Contents lists available at ScienceDirect

Machine Learning with Applications

journal homepage: www.elsevier.com/locate/mlwa

Pre-trained transformers: an empirical comparison

Silvia Casola [a,b,*], Ivano Lauriola [c,1], Alberto Lavelli [b]

[a] Università di Padova, Human Inspired Technology Research Centre, Via Luzzatti, 4, 35121 Padova, Italy
[b] Fondazione Bruno Kessler, Via Sommarive, 18, Trento, 38123, Italy
[c] Amazon Alexa AI, United States of America

ARTICLE INFO

Keywords:
Natural Language Processing
Deep Neural Networks
Transformer models
Pre-training
Reproducibility
Survey

ABSTRACT

Pre-trained transformers have rapidly become very popular in the Natural Language Processing (NLP) community, surpassing the previous state of the art in a wide variety of tasks. While their effectiveness is indisputable, these methods are expensive to fine-tune on the target domain due to the high number of hyper-parameters; this aspect significantly affects the model selection phase and the reliability of the experimental assessment. This paper serves a double purpose: we first describe five popular transformer models and survey their typical use in previous literature, focusing on reproducibility; then, we perform comparisons in a controlled environment over a wide range of NLP tasks. Our analysis reveals that only a minority of recent NLP papers that use pre-trained transformers reported multiple runs (20%), standard deviation or statistical significance (10%), and other crucial information, seriously hurting replicability and reproducibility. Through a vast empirical comparison on real-world datasets and benchmarks, we also show how the hyper-parameters and the initial seed impact results, and highlight the low models' robustness.

Contents

1. Introduction

Pre-trained transformer models (Vaswani et al., 2017) have recently shown their potential on a plethora of Natural Language Processing (NLP) tasks, including Neural Machine Translation (Imamura & Sumita, 2019), Question Answering (Devlin et al., 2019; Garg et al., 2020), Sequence Classification (Sun et al., 2019), and Sentiment Analysis (Hoang et al., 2019), to name a few.

These models owe their luck to two pillars of modern NLP based on Deep Learning, i.e., pre-training and self-attention. On the one

[*] Corresponding author at: Fondazione Bruno Kessler, Via Sommarive, 18, Trento, 38123, Italy.
E-mail addresses: scasola@fbk.eu (S. Casola), lauivano@amazon.com (I. Lauriola), lavelli@fbk.eu (A. Lavelli).
[1] The work of I.L. was done prior to joining Amazon Alexa AI.

https://doi.org/10.1016/j.mlwa.2022.100334
Received 23 January 2022; Received in revised form 5 May 2022; Accepted 14 May 2022
Available online 3 June 2022

hand, intensive unsupervised pre-training allows the network to produce robust contextualized word and sentence vectors; on the other hand, self-attention mechanisms draw global dependencies between words, keeping track of the whole input sequence. Besides the exceptional empirical performance, pre-trained transformer models can be trained significantly faster than architectures based on recurrent or convolutional layers.

One of the most popular pre-trained transformer models is BERT (Bidirectional Encoder Representations from transformers) (Devlin et al., 2019). Inspired by BERT, several pre-trained transformers have been proposed, as is the case of RoBERTa (Liu et al., 2019), ALBERT (Lan et al., 2020), and DistilBERT (Sanh et al., 2019).

Although these methods' effectiveness is indisputable, there is a severe lack of rigorous comparisons and controlled *sandbox* experiments due to multiple reasons. First, transformer models are often used as part of a more complex system that merges multiple technologies, such as knowledge bases, ontologies, grammatical features, reasoning, and databases. Consequently, it is hard to compare the single pre-trained transformer against other methods. Secondly, NLP is historically characterized by a considerable effort in competitions and shared tasks, where multiple teams are challenged to solve specific problems with unannotated test sets and public leaderboards.[2] Although this has proven crucial for NLP research's progress, the comparison among models is not always reliable (Lin et al., 2021), as even the fine-tuning of the initial random seed matters. Finally, large pre-trained transformer models require considerable effort in fine-tuning and dedicated hardware, i.e., multiple GPUs. Consequently, published results, either on leaderboards or scientific papers, often refer to a single run and do not take variability into account. Moreover, they tend to simplify the selection of hyper-parameters, which often refer to default or standard values. The comparison between different transformers is therefore tricky, and the results are not always reliable. After a manual check, we observed that only about 20% of scientific papers that use pre-trained transformers consider multiple runs in their experiments. These premises raise concerns about reproducibility in NLP, which has already been found problematic (Cohen et al., 2018; Dror et al., 2018). In particular, the lack of detailed controlled experiments might impact experiments' *replicability* and their results' *reproducibility* – as statistical significance, for example, is rarely taken into account.

This work tries to fill this gap, with a double aim: a) we survey the literature to understand how pre-trained transformers are used in practice, focusing on how models are tuned and results on downstream tasks are reported and b) we provide an exhaustive and reliable empirical comparison of various pre-trained transformer models on multiple scenarios and tasks, including binary and multi-class sequence classification, Named Entity Recognition, Question Answering, and the General Language Understanding Evaluation (GLUE) benchmark. Additionally, we analyze models' robustness in terms of standard deviation and sensitivity to hyper-parameters. While some surveys of pre-trained transformers and language models exist (Liu et al., 2020b; Qiu et al., 2020), our focus is specifically on directly comparing popular pre-trained transformers in a controlled environment to emphasize their empirical differences.

Our results suggest that pre-trained transformer models are very sensitive to hyper-parameters, making their selection extremely important.

The paper is organized as follows. Section 2 gives an overview of the transformer model architecture and describes popular pre-trained transformer models. Section 3 exposes a brief literature review of works mentioning such models. Section 4 describes the experimental protocol and reports results from multiple benchmarks and real-world corpora. Results are discussed and interpreted in Section 5, where we draw our conclusions.

2. Transformer models

As opposed to recurrent neural networks, the attention mechanism is at the core of the transformer (Vaswani et al., 2017) model. Intuitively, one can think that different text elements have different importance for a given task. For example, it is reasonable to speculate that "shopping" or "offer" are relevant words when filtering spam emails, much less so when considering sentiment. The attention mechanism learns such relevance and allows the model to focus on specific aspects of the text. In practice, attention computes the relative importance of an input sequence (key) for an output (query); this importance is computed by multiplying a dynamically learned score to each element of the input sequence (value). More formally, attention maps a set of query, key, and value elements to an output, computed as a weighted sum of the values; each weight measures the compatibility (via the dot product) between the query and the corresponding key:

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

where Q, K and V are the query, key and value matrices and d_k is the dimension of a key vector (and its corresponding query). In its multi-head formulation, the input tokens are first projected h times with learned coefficients to obtain h queries, keys and values matrices; attention is then computed in parallel on the Q, K, and V triplet and the resulting outputs are concatenated and projected back, to obtain a single attention matrix. This mechanism allows the model to use multiple representation subspaces and attend to different positions (see Fig. 1).

The transformer architecture comprises an encoder and a decoder. The encoder is a stack of N identical layers, each composed of a multi-head self-attention mechanism and a fully connected feed-forward sub-layer; self-attention (i.e., attention in which $Q = K = V$) is used to compute a representation of the input without recurrency or convolution. The decoder has a similar architecture, with an additional intermediate masked attention sub-layer over the encoder output.

Along with the architectural changes, pre-training has also played an essential role in performance improvement. While pre-training is extremely resource-intensive, the resulting model – which obtains a general language representation – is often made publicly available to the NLP community. The same model can thus be fine-tuned on different downstream tasks, which is cheap in training time and data size, therefore affordable even in low-resource scenarios. Following the pre-training fine-tuning paradigms, transformers are currently the most popular architecture for the vast majority of NLP tasks, that are very diverse in terms of modeling and domains.

In the following, we will describe some pre-trained transformers, highlighting their fundamental differences, as summarized in Table 1.

BERT (Devlin et al., 2019): based on the original transformer architecture, BERT (Bidirectional Encoder Representations from transformers) is pre-trained using two objectives: masked language model (MLM) and next sentence prediction (NSP). MLM consists of randomly masking a percentage of the input tokens and using the left and right context to predicting such tokens. NSP is a binary classification task in which the model has to infer if two sentences are consecutive. BooksCorpus (Zhu et al., 2015) and the English Wikipedia (16 GB in total) were used as training corpora. BERT showed impressive results at publication time, surpassing state of the art in eleven NLP tasks, including GLUE, SQuAD, and SWAG (Zellers et al., 2018).

RoBERTa (Liu et al., 2019): based on BERT, RoBERTa (Robustly-optimized BERT approach) improves the training procedure and data. The NSP task is removed, and a different set of hyper-parameters is used (e.g., larger batches). The model is trained on

[2] A popular shared task is SQuAD: https://rajpurkar.github.io/SQuAD-explorer/[Last accessed: March 2021].

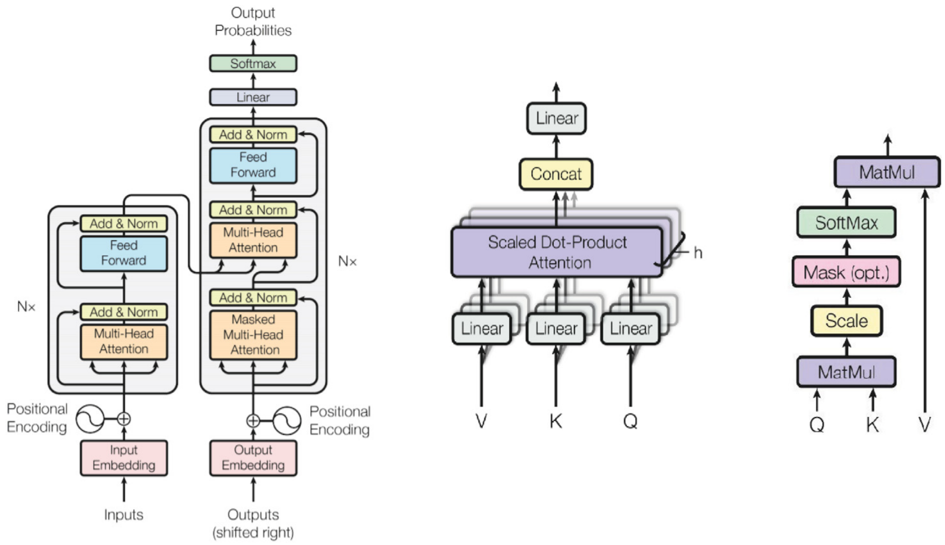

Fig. 1. Transformer architecture (left), Multi-head attention (middle), attention (right).
Source: From Vaswani et al. (2017).

Table 1
Pre-trained transformers.

Model	Architecture	#Parameters	Loss	Corpus
BERT-base	Transformer	110M	MLM, NSP	BookCorpus, Wikipedia
RoBERTa-base	Transformer	125M	MLM	BookCorpus, Wikipedia, CC-news, OPENWEBTEXT, STORIES
DistilBERT	Transformer	66M	MLM, distillation, cosine embedding	BookCorpus, Wikipedia
XLNet-base	Transformer + two-stream attention + target-aware representations	110M	PLM	BookCorpus, Wikipedia, Giga-5, ClueWeb 2012-B, CommonCrowl
ALBERT-base	Transformer + embedding factorization + weight sharing	12M	MLM, SOP	BookCorpus, Wikipedia, additional data

a larger corpus, including BERT's datasets, CC-News,[3] the OPENWEBTEXT (Gokaslan & Cohen, 2019), and STORIES (Trinh & Le, 2018), with 160 GB of uncompressed text in total. RoBERTa showed improved performance over several benchmarks, including GLUE (Wang et al., 2018) and SQuAD (Rajpurkar et al., 2016).

XLNet (Yang et al., 2019): the model introduces Permutated Language Modeling (PLM), merging MLM's advantages and autoregression. MLM predicts each masked token in a sentence independently, causing a discrepancy between train and fine-tuning (as no MASK token – used during training – appears when fine-tuning). In contrast, PLM defines a random permutation of the input tokens and predicts the target one by only using antecedent tokens in the permutation order. Architecturally, several variations to the standard transformer structure are introduced, including two-stream self-attentions and target-aware representations. In addition to the datasets used in BERT, XLNet is trained on Giga5 (Graff et al., 2003), ClueWeb 2012-B[4], and Common Crawl[5] (158 GB in total).

DistilBERT (Sanh et al., 2019): the model is a distilled version of BERT, whose size is decreased by 40% at the expense of slight performance deterioration. Knowledge distillation (Hinton et al., 2015) is a compression technique in which the student model (smaller) behaves similarly to the teacher model (larger) by learning to reproduce its output distribution. DistilBERT's (student) architecture has minor differences to BERT, but a smaller size; a triple loss (distillation loss, MLM loss, cosine embedding loss) is used for training. BERT's training corpus is employed. DistilBERT showed comparable results to BERT (97% of performance retained) while being smaller and faster.

ALBERT (Lan et al., 2020): ALBERT (A Lite BERT) is optimized for reducing memory consumption and improving speed. The model uses factorized embedding parametrization and cross-layer parameter sharing to reduce the number of parameters. In addition to the MLM loss, ALBERT introduces a Sentence Order Prediction (SOP) loss, used to learn if two sentences are consecutive or their order is swapped. Besides BERT's corpus, the model is trained on additional data.

2.1. Other pre-trained transformers

While the models described above are mainly used for discriminative tasks, transformers have also gained huge popularity in text generation. GPT (Generative Pre-Training) (Radford et al., 2018) and its later improvements GPT-2 (Radford et al., 2019) and GPT-3 (Brown et al., 2020), in particular, are relatively standard transformer architectures trained with a language model objective. Their main success

[3] https://commoncrawl.org/2016/10/news-dataset-available/ [Last accessed: March 2021].

[4] https://lemurproject.org/clueweb12/ [Last accessed: March 2021].

[5] https://commoncrawl.org/ [Last accessed: March 2021].

was in Natural Language Generation, but they also achieved impressive results in other tasks. Notably, GPT-3 has shown effective in zero- and few-shot settings, with no fine-tuning. A somewhat similar approach has been followed by T5 (Text-To-Text Transfer Transformer) (Raffel et al., 2020), for which all tasks are framed in a text-to-text generative approach (for classification problems, for example, the model learns to generate the correct labels given the input). A different direction of research focuses on finding effective training objectives. For example, Electra (Efficiently Learning an Encoder that Classifies Token Replacements Accurately) (Clark et al., 2020) learns whether words in a sequence were replaced by a generator (replaced token detection). The main advantage of this approach is that it is defined over all input tokens while still being bidirectional (in contrast, only 15% of tokens are corrupted in BERT, when using MLM); thus, fewer data are necessary for training. Efficiency is the core objective for the Reformer (Kitaev et al., 2020), the Longformer (Beltagy et al., 2020), and Big Bird (Zaheer et al., 2020), which were designed to handle very long sequences (too expensive for standard attention, which scales quadratically in the input length).

Other transformer models include CTRL (Keskar et al., 2019), Bart (Lewis et al., 2020), Pegasus (Zhang et al., 2020), and Funnel (Dai et al., 2020), among others. Interestingly, the research community released several pre-trained Transformer models for languages other than English (see, for example, (Cui et al., 2019) for Chinese or (Le et al., 2020) for French); moreover, an interesting trend is that of multilingual transformers (Liu et al., 2020a).

3. Literature review

Pre-trained transformer models have received an unpredictable interest in the literature. To better understand their impact, we retrieved and analyzed papers with a mention of transformers models from various sources. While we do not aim at performing a systematic literature review, we can probe previous work to understand how (a) pre-trained transformers models are typically used and fine-tuned and (b) results are reported, and whether stochasticity is taken into account.

3.1. Survey method

To quantify transformers' impact in NLP literature, we surveyed three different sources:

ACL Anthology The ACL Anthology is one of the main repositories for NLP, collecting manuscripts and proceedings from all of the major NLP conferences and journals. This repository provides the possibility to perform a search via a GUI and public APIs and meta-data to perform custom searches.

arXiv arXiv is an open-access archive containing over 2 million preprints (not necessarily peer-reviewed). Differently from the ACL Anthology, arXiv is not strictly focused on NLP. Hence, we can better estimate the impact of these methods on different domains. For instance, BioBERT (Lee et al. 2020), a BERT model pre-trained on Biomedical documents, has proven its supremacy on several Biomedical tasks often out of the scope of the ACL Anthology. We observe that arXiv contains a vast amount of authoritative pre-prints mainly due to the frenzy and the need to publish quickly in NLP.

Google Scholar Google Scholar is a web engine that indexes academic literature. Compared to the previous two sources, the documents' nature and scope are even broader (for example, it indexes thesis and technical reports). We include this source to have a broader vision of transformers' impact.

For all sources, we searched for "pre-trained Transformer" in the 2017–2021 time frame.

3.2. Survey results

The ACL Anthology retrieves 14,300 results for our query.[6]

Most of those papers extend pre-trained transformer models in several ways, such as by adding external features or analyzing a specific aspect, e.g., pre-training.

To better analyze the spread of pre-trained transformer models in the literature, we manually checked 60 randomly selected papers from the 2017–2021 time frame.[7] Our goal is to better understand how transformer models are used in the literature and the main problems in the evaluation and comparison process.

We observed that most of the selected papers are task-oriented, i.e., their goal is to leverage the pre-trained transformer to solve a specific task. This is the case of 77% of manuscripts, which use the transformer with no or limited architectural improvements. The remaining 23% perform in-depth analysis applied to various topics, including multilinguality or human–machine comparison. From an empirical point of view, only 20% of these scientific papers declared that they performed multiple runs in their experiments. Moreover, only six articles, i.e., 10% of analyzed papers, report the standard deviation or a statistical significance test. Concerning hyper-parameters optimization, we found that 80% of manuscripts do not show a clear and complete model selection procedure, limiting the reproducibility of the exposed results. Specifically, 25% of manuscripts do not mention the model selection strategy adopted or the network's configuration. Additionally, 38% of articles show the optimal tuned configuration, i.e., the selected value for some hyper-parameters. However, they do not describe how these optimal values are selected, i.e., the search strategy (manual- grid- or random-search) or the search space (which value has been tested for each hyper-parameter). 10% of manuscripts used the validation data to select the number of epochs through an early stopping procedure. Another 10% only tuned a single hyper-parameter of a general system that uses the transformer rather than the transformer itself. In other words, the transformer is fine-tuned in its "default" configuration. Differently, the remaining papers (20%) fine-tuned one or multiple hyper-parameters, and they provide a complete description of the experimental setting, allowing the replication of the results. Finally, we observed that 25% of the analyzed papers use a single dataset in their experiments, further limiting the generalization of results.

As we already discussed, these two aspects, i.e., single-run and missing hyper-parameters, heavily affect the overall results and conclusions in NLP research. Indeed, it is known that these methods can be highly unstable (Zhou et al., 2020). We will better discuss and analyze this aspect of transformer models in the experimental part of this paper.

Fig. 2 shows the number of retrieved results per year using the arXiv and the Google Scholar internal search engine. Notice the strongly increasing trend.

4. Experimental comparison

This section exposes the empirical experiments we carried out to assess multiple transformer-based pre-trained models. Specifically, we evaluated BERT, RoBERTa, DistilBERT, XLNet, and ALBERT with multiple benchmarks and real-world tasks, including sequence classification, word sequence labeling, and question answering.

We selected these five models for two main reasons. Firstly, to the best of our knowledge, these models are well-established among NLP practitioners and researchers. For example, a fine-tuned BERT is usually the go-to baseline in many diverse NLP applications, e.g., in the vast majority of recent shared tasks. The same applies to other models in different contexts (for example, distilled models are used when the resources are limited). Secondly, the five models are structurally diverse,

[6] Last updated: 14 March 2022

[7] We believe that this number is large enough to extract useful insights.

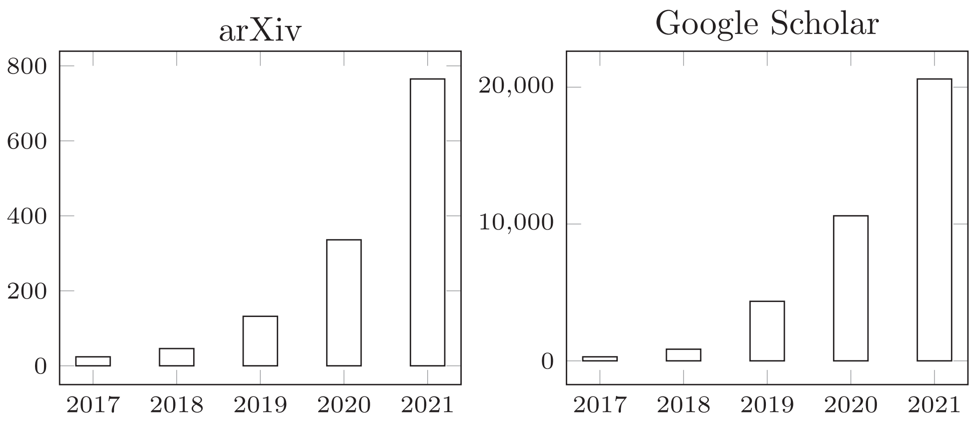

Fig. 2. Number of papers retrieved from arXiv (left) and Google Scholar (right).

allowing us to cover relevant scenarios and use-cases. For instance, RoBERTa and DistilBERT maintain BERT's original architecture, while XLNet and ALBERT try to improve it; DistilBERT and ALBERT are specifically designed to make the model smaller and more efficient, while other models aim at maximizing the performance. We believe that experimenting on a diverse set of models might allow us to investigate which aspect impacts performance in different scenarios.

In the remainder of this section, we describe our empirical setting to evaluate and compare these pre-trained transformer-based models.

4.1. Training procedure and model selection

This sub-section describes the unified setting that we used to train our models, select the hyper-parameters, and evaluate the performance in all of the experiments and tasks we carried out.

Starting from the available pre-trained checkpoints,[8] we fine-tuned the transformer (base, uncased) models on the target tasks using only task-specific training data. A maximum of 40 epochs was run for problems with less than 250,000 datapoints, while at most 15 epochs were allotted for bigger training sets. The learning rate of the optimization, that is probably the most crucial hyper-parameter, has been selected in validation with values in $\{10^i : i = -3 \cdots -6\}$ for datasets with less than 100,000 training examples and in $\{10^{-3}, 10^{-4}\}$ for larger ones (due to our computational capacity); in a few specific cases, as specified in the following, we enlarged the search grid for analytical purposes. We set the batch size to 32 and applied a linear warm-start with 1 epoch as we observed it slightly improved the performance computed on the validation data. The validation sets were used to early stop the training by observing the validation loss (cross-entropy) and to select the hyper-parameters. Eventually, we used the model with the minimum validation loss during inference. We repeated the same procedure 5 times and averaged results on the test sets.

This operational assessment allows us to evaluate (i) the empirical effectiveness, (ii) the robustness and reliability of the models, and (iii) the statistical significance of results[9](Dror et al., 2018).

We divided our experiments into multiple parts. Firstly, we show the comparison on various sequence classification corpora. Then, we analyze those methods when applied to Named Entity Recognition (NER) in the Biomedical domain - a conceptually different task compared to sequence classification. Next, we show a few results on SQuAD, a popular dataset for Question Answering. Finally, we show an in-depth analysis on the popular GLUE benchmarks.

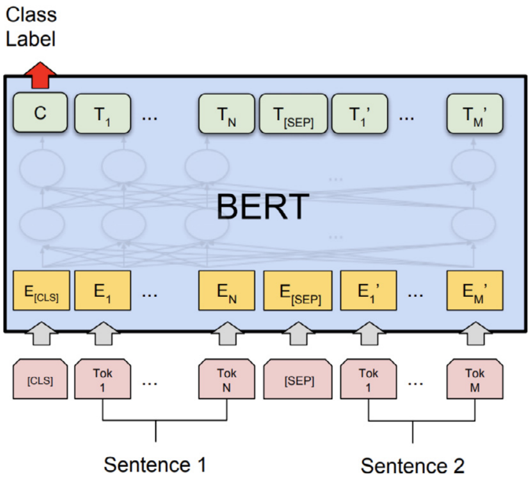

Fig. 3. BERT for sequence classification. The representation developed in the last layer corresponding to the special token *[CLS]* contains a contextualized embedding of the input sequence. In the case of sequence classification, the input consists of a single sentence. However, other tasks, such as QA, need to encode multiple sentences, delimited by a special token.

4.2. Sequence classification

Sequence (or text) classification assigns a class label $y \in \mathcal{Y} = \{c_1, \dots c_k\}$ to each input sequence $\boldsymbol{x} = (x_1, \dots, x_l) \in \mathcal{X}$ from a given distribution. When dealing with transformer-based architectures, the representation corresponding to the initial special token (*[CLS]* for BERT) of the last layer contains the contextualized view of the entire sequence; thus, it is commonly used to predict the class label. This scenario is depicted in Fig. 3.

We used multiple small- and medium-size sequence classification corpora in this analysis, with up to 25,000 training examples. The corpora have different characteristics, and we selected them to cover multiple domains; they are:

Decop (Capuozzo et al., 2020): This corpus contains various personal opinions on hot topics such as immigration, gay marriage, abortion, euthanasia, and cannabis legalization. The binary task consists in identifying deceiving and truthful opinions. The corpus includes 1700, 300, and 500 training, validation, and test opinions.

Clickbait (Chakraborty et al., 2016): This corpus was used to develop a browser extension that warns the readers of different media sites about clickbait headlines. It consists of 16,000 non-clickbait

[8] https://huggingface.co/transformers/index.html [Last accessed: March 2021].

[9] t test on all pairs; $\alpha = 0.05$.

Table 2
Average test accuracy computed on sequence classification tasks. The best results are in bold. The last row shows the average rank to summarize results (lower is better). Not statistically different results are marked with the same symbol.

Dataset	BERT	RoBERTa	DistilBERT	XLNet	ALBERT
Decop	85.60$_{\pm 3.88}$	**88.00**$_{\pm 2.19}$	81.20$_{\pm 7.44}$	86.40$_{\pm 4.08}$	82.40$_{\pm 6.86}$
Clickbait	99.40$_{\pm 0.80}$	**99.60**$^{\star}_{\pm 0.80}$	98.80$_{\pm 0.98}$	**99.60**$^{\star}_{\pm 0.80}$	98.00$_{\pm 2.19}$
Acl-imdb	91.60$_{\pm 1.96}$	92.40$_{\pm 1.50}$	90.80$_{\pm 2.40}$	**96.00**$_{\pm 2.53}$	95.20$_{\pm 2.04}$
Subjectivity	95.60$^{\star}_{\pm 1.50}$	95.60$^{\star}_{\pm 1.96}$	94.80$_{\pm 2.99}$	95.20$_{\pm 2.71}$	**96.80**$_{\pm 2.40}$
Website	**94.00**$^{\star}_{\pm 3.35}$	93.60$^{\circ}_{\pm 2.94}$	92.80$_{\pm 2.71}$	**94.00**$^{\star\circ}_{\pm 3.58}$	91.20$_{\pm 3.92}$
Scicite	79.20$_{\pm 0.98}$	78.00$_{\pm 3.35}$	75.60$_{\pm 4.80}$	77.20$_{\pm 4.66}$	**81.60**$_{\pm 3.20}$
Av. rank	2.50	**2.17**	4.67	**2.17**	3.00

and 16,000 clickbait headlines that we randomly divided into training (25,000), validation (3000), and test (4000).

Acl-imdb (Maas et al., 2011): This corpus collects 25,000 training and 25,000 test IMDB reviews. The positive and negative classes consist of high polarized reviews. A negative review has a score ≤ 4 out of 10, whereas a positive review has a score ≥ 7. Neutral reviews are not included. We randomly split the training set into training (23,000) and validation (2000).

Subjectivity (Pang & Lee, 2004): The task consists in dichotomizing subjective and objective sentences in a movie-related context. 5000 objective sentences were mined from movie plots, whereas subjective sentences were collected from movie reviews. We randomly split data into training (8000), validation (1000), and test (1000).

Website (Kotzias et al., 2015): The corpus includes 3000 reviews from three different sources, i.e., IMDB, Yelp, and Amazon (1500 positives, 1500 negatives). We randomly divided data into training (2000), validation (500), and test (500).

Scicite (Cohan et al., 2019): The corpus consists of sentences from scientific papers containing a citation. The multi-class task consists of identifying the intent of a given citation in a sentence, i.e., background information, use of methods, or comparing results.

We applied the pre-trained transformer models as described in Section 4.1. We used accuracy to evaluate the performance. The results computed on the test sets and averaged over 5 different runs are shown in Table 2. The standard deviation and the average rank are also exposed; the average rank is computed by ranking models performance on the individual datasets (1 to 5, lower is better) and then averaging results on all datasets. We mark non statistically different results with the same symbol (as is the case, for example, of RoBERTa and XLNet for the Clickbait dataset).

Reported results are not surprising. RoBERTa and XLNet, which developed various strategies to improve the accuracy of the simple BERT, are the two methods that achieve, on average, better results (2.17 of average rank). DistilBERT and ALBERT, which focus on improving BERT's efficiency, achieve considerably lower results (4.67 and 3.00 of average rank). However, despite the average ranking datasets, ALBERT achieves top results on two tasks: Subjectivity and Scicite. In the last case, the model improves the absolute accuracy computed by the other methods by 2–3 points. Reasonably, BERT achieves median results (2.50 of average rank). However, the variability of results (i.e., the standard deviation) reported in the table prevents, in some cases, from defining the most performing model and emphasizes several interesting clues.

Let us consider the case of Decop that is, on average, the corpus with the largest standard deviation: the highest accuracy is reached by RoBERTa (88.00), which also corresponds to the lowest standard deviation (2.19), whereas the lowest accuracy is reached by DistilBERT (81.20) with the highest variability (7.44). This result suggests that DistilBERT may suffer from fine-tuning issues, and some runs may fail due to a bad hyper-parametrization or the presence of local minima. Differently from Decop, results computed on Clickbait are quite similar. In the case of Website, the standard deviation prevents electing a winning model, and BERT and RoBERTa perform statistically similarly to XLNet.

Table 3
BNER corpora description. The values in Training, Validation, and Test, refers to the number of sequences belonging to the associated portion of data.

Corpus	Training	Valid.	Test	Entity types
BioNLP11-ID	2496	721	1961	chemical, ggp, species
BioNLP13-PC	2499	857	1695	chemical, ggp, cellular comp.
BioNLP13-CG	3033	1003	1906	chemical, cells, cellular comp., species
BioNLP13-GE	2449	2737	3391	gene/protein
Ex-PTM	1377	437	1839	gene/protein

Table 4
Experimental results on Biomedical NER tasks.

Dataset	BERT	RoBERTa	DistilBERT	XLNet	ALBERT
BioNLP11-ID	81.17$_{\pm 0.40}$	87.13$_{\pm 0.31}$	80.95$_{\pm 1.03}$	**87.46**$_{\pm 0.49}$	77.66$_{\pm 0.93}$
BioNLP13-PC	88.71$_{\pm 1.03}$	**88.91**$_{\pm 0.95}$	88.11$_{\pm 0.75}$	88.31$_{\pm 1.14}$	77.90$_{\pm 3.11}$
BioNLP13-CG	82.95$_{\pm 1.44}$	84.65$_{\pm 0.36}$	82.54$_{\pm 1.02}$	**85.03**$_{\pm 0.81}$	78.59$_{\pm 1.98}$
BioNLP13-GE	**77.10**$^{\star}_{\pm 0.65}$	77.06$^{\star}_{\pm 1.13}$	75.45$_{\pm 1.19}$	74.80$_{\pm 1.69}$	69.60$_{\pm 1.86}$
Ex-PTM	74.61$_{\pm 1.17}$	74.50$_{\pm 1.20}$	70.87$_{\pm 3.04}$	**75.89**$_{\pm 1.17}$	60.48$_{\pm 8.69}$
Av. rank	2.2	**2**	3.8	**2**	5

4.3. Named Entity Recognition

Named Entity Recognition (NER) is a challenging and popular task in the NLP community. NER finds relevant entities in unstructured documents, such as persons' names, locations, and dates. Additionally, NER is a preliminary step for more complex tasks, such as relation extraction, sentiment analysis, dialogue, and knowledge-base maintenance. Unlike sequence classification and question answering, NER is a word sequence labeling task, and each word of the input sequence is assigned to a specific class of entities.

This paper specifically addresses NER in the biomedical domain (BioNER), a task whose aim is to extract biomedical entities, such as proteins, chemical compounds, or organism names, from biomedical documents. We assessed the pre-trained transformer-based architectures on several Biomedical NER corpora (Crichton et al., 2017) publicly available on GitHub.[10] We briefly describe these corpora in Table 3.

Similarly to previous experiments, we selected the learning rate in validation with values $10^{-i}, i = 3 \dots 6$ and averaged the results over 5 different runs. Results of the comparison are exposed in Table 4.

As the table shows, RoBERTa and XLNet perform better than other models, while lighter models perform significantly worse. Finally, note that the gap between RoBERTa and BERT is minimal on most of the BioNER corpora (even not significant in the case of BioNLP13-GE).

4.4. Question Answering

Question Answering (QA) is a popular task in NLP consisting in finding an answer given a question and a source text. QA mainly consists of two distinct tasks: Answer Sentence Selection (Wang et al., 2007) (AS2) and the most popular Machine Reading (Chen et al., 2017) (MR). The former identifies the whole sentence containing the answer to a given question in relevant documents, e.g., retrieved by a search engine. The latter finds an exact text span in a document (or typically a paragraph) containing the answer.

[10] https://github.com/cambridgeltl/MTL-Bioinformatics-2016 [Last accessed March 2021].

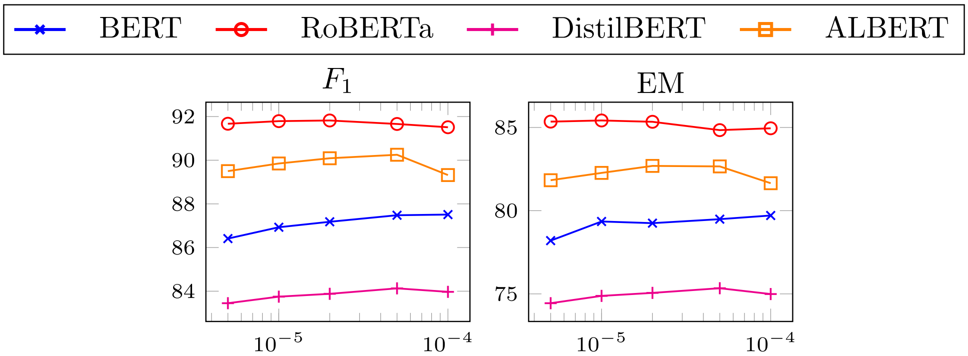

Fig. 4. Empirical performance of transformer-based models on SQuAD-1.1 dev. set while increasing the learning rate. Each configuration has been run 3 times. We omit the standard deviation for the sake of readability.

Table 5
Experimental results on the SQuAD-1.1 development set.

Metric	BERT	RoBERTa	DistilBERT	ALBERT
F_1	$87.51_{\pm 0.16}$	$\mathbf{91.82}_{\pm 0.07}$	$84.13_{\pm 0.37}$	$90.25_{\pm 0.12}$
EM	$79.71_{\pm 0.32}$	$\mathbf{85.42}_{\pm 0.05}$	$75.34_{\pm 0.51}$	$82.69_{\pm 0.08}$
Av. rank	3	**1**	4	2

In these experiments, we focus on MR for multiple reasons. Firstly, MR is attracting much more attention in the scientific literature. Secondly, AS2, that consists in ranking question/sentence pairs to find the most relevant, is relatively similar to sentence classification.

In particular, we perform experiments on SQuAD-1.1. The dataset consists of human-posed questions on Wikipedia articles, and it includes over 100,000 questions. Each question is answerable and a text span representing the answer is always contained in the associated paragraph. We used the code from the Huggingface Transformers 4.2 repository[11] for this task. We did not evaluate XLNet as it is not supported with the currently available script. We selected the learning rate with values in {5e-6, 1e-5, 2e-5, 5e-5 1e-4}. We discarded smaller learning rates used in the previous experiments as we observed that they do not improve the validation performance. We used a constant learning rate without warm-up in the first step of these experiments. For computational reasons, we distributed the computation over 8 GPUs with a batch size of 64 sequences per GPU.

Table 5 contains the Exact Match (EM) and the F_1 scores averaged over 3 different runs.

We observe that RoBERTa is the model achieving the highest F_1 and EM. It is also the most stable model as the standard deviation computed over 3 different runs is minimal compared to other transformers.

Additionally, we evaluated the stability of these transformers when varying hyper-parameters. Fig. 4 shows the empirical performance on the development set when increasing the learning rate. Note that RoBERTa is extremely robust against this hyper-parameter as the difference between the best (85.42 EM and 91.82 F_1) and the worst (84.84 EM and 91.51 F_1) configuration is negligible.

Next, we analyzed the effect of different learning rate schedulers, which are: (i) constant, as is the case of previous experiments, (ii) constant with warm-up, where the learning rate grows during the first epoch, and (iii) linear, that linearly decreases the learning rate each iteration. We evaluated the schedulers with the best learning rate selected in the previous phase to make the computation tractable.

This comparison is reported in Fig. 5.

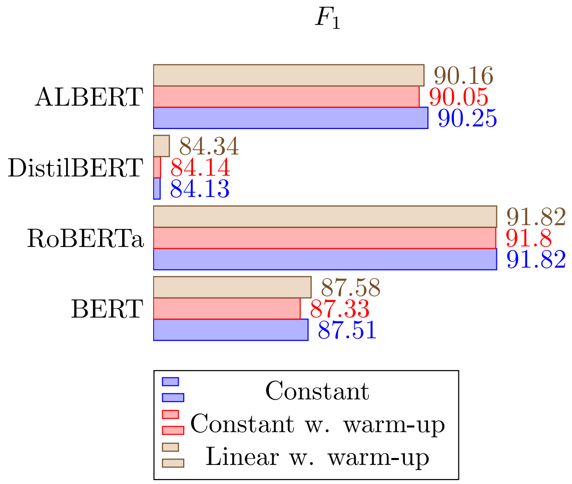

Fig. 5. Effect of different learning rate schedulers on SQuAD (development set).

Interestingly, different schedulers do not produce considerable differences in F_1. This aspect clearly indicates that the absolute value of the learning rate can be much more relevant than the scheduler used to alter it during training. However, we cannot extend this result to other tasks and corpora. Indeed, it is known that the learning rate scheduler may, in some circumstances, affect the final results (Howard & Ruder, 2018).

4.5. GLUE

The General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2018) is a collection of tasks for evaluating models' performance on a set of Natural Language Understanding tasks in English.

The nine tasks are diverse from several points of view. First, they include single-sentence (linguistic acceptability, sentiment analysis) and sentence-pair tasks (similarity and paraphrase, linguistic inference). Secondly, the size of the corresponding datasets varies widely, ranging from a few hundred to a few hundred thousand examples. Third, the content of the datasets is extracted from several domains, including news, social media, books, and Wikipedia. Finally, the evaluation metric varies with the task and the dataset characteristics.

Table 6 shows a synthetic representation of the benchmark features. Given the wide variety of the benchmark, GLUE is regarded as a tool to evaluate models' ability to learn general linguistic knowledge and has become increasingly relevant in the Natural Language Processing

[11] https://github.com/huggingface/transformers/tree/master/examples/question-answering [Last accessed: March 2021].

Table 6
GLUE benchmark features.

Corpus	|Train|	Task	Metric	Domain
CoLA	8.5K	acceptability	Matthews corr	wide
SST-2	67K	sentiment	accuracy	movie reviews
MRPC	3.7K	paraphrase	accuracy/F_1	news
STS-B	7K	sentence similarity	Pearson/Spearman corr	wide
QQP	364K	paraphrase	accuracy/F_1	social questions
MNLI	393K	NLI	matched/mismatched acc	wide
QNLI	105K	QA/NLI	accuracy	Wikipedia
RTE	2.5K	NLI	accuracy	news, Wikipedia
WNLI	634	coreference/NLI	accuracy	fiction book

community for general-purpose models evaluation. All transformers considered in this paper (and most of their variations) report their scores on the GLUE datasets in their original publication.

Table 7 reports the results on the GLUE benchmark (dev sets) obtained under our experimental protocol. Performance-wise, RoBERTa obtains the highest rank and achieves the best results for at least one metric for six datasets out of nine. RoBERTa seems to perform particularly well on small datasets, resulting in the best model for four out of the five datasets with less than 10,000 training examples; for all datasets, the maximum difference with the best model's performance is less than 1.35 points. XLNet results in the best model for three datasets and obtains the second-best rank; however, specific datasets show very poor performance, possibly due to high sensitivity to hyper-parameters. BERT and ALBERT obtain the same average rank; however, ALBERT shows a slightly higher variability (rank std = 1.17) with respect to BERT (std = 1.05). Lastly, DistilBERT achieves the lowest performance, as expected. Fig. 6 shows the accuracy variability with respect to the learning rate. Note that, differently from what we observed on SQuAD, these tasks are very sensitive to the learning rate. We notice that our results are considerably different from those published in the reference transformer's paper in a few cases. We attribute these differences mainly to the choice of hyper-parameters and our base-model, single task, and single dataset setting; finally, we select the final model based on the loss instead of the metric. We also remind that we computed the results on GLUE on the development set.[12] Hence, it is easy to overfit models showing higher results without a predefined protocol.

Finally, we also analyzed the effect of the batch size on the accuracy for the RTE dataset. We selected the best learning rate and repeated experiments five times for each batch for each model. Surprisingly, the accuracy achieved with different batch sizes may vary significantly. Even if we consider RoBERTa, which proved to be the most robust model in the previous experiments, there is an average improvement of 3 points in accuracy with optimal batch size. Unfortunately, we do not observe a clear and interpretable trend (see Fig. 7).

5. Discussion and conclusion

Transformer models have rapidly become the new standard in most NLP tasks. Their architectures (with or without adjustments) are used as building blocks of more complex systems for research purposes and as the to-go model for state-of-the-art performance in applications. Starting from BERT, pre-trained and then fine-tuned models have been used as a baseline in various tasks, a baseline that is often tough to beat.

This paper presents a survey of pre-trained language transformer models, describing their key similarities and differences. In particular, we analyzed BERT and four of its variations: RoBERTa, DistillBERT, XLNet, and ALBERT.

This paper served a double purpose: surveying the existing literature and sharing practices from one side, and comparing the models in a controlled environment from the other.

Looking at the recent NLP literature, we found that transformer models are highly popular, with most papers being task-oriented. However, our analysis suggested that a comparison between models is often problematic, as, for example, many papers only present single runs and do not take stochasticity into account. Moving from these considerations, we performed a set of controlled experiments investigating pre-trained transformer models' performance on real-world datasets and benchmarks. The obtained results align with the previous literature in terms of absolute performance. In particular, we found RoBERTa to perform better, on average, on text classification, sequence tagging, question answering, and on the GLUE benchmark; RoBERTa also seems to work particularly well with small datasets. XLNet performs similarly on text classification and Named Entity Recognition, even though its performance on GLUE is lower than that of RoBERTa; in particular, XLNet seems to perform particularly poorly on some individual datasets (e.g., CoLA). BERT shows median performance in most tasks. Among the smallest models, ALBERT has the best performance.

While the mean performance justifies the model's popularity, our analysis also highlighted considerable instability. In particular, we analyzed the performance variations with respect to the learning rate, the batch size, and the learning rate scheduler. The learning rate and the batch size strongly influence performance; occasionally, substantial variability is shown even for close learning rates, and performance often does not vary consistently with the learning rate increase or decrease. For example, the ALBERT model achieves its best performance (69.32) with a learning rate of 5e-6; this value drops by more than four points using a learning rate of 1e-5 (65.27) to increase again with a learning rate of 5e-5 (68.74).

The models' lack of robustness to hyper-parameters is problematic for several reasons: first, it highlights the importance of a comprehensive parameter search, which is not always feasible, especially in scenarios with limited computational resources. Given the recent NLP focus on state-of-the-art performance, parameter tuning plays an even more critical role since the possibility of performing an extensive parameter search is related to better absolute results. Moreover, this attention to tuning should be paid even when comparing results to previously published models.

Our experiments also showed that, when considering multiple runs, the standard deviation is often high; this means that even the seed plays a non-negligible role. An example is the performance of RoBERTa, which obtains a minimum Matthews correlation coefficient of 49.86 and a maximum of 63.14 (with an increase of more than 13 points) in 5 separate runs of the CoLA datasets with fixed hyper-parameters; we obtain similar results for other models and datasets. This variability underlines how important performing several runs is, and that statistical significance should always be tested when comparing models.

Overall, our work highlights the need for a shared, reliable framework for model comparison in the NLP community, which should define and encourage good practices in terms of reproducibility and replicability of results, taking into account the intrinsic stochastic nature of modern NLP models.

CRediT authorship contribution statement

Silvia Casola: Experiments, Models description and comparison. **Ivano Lauriola:** Experiments, Previous work review. **Alberto Lavelli:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

[12] Indeed, the test set is not available as it is currently used for shared tasks and to compute the rank on a public leaderboard.

Table 7
Experimental results on the GLUE benchmark. When multiple metrics are considered, we use their mean to compute the rank.

Dataset	BERT	RoBERTa	Distil-BERT	XLNet	ALBERT
CoLA (MCC)	55.66$_{\pm 2.76}$	**57.16**$_{\pm 5.24}$	47.94$_{\pm 2.22}$	35.33$_{\pm 2.94}$	53.71$_{\pm 1.29}$
SST-2 (acc)	90.94$_{\pm 0.31}$	93.44$_{\pm 0.19}$	89.86$_{\pm 0.48}$	**93.67**$_{\pm 0.88}$	91.58$_{\pm 0.54}$
MRPC (acc/F_1)	85.05$_{\pm 2.01}$	**87.55**$_{\pm 1.12}$	84.51$_{\pm 2.00}$	87.11$_{\pm 1.09}$	87.25$_{\pm 0.52}$
	89.35$_{\pm 1.66}$	90.28$_{\pm 1.60}$	88.67$_{\pm 1.68}$	90.47$_{\pm 0.87}$	**90.84**$_{\pm 0.38}$
STS-B (r_p/ r_s)	89.15$_{\pm 0.34}$	90.54$_{\pm 0.33}$	87.21$_{\pm 0.22}$	88.33$_{\pm 0.31}$	**91.01**$_{\pm 0.16}$
	88.82$_{\pm 0.35}$	90.35$_{\pm 0.34}$	87.00$_{\pm 0.23}$	88.09$_{\pm 0.31}$	**90.70**$_{\pm 0.17}$
QQP (acc/F_1)	90.28$_{\pm 0.17}$	89.33$_{\pm 0.35}$	89.35$_{\pm 0.27}$	**90.53**$_{\pm 0.28}$	89.67$_{\pm 0.18}$
	87.07$_{\pm 0.21}$	86.03$_{\pm 0.50}$	85.78$_{\pm 0.32}$	**87.37**$_{\pm 0.57}$	86.15$_{\pm 0.31}$
MNLI (m/mm acc)	83.77$_{\pm 0.33}$	86.28$_{\pm 0.15}$	81.26$_{\pm 0.35}$	**86.41**$_{\pm 0.54}$	84.31$_{\pm 0.34}$
	84.23$_{\pm 0.22}$	**86.48**$_{\pm 0.36}$	81.66$_{\pm 0.13}$	85.99$_{\pm 0.53}$	84.76$_{\pm 0.27}$
QNLI (acc)	90.17$_{\pm 0.74}$	**91.72**$_{\pm 0.19}$	87.44$_{\pm 0.61}$	90.83$_{\pm 0.47}$	60.38$_{\pm 6.08}$
RTE (acc)	66.35$_{\pm 4.50}$	**70.11**$_{\pm 2.69}$	58.56$_{\pm 2.10}$	67.00$_{\pm 2.26}$	67.73$_{\pm 1.41}$
WNLI (acc)	**56.34**$_{\pm 00.00}$	**56.34**$_{\pm 0.0}$	55.77$^{\star}_{\pm 1.26}$	55.49$^{\star\circ\bullet}_{\pm 1.89}$	55.49$^{\circ\bullet}_{\pm 1.89}$
Av. rank	2.89	**1.78**	4.50	2.72	2.89

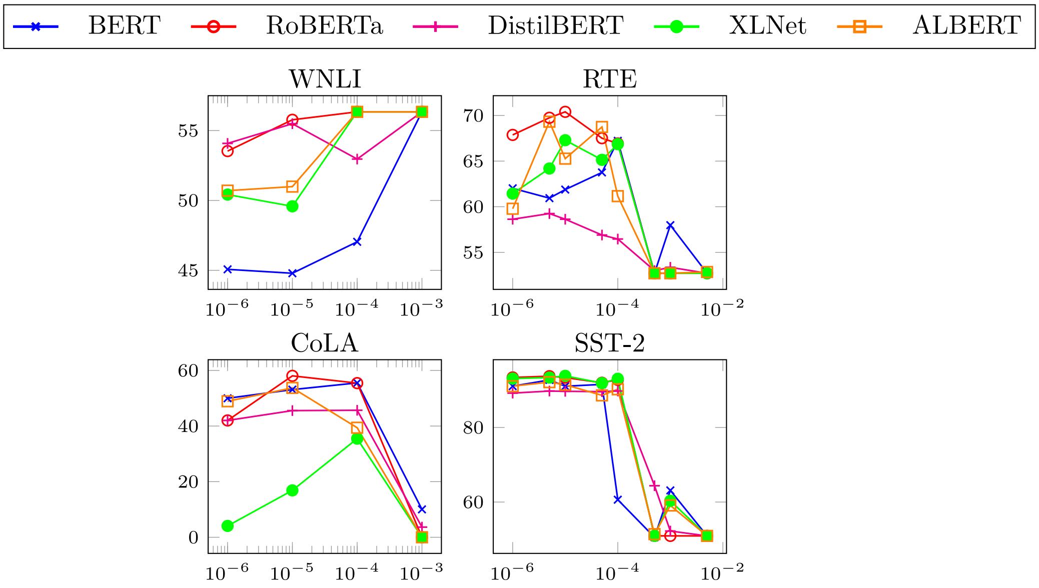

Fig. 6. Accuracy variability with respect to the learning rate. For RTE and SST-2 we enlarged the search grid to {5e-6, 1e-6, 5e-5, …, 1e-3, 5e-3}.

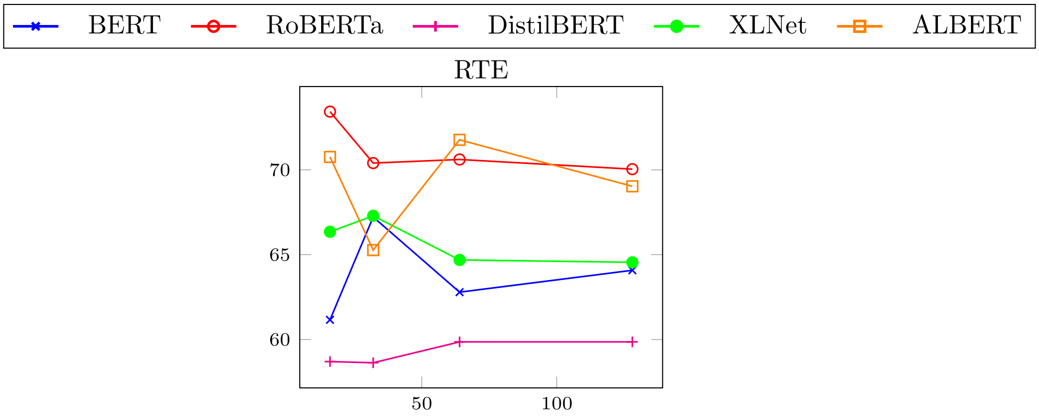

Fig. 7. Effect of different batch sizes on RTE (development set). For each model, the best learning rate was selected (1e-5 for all models except BERT, for which 1e-4 was selected). The plot reports the mean of 5 runs.

References

Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The long-document transformer. *CoRR*, arXiv:2004.05150, URL https://arxiv.org/abs/2004.05150.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., ... Amodei, D. (2020). Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, december 6-12, 2020, virtual*. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Capuozzo, P., Lauriola, I., Strapparava, C., Aiolli, F., & Sartori, G. (2020). Decop: A multilingual and multi-domain corpus for detecting deception in typed text. In *Proceedings of the 12th Language Resources and Evaluation Conference* (pp. 1423–1430).

Chakraborty, A., Paranjape, B., Kakarla, S., & Ganguly, N. (2016). Stop clickbait: Detecting and preventing clickbaits in online news media. In *Advances in Social Networks Analysis and Mining (ASONAM), 2016 IEEE/ACM international conference on* (pp. 9–16). IEEE.

Chen, D., Fisch, A., Weston, J., & Bordes, A. (2017). Reading Wikipedia to answer open-domain questions. In R. Barzilay, & M. Kan (Eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers* (pp. 1870–1879). Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/P17-1171.

Clark, K., Luong, M., Le, Q. V., & Manning, C. D. (2020). ELECTRA: Pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, URL https://openreview.net/forum?id=r1xMH1BtvB.

Cohan, A., Ammar, W., van Zuylen, M., & Cady, F. (2019). Structural scaffolds for citation intent classification in scientific publications. 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)* (pp. 3586–3596). Minneapolis, Minnesota: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/N19-1361, URL https://www.aclweb.org/anthology/N19-1361.

Cohen, K. B., Xia, J., Zweigenbaum, P., Callahan, T., Hargraves, O., Goss, F., Ide, N., Névéol, A., Grouin, C., & Hunter, L. E. (2018). Three dimensions of reproducibility in natural language processing. In N. C. C. chair), K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, & T. Tokunaga (Eds.), *Proceedings of the eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA).

Crichton, G., Pyysalo, S., Chiu, B., & Korhonen, A. (2017). A neural network multi-task learning approach to biomedical named entity recognition. *BMC Bioinformatics*, *18*(1), 368.

Cui, Y., Che, W., Liu, T., Qin, B., Yang, Z., Wang, S., & Hu, G. (2019). Pre-training with whole word masking for Chinese BERT. *CoRR*, arXiv:1906.08101, arXiv:1906.08101, URL http://arxiv.org/abs/1906.08101.

Dai, Z., Lai, G., Yang, Y., & Le, Q. (2020). Funnel-transformer: Filtering out sequential redundancy for efficient language processing. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, Virtual*. URL https://proceedings.neurips.cc/paper/2020/hash/2cd2915e69546904e4e5d4a2ac9e1652-Abstract.html.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (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)* (pp. 4171–4186). Minneapolis, Minnesota: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/N19-1423, URL https://www.aclweb.org/anthology/N19-1423.

Dror, R., Baumer, G., Shlomov, S., & Reichart, R. (2018). The Hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long papers)* (pp. 1383–1392). Melbourne, Australia: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/P18-1128, URL https://www.aclweb.org/anthology/P18-1128.

Garg, S., Vu, T., & Moschitti, A. (2020). TANDA: Transfer and adapt pre-trained transformer models for answer sentence selection. In *Proceedings of AAAI*.

Gokaslan, A., & Cohen, V. (2019). OpenWebText corpus. http://Skylion007.github.io/OpenWebTextCorpus.

Graff, D., Kong, J., Chen, K., & Maeda, K. (2003). English gigaword. *Linguistic Data Consortium, Philadelphia*, *4*(1), 34.

Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*. URL http://arxiv.org/abs/1503.02531.

Hoang, M., Bihorac, O. A., & Rouces, J. (2019). Aspect-based sentiment analysis using BERT. In *Proceedings of NoDaLiDa*.

Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long papers)* (pp. 328–339). Melbourne, Australia: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/P18-1031, URL https://www.aclweb.org/anthology/P18-1031.

Imamura, K., & Sumita, E. (2019). Recycling a pre-trained BERT encoder for neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation* (pp. 23–31).

Keskar, N. S., McCann, B., Varshney, L., Xiong, C., & Socher, R. (2019). CTRL - A conditional transformer language model for controllable generation. arXiv preprint arXiv:1909.05858.

Kitaev, N., Kaiser, L., & Levskaya, A. (2020). Reformer: The efficient transformer. In *8th International Conference on Learning Rpresentations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, URL https://openreview.net/forum?id=rkgNKkHtvB.

Kotzias, D., Denil, M., De Freitas, N., & Smyth, P. (2015). From group to individual labels using deep features. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 597–606).

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, URL https://openreview.net/forum?id=H1eA7AEtvS.

Le, H., Vial, L., Frej, J., Segonne, V., Coavoux, M., Lecouteux, B., Allauzen, A., Crabbé, B., Besacier, L., & Schwab, D. (2020). FlauBERT: Unsupervised language model pre-training for French. In *LREC*.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In D. Jurafsky, J. Chai, N. Schluter, & J. R. Tetreault (Eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020* (pp. 7871–7880). Association for Computational Linguistics.

Lin, J., Campos, D., Craswell, N., Mitra, B., & Yilmaz, E. (2021). Significant improvements over the state of the art? A case study of the MS MARCO document ranking leaderboard. arXiv preprint arXiv:2102.12887.

Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., Lewis, M., & Zettlemoyer, L. (2020). Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, *8*, 726–742, URL https://transacl.org/ojs/index.php/tacl/article/view/2107.

Liu, Q., Kusner, M. J., & Blunsom, P. (2020). A survey on contextual embeddings. arXiv preprint arXiv:2003.07278.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human language technologies* (pp. 142–150). Portland, Oregon, USA: Association for Computational Linguistics, URL http://www.aclweb.org/anthology/P11-1015.

Pang, B., & Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics* (pp. 271–278). Barcelona, Spain: http://dx.doi.org/10.3115/1218955.1218990, URL https://www.aclweb.org/anthology/P04-1035.

Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., & Huang, X. (2020). Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 1872–1897.

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.

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

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, *21*(140), 1–67, URL http://jmlr.org/papers/v21/20-074.html.

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, http://dx.doi.org/10.18653/v1/d16-1264.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, A distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019). How to fine-tune BERT for text classification? In *China national conference on Chinese computational linguistics* (pp. 194–206). Springer.

Trinh, T. H., & Le, Q. V. (2018). A simple method for commonsense reasoning. arXiv preprint arXiv:1806.02847.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems, Vol. 30* (pp. 5998–6008). Curran Associates, Inc., URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. (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* (pp. 353–355).

Wang, M., Smith, N. A., & Mitamura, T. (2007). What is the jeopardy model? A quasi-synchronous grammar for QA. In *EMNLP-CONLL* (pp. 22–32). Prague, Czech Republic: Association for Computational Linguistics, URL https://www.aclweb.org/anthology/D07-1003.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). XLNet: Generalized autoregressive pretraining for language understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems, Vol. 32* (pp. 5753–5763). Curran Associates, Inc..

Zaheer, M., Guruganesh, G., Dubey, K. A., Ainslie, J., Alberti, C., nón, S. O., Pham, P., Ravula, A., Wang, Q., Yang, L., & Ahmed, A. (2020). Big Bird: Transformers for longer sequences. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, Virtual*. URL https://proceedings.neurips.cc/paper/2020/hash/c8512d142a2d849725f31a9a7a361ab9-Abstract.html.

Zellers, R., Bisk, Y., Schwartz, R., & Choi, Y. (2018). SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *EMNLP*.

Zhang, J., Zhao, Y., Saleh, M., & Liu, P. J. (2020). PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of Machine Learning Research*: *vol. 119*, *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event* (pp. 11328–11339). PMLR, URL http://proceedings.mlr.press/v119/zhang20ae.html.

Zhou, X., Nie, Y., Tan, H., & Bansal, M. (2020). The curse of performance instability in analysis datasets: Consequences, source, and suggestions. In B. Webber, T. Cohn, Y. He, & Y. Liu (Eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020* (pp. 8215–8228). Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2020.emnlp-main.659.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE international conference on computer vision*.