language divide<sup>2</sup> and enable the application of NLP methods to many of the world’s other 6,000 languages where labelled data is less plentiful.

Recently, cross-lingual extensions of these LMs have been proposed that train on multiple languages jointly (Artetxe and Schwenk, 2018; Lample and Conneau, 2019). These models are able

to perform zero-shot learning, only requiring labelled data in the source language. However, source data in another language may often not be available, whereas obtaining a small number of labels is typically straightforward.

Furthermore such models have several downsides: a) some variants rely on large amounts of parallel data, which may not be available for truly low-resource languages; b) they require a huge amount of compute for training<sup>3</sup>; and c) cross-lingual models underperform on low-resource languages—precisely the setting where they would be most useful. We are aware of two possible reasons for this: 1) Languages that are less frequently seen during training are underrepresented in the embedding space.<sup>4</sup> 2) Infrequent scripts are over-segmented in the shared word piece vocabulary (Wang et al., 2019).

In this work, we show that small monolingual LMs are able to outperform expensive cross-lingual models both in the zero-shot and the supervised setting. We propose Multi-lingual language model Fine-tuning (MultiFiT) to enable practitioners to train and fine-tune language models efficiently.<sup>5</sup> Our model combines universal language model fine-tuning (ULMFiT; Howard and Ruder, 2018) with the quasi-recurrent neural network (QRNN; Bradbury et al., 2017) and subword tokenization (Kudo, 2018) and can be pretrained on a single Tesla V100 GPU in a few hours. In addition, we propose to use a pretrained cross-lingual model’s predictions as pseudo labels to adapt the

<sup>1</sup><http://nlp.fast.ai>

\*The first four authors contributed equally.

†Corresponding author: [eisenjulian@gmail.com](mailto:eisenjulian@gmail.com)

‡Sebastian is now affiliated with DeepMind.

<sup>2</sup><http://labs.theguardian.com/digital-language-divide/>

<sup>3</sup>The training cost is amortized over time as pretraining only needs to be performed once and fine-tuning is much cheaper. However, if a model needs to be applied to a new language or a domain not covered by the model, a new model needs to be trained from scratch.

<sup>4</sup>This is similar to how word embeddings are known to underperform on low-frequency tokens (Gong et al., 2018).

<sup>5</sup>We use ‘multilingual’ as referring to training *independent* models in multiple languages. We use ‘cross-lingual’ to refer to training a *joint* model across multiple languages.

monolingual language model to the zero-shot setting. We evaluate our models on two widely used cross-lingual classification datasets, MLDoc (Schwenk and Li, 2018) and CLS (Prettenhofer and Stein, 2010) where we outperform the state-of-the-art zero-shot model LASER (Artetxe and Schwenk, 2018) and multi-lingual BERT (Devlin et al., 2018) in the supervised setting—even without any pretraining. In the zero-shot setting, we outperform both models using pseudo labels—and report significantly higher performance with as little as 100 examples. We finally show that information from monolingual and cross-lingual language models is complementary and that pretraining makes models robust to noise.

## 2 Related work

**Pretrained language models** Pretrained language models based on an LSTM (Peters et al., 2018a; Howard and Ruder, 2018) and a Transformer (Radford et al., 2018; Devlin et al., 2018) have been proposed. Recent work (Peters et al., 2018b) suggests that—all else being equal—an LSTM outperforms the Transformer in terms of downstream performance. For this reason, we use a variant of the LSTM as our language model.

### Cross-lingual pretrained language models

The multi-lingual BERT model is pretrained on the Wikipedias of 104 languages using a shared word piece vocabulary. LASER (Artetxe and Schwenk, 2018) is trained on parallel data of 93 languages with a shared BPE vocabulary. XLM (Lample and Conneau, 2019) additionally pretrains BERT with parallel data. These models enable zero-shot transfer, but achieve lower results than monolingual models. In contrast, we focus on making the training of monolingual language models more efficient in a multi-lingual context. Concurrent work (Mulcaire et al., 2019) pretrains on English and another language, but shows that cross-lingual pretraining only helps sometimes.

**Multi-lingual language modeling** Training language models in non-English languages has only recently received some attention. Kawakami et al. (2017) evaluate on seven languages. Cotterell et al. (2018) study 21 languages. Gerz et al. (2018) create datasets for 50 languages. All of these studies, however, only create small datasets, which are inadequate for pretraining language models. In contrast, we are among the first to report the

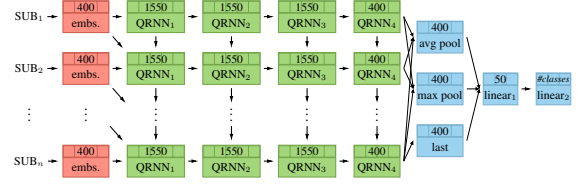


Figure 1: The MultiFiT language model with classifier consisting of a subword embedding layer, four QRNN layers, an aggregation layer, and two linear layers.

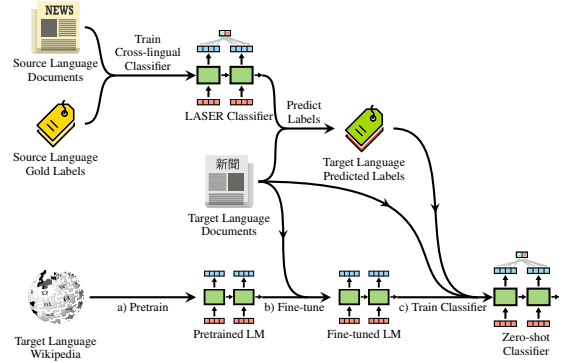


Figure 2: The steps of our cross-lingual bootstrapping method for zero-shot cross-lingual transfer. a) A monolingual language model (LM) is pretrained on target language data; b) the LM is fine-tuned on target language documents; and c) the LM is fine-tuned as a classifier using the zero-shot predictions from a linear classification layer fine-tuned on top of cross-lingual representations from LASER.

performance of monolingual language models on downstream tasks in multiple languages.

## 3 Our method

### 3.1 Multi-lingual Fine-Tuning

We propose Multi-lingual Fine-tuning (MultiFit). Our method uses the ULMFiT model (Howard and Ruder, 2018) with discriminative fine-tuning as foundation. ULMFiT is based on a 3-layer AWD-LSTM (Mer, 2017) language model. The AWD-LSTM is a regular LSTM (Hochreiter and Schmidhuber, 1997) with tuned dropout hyperparameters. To enable faster training and fine-tuning of the model, we replace it with a QRNN (Bradbury et al., 2017). The QRNN alternates convolutional layers, which are parallel across timesteps, and a recurrent pooling function, which is parallel across channels. It has been shown to outperform LSTMs, while being up to  $16\times$  faster at train and test time. ULMFiT in addition is restricted to words as input. To make our model

more robust across languages, we use subword tokenization based on a unigram language model (Kudo, 2018), which is more flexible compared to byte-pair encoding (Sennrich et al., 2016). We additionally employ label smoothing (Szegedy et al., 2016) and a novel cosine variant of the one-cycle policy (Smith, 2018)<sup>6</sup>, which we found to outperform ULMFiT’s slanted triangular learning rate schedule and gradual unfreezing. The full model can be seen in Figure 1.

### 3.2 Cross-lingual Bootstrapping

Prior methods have employed cross-lingual training strategies relying on parallel data and a shared BPE vocabulary. These can be combined with our language model, but increase its training complexity. For the case where an existing pre-trained cross-lingual model and source language data are available, we propose a bootstrapping method (Ruder and Plank, 2018) that uses the pre-trained model’s zero-shot predictions as pseudo labels to fine-tune the monolingual model on target language data.

The steps of the method can be seen in Figure 2. Specifically, we first fine-tune a linear classification layer on top of pretrained cross-lingual representations on source language training data. We then apply this cross-lingual classifier to the target language data and store its predicted label for every example. We now fine-tune our pretrained LM on the target language data and these pseudo labels<sup>7</sup>. Importantly, this method enables our monolingual LM to significantly outperform its cross-lingual teacher in the zero-shot setting (§5).

## 4 Experimental setup

This section provides an overview of our experimental setup; see the appendix for full details.

**Data** We evaluate our models on the Multilingual Document Classification Corpus (MLDoc; Schwenk and Li, 2018)<sup>8</sup>—a new subset of Reuters Corpus Volume 2 (Lewis et al., 2004) with balanced class priors for eight languages—and on the Cross-Lingual Sentiment dataset (CLS; Prettenhofer and Stein, 2010)<sup>9</sup> consisting of Amazon

<sup>6</sup>The idea is due to private conversation with the author.

<sup>7</sup>Distillation (Hinton et al., 2015) yielded similar results.

<sup>8</sup><https://github.com/facebookresearch/MLDoc>

<sup>9</sup><https://webis.de/data/webis-cls-10.html>

	Domain	Languages	Train	Dev	Test
MLDoc	News	EN, DE, ES, FR, IT, JA, RU, ZH	1k / 2k / 5k / 10k	2k	10k
CLS	Product reviews	EN, DE, FR, JA	2k	-	2k

Table 1: The domain, languages, and number of training, development, and test examples in each dataset.

	DE	ES	FR	IT	JA	RU	ZH
<i>Zero-shot (1,000 source language examples)</i>							
MultiCCA	81.20	72.50	72.38	69.38	67.63	60.80	74.73
LASER, paper	86.25	<b>79.30</b>	78.30	70.20	60.95	67.25	70.98
LASER, code	87.65	75.48	84.00	71.18	64.58	66.58	76.65
MultiBERT	82.35	74.98	83.03	68.27	64.58	<b>71.58</b>	66.17
MultiFiT, pseudo	<b>91.62</b>	79.10	<b>89.42</b>	<b>76.02</b>	<b>69.57</b>	67.83	<b>82.48</b>
<i>Supervised (100 target language examples)</i>							
MultiFit	90.90	89.00	85.03	80.12	80.55	73.55	88.02
<i>Supervised (1,000 target language examples)</i>							
MultiCCA	93.70	94.45	92.05	85.55	85.35	85.65	87.30
LASER, paper	92.70	88.75	90.80	85.93	85.15	84.65	88.98
MultiBERT	94.00	95.15	93.20	85.82	87.48	86.85	90.72
Monolingual BERT	94.93	-	-	-	-	-	92.17
MultiFiT, no wiki	95.23	95.07	94.65	89.30	88.63	87.52	90.03
MultiFiT	<b>95.90</b>	<b>96.07</b>	<b>94.75</b>	<b>90.25</b>	<b>90.03</b>	<b>87.65</b>	<b>92.52</b>

Table 2: Comparison of zero-shot and supervised methods on MLDoc.

product reviews in four languages. We provide an overview of the datasets in Table 1.

**Pretraining** We pretrain our models on 100M tokens extracted from the Wikipedia of the corresponding language for 10 epochs. As fewer tokens might be available for some languages, we also compare against a version (no wiki) that uses no pretraining. For all models, we fine-tune the LMs on the target data of the same language for 20 epochs. We perform subword tokenization with the unigram language model (Kudo, 2018).

**Evaluation settings** We compare two settings based on the availability of source and target language data: supervised and zero-shot. In the supervised setting, every model is fine-tuned and evaluated on examples from the target language. In the zero-shot setting, every model is fine-tuned on source language examples and evaluated on target language examples. In all cases, we use English as the source language.

**Baselines** We compare against the state-of-the-art cross-lingual embedding models LASER (Artetxe and Schwenk, 2018), which uses a large parallel corpus, multilingual BERT (MultiB-

		DE			FR			JA		
		Books	DVD	Music	Books	DVD	Music	Books	DVD	Music
<i>Zero-shot</i>	LASER, code	84.15	78.00	79.15	83.90	83.40	80.75	74.99	74.55	76.30
	MultiBERT	72.15	70.05	73.80	75.50	74.70	76.05	65.41	64.90	70.33
	MultiFiT, pseudo	<b>89.60</b>	<b>81.80</b>	<b>84.40</b>	<b>87.84</b>	<b>83.50</b>	<b>85.60</b>	<b>80.45</b>	<b>77.65</b>	<b>81.50</b>
<i>Translat.</i>	MT-BOW	79.68	77.92	77.22	80.76	78.83	75.78	70.22	71.30	72.02
	CL-SCL	79.50	76.92	77.79	78.49	78.80	77.92	73.09	71.07	75.11
	BiDRL	84.14	84.05	84.67	84.39	83.60	82.52	73.15	76.78	78.77
<i>Super.</i>	MultiBERT	86.05	84.90	82.00	86.15	86.90	86.65	80.87	82.83	79.95
	MultiFiT	<b>93.19</b>	<b>90.54</b>	<b>93.00</b>	<b>91.25</b>	<b>89.55</b>	<b>93.40</b>	<b>86.29</b>	<b>85.75</b>	<b>86.59</b>

Table 3: Comparison of zero-shot, translation-based and supervised methods (with 2k training examples) on all domains of CLS. MT-BOW and CL-SCL results are from (Zhou et al., 2016).

	LSTM	QRNN
Language model pretraining	143	<b>71</b>
Classifier fine-tuning	467	<b>156</b>

Table 4: Comparison of LSTM and QRNN per-batch training speed on a Tesla V100 (in ms) in MultiFiT.

ERT)<sup>10</sup>, and monolingual BERT<sup>11</sup>. We also compare against the best models on each dataset, MultiCCA (Ammar et al., 2016), a cross-lingual word embedding model, and BiDRL (Zhou et al., 2016), which translates source and target data.

**Our methods** We evaluate our monolingual LMs in the supervised setting (MultiFiT) and our LMs fine-tuned with pseudo labels from LASER in the zero-shot setting (pseudo).

	DE	ES	ZH
ULMFiT	94.19	95.23	66.82
MultiFiT, no wiki	95.23	95.07	90.03
MultiFiT, small Wiki	95.37	95.30	89.80
MultiFiT	<b>95.90</b>	<b>96.07</b>	<b>92.52</b>

Table 5: Comparison of MultiFiT results with different pretraining corpora and ULMFiT, fine-tuned with 1k labels on MLDoc.

## 5 Results

**MLDoc** We show results for MLDoc in Table 2. In the zero-shot setting, MultiBERT underper-

<sup>10</sup><https://github.com/google-research/bert/blob/master/multilingual.md>

<sup>11</sup>Models are available for English, Chinese, and German (<https://deepset.ai/german-bert>).

forms the comparison methods as the shared embedding space between many languages is overly restrictive. Our monolingual LMs outperform their cross-lingual teacher LASER in almost every setting. When fine-tuned with only 100 target language examples, they are able to outperform all zero-shot approaches except MultiFiT on DE and FR. This calls into question the need for zero-shot approaches, as fine-tuning with even a small number of target examples is able to yield superior performance. When fine-tuning with 1,000 target examples, MultiFiT—even without pretraining—outperforms all comparison methods, including monolingual BERT.

**CLS** We show results for CLS in Table 3. MultiFiT is able to outperform its zero-shot teacher LASER across all domains. Importantly, the bootstrapped monolingual model also outperforms more sophisticated models that are trained on translations across almost all domains. In the supervised setting, MultiFiT similarly outperforms multilingual BERT. For both datasets, our methods that have been pretrained on 100 million tokens outperform both multilingual BERT and LASER, models that have been trained with orders of magnitude more data and compute.

## 6 Analysis

**Speed** We compare the LSTM and QRNN cell in MultiFiT based on the speed for processing a single batch for pretraining and fine-tuning in Table 4. MultiFiT with a QRNN pretrains and fine-tunes about 2× and 3× faster respectively.

**MultiFiT vs. ULMFiT** We compare MultiFiT pretrained on 100M Wikipedia tokens against



ULMFiT pretrained on the same data using a 3-layer LSTM and spaCy tokenization<sup>12</sup> as well as MultiFiT pretrained on 2M Wikipedia tokens, and MultiFiT with no pretraining in Table 5. Pretraining on more data generally helps. MultiFiT outperforms ULMFiT significantly; the performance improvement is particularly pronounced in Chinese where ULMFiT’s word-based tokenization underperformed.

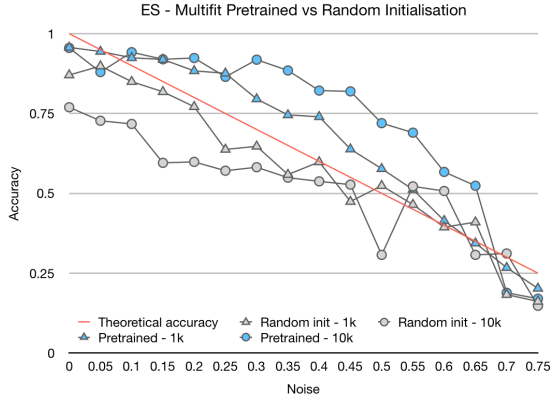


Figure 3: Comparison of MultiFiT’s robustness to label noise on MLDoc with and without pretraining.

	DE	ES	FR	IT	JA	RU	ZH
LASER, code	87.65	75.48	84.00	71.18	64.58	66.58	76.65
Random init. (1k)	77.80	70.50	75.65	68.52	68.50	61.37	79.19
Random init. (10k)	90.53	69.75	87.40	72.72	67.55	63.67	81.44
MultiFiT, pseudo (1k)	<b>91.34</b>	<b>78.92</b>	<b>89.45</b>	<b>76.00</b>	<b>69.57</b>	<b>68.19</b>	<b>82.45</b>

Table 6: Bootstrapping results on MLDoc with and without pretraining, trained on 1k/10k LASER labels.

	DE	ES	FR	IT	RU
Word-based	95.28	95.97	94.72	89.97	<b>88.02</b>
Subword	<b>96.10</b>	<b>96.07</b>	<b>94.75</b>	<b>94.75</b>	87.65

Table 7: Comparison of different tokenization strategies for different languages on MLDoc.

**Robustness to noise** We suspect that MultiFiT is able to outperform its teacher as the information from pretraining makes it robust to label noise. To test this hypothesis, we train MultiFiT and a randomly initialized model with the same architecture on 1k and 10k examples of the Spanish MLDoc. We randomly perturb labels with a probability ranging from 0-0.75 and show results in Figure 3. The pretrained MultiFiT is able to partially

<sup>12</sup><https://spacy.io/api/tokenizer>

ignore the noise, up to 65% of noisy training examples. Without pretraining, the model does not exceed the theoretical baseline (the percentage of correct examples). In addition, we compare MultiFiT with and without pretraining in Table 6. Pretraining enables MultiFiT to achieve much better performance compared to a randomly initialised model. Both results together suggest a) that pretraining increases robustness to noise and b) that information from monolingual and cross-lingual language models is complementary.

**Tokenization** Subword tokenization has been found useful for language modeling with morphologically rich languages (Czapla et al., 2018; Mielke and Eisner, 2019) and has been used in recent pretrained LMs (Devlin et al., 2018), but its concrete impact on downstream performance has not been observed. We train models with the best performing vocabulary sizes for subword (15k) and regular word-based tokenization (60k) with the Moses tokenizer (Koehn et al., 2007) on German and Russian MLDoc and show results in Table 7. Subword tokenization outperforms word-based tokenization on most languages, while being faster to train due to the smaller vocabulary size.

## 7 Conclusion

We have proposed novel methods for multilingual fine-tuning of languages that outperform models trained with far more data and compute on two widely studied cross-lingual text classification datasets on eight languages in both the zero-shot and supervised setting.

## References

- 2017. Pointer Sentinel Mixture Models. In *Proceedings of ICLR 2017*.
- Waleed Ammar, George Mulcaire, Yulia Tsvetkov, Guillaume Lample, Chris Dyer, and Noah A. Smith. 2016. *Massively Multilingual Word Embeddings*.
- Mikel Artetxe and Holger Schwenk. 2018. *Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond*. *arXiv preprint arXiv:1812.10464*.
- James Bradbury, Stephen Merity, Caiming Xiong, and Richard Socher. 2017. *Quasi-Recurrent Neural Networks*. In *Proceedings of ICLR 2017*.
- Ryan Cotterell, Sebastian Mielke, Jason Eisner, and Brian Roark. 2018. Are all languages equally hard to language-model? In *Proceedings of NAACL-HLT 2018*, pages 536–541.

- Piotr Czapla, Jeremy Howard, and Marcin Kardas. 2018. Universal language model fine-tuning with subword tokenization for polish. In *Proceedings of PolEval 2018 Workshop*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. [BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding](#).
- Daniela Gerz, Ivan Vulić, Edoardo Ponti, Jason Naradowsky, Roi Reichart, and Anna Korhonen. 2018. Language Modeling for Morphologically Rich Languages: Character-Aware Modeling for Word-Level Prediction. *Proceedings of TACL 2018*.
- Chengyue Gong, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2018. [FRAGE: Frequency-Agnostic Word Representation](#). In *Proceedings of NIPS 2018*.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. [Distilling the Knowledge in a Neural Network](#). *arXiv preprint arXiv:1503.02531*.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. [Long Short-Term Memory](#). *Neural Computation*, 9(8):1735–1780.
- Jeremy Howard and Sebastian Ruder. 2018. [Universal Language Model Fine-tuning for Text Classification](#). In *Proceedings of ACL 2018*.
- Kazuya Kawakami, Chris Dyer, and Phil Blunsom. 2017. [Learning to Create and Reuse Words in Open-Vocabulary Neural Language Modeling](#). In *ACL 2017*.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of ACL 2007*, pages 177–180.
- Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of ACL 2018*, pages 66–75.
- Guillaume Lample and Alexis Conneau. 2019. [Cross-lingual Language Model Pretraining](#). *arXiv preprint arXiv:1901.07291*.
- David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. 2004. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397.
- Sebastian J. Mielke and Jason Eisner. 2019. [Spell Once, Summon Anywhere: A Two-Level Open-Vocabulary Language Model](#). In *Proceedings of AAAI 2019*.
- Phoebe Mulcaire, Jungo Kasai, and Noah Smith. 2019. [Polyglot Contextual Representations Improve Crosslingual Transfer](#). In *Proceedings of NAACL 2019*.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018a. [Deep contextualized word representations](#). In *Proceedings of NAACL-HLT 2018*.
- Matthew E. Peters, Mark Neumann, Luke Zettlemoyer, Wen-tau Yih, Paul G Allen, and Computer Science. 2018b. [Dissecting Contextual Word Embeddings: Architecture and Representation](#). In *Proceedings of EMNLP 2018*.
- Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 157–163.
- Peter Prettenhofer and Benno Stein. 2010. [Cross-Language Text Classification using Structural Correspondence Learning](#). In *ACL '10 Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL '10)*, pages 1118–1127.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving Language Understanding by Generative Pre-Training.
- Sebastian Ruder and Barbara Plank. 2018. [Strong Baselines for Neural Semi-supervised Learning under Domain Shift](#). In *Proceedings of ACL 2018*.
- Holger Schwenk and Xian Li. 2018. [A Corpus for Multilingual Document Classification in Eight Languages](#). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. [Neural Machine Translation of Rare Words with Subword Units](#). In *Proceedings of ACL 2016*.
- Leslie N Smith. 2018. A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*.
- Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. [Rethinking the Inception Architecture for Computer Vision](#). In *Proceedings of CVPR 2016*.
- Xinyi Wang, Hieu Pham, Philip Arthur, and Graham Neubig. 2019. [Multilingual neural machine translation with soft decoupled encoding](#). In *Proceedings of ICLR 2019*.
- Xinjie Zhou, Xianjun Wan, and Jianguo Xiao. 2016. Cross-Lingual Sentiment Classification with Bilingual Document Representation Learning. *Proceedings of ACL 2016*, pages 1403–1412.

## A Hyper-parameters

The MultiFit architecture has 4 QRNN layers with a hidden dimensionality of 1550, a vocabulary size of 15,000 subword tokens, and an embedding size of 400. The vocabularies were computed using the SentencePiece<sup>13</sup> unigram language model (Kudo, 2018) with 99% character coverage for Chinese and Japanese and 100% for the rest. The encoder’s and decoder’s weights are shared (Press and Wolf, 2017). The output of the last QRNN layer (the last time step concatenated with an average and maximum pooled over time steps) is passed to the classifier with 2 dense layers.

Our language models were trained for 10 epochs on 100 million tokens of Wikipedia articles and then fine-tuned for 20 epochs on the corresponding dataset (MLDoc or CLS). The classifier was fine-tuned for 4 to 8 epochs. Results of the best model based on accuracy on the validation set are reported. We used a modified version of 1-cycle learning rate schedule (Smith, 2018) that uses cosine instead of linear annealing, cyclical momentum and discriminative finetuning (Howard and Ruder, 2018). Our batch size for language model training was 50 and for classification tasks 18. We were using BPTT of length 70. Due to the large amount of available training data our pretrained language models were trained without any dropout. We used the same dropout values as (Howard and Ruder, 2018) multiplied by 0.3 and 0.5 for fine-tuning of language models and the classification task respectively. We used weight decay of 0.01 for both tasks. The final regularization method was label smoothing (Szegedy et al., 2016) with epsilon of 0.1.

## B Speed comparison hyper-parameters

For the speed comparison, we use the same architecture and only change the underlying RNN cell (QRNN or LSTM). We pretrain and fine-tune both models on 15k tokens on a Tesla V100. For pre-training, we use a BPTT size of 70 and a batch size of 64. For fine-tuning, we use a batch size of 32.

---

<sup>13</sup><https://github.com/google/sentencepiece>