On the comparability of pre-trained language models

Matthias Aßenmacher
Department of Statistics
Ludwig-Maximilians-Universität
Munich, Germany

Christian Heumann
Department of Statistics
Ludwig-Maximilians-Universität
Munich, Germany

{matthias,chris}@stat.uni-muenchen.de

Abstract

Recent developments in unsupervised representation learning have successfully established the concept of transfer learning in NLP. Instead of simply plugging in static pre-trained representations, end-to-end trainable model architectures are making better use of contextual information through more intelligently designed language modelling objectives. Along with this, larger corpora are used for self-supervised pre-training of models which are afterwards fine-tuned on supervised tasks. Advances in parallel computing made it possible to train these models with growing capacities in the same or even in shorter time than previously established models. These developments agglomerate in new state-of-the-art results being revealed in an increasing frequency. Nevertheless, we show that it is not possible to completely disentangle the contributions of the three driving forces to these improvements.

We provide a concise overview on several large pre-trained language models, which achieved state-of-the-art results on different leaderboards in the last two years, and compare them with respect to their use of new architectures and resources. We clarify where the differences between the models are and attempt to gain some insight into the single contributions of lexical and computational improvements as well as those of architectural changes. We do not intend to quantify these contributions, but rather see our work as an overview in order to identify potential starting points for benchmark comparisons.

1 Introduction

For solving NLP tasks, most researchers turn to using pre-trained word embeddings (Mikolov et al., 2013; Pennington et al., 2014; Bojanowski et al., 2017) as a key component of their models. These representations map each word of a sequence to a real valued vector of fixed dimension. Drawbacks of these kinds of externally learned features are that they are (i) fixed, i.e. can not be adapted to a specific domain they are used in, and (ii) context independent, i.e. there's only one embedding for a word by which it is represented in any context.

More recently, transfer learning approaches, as for example convolutional neural networks (CNNs) pre-trained on ImageNet (Krizhevsky et al., 2012) in computer vision, have entered the discussion. Transfer learning in the NLP context means pre-training a network with a self-supervised objective on large amounts of plain text and fine-tuning its weights afterwards on a task specific, labelled data set. For a comprehensive overview on the current state of transfer learning in NLP, we recommend the excellent tutorial and blog post by Ruder et al. (2019)[1].

With ULMFiT (**U**niversal **L**anguage **M**odel **Fi**ne **T**uning), Howard and Ruder (2018) proposed a LSTM-based (Hochreiter and Schmidhuber, 1997) approach for transfer learning in NLP using AWD-LSTMs (Merity et al., 2017). This model can be characterised as unidirectional contextual, while a bidirectionally contextual LSTM-based model was presented in ELMo (**E**mbeddings from **L**anguage **Mo**dels) by Peters et al. (2018).

The bidirectionality in ELMo is achieved by using

[1] https://ruder.io/state-of-transfer-learning-in-nlp/

biLSTMs instead of AWD-LSTMs. On the other hand, ULMFiT uses a more "pure" transfer learning approach compared to ELMo, as the ELMo-embeddings are extracted from the pre-trained model and are *not* fine-tuned in conjunction with the weights of the task-specific architecture.

The OpenAI GPT (**G**enerative **P**re-**T**raining, Radford et al., 2018) is a model which resembles the characteristics of ULMFiT in two crucial points. It is a unidirectional language model and it allows stacking task specific layers on top after pre-training, i.e. it is fully end-to-end trainable. The major difference between them is the internal architecture, where GPT uses a Transformer decoder architecture (Vaswani et al., 2017).

Instead of processing one input token at a time, like recurrent architectures (LSTMs, GRUs) do, Transformers process whole sequences all at once. This is possible because they utilize a variant of the *Attention* mechanism (Bahdanau et al., 2014), which allows modelling dependencies without having to feed the data to the model sequentially. At the same time, GPT can be characterised as unidirectional as it just takes into account the left side of the context. Its successor OpenAI GPT2 (Radford et al., 2019) possesses (despite some smaller architectural changes) the same model architecture and thus can also be termed as unidirectional contextual.

BERT (**B**idirectional **E**ncoder **R**epresentations from **T**ransformers, Devlin et al., 2019), and consequently the other two BERT-based approaches discussed here (Liu et al., 2019; Lan et al., 2019) as well, differ from the GPT models by the fact that they are bidirectional Transformer encoder models. Devlin et al. (2019) proposed *Masked Language Modelling* (MLM) as a special training objective which allows the use of a bidirectional Transformer encoder without compromising the language modelling objective. XLNet (Yang et al., 2019) on the contrary relies on an objective which the authors call *Permutation Language Modelling* (PLM) and is also able to model a bidirectional context despite being an auto-regressive model.

2 Related work

In their stimulating paper, Raffel et al. (2019) take several steps in a similar direction by trying to ensure comparability among different Transformer-based models. They perform various experiments with respect to the transfer learning ability of a Transformer encoder-decoder architecture by varying the pre-training objective (different variants of denoising vs. language modelling), the pre-training resources (their newly introduced C4 corpus vs. variants thereof) and the parameter size (from 200M up to 11B). Especially, their idea of introducing a new corpus and creating subsets resembling previously used corpora like RealNews (Zellers et al., 2019) or OpenWebText (Gokaslan and Cohen, 2019) is a promising approach in order to ensure comparability.

However, their experiments do not cover an important point we are trying to address with our work: Focussing on only one specific architecture does not yield an answer to the question which components explain the performance differences between models where the overall architecture differs (e.g. Attention-based vs. LSTM-based). Yang et al. (2019) also address comparability to some extent by performing an ablation study to compare their XLNet explicitly to BERT. They train six different XLNet-based models where they modify different parts of their model in order to quantify how these design choices influence performance. At the same time they restrict themselves to an architecture of the same size as BERT-BASE and use the same amount of lexical resources for pre-training. Liu et al. (2019) vary RoBERTa with respect to model size and amount of pre-training resources in order to perform an ablation study also aiming at comparability to BERT. Lan et al. (2019) go one step further with ALBERT by also comparing their model to BERT with regard to run time as well as width and depth of the model.

Despite all these experiments are highly valuable steps into the direction of better comparability, there are still no clear guidelines on which comparisons to perform in order to ensure a maximum degree of comparability with respect to multiple potentially influential factors at the same time.

3 Materials and Methods

First, we present the different corpora which were utilised for pre-training the models and compare them with respect to their size and their accessibility (cf. Tab. 1). Subsequently, we will briefly introduce benchmark data sets which the models are commonly fine-tuned and evaluated on.

While conceptual differences between the evaluated models have been addressed in the introduction, the models will now be described in more detail. This is driven by the intention to emphasise

differences beyond the obvious, conceptual ones.

3.1 Pre-training corpora

English Wikipedia Devlin et al. (2019) state that they used data from the English Wikipedia and provide a manual for crawling it, but no actual data set. Their version encompassed around 2.5B words. Wikipedia data sets are available in the Tensorflow `Datasets`-module.

CommonCrawl Among other resources, Yang et al. (2019) used data from CommonCrawl. Besides stating that they filtered out short or low-quality content, no further information is given. Since CommonCrawl is a dynamic database, which is updated on a monthly base (and the extracted amount of data always depends on the user) we can not provide a word count for this source in Tab. 1.

ClueWeb (Callan et al., 2009), Giga5 (Parker et al., 2011) The information about ClueWeb and Giga5 is similarly sparse as for CommonCrawl. ClueWeb was obtained by crawling $\sim$ 2.8M web pages in 2012, Giga5 was crawled between 01/2009 and 12/2010.

1B Word Benchmark[2] **(Chelba et al., 2013)** This corpus, actually introduced as a benchmark data set by Chelba et al. (2013), combines multiple data sets from the EMNLP 2011 workshop on Statistical Machine Translation. The authors normalised and tokenized the corpus and performed further pre-processing steps in dropping duplicate sentences as well as discarding words with a count below three. Additionally, they randomised the ordering of the sentences in the corpus. This constitutes a corpus with a vocabulary of 793.471 words and a total word count of 829.250.940 words.

BooksCorpus[3] **(Zhu et al., 2015)** In 2015, Zhu et al. introduced the BooksCorpus, which is heavily used for pre-training language models (cf. Tab. 1). In their work, they used the BooksCorpus in order to train a model for retrieving sentence similarity. Overall, the corpus comprises 984.846.357 words in 74.004.228 sentences obtained from analysing 11.038 books. They report a vocabulary consisting of 1.316.420 unique words, making the corpus lexically more diverse than the 1B Word Benchmark, as it possesses a by 66% larger vocabulary whereas having a word count which is only 19% higher.

Wikitext-103 (Merity et al., 2016a,b) The authors emphasised the necessity for a new large scale language modelling data set by stressing the shortcomings of other corpora. They highlight the occurrence of complete articles, which allows learning long range dependencies, as one of the main benefits of their corpus. This property is, according to the authors, not given in the 1B Word Benchmark as the sentence ordering is randomised there. With a count of 103.227.021 tokens and a vocabulary size of 267.735, it is about one eighth of the 1B Word Benchmark's size concerning token count and about one third concerning the vocabulary size. Note, that there is also the smaller Wikitext-2 corpus (Merity et al., 2016c) available, which is a subset of about 2% of the size of Wikitext-103.

CC-News (Nagel, 2016) This corpus was presented and used by Liu et al. (2019). They used a web crawler proposed by Hamborg et al. (2017) to extract data from the CommonCrawl News data set (Nagel, 2016) and obtained a data set similar to the RealNews data set (Zellers et al., 2019).

Stories[4] **(Trinh and Le, 2018)** The authors built a specific subset of the CommonCrawl data based on questions from common sense reasoning tasks. They extracted nearly 1M documents, most of which are taken from longer, coherent stories.

WebText (Radford et al., 2019) This pre-training corpus, obtained by creating "a new web scrape which emphasised document quality" (Radford et al., 2019), is not publicly available.

OpenWebText (Gokaslan and Cohen, 2019) As a reaction to Radford et al. (2019) *not* releasing their pre-training corpus, Gokaslan and Cohen (2019) started an initiative to emulate an open-source version of the WebText corpus.

It becomes obvious that there is a lot of heterogeneity with respect to the observed combinations of availability, quality and corpus size. Thus, we can state that there is some lack of transparency when it comes to the lexical resources used for per-training. Especially, the missing standardised availability of the BooksCorpus is problematic as this corpus is heavily used for pre-training.

[2] https://research.google/pubs/pub41880/
[3] https://yknzhu.wixsite.com/mbweb
[4] https://console.cloud.google.com/storage/browser/commonsense-reasoning/reproduce/stories_corpus

Corpora	Word-count$^{\heartsuit}$	Accessibility	Used by
English Wikipedia	$\sim$ 2.500M	Fully available	BERT; XLNet; RoBERTa; ALBERT
CommonCrawl	Unclear	Fully available	XLNet
ClueWeb 2012-B, Giga5	Unclear	Fully available ($$)	XLNet
1B Word Benchmark	$\sim$ 830M	Fully available	ELMo
BooksCorpus	$\sim$ 985M	Not available	GPT; BERT; XLNet; RoBERTa; ALBERT
Wikitext-103	$\sim$ 103M	Fully available	ULMFit
CC-News	Unclear	Crawling Manual	RoBERTa
Stories	$\sim$ 7.000M$^{\diamondsuit}$	Fully available	RoBERTa
WebText	Unclear	Not available	GPT2
OpenWebText	Unclear	Fully available	RoBERTa

Table 1: Pre-training resources (sorted by date). *Crawling Manual* means the authors did not provide data, but at least a manual for crawling it. Dollar signs signify the necessity of a payment in order to get access. RealNews (Zellers et al., 2019) and C4 (Raffel et al., 2019) are not included as they were not used by the evaluated models.
$^{\heartsuit}$ We report the word-count as given in the respective articles proposing the corpora. Note that the number of *tokens* reported in other articles depends on the tokenization scheme used by a specific model.
$^{\diamondsuit}$ Stated by one of the authors on twitter: https:/twitter.com/thtrieu_/status/1096672446864748545

3.2 Benchmark data sets for fine-tuning

GLUE[5] **(Wang et al., 2018)** The *General Language Understanding Evaluation* (GLUE) benchmark is a freely available collection of nine data sets on which models can be evaluated. It provides a fixed train-dev-test split with held out labels for the test set, as well as a leaderboard which displays the top submissions and the current state-of-the-art (SOTA). The relevant metric for the SOTA is an aggregate measure of the nine single task metrics. The benchmark includes two binary classification tasks with single-sentence inputs (CoLa [Warstadt et al., 2018] and SST-2 [Socher et al., 2013]) and five binary classification tasks with inputs that consist of sentence-pairs (MRPC [Dolan and Brockett, 2005], QQP [Shankar et al., 2017], QNLI, RTE and WNLI [all Wang et al., 2018]). The remaining two tasks also take sentence-pairs as input but have a multi-class classification objective with either three (MNLI [Williams et al., 2017]) or five classes (STS-B [Cer et al., 2017]).

SuperGLUE[6] **(Wang et al., 2019)** As a reaction to human baselines being surpassed by the top ranked models, Wang et al. (2019) proposed a set of benchmark data sets similar to, but, according to the authors, more difficult than GLUE. It did not make sense to include it as a part of our model comparison, as (at the time of writing) only two of the discussed models were evaluated on SuperGLUE.

SQuAD[7] **(Rajpurkar et al., 2016, 2018)** The **S**tanford **Qu**estion **A**nswering **D**ataset (SQuAD) 1.1 consists of 100.000+ questions explicitly designed to be answerable by reading segments of Wikipedia articles. The task is to correctly locate the segment in the text which contains the answer. A shortcoming is the omission of situations where the question is not answerable by reading the provided article. Rajpurkar et al. (2018) address this problem in SQuAD 2.0 by adding 50.000 handcrafted unanswerable questions to SQuAD 1.1. The authors provide a train and development set as well as an official leaderboard. The test set is completely held out, participants are required to upload their models to CodaLab. The SQuAD 1.1 data is, in an augmented form (QNLI), also part of GLUE.

RACE[8] **(Lai et al., 2017)** The Large-scale **R**e**A**ding **C**omprehension Dataset From **E**xaminations (RACE) contains English exam questions for Chinese students (middle/high school). In most of the articles using RACE for evaluation, it is described to be especially challenging due to (i) the length of the passages, (ii) the inclusion of reasoning questions and (iii) the intentionally tricky design of the questions in order to test a human's ability in reading comprehension. The data set can be subdivided

[5] https://gluebenchmark.com/
[6] https://super.gluebenchmark.com/
[7] https://rajpurkar.github.io/SQuAD-explorer/
[8] http://www.qizhexie.com/data/RACE_leaderboard.html

in RACE-M (middle school examination) and RACE-H (high school examination) and comprises a total of 97.687 questions on 27.933 passages of text.

3.3 Evaluated Models

ULMFit (Howard and Ruder, 2018) The AWD-LSTMs in this architecture make use of DropConnect (Wan et al., 2013) for better regularisation and apply averaged stochastic gradient descent (ASGD) for optimization (Polyak and Juditsky, 1992). The model consists of an embedding layer followed by three LSTM layers with a softmax classifier on top for pre-training. It is complemented by a task specific final layer during fine-tuning. The vocabulary size is limited to 30k words as in Johnson and Zhang (2017).
ULMFiT was not evaluated on GLUE, but on several other data sets (IMDb [Maas et al., 2011], TREC-6 [Voorhees and Tice, 1999], Yelp-bi, Yelp-full, AG's news, DBpedia [all Zhang et al., 2015]).

ELMo (Peters et al., 2018) Consisting of multiple biLSTM layers, one can extract multiple intermediate-layer representations from ELMo. These representations are used for computing a (task-specific) weighted combination, which is concatenated with external, static word embeddings. During the training of the downstream model, ELMo embeddings are not updated, only the weights for combining them are. For the GLUE benchmark there are multiple ELMo-based architectures available on the leaderboard. In Tab. 3, we report the best-performing model, an ELMo-based BiLSTM-model with Attention (Wang et al., 2018).

OpenAI GPT (Radford et al., 2018) The OpenAI GPT is a pure attention-based architecture that does not make use of any recurrent layers. Pre-training is performed by combining Byte-Pair encoded (Sennrich et al., 2015) token embeddings with learned position embeddings, feeding them into a multi-layer transformer decoder architecture with a standard language modelling objective. Fine-tuning was, amongst others, performed on the nine tasks that together form the GLUE benchmark.

BERT (Devlin et al., 2019) BERT can be seen as a reference point for everything that came thereafter. Similar to GPT it uses Byte-Pair Encoding (BPE) with a vocabulary size of 30k. By introducing the MLM objective, the authors were able to combine deep bidirectionality with Self-Attention for the first time. Additionally, BERT also utilizes the next-sentence prediction (NSP) objective, the usefulness of which has been debated in other research papers (Liu et al., 2019). The BERT-BASE model consists of 12 bidirectional transformer-encoder blocks (24 for BERT-LARGE) with 12 (16 respectively) attention heads per block and an embedding size of 768 (1024 respectively).

OpenAI GPT2 (Radford et al., 2019) Compared to its predecessor GPT, it contains some smaller changes concerning the placement of layer normalisation and residual connections. Overall, there are four different versions of GPT2 with the smallest one being equal to GPT, the medium one being of similar size as BERT-LARGE and the xlarge one being released as the actual GPT2 model with 1.5B parameters.

XLNet (Yang et al., 2019) In order to overcome (what they call) the *pretraining-finetune discrepancy*, which is a consequence of BERT's MLM objective, and to simultaneously include bidirectional contexts, Yang et al. (2019) propose the PLM objective . They use two-stream self-attention for preserving the position information of the token to be predicted, which would otherwise be lost due to the permutation. While the *content stream attention* resembles the standard Self-Attention in a transformer-decoder, the *query stream attention* doesn't allow the token to see itself but just the preceding tokens of the permuted sequence.

RoBERTa (Liu et al., 2019) With RoBERTa (**R**obustly **o**ptimized **BERT** **a**pproach), Liu et al. (2019) introduce a replicate of BERT with tuned hyperparameters and a larger corpus used for pre-training. The masking strategy is changed from static (once during pre-processing) to dynamic (every sequence just before feeding it to the model), the additional NSP objective is removed, the BPE vocabulary is increased to 50k and training is performed on larger batches than BERT. These adjustments improve performance of the model and make it competitive to the performance of XLNet.

ALBERT (Lan et al., 2019) By identifying that the increase of the model size is a problem, ALBERT (**A** **L**ite **BERT**) goes into another direction compared to most of post-BERT architectures. Parameter-reduction techniques are applied in order to train a faster model with lower memory demands that, at the same time, yields a comparable

	Compute			Resources	
Model	Hardware	Training time	pfs-days ♡	#parameters	lexical
ULMFiT	NA	NA	NA	33M	0.18GB
GPT	8 GPUs (P600)	∼ 30 days	0.96	117M	< 13GB
BERT-BASE	4 Cloud TPUs	∼ 4 days	0.96 [2.24] ♢	110M	13GB
BERT-LARGE	16 Cloud TPUs	∼ 4 days	3.84 [8.96] ♢	340M	13GB
GPT2-MEDIUM	NA	NA	NA	345M	40GB
GPT2-XLARGE	8 v3 Cloud TPUs	∼ 7 days	7.84	1.500M	40GB
XLNet-LARGE	128 v3 Cloud TPUs	∼ 2.5 days	44.8	340M	126GB
RoBERTa	DGX-1 GPUs (8xV100) ♣	NA ♣	NA	360M	160GB
	1024 32GB V100 GPUs ♠	∼ 1 day ♠	4.78	360M	16GB
ALBERT	64 – 1024 v3 Cloud TPUs	NA	NA	233M	16GB

Table 2: Usage of compute and pre-training resources alongside with model size for the evaluated model architectures. With *lexical resources* we refer to the size of the pre-training corpus. ELMo not included as it is not end-to-end trainable (Size depends on the used model after obtaining the embeddings). The size of ULMFiT is assumed to be the larger value from Merity et al. (2017), since Howard and Ruder (2018) use AWD-LSTMs with a vocabulary size of 30k tokens (Johnson and Zhang, 2016, 2017). Values for GPT2-XLARGE are taken from Strubell et al. (2019).
♡ *Petaflop-days*: Estimation according to the formula proposed on https://openai.com/blog/ai-and-compute/:
pfs-days = number of units × PFLOPS/unit × days trained × utilization, with an assumed utilization of $\frac{1}{3}$. PFLOPS/unit for TPUs from https://cloud.google.com/tpu/.
♢ Unclear, whether v2 or v3 TPUs were used. Thus, we provide calculations for both: v2[v3]
♣ Full RoBERTa model (Liu et al., 2019) ♠ RoBERTa variant utilizing less pre-training resources

performance to SOTA models. We will always refer to the best performing ALBERT-XXLARGE, despite also the smaller ALBERT models yield results comparable to BERT.

4 Model comparison

Tab. 2 gives an overview on the amount of computational power needed to pre-train a given architecture on given pre-training (lexical) resources. In Tab. 3 we will directly try to relate model architecture and size as well as usage of lexical resources to model performance.
One thing we can learn from Tab. 2 is the lack of details when it comes to reporting the computational resources used for pre-training. While Howard and Ruder (2018) do not provide any information on the computational power utilised for pre-training, the other articles report it to different degrees. Unfortunately, there are no clear guidelines on how to appraise this when it comes to evaluating and comparing models. This may be attributed to the rapidly growing availability of hardware, but in our opinion it should nevertheless be accounted for, since it might pose environmental issues (Strubell et al., 2019) and also limits portability to smaller devices.

Further, it is important to consider the differences displayed in the Tab. 2 and Tab. 3 when comparing the model performances. Considering two models of approximately the same size (BERT-BASE vs. GPT), the superior performance of BERT-BASE seems to originate purely from its more elaborated architecture because of the similar size. But one should also be aware of the larger lexical resources (BERT-BASE uses at least twice as much data for pre-training) and the unknown differences in usage of computational power. We approximated the latter as the *pfs-days* (cf. Tab. 2), resulting in an estimation for BERT-BASE being not less than the one for GPT.
Another aspect which should not be ignored when evaluating performance is ensembling. As can be seen in the first column of Tab. 3, the three model ensembles outperform both of the BERT models by a large margin. Only parts of these differences may be attributed to the model architecture or the hyperparameter settings, as the ensembling as well as the larger pre-training resources might give an

	GLUE		SQuAD		RACE	Resources	
Model	leaderboard	dev$^{\heartsuit}$	v1.1 (dev)	v2.0 (dev)	test	#parameters	lexical
BERT-BASE	*78.3*	–	*88.5*	*76.3* ♣	*65.0* ♠	*110M*	*13GB*
ELMo-based	- 8.3	–	- 2.9	–	–	–	–
GPT	- 5.5	–	–	–	- 6.0	1.1x	< 0.5x
BERT-LARGE	+ 2.2	*84.05*	+ 2.4	+ 5.6	+ 7.0 ♠	3.1x	1.0x
XLNet-BASE	–	–	–	+ 5.03	+ 1.05	∼ 1.0x	1.0x
XLNet-LARGE	+ 10.1 $^{\diamondsuit}$	+ 3.39	+ 6.0	+ 12.5	+ 16.75	3.1x	9.7x
RoBERTa	+ 10.2 $^{\diamondsuit}$	+ 5.19	**+ 6.1**	+ 13.1	+ 18.2	3.3x	12.3x
RoBERTa-BASE	–	+ 2.30	–	–	–	1.0x	12.3x
RoBERTa ‡	–	+ 3.79	+ 5.1	+ 11.0	–	3.3x	1.2x †
ALBERT	**+ 11.1** $^{\diamondsuit}$	**+ 5.91**	+ 5.6	**+ 13.9**	**+ 21.5**	2.1x	1.2x †

Table 3: Performance values as well as model size and resource usage (Reference in *italics*, highest improvements in **bold**). Performance differences are given in percentage points (%pts), differences in size/resources as factors. ULMFiT and GPT2 are omitted as there are no performance values on these data sets publicly available. No model size for ELMo provided, since the performance values are from different models (cf. Sec. 3.3).
Displayed performance measures are Matthews Correlation (GLUE), F1 score (SQuAD) and Accuracy (RACE).
$^{\heartsuit}$ Own calculations based on Lan et al. (2019), Tab. 13; WNLI is excluded $^{\diamondsuit}$ Ensemble performance
♣ Values taken from Yang et al. (2019), Tab. 6 ♠ Values taken from Zhang et al. (2019), Tab. 2
† Liu et al. (2019) and Lan et al. (2019) specify the BooksCorpus + English Wikipedia as 16GB
‡ This variant of RoBERTa uses only BooksCorpus + English Wikipedia for pre-training

advantage to these models. As there are no performance values of *single* models available for XLNet, RoBERTa and ALBERT on the official GLUE leaderboard, we also compare the single model performances from Lan et al. (2019) obtained on the dev sets. From this comparison we get an impression of how high the contribution of ensembling might be: The difference between BERT-LARGE and the XLNet ensemble in the official score (7.9 %pts) is more than twice as high as the difference in dev score (3.4 %pts).

In order to address the differences in size of the pre-training resources, Yang et al. (2019) make the extremely insightful effort to compare a XLNet-BASE variant to BERT-BASE using the same pre-training resources. While the F1 score on SQuAD v2.0 is still remarkably higher than for BERT-BASE (comparable to BERT-LARGE) it does not show a large improvement on RACE (which might have been expected due to the large improvement of XLNet-LARGE over both BERT models).

The comparability of RoBERTa from the GLUE leaderboard (ensemble + larger pre-training resources) to BERT-LARGE is limited, but the authors perform several experiments in order to show the usefulness of their optimisations. Pre-training a single model on comparable lexical resources (13GB for BERT vs. 16GB for RoBERTa), the RoBERTa model shows a smaller (compared to the RoBERTa ensemble), but still remarkable, improvement over BERT-LARGE. In another ablation study, Liu et al. (2019) train a RoBERTa-BASE variant on larger pre-training resources. Even though comprising only about one third of the size of BERT-LARGE, the larger pre-training corpus in conjunction with the optimised training leads to a slightly better performance on the GLUE dev set. We are not able to compare RoBERTa-BASE to BERT-BASE, as neither the "official" leaderboard score for RoBERTa-BASE nor the "inofficial" dev set score for BERT-BASE are available.

In order to set the results of ULMFiT into context, we present the results published by Yang et al. (2019) alongside with information on size and pre-training resources in Tab. 4. Despite being much larger and pre-training on some orders of magnitude larger corpora, BERT-LARGE and XLNet-LARGE do not exhibit that large improvements over the performance of ULMFiT. This might partly originate from the relative simplicity of the tasks, but partly also from the already achieved high performances.

	Sentiment			Topic		Resources	
Model	IMDb	Yelp-bi	Yelp-full	AG's news	DBpedia	size	lexical
ULMFiT	*95.40*	*97.84*	*70.02*	*94.99*	*99.20*	*33M*	*0.18GB*
BERT-LARGE	+ 0.09	+ 0.27	+ 0.66	–	+ 0.16	10.3x	72.2x
XLNet-LARGE	**+ 0.81**	**+ 0.61**	**+ 2.28**	**+ 0.52**	**+ 0.18**	10.3x	222.2x

Table 4: Performance comparison (+ model size and resource usage) on the benchmark data sets used by Howard and Ruder (2018). Specification of the differences and highlighting as in Tab. 3. We report accuracies, as opposed to Howard and Ruder (2018); Yang et al. (2019), in order to facilitate a similar interpretation compared to Tab. 3.

5 Discussion

This chapter reflects the main takeaways from the above comparisons and raises some issues for research practices. We do not claim to have a solution to these potentially problematic aspects, but rather think that these points are highly debatable.

Why no benchmark corpus for pre-training? It is good practice to use benchmark data sets for comparing the performance of pre-trained language models on different types of *Natural language understanding* (NLU) tasks. Many recently published articles (Liu et al., 2019; Yang et al., 2019; Lan et al., 2019) perform (partly extensive) ablation studies controlling for pre-training resources in order to make (versions of) their models comparable to BERT, which is really important as it helps to get an intuition for the impact of pre-training resources. Nevertheless, it is unfortunately not perfect due to two critical issues: (i) BERT and all of its successors make use of the BooksCorpus (Zhu et al., 2015) which is not publicly available and (ii) this only leads to model comparisons in a low pre-training resource environment (compared to more recent models) and yields no insight on the behaviour of the reference model (e.g. BERT) in a medium or high resource context. So we view statements of the type *"Model architecture A is superior to model architecture B on performing task X."* somewhat critical and propose to phrase it more like the following statement: *"Model architecture A is superior to model architecture B on performing task X, when pre-trained on a small/medium/large corpus of low/high quality data from domain Y for pre-training time Z."*

Why no standardised description of (computational) resources? When writing this article, it turned out difficult to get one unified measure for the amount of the computational power used for pre-training. In our opinion, this is not a carelessness of the authors but rather the lack of a clear reporting standard. We found ourselves confronted with the following situations:

a) No information at all (Radford et al., 2019)

b) Hardware (Liu et al., 2019; Lan et al., 2019)

c) Hardware and training time (Devlin et al., 2019; Yang et al., 2019)

d) Standardised measure (Radford, 2018)

While *a)* is clearly unsatisfactory and should be avoided, *b)* and *c)* provide most of the necessary information but miss out on going the last final step to *d)*, where the reporting reaches universal comparability across different articles. The measure we computed (cf. Tab. 2) is of course not as exact as a computation based on the counts of operations in a network, but requires no deep insight into the model architecture and is thus applicable to a wide range of architectures without much effort.

Shouldn't performance be evaluated in relation to size and resource usage? As larger models have a higher capacity for learning representations and using larger pre-training resources should improve their quality, varying these two components simultaneously with the model architecture might lead to interference between the individual effects on model performance. This aspect has a slight overlap with the question raised above, but while the above is more or less about introducing some reference, this is about carefully varying and evaluating the effects of different model parts.

6 Conclusion

As can be seen from the above analysis, there is a lack of a concise guideline for *fair* comparisons of

large pre-trained language models. It is not sufficient to just rank models by their performance on the common benchmark data sets as this does not take into account all the other factors mentioned in this analysis. Further aspects worth reporting are the use of resources (time and compute) spent on model development (including all experimental runs and trials) and hyperparameter tuning during pre-training. In our opinion, this is important with respect to two facets: On the one hand side it is important to take into account environmental considerations when training deep learning models (Strubell et al., 2019), on the other hand side it is also a signal to the reader/user how difficult it is to train (and to fine-tune) the model. This might have implications for the usage of a model as transfer learning model for diverse downstream tasks. Models that have already been tuned to a high degree during pre-training to reach a certain level of performance, may have, in the long run, less potential for further improvements compared to models which do so without much hyperparameter tuning. To conclude, we unfortunately cannot say with determination which one of the influential factors (architecture or amount of pre-training resources) is more important, but we think that a substantial amount of the recent improvements can be attributed to larger pre-training resources. A detailed disentanglement of the influence of the different components stays an open research question which might be answerable by carefully designed benchmark studies.

Acknowledgments

We would like to thank the three anonymous reviewers for their insightful comments and their feedback on our work.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Jamie Callan, Mark Hoy, Changkuk Yoo, and Le Zhao. 2009. Clueweb09 data set.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2013. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Aaron Gokaslan and Vanya Cohen. 2019. Openwebtext corpus.

Felix Hamborg, Norman Meuschke, Corinna Breitinger, and Bela Gipp. 2017. News-please: a generic news crawler and extractor. In *15th International Symposium of Information Science (ISI 2017)*, pages 218–223.

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. *arXiv preprint arXiv:1801.06146*.

Rie Johnson and Tong Zhang. 2016. Convolutional neural networks for text categorization: Shallow word-level vs. deep character-level. *arXiv preprint arXiv:1609.00718*.

Rie Johnson and Tong Zhang. 2017. Deep pyramid convolutional neural networks for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 562–570.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut.

2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pages 142–150. Association for Computational Linguistics.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2017. Regularizing and optimizing lstm language models. *arXiv preprint arXiv:1708.02182*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016a. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016b. Wikitext-103. Accessed: 2020-02-10.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016c. Wikitext-2. Accessed: 2020-02-10.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Sebastian Nagel. 2016. Cc-news.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English gigaword fifth edition, june. *Linguistic Data Consortium, LDC2011T07*, 12.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Boris T Polyak and Anatoli B Juditsky. 1992. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855.

Alec Radford. 2018. Improving language understanding with unsupervised learning. Accessed: 2020-02-10.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Sebastian Ruder, Matthew E. Peters, Swabha Swayamdipta, and Thomas Wolf. 2019. Transfer learning in natural language processing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, pages 15–18, Minneapolis, Minnesota. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Iyer Shankar, Dandekar Nikhil, and Csernai Kornél. 2017. First quora dataset release: Question pairs. Accessed: 2020-02-10.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*.

Trieu H Trinh and Quoc V Le. 2018. A simple method for commonsense reasoning. *arXiv preprint arXiv:1806.02847*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Ellen M Voorhees and Dawn M Tice. 1999. The trec-8 question answering track evaluation. In *TREC*, volume 1999, page 82. Citeseer.

Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. 2013. Regularization of neural networks using dropconnect. In *International conference on machine learning*, pages 1058–1066.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint arXiv:1905.00537*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2018. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *arXiv preprint arXiv:1905.12616*.

Shuailiang Zhang, Hai Zhao, Yuwei Wu, Zhuosheng Zhang, Xi Zhou, and Xiang Zhou. 2019. Dual comatching network for multi-choice reading comprehension. *arXiv preprint arXiv:1901.09381*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.