Universal Language Model Fine-Tuning for Polish Hate Speech Detection

Piotr Czapla (n-waves)
Sylvain Gugger (fast.ai)
Jeremy Howard (fast.ai)
Marcin Kardas (n-waves)

Abstract

Transfer learning in NLP allows for using large amounts of unlabelled text in unsupervised manner to dramatically reduce data necessary for a target task. However, the high performance of model on the source task does not indicate whether the final model will perform well on the target task. Our experiments with Universal Language for Fine-tuning (Howard i Ruder 2018) architecture run on PolEval 2019 Harmful Speech Detection task show that initial weights of language model play an important role in model performance on the target task. Interestingly, the language model's perplexity was not affected by the initial weights and in both studied cases the models performed equally well on the source task even though the performance differ significantly for the target task. We propose a simple mechanism to test if the sampled initial weights are well suited for the target task.

Finally, we present our solution for Harmful Speech Detection that achieves state-of-the-art performance and took first place in Task 6.1 of the PolEval'19 competition. Our model and source code are publicly available.[1]

Keywords

natural language processing, document classification, offensive speech detection

1. Introduction

Offensive speech is a growing problem on the Internet, amplified by the use of social media. According to Wirtualne Media (2018) in February 2018 there were 4.61 million active Polish Twitter users, which constituted 16.51% of all Polish Internet users. 4.52% were under 15

[1] `https://n-waves/ulmfit-multilingual`

years old. A popular approach of automatic detection of offensive speech is to use a curated list of forbidden words. The method often is ineffective at detecting instances of direct insults, cyber-bullying, or hate speech.

Recent work that uses language modeling as a source for transfer learning to classification tasks makes it possible to achieve higher performance than previously known transfer learning techniques. We present an extension to the Howard i Ruder (2018) ULMFiT architecture adapted to the morphological rich languages using subword tokenization (Kudo 2018), that let us win the first place on Task 6.1 of PolEval 2019 competition with an F1 score of 58.6%. The result were further improved during ablation studies and our best performing model achieves 62% F1 score.

We show how selection of the pretraining dataset is key to the good performance. Our ablation studies suggest that perplexity of the language model does not provide a strong indication of performance on down stream tasks. We show evidence that fine-tuning is ineffective to combat bad luck during initialization of language model weights, and the difference in performance between two initialization does not change when the pretraining dataset is changed. We propose an alternative way to quickly measure the applicability of the drawn weights on the downstream task. The method requires further testing. Our results align with the recent findings of Frankle i Carbin (2018) that highlight the importance of the weights drawn during initialization.

2. Related Work

Pretrained language models Pretrained language models based on an LSTM (Howard i Ruder 2018) and a Transformer (Devlin i in. 2018, Radford i in. 2019) have been proposed. Howard i Ruder (2018) used an English Wikipedia as a pretraining corpus to achieve state-of-the-art accuracy on several sentiment analysis datasets. Recent work by Peters i in. (2018) suggests that—all else being equal—an LSTM outperforms the Transformer in terms of downstream performance. For this reason, we use LSTM as our language model.

The importance of initialization The importance of the initial connections and the numbers returned by the random generator were mentioned previously by Frankle i Carbin (2018), Zhou i in. (2019). Zhang i in. (2019) also show that the upper layers of neural networks do not change much from their initial random weights. All of these findings inclined us to pretrain multiple language models. Our study confirms the importance of luck during initialization of the weights. We show that two sets of weights can have similar perplexity, but one will perform significantly better on the downstream classification task. This relation holds even when the underlying text corpus, tokenization and the order of training examples are changed.

Tanti i in. (2019) experimented with transfer learning for image caption generation. Similar to our findings, they noticed that the best language models (in terms of perplexity) do not result in the best caption generators after transfer learning.

Subword Tokenization for Polish Due to rich morphology of Polish language word-based models require much larger vocabularies and training data compared to English. This is why it is more common for such languages to use a subword tokenization. Czapla i in. (2018) used ULMFiT with subword tokenization for Polish language modelling achieving state-of-the-art perplexity on PolEval'18 LM corpus. The model with vocabulary consisting of 25K subword tokens was able to generalize conjugation and declension forms for words in new contexts that were not present in training corpus.

3. Experiments

Our solution uses Universal Language Model for Fine-tuning ULMFiT (Howard i Ruder 2018) with Sentence Piece tokenization, as in (Czapla i in. 2018). We use ULMFiT implementation from fast.ai library (Howard i in. 2019). It was pretrained using the Polish language part of reddit.com. We use weighted binary cross entropy as a loss function to handle class imbalance, and early stopping to minimize overfitting. In ablation studies we show that all of these decisions except for early stopping were critical to achieving good performance on the test set.

3.1. Weights of Language Model

The specific instance of weights has a significant impact on the performance of the downstream task; the relation holds even when other aspects of the training varies. We noticed this when training 4 ULMFiT models with weights sampled from random generator initialized with seed 0 and 1 for the Wikipedia and reddit pretraining corpuses. These pretrained models were then used to train 298 classification models that differed from each other in weights for classification heads, tokenization (different SentencePiece models), the number of the fine-tuning epochs (0, 6 and 20 epochs) on the PolEval dataset and the order of training examples. In every subset of the experiments, the seed 0 under-performed on the test set compared to the seed 1. The Table 1 and the histogram in Figure 1 show statistics across all classification models with respect to the initial seed used to initialize language model weights. This observation aligns with the recent work describing the importance of the model initialization (Frankle i Carbin 2018, Zhang i in. 2019).

The difference in the performance can be observed even when the language model was pretrained only for one epoch (instead of 10), see histogram in Figure 2. This suggests a quick way to search for the optimal weights of a language model for a particular task. Our experiments were done only on two sets of weights, and the validation set used in early stopping had training set distribution that was significantly different from test distribution which makes this results inconclusive but promising. We hypothesize that if this phenomenon is consistent, it may explain why larger models such as BERT (Devlin i in. 2018) underperform on classification tasks compared to (Howard i Ruder 2018), as such models are only fine-tuned for each classification task without new random initialization and pretraining which might be important for specific tasks.

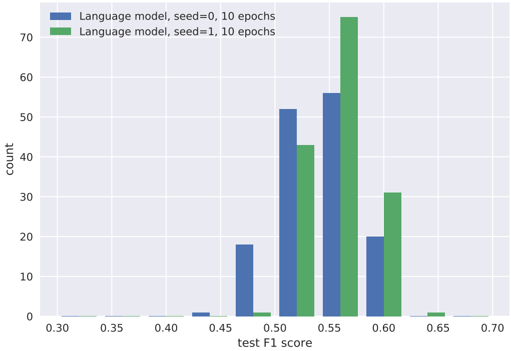

Figure 1: Comparison of distribution of F1 score on test set with seed=1 (green) and seed=0 (blue) for all experiments.

Table 1: Table showing how Seed 0 consistently under-performs compared to Seed 0.

	10 epochs of training on reddit & wiki			1 epochs of training on wiki		
	seed 1	seed 0	diff	seed 1	seed 0	diff
count	151	147		10	10	
mean	0.555511	0.539647	0.015865	0.525926	0.469891	0.056035
std	0.028428	0.033603	-0.005174	0.024596	0.03878	-0.014185
min	0.488479	0.451613	0.036866	0.495798	0.394231	0.101568
25%	0.536181	0.515420	0.020761	0.504658	0.443662	0.060996
50%	0.558333	0.541485	0.016849	0.526767	0.47806	0.048707
75%	0.576201	0.563492	0.012709	0.53933	0.490383	0.048947
max	0.622222	0.614232	0.007990	0.564885	0.523809	0.041076

3.2. Datasets

Harmful Speech dataset

The dataset consists of 10K tweets in the training set and 1K tweets in the test set, all labelled either as harmful or non-harmful. The training dataset was used for unsupervised fine-tuning of language models. The test set had 13.4% of harmful tweets which is more than the training set and less retweets compared to the training set.

Reddit comments

We used Google BigQuery and a public dataset[2] of comments from Reddit, a social media platform, to extract comments from all subreddits marked as Polish. According to OpenNLP

[2] https://bigquery.cloud.google.com/table/fh-bigquery:reddit_comments.2015_05

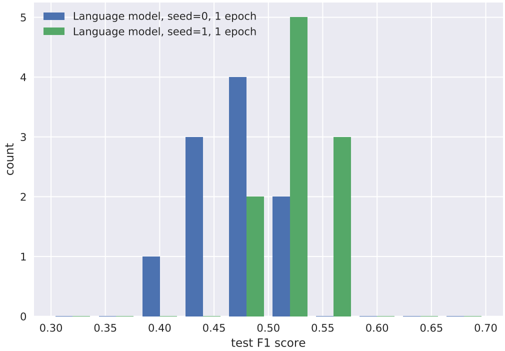

Figure 2: Comparison of distribution of F1 score on test set of classifiers based on two language models, initialized with seed=1 (green) and seed=0 (blue), that were pretrained for 1 epoch.

Table 2: Summary of PolEval 2019 Harmful Speech datasets. In deduplication tweet and retweet are considered the same.

dataset	tweets	harmful	tokens / tweet
train	10041	8.48%	12.4
train (dedup.)	9400	8.05%	12.2
test	1000	13.40%	12.2
test (dedup.)	946	12.79%	12.0

language detector, 67% of obtained comments use Polish and 23% use English. The reddit dataset is preprocessed with the default fastai.text (v1.0.51) transformations.

Wikipedia

The wiki dataset was downloaded from the Mediawiki dumps. It was pre-tokenized using Moses Tokenizer for consistency with WikiText-103 (Merity i in. 2016) and transformed using fastai.text (v1.0.51) transformations (see Howard i in. (2019)).

3.3. Architecture

We use Universal Language Model for Fine-Tuning (Howard i Ruder 2018) with hyperparameters as presented in Table 3.

Tokenization and preprocessing

We used sentence piece unigram model (Kudo 2018) for tokenization, following architecture described in (Czapla i in. 2018). The unigram model was trained on the language model

Table 3: Details of our submission.

language model	vocabulary size	25 K
	RNN type	LSTM
	recurrent layers	4
	embeddings dimension	400
	hidden state dimension	1150
	training time	12 epochs
	peak learning rate	0.01
	batch size	160
	BPTT	70
	data set	reddit comments
fine-tuning	training time	6 epochs
	dropout	no
	peak learning rate	0.001
	batch size	128
classifier	training time	8 epochs
	loss	weighted cross entropy
	dropout	0.1
	linear layers	2
	batch size	320
results	precision	66.67%
	recall	52.24%
	F1 score	58.58%
	accuracy	90.10%

pretraining corpus with 25K subword tokens limit, including 100% characters in the corpus alphabet. We do not use subword regularization during training or inference. The goal of preprocessing step was mainly to normalize texts between language model training corpus and tweets, as well as to remove parts that we considered noise (links, user names, numbers). We also replaced emojis and some emoticons with their descriptions taken from The Unicode Consortium and Wikipedia's list of emoticons. We removed duplicated tweets in an attempt to make make the training and validation sets independent.

Pretraining and fine-tuning

Our models were pretrained for 10–12 epochs on our reddit dataset. The training is relatively quick and takes only 4 hours to complete on a single GPU, which allowed us to experiment with different modifications to the architecture. The sentence piece tokenization model is trained on the first dataset and it is left unchanged during the fine-tuning and classification. It is one of the reasons why we used reddit instead of Wikipedia. The corpus was close enough to the Poleval dataset that the fine-tuning step was not necessary, and both models with and without fine-tuning performed well. On the other hand, language models trained on

Wikipedia during ablation studies performed worse without fine-tuning. See Table 5 for more details.

Classifier

As shown in Table 2 the datasets are highly unbalanced. To mitigate the fact we used weighted binary cross entropy as a loss function:

$$\mathscr{L}(y,\hat{y}) = -\frac{1}{m}\sum_{i=1}^{m}(30y_i \ln \hat{y}_i + 0.5(1-y_i)\ln(1-\hat{y}_i)),$$

where m is a mini-batch size, y_i is a true label of the i-th training example and $\hat{y}_i$ is model's prediction.

3.4. Submission

We selected our model for submission by looking at the F1 score on a validation set of 10% of the training set. The model was one of the first that we trained during competition. The later models were exploring a number of alternative methods to improve our accuracy, including use of an English hate speech corpus to train multilingual models, such as Laser by Artetxe i Schwenk (2018). The attempts where not successful. During ablation studies we noticed that ULMFiT has high variability in performance, depending on the weight initialization of both the classifier and the language model. To draw meaningful conclusions we trained around 500 classifiers. Some of them had much better performance. Unfortunately we noticed that the F1 performance on our validation set is slightly negatively correlated with the performance on the test set. We performed a number of experiments in order to align the validation set with test set. The only successful attempt that gave us a positive correlation was using half of the test set as the validation set. Unfortunately, this makes the selection of further models impossible without risking over-fitting to the test set. As shown in Table 2 the training and test sets have significantly different fraction of tweets labelled as harmful. It could be simply a result of increased hate speech rate during the time the test data was acquired. However, the difference in performance between validation and test sets in our experiments suggest that there might be a mismatch between distributions of labels, f.e., due to different sensitivity of annotators annotating each dataset.

4. Ablation studies

Our architecture have high variance of results between runs, even with all hyper parameters fixed. In order to mitigate the issue during our experiments we forced all executions to be deterministic. We fix seed values at 4 stages of our pipeline:

— at the beginning of pretraining, before language model weights are sampled

— at the beginning of fine-tuning, to fix the order in which tweets are shuffled

- at the beginning of classifier initialization, before classifier weights are sampled
- at the beginning of classifier training, to fix the order in which tweets are shuffled

For each experiment we used at least two pretrained language models, and trained 10 classification models for each model. Our results of the ablation studies are presented below in table 4. We found that increasing dropout does not improve the performance of the classifiers. The weighted cross entropy was crucial to achieve good results. Without weights the best results are worse than the average result trained with weighted cross-entropy. Early stopping was not necessary for the language model with seed 1 but was crucial for the language model with seed 0. Table 5 shows the summary of the experiment we executed in order to see if the fine-tuning was necessary to achieve good performance on the classification task. We fine-tuned the all 4 language models trained on reddit and wikipedia for 0 epochs (ie. no finetune) 6 and 20. The finetuning was not necessary for reddit to achieve good performance but was crucial for language models pretarined on wikipedia.

Table 4: Summary of ablation studies

dataset	exp_type	lmseed	mean	std	max	75%
reddit	dropmul = 0.5	0	0.519352	0.030414	0.589552	0.533757
reddit	dropmul = 0.5	1	0.538298	0.030352	0.602151	0.557069
wiki	1 epoch pretraining	0	0.469891	0.038780	0.523809	0.490383
wiki	1 epoch pretraining	1	0.525926	0.024596	0.564885	0.539330
wiki	cross entropy w/o weights	0	0.433285	0.062494	0.521739	0.487437
wiki	cross entropy w/o weights	1	0.451950	0.050503	0.539535	0.490566
wiki	w/o early stopping	0	0.516319	0.033725	0.564315	0.546150
wiki	w/o early stopping	1	0.564405	0.019395	0.608392	0.574534
wiki	our model	0	0.523124	0.027906	0.570470	0.540592
wiki	our model	1	0.550656	0.027349	0.608392	0.573604
reddit	our model	0	0.553660	0.029906	0.614232	0.575757
reddit	our model	1	0.560869	0.028504	**0.622222**	**0.580522**

Table 5: Performance of models with and without fine-tuning,

data set	fine-tuning	mean	std	max	75%
wiki	no	0.522515	0.026661	0.582781	0.541998
wiki	yes	0.536890	0.030744	0.608392	0.558897
reddit	yes	0.561675	0.027365	0.622222	0.581680
reddit	no	0.573931	0.016832	0.603390	0.581451

We further explored the difference between language model and different weight initialization, and noticed a a possible reverse correlation between perplexity and the performance on the downstream task (see 3). However, 2 random initializations is not enough to draw conclusive results.

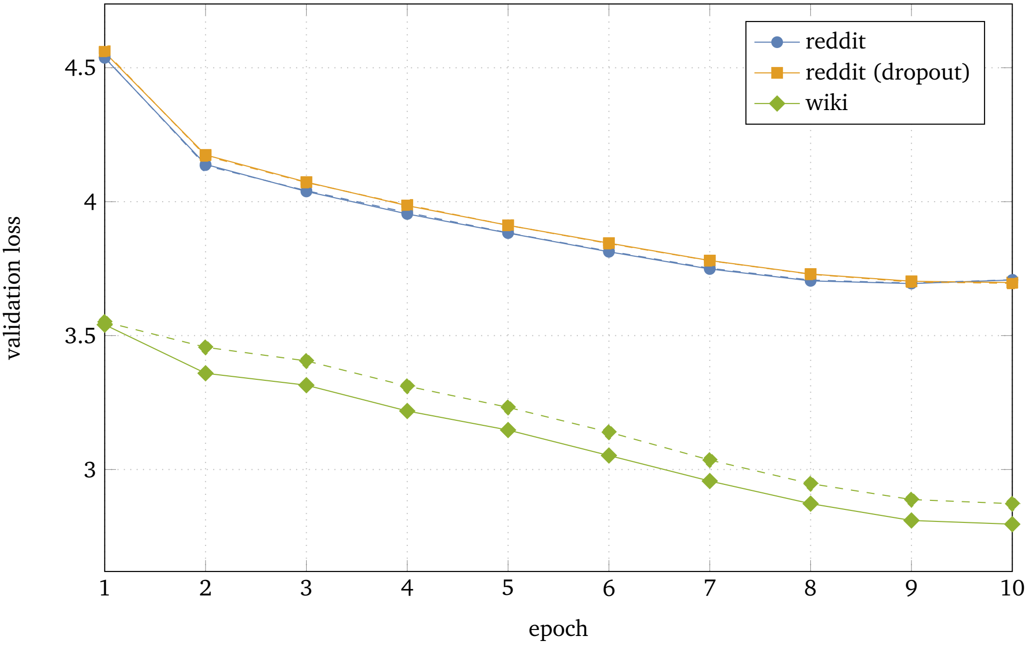

Figure 3: Validation loss of language models. Each setting was trained twice with seed values: 0 (solid lines) and 1 (dashed lines). The models with seed value 1 performed better than the models with seed value 0. The models pretrained on Reddit were performing better than models pretrained on Wikipedia.

5. Final Remarks

In Czapla i in. (2018) we showed that Universal Language Model for Fine-tuning complemented with subword tokenization achieves state-of-the-art perplexity in Polish language modelling. In this paper we presented experimental evidence that ULMFiT pretrained on Polish corpus can be successfully used for Polish documents classification.

It remained an open question whether high performance of ULMFiT on the language modelling task will translate to high performance on downstream tasks. Our experiments present, in accordance with findings from Tanti i in. (2019), evidence that this may not be the case. Therefore, to evaluate a model one is required to go through the whole iteration from pretraining language model through fine-tuning to training the model on the downstream task. We showed an alternative way of measuring the performance of sampled language model weights. The work is inconclusive but promising and should be further explored.

References

Artetxe M. i Schwenk H. (2018). *Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond*. „CoRR", abs/1812.10464.

Czapla P., Howard J. i Kardas M. (2018). *Universal language model fine-tuning with subword tokenization for polish*. [W:] *Proceedings of PolEval 2018 Workshop*.

Devlin J., Chang M.-W., Lee K. i Toutanova K. (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. „arXiv:1810.04805 [cs]". 00002 arXiv: 1810.04805.

Frankle J. i Carbin M. (2018). *The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks*. „arXiv:1803.03635 [cs]". 00033 arXiv: 1803.03635.

Howard J. i Ruder S. (2018). *Universal language model fine-tuning for text classification*. [W:] *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, s. 328–339. Association for Computational Linguistics.

Howard J., Gugger S. i in. (2019). *fastai*. https://github.com/fastai/fastai.

Kudo T. (2018). *Subword regularization: Improving neural network translation models with multiple subword candidates*. [W:] *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, s. 66–75. Association for Computational Linguistics.

Merity S., Xiong C., Bradbury J. i Socher R. (2016). *Pointer Sentinel Mixture Models*. „arXiv:1609.07843 [cs]". 00229 arXiv: 1609.07843.

Peters M. E., Neumann M., Zettlemoyer L., Yih W.-t., Allen P. G. i Science C. (2018). *Dissecting Contextual Word Embeddings: Architecture and Representation*. [W:] *Proceedings of EMNLP 2018*.

Radford A., Wu J., Child R., Luan D., Amodei D. i Sutskever I. (2019). *Language models are unsupervised multitask learners*.

Tanti M., Gatt A. i Camilleri K. P. (2019). *Transfer learning from language models to image caption generators: Better models may not transfer better*. „CoRR", abs/1901.01216.

Wirtualne Media (2018). *Wśród polskich użytkowników twittera przeważają mężczyźni, osoby z dużych miast i ze średnim lub wyższym wykształceniem (analiza)*.

Zhang C., Bengio S. i Singer Y. (2019). *Are All Layers Created Equal?* „arXiv:1902.01996 [cs, stat]". 00003 arXiv: 1902.01996.

Zhou H., Lan J., Liu R. i Yosinski J. (2019). *Deconstructing Lottery Tickets: Zeros, Signs, and the Supermask*. 00000.