Transformers: “The End of History” for NLP?

Anton Chernyavskiy[1], **Dmitry Ilvovsky**[1] and **Preslav Nakov**[2]
[1]National Research University Higher School of Economics, Moscow, Russia
[2]Qatar Computing Research Institute, HBKU, Doha, Qatar
aschernyavskiy_1@edu.hse.ru, dilvovsky@hse.ru
pnakov@hbku.edu.qa

Abstract

Recent advances in neural architectures, such as the Transformer, coupled with the emergence of large-scale pre-trained models such as BERT, have revolutionized the field of Natural Language Processing (NLP), pushing the state-of-the-art for a number of NLP tasks. A rich family of variations of these models has been proposed, such as RoBERTa, ALBERT, and XLNet, but fundamentally, they all remain limited in their ability to model certain kinds of information, and they cannot cope with certain information sources, which was easy for pre-existing models. Thus, here we aim to shed some light on some important theoretical limitations of pre-trained BERT-style models that are inherent in the general Transformer architecture. First, we demonstrate in practice on two general types of tasks—segmentation and segment labeling—and four datasets that these limitations are indeed harmful and that addressing them, even in some very simple and naïve ways, can yield sizable improvements over vanilla RoBERTa and XLNet. Then, we offer a more general discussion on desiderata for future additions to the Transformer architecture that would increase its expressiveness, which we hope could help in the design of the next generation of deep NLP architectures.

1 Introduction

The history of Natural Language Processing (NLP) has seen several stages: first, rule-based, e.g., think of the expert systems of the 80s, then came the statistical revolution, and now along came the neural revolution. The latter was enabled by a combination of deep neural architectures, specialized hardware, and the existence of large volumes of data. Yet, the revolution was going slower in NLP compared to other fields such as Computer Vision, which were quickly and deeply transformed by the emergence of large pre-trained models, which were enabled by the emergence of large-scale datasets such as ImageNet.

Things changed in 2018, when NLP finally got its “ImageNet moment” with the invention of BERT (Devlin et al., 2019).[1] This was enabled by recent advances in neural architectures, such as the Transformer (Vaswani et al., 2017), followed by the emergence of large-scale pre-trained models such as BERT, eventually revolutionizing NLP and pushing the state-of-the-art for a number of NLP tasks. A rich family of variations of these models have been proposed, such as RoBERTa (Liu et al., 2019b), ALBERT (Lan et al., 2019), and XLNet (Yang et al., 2019). For some researchers, it felt like this might very well be the “End of History” for NLP (à la Fukuyama[2]).

It was not too long before researchers started realizing that BERT and Transformer architectures in general, despite their phenomenal success, remain fundamentally limited in their ability to model certain kinds of information, which was natural and simple for the old-fashioned feature-based models. Although BERT encodes some syntax, semantic, and linguistic features, it may not use them in downstream tasks (Kovaleva et al., 2019). It ignores negation (Ettinger, 2020), and it might need to be combined with Conditional Random Fields (CRF) to improve its performance for some tasks and languages, most notably for sequence labeling (Souza et al., 2019). There is a range of sequence tagging tasks where entities have different lengths (not 1-3 words as in the classical NER formulation), and sometimes their continuity is required, e.g., for tagging in court papers. Moreover, in some problem formulations, it is important to accurately process the boundaries of the spans (in particular,

[1]A notable previous promising attempt was ELMo (Peters et al., 2018), but it became largely outdated in less than a year.

[2]http://en.wikipedia.org/wiki/The_End_of_History_and_the_Last_Man

the punctuation symbols), which turns out to be something that Transformers are not particularly good at (as we will discuss below).

In many sequence classification tasks, some classes are described by specific features. Besides, a very large contextual window may be required for the correct classification, which is a problem for Transformer models because of the quadratic complexity of calculating attention weights.[3]

Is it possible to guarantee that BERT-style models will carefully analyze all these cases? This is what we aim to explore below. Our contributions can be summarized as follows:

- We explore some theoretical limitations of pre-trained BERT-style models when applied to sequence segmentation and labeling tasks. These limitations are not limitations of a specific model, but are due to the general Transformer architecture.
- We demonstrate in practice on two different tasks (one on segmentation, and one on segment labeling) and four datasets that it is possible to improve over state-of-the-art models such as BERT, RoBERTa, XLNet, and this can be achieved with simple and naïve approaches, such as feature engineering and post-processing.
- We propose desiderata for attributes to add to the Transformer architecture in order to increase its expressiveness, which could guide the design of the next generation of deep NLP architectures.

The rest of our paper is structured as follows. Section 2 summarizes the related prior research. Section 3 describes the tasks we address. Section 4 presents the models and the modifications thereof. Section 5 outlines the experimental setup. Section 6 describes the experiments and the evaluation results. Section 7 provides key points that lead to further general potential improvements of Transformers. Section 8 concludes and points to possible directions for future work.

2 Related Work

Studies of what BERT learns and what it can represent There is a large number of papers that study what kinds of information can be learned with BERT-style models and how attention layers capture this information; a survey is presented in (Rogers et al., 2020). It was shown that BERT learns syntactic features (Goldberg, 2019; Liu et al., 2019a), semantic roles and entities types (Tenney et al., 2019), linguistic information and subject-verb agreement (Jawahar et al., 2019). Note that the papers that explore what BERT-style models can encode do not indicate that they directly use such knowledge (Kovaleva et al., 2019). Instead, we focus on what is *not* modeled and explore some general limitations.

Limitations of BERT/Transformer Indeed, Kovaleva et al. (2019) revealed that vertical self-attention patterns generally come from pretraining tasks rather than from task-specific linguistic reasoning and model is overparametrized. Ettinger (2020) demonstrated that BERT encodes some semantics, but fully insensitive to negation. Sun et al. (2020) showed that BERT-style models are erroneous in simple cases, e.g. they do not correctly process word sequences with misspellings. They have bad representations of numbers with the floating point for the same tokenization reason (Wallace et al., 2019). Moreover, it is easy to attack them with adversarial examples (Jin et al., 2019). Durrani et al. (2019) showed that BERT subtoken-based representations are better for modeling syntax, while ELMo character-based representations are preferable for modeling morphology. It also should be noticed that hyperparameters tuning is very non-trivial task not only for NLP engineers but also for advanced NLP researchers (Popel and Bojar, 2018). Most of these limitations are low-level and technical, or specific architecture (BERT) is considered. In contrast, we single out the general limitations of Transformer at a higher level, but which can be technically confirmed, and provide desiderata for their elimination.

Fixes of BERT/Transformer A lot of improvements of the original BERT model have been proposed: RoBERTa (changed language model masking, learning rate, dataset size), DistilBERT (Sanh et al., 2019) (distillation to significantly reduce the number of parameters), ALBERT (cross-layer parameter sharing, factorized embedding parameterization), Transformer-XL (Dai et al., 2019) (recurrence mechanism and relative positional encoding

[3] Some solutions were proposed such as Longformer (Beltagy et al., 2020), Performer (Choromanski et al., 2020), Linformer (Wang et al., 2020), Linear Transformer (Katharopoulos et al., 2020), and Big Bird (Zaheer et al., 2020).

to improve sequence modeling), XLNet (permutation language modeling to better modeling bidirectional relations), BERT-CRF (Arkhipov et al., 2019; Souza et al., 2019) (dependencies between the posteriors for structure prediction helped in some tasks and languages), KnowBERT (Peters et al., 2019) (incorporates external knowledge). Most of these models pay attention only to the one-two concrete fixes, whereas our paper aims at more general Transformer limitations.

3 Tasks

3.1 Propaganda Detection

We choose the task of Detection of Propaganda Techniques in News Articles (SemEval 2020 Task 11)[4] as the main for experiments. Generally, it is formulated as finding and classifying all propagandistic fragments in the text (Da San Martino et al., 2020). To do this, two subtasks are proposed: (*i*) *span identification (SI)*, i.e. selection of all propaganda spans within the article, (*i*) *technique classification (TC)*, i.e. multi-label classification of each span into 14 classes. The corpus with a detailed description of propaganda techniques is presented in (Da San Martino et al., 2019).

The motivation for choosing this task is triggered by several factors. Firstly, two technically different problems are considered, which can be formulated at a general level (multi-label sequence classification and binary token labeling). Secondly, this task has specificity necessary for our research, unlike standard named entity recognition. So, traditional NLP methods can be applied over the set of hand-crafted features: sentiment, readability scores, length, etc. Here length is the strong feature due to the data statistics (Da San Martino et al., 2019). Moreover, spans can be nested in each other, while span borders vary widely and may include punctuation symbols. In addition, sometimes Transformed-based models face the problem of limited input sequence length. In this task, such problem appears with the classification of "Repetition" spans. By definition, this class includes spans that have an intentional repetition of the same information. This information can be repeated both in one sentence and in very distant parts of the text.

3.2 Keyphrases Extraction

In order to demonstrate the transferability of the studied limitations between datasets, we further experimented with the task of Extracting Keyphrases and Relations from Scientific Publications, using the dataset from SemEval-2017 Task 10 (Augenstein et al., 2017). We focus on the following two subtasks: (*i*) *keyphrases identification (KI)*, i.e., search of all keyphrases within the text, (*ii*) *keyphrases classification (KC)*, i.e., multi-class classification of given keyphrases into three classes. According to the data statistics, the length of the phrases is a strong feature. Also, phrases can be nested inside one other, and many of them are repeated across different articles. So, these subtasks allow us to demonstrate a number of issues.

[4]The official task webpage: `https://propaganda.qcri.org/semeval2020-task11/`

4 Method

Preliminarily, we selected the most successful approach as the baseline from Transformer-based models (BERT, RoBERTa, ALBERT and XLNet). In both propaganda detection subtasks, it turned out to be RoBERTa, which is an optimized version of the standard BERT with a modified pre-training procedure. Whereas in both keyphrases extraction subtasks, it turned out to be XLNet, which is a language model that aims to better study bidirectional links or relationships in a sequence of words. From a theoretical point of view, investigated results and researched problems should be typical for other Transformer-based models, such as BERT and DistilBERT. Nonetheless, we additionally conduct experiments with both XLNet and RoBERTa in both tasks for a better demonstration of the universality of our findings.

4.1 Token Classification

We reformulate the SI and the KI tasks as a kind of a "named entity recognition" task. Specifically, in the SI task, for each span, all of its internal tokens are assigned to the "`PROP`" class and the rest to "`O`" (**O**utside). Thus, this is a binary token classification task. At the same time, various types of encoding formats are studied. Except for the described Inside-Outside classification, we consider BIO (**B**egin) and BIEOS (**B**egin, **E**nd, **S**ingle are added) tags encodings. Such markups theoretically can provide better processing for border tokens (Ratinov and Roth, 2009).

In order to ensure the sustainability of the trained models, we create an ensemble of three models trained with the same hyperparameters but using different random seeds. We merge the intersecting spans during the ensemble procedure (intervals union).

End-to-end Training with CRFs Conditional Random Fields (CRF) (Lafferty et al., 2001) can qualitatively track the dependencies between the tags in the markup. Therefore, this approach has gained great popularity in solving the problem of extracting named entities with LSTMs or RNNs. Advanced Transformer-based models generally can model relationships between words at a good level due to the attention mechanism, but adding a CRF layer theoretically is not unnecessary. The idea is that we need to model the relationships not only between tokens but also labels.

Our preliminary research showed that both RoBERTa and XLNet are capable of making classification errors even when choosing tags in NER encodings with clear rules. For example, in the case of BIO, the "`I-PROP`" tag can only go after the "`B-PROP`" tag. However, RoBERTa produced results with a sequence of tags "`O-PROP I-PROP O-PROP`" for some inputs. Here it is hard to determine where the error was, but the CRF handles such cases from a probabilistic point of view.

We use the CRF layer instead of the standard model classification head to apply the end-to-end training. Here we model connections only between neighboring subtokens since our main goal is the proper sequence analysis. So, the subtokens that are not placed at the beginning of words are ignored (i.e of the format *##smth*).

RoBERTa, XLNet, and Punctuation Symbols In the SI task, there is one more problem which even the CRF layer cannot always handle. It is the processing of punctuation and quotation marks at span borders. Clark et al. (2019) and Kovaleva et al. (2019) showed that BERT generally has high token-token attentions to the `[SEP]`, periods and commas, as they are most frequent tokens. But we find out that only large attention weights to punctuation may still not be enough in some tasks.

A simple rule can be formulated to solve the problem: a span cannot begin/end with a punctuation symbol unless it is enclosed in quotation marks. Thus, we apply post-processing adding possible quotation marks at the borders and filtering punctuation symbols in case they are absent.

4.2 Sequence Classification

We model the TC task in the same way as the KC task, that is, as a multi-class sequence classification problem. We create a fairly strong baseline approach to achieve better results. First, the context is used, since spans in both tasks can have various meanings in different contexts. So, we select the entire sentence which contains the span for this purpose. In this case, we consider two possible options: (*i*) highlight the span with the special limiting tokens and submit to model only one input; (*ii*) make two inputs: one for the span and the other for the context. Moreover, to provide a better initialization of the model and share some additional knowledge from other data in the TC task, we apply the transfer learning strategy from the SI task.

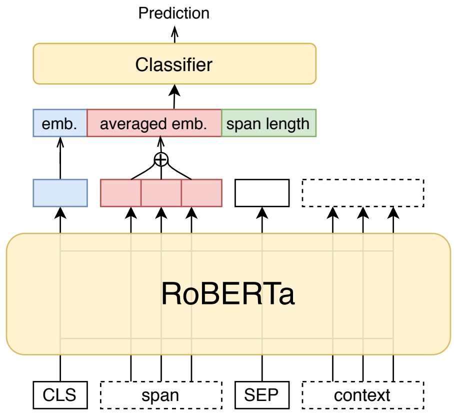

Figure 1: The RoBERTa model takes an input span and the context (sentence with the span). It combines the embedding of the `[CLS]` token, averaged embedding of all span tokens and span length feature.

Just like in the token classification problem, we compose an ensemble of the same model, but with three different random seed initializations. This is done in order to stabilize the model, and it is not a typical ensemble of different models.

Input Length BERT does not have a mechanism to perform explicit character/word/subword counting. Exactly this problem and the lack of good consistency between predicted tags may cause a problem with punctuation (quotation marks) in the sequence tagging task, since BERT theoretically cannot accurately account for the number of opening/closing quotation marks (as it cannot count).

To explicitly take into account the input sequence size in the model, we add a length feature to the `[CLS]` token embedding, as it should contain all the necessary information for the task solution (Figure 1). It may be also useful to pre-process length feature through binning. In this case, it is possible to additionally create trainable embeddings associated with each bin or directly add an external knowledge from the "gazetteer" containing relevant information about the dataset according to the given bin (we will consider gazetteers below).

In addition to the input length in characters (or in tokens), it may be useful to add other quantitative features: for instance, the number of question or exclamation symbols.

Span Embeddings In the TC task, we concatenate the `[CLS]` token representation with the span embedding obtained by averaging over all tokens embeddings from the last layer to submit to the classifier (end-to-end training). Note that the added embedding contains information about the "degree of propaganda" in the classified span as the initialization, since we transfer a model from another task. Moreover, this model can reconfigure it to serve other features during the training process. Also, it may be useful to join embeddings obtained by the max-pool operation or taken from other layers.

Train hand-crafted "gazetteer" Theoretically, gazetteers provide external relevant information about entities in the NER tasks. We exploit a similar idea, but use the training data itself to construct our "gazetteer". We create the hash map, where the keys are spans pre-processed by the Porter Stemmer (Porter, 1980), and the values are distributions of the classes in which spans are present in the training dataset.

There are several ways to apply this "gazetteer". Firstly, we can use these frequency representations as additional features and concatenate them with the `[CLS]` token in the same way as it was described for the length and span embedding. Nevertheless, in this case, over-fitting may occur, since such a "feature" will contain a correct label. The second method is based on post-processing. The proposed idea is to increase the probability of each class of span by some value (+0.5) if the span of this class is present in the "gazetteer". It is a general problem of not 100% quality on the training data. However, it is important to solve it in some tasks, like ours, where each propaganda technique is often described by the same words.

Classes Insertions Earlier we described the problem of non-perfect spatially consistent class predictions in the token labeling task. For the sequence classification task, it may be expressed as the incorrect "nesting" of classes. That is, may appear situations that span of class `A` is nested in some span of class `B`, but there are no such cases in the training data. If we believe that the training sample gives us an almost complete description of the researched problem, such a classification obviously cannot be correct.

The simplest solution is again post-processing. One variant is to choose a pair of spans that have maximal predicted probability and the correct "nesting". The other one is to choose a pair of classes with maximal probability $p(x)p(y)p(A)$. Here $p(x)$ is the predicted probability that span has the label x and $p(A)$ is the estimated probability of the nesting case A, where a span of class x is inside the span of class y. To estimate $p(A)$, we calculate the co-occurrences matrix of nesting classes in the training set and we apply softmax with temperature t over this matrix to get probabilities. The temperature parameter is adjusted for each model on validation. We use the first approach in the TC task. As there are only three classes and all classes insertions are possible, we apply the second approach with $t = 0.26$ in the KC task.

Specific classes – "Repetition" It is clear, the whole text can be a context, rather than a particular single sentence to correctly classify such span in the TC task. Theoretically, the source model uses this class as a padding one, because it often does not see this whole context.

As a solution, we apply special post-processing step. Let k be the number of occurrences of the considered span in the set of spans allocated for prediction within the article and p – the probability of the "Repetition" class predicted by the source model. So, we apply the following formula:

$$\hat{p} = \begin{cases} 1, & \text{if } k \geq 3 \text{ or } (k = 2 \text{ and } p \geq t_1) \\ 0, & \text{if } k = 1 \text{ and } p \leq t_2 \\ p, & \text{otherwise} \end{cases} \quad (1)$$

Here we have threshold values for probability $t_1 = 0.001$ and $t_2 = 0.99$. Since the repetition may be contained in the span itself, it is incorrect, to nullify the probabilities of the unique spans.

Multi-label If the same span can have different labels, it is necessary to apply supplementary post-processing of the predictions. So, if the same span is asked several times during the testing process (the span is determined by its coordinates in the text), then we give different labels, i.e. the top of most likely predictions.

5 Experimental Setup

5.1 Data

Propaganda Detection The dataset provided for the competition SemEval2020-Task11 contains 371 training, 75 development and 90 testing English articles. In total train and test sets have 6129 annotated spans. Since there is original partitioning on train, dev and test subsamples, but the only one-time evaluation was available for the test, we additionally randomly split train in 80 : 20 ratio to obtain new training and validation subsamples.

The core evaluation metric for span identification is the adopted F_1-score (is described in (Da San Martino et al., 2019)). Generally, it correlates with the standard F_1-score for tokens. Micro-averaged F_1 is used in the technique classification task, which is equivalent to the accuracy score.

Keyphrases Extraction The dataset provided for SemEval-2017 task 10 contains 350 training, 50 development, and 100 testing English documents. In total, the training and the testing sets have 9,945 annotated keyphrases. The evaluation measure for both subtasks is micro-averaged F_1-score.

5.2 Parameters setting

We use HuggingFace transformers library to implement and train proposed approaches[5]. RoBERTa$_{\text{LARGE}}$ and XLNet$_{\text{LARGE}}$ models are used, as they were more powerful than their BASE versions in our preliminary task research.

We select hyper-parameters according to the recommendations in the original papers using our validation subsample and make about 10–20 runs to find out the best configuration. Grid-search over {5e-6, 1e-5, 2e-5, 3e-5, 5e-5} is used to find the optimal learning rate. So, we fix the following in the propaganda detection problem: learning rate of 2e-5 (3e-5 for XLNet in the TC task), batch size of 24, the maximum sequence length of 128 (128 is fixed as it is long enough to encode the span; besides, there are very few long sentences in our datasets), Adam optimizer with a linear warm-up of 500 steps in. The sequence length and the batch size are selected as the maximum possible for the utilized GPU machine (3 GeForce GTX 1080 GPUs). We perform training for 30 epochs with savings every 2 epochs and select the best checkpoints on the validation subsample (typically, it was 10–20 epochs). We find that the models are case-sensitive, so the uncased model should be used to solve the SI task, whereas the cased model is better for the TC task.

We use the following settings in the keyphrases extraction task: learning rate of 2e-5 (3e-5 for RoBERTa-CRF), batch size of 12, the maximum sequence length of 64, Adam optimizer with a linear warm-up of 60 steps in to solve the KI task; learning rate of 2e-5 (1e-5 for XLNet-Length) and head learning rate of 1e-4 (in cases with the *Length* feature), batch size of 20 (10 for XLNet-Length), the maximum sequence length of 128, Adam optimizer with a linear warm-up of 200 steps in to solve the KC task. We perform training for 10 epochs, saving each epoch and selecting the best one on the validation set.

The training stage in a distributed setting takes approximately 2.38 minutes per epoch (+0.05 for the "avg. embedding" modification) for the TC task; 6.55 minutes per epoch for RoBERTa (+1.27 for CRF) and 6.75 minutes per epoch for XLNet (+1.08 for CRF) for the SI task.

6 Experiments and Results

6.1 Token Classification

Although the BIEOS encoding is the most complete according to the set of described rules against IO and BIO, the most successful approach during the validation was BIO. At the same time, it was optimal in both cases with using CRF and without it. So, we fixed the BIO tags encoding for comparison purposes. We observed a much better recall score just with the small losses in the precision metric for our ensembling strategy with spans merging.

The comparison of the described approaches for SI and KI tasks is presented in Table 1. Although the predictions of source models are smooth enough, adding a CRF layer on top allows achieving higher scores. Manual analysis of the output in the SI task has revealed that about 3.5% of the predicted tags were illegal sequence continuations, e.g., an "`I-PROP`" tag following an "`O`" tag.

Also, we figured out that neither XLNet nor RoBERTa could learn the described rule for quotes and punctuation symbols. Moreover, adding CRF also does not help solve the problem according to the better "overall" score in the table. We analyzed the source of these errors. Indeed, there were some annotation errors. However, the vast majority of the errors related to punctuation at the bound-

[5]Pretrained models checkpoints and baselines implementation are taken from https://github.com/huggingface/transformers

Task	Approach	F1
SI	RoBERTa (BIO encoding)	46.91
	+ CRF	48.54 $_{\uparrow 1.63}$
	+ punctuation post-processing	47.54 $_{\uparrow 0.63}$
	Overall	48.87 $_{\uparrow 1.96}$
	XLNet (BIO encoding)	46.47
	+ CRF	46.68 $_{\uparrow 0.21}$
	+ punctuation post-processing	46.76 $_{\uparrow 0.29}$
	Overall	47.05 $_{\uparrow 0.58}$
KI	RoBERTa (BIO encoding)	57.85
	+ CRF	58.59 $_{\uparrow 0.74}$
	XLNet (BIO encoding)	58.80
	+ CRF	60.11 $_{\uparrow 1.31}$

Table 1: Analysis of RoBERTa and XLNet upgrades for the token classification (span identification and keyphrases identification) tasks solution. *Overall* means the simultaneous application of two upgrades.

Technique Classification		
Approach	**RoBERTa**	**XLNet**
Baseline	62.75	58.23
+ length	63.50 $_{\uparrow 0.75}$	59.64 $_{\uparrow 1.41}$
+ averaged span embed.	62.94 $_{\uparrow 0.19}$	59.64 $_{\uparrow 1.41}$
+ multi-label	63.78 $_{\uparrow 1.03}$	59.27 $_{\uparrow 1.04}$
+ "gazetteer" p.p.	62.84 $_{\uparrow 0.10}$	58.33 $_{\uparrow 0.10}$
+ "Repetition" p.p.	66.79 $_{\uparrow 4.04}$	62.46 $_{\uparrow 3.67}$
+ classes insertions	62.65 $_{\downarrow 0.10}$	57.85 $_{\downarrow 0.38}$

Table 2: Analysis of RoBERTa and XLNet upgrades for the TC task solution on the development sample. The micro-F_1 score is presented.[6]

aries were actually model errors. E.g., in an example like <`"It is what it is."`>, where the entire text (including the quotation marks) had to be detected, the model would propose sequences like <`"It is what it is`> or <`It is what it is.`>.

Thus, there is a common problem for all Transformer-based models—lack of strong spatial consistency in tag predictions.

6.2 Sequence Classification

We took models with separated inputs (span and context) for all experiments, as such baseline model demonstrated the better quality on the validation sample. The results for the customized models are presented in Tables 2 and 3 for TC and KC tasks, respectively. We also studied effects for the natural multi-label formulation of the TC task (see Table 4). All directions of quality changes were the same.

Although positional embeddings are used in BERT-like models, our experiments showed that they are not enough to fully model the length of the input (span). So, scores of the systems that explicitly use length increased both for RoBERTa and XLNet in both tasks.

[6]p.p. here and in other tables denotes post-processing

Keyphrases Classification		
Approach	**RoBERTa**	**XLNet**
Baseline	77.18	78.50
+ length	77.38 $_{\uparrow 0.20}$	78.65 $_{\uparrow 0.15}$
+ "gazetteer" p.p.	77.43 $_{\uparrow 0.25}$	78.69 $_{\uparrow 0.19}$
+ classes insertions	77.82 $_{\uparrow 0.64}$	78.69 $_{\uparrow 0.19}$

Table 3: Analysis of RoBERTa and XLNet upgrades for the KC task on the development sample in the multi-label mode. The micro-F_1 score is presented.

Technique Classification		
Approach	**RoBERTa**	**XLNet**
Baseline+multi-label	63.78	59.27
+ length	64.72 $_{\uparrow 0.94}$	60.68 $_{\uparrow 1.41}$
+ averaged span embed.	64.25 $_{\uparrow 0.47}$	60.77 $_{\uparrow 1.50}$
+ "gazetteer" p.p.	63.87 $_{\uparrow 0.09}$	59.36 $_{\uparrow 0.09}$
+ "Repetition" p.p.	67.54 $_{\uparrow 3.76}$	63.50 $_{\uparrow 4.23}$
+ classes insertions	63.69 $_{\downarrow 0.09}$	58.89 $_{\downarrow 0.38}$

Table 4: Analysis of RoBERTa and XLNet upgrades for the TC task solution on the development sample in the multi-label mode. The micro-F_1 score is presented.

Technique Classification	
Approach	**F1-score**
RoBERTa	62.08
+ length and averaged span embed.	62.27 $_{\uparrow 0.19}$
+ multi-label correction	63.50 $_{\uparrow 1.23}$
+ classes insertions	63.69 $_{\uparrow 0.19}$
+ "Repetition" p.p.	66.89 $_{\uparrow 3.20}$
+ "gazetteer" p.p.	67.07 $_{\uparrow 0.18}$

Table 5: An incremental analysis of the proposed approach for the TC task on the dev. sample.

According to the source implementation of RoBERTa, XLNet, and other similar models, only `[CLS]` token embedding is used for the sequence classification. However, in the TC task, it turned out that the remaining tokens can also be useful, as in the averaging approach.

Post-processing with "gazetteer" stably improved the score of both models. Note that this can add errors since it does not take into account the context of the spans. For the same reason, we did not set 100% probabilities for corrected classes.

Correcting the nesting of spans by classes can also have a positive impact (see Table 3). However, correct nesting does not guarantee correct final markup, since we only post-process predictions. Better results can be achieved if the model tries to learn it at the training stage.

The presented tables show that the highest quality increase in the TC task was achieved by correcting the “Repetition” class. The reason is that this class is very frequent, but for its classification often a large span context is needed.

We also examined the effect of each modification for the RoBERTa model in the TC task applying an incremental analysis on the development sample (Table 5). It confirms that all researched upgrades are not mutually exclusive.

Note that while better pre-training could make some of the discussed problems less severe, it is still true that certain limitations are more “theoretical” and they would not be resolved by simple pre-training. For example, there is nothing in the Transformer model that would allow it to model the segment length, etc.

7 Discussion

Below we propose some desiderata to add to the Transformer in order to increase its expressiveness, which could guide the design of the next generation of general Transformer architectures.

Length We have seen that length is important for the sequence labeling task. However, it would be important for a number of other NLP tasks, e.g., in seq2seq models. For example, in Neural Machine Translation, if we have an input sentence of length 20, it might be bad to generate a translation of length 2 or of length 200 words. Similarly, in abstractive neural text summarization, we might want to be able to inform the model about the expected target length of the summary: should it be 10 length long? 100-word long?

External Knowledge Gazetteers are an important source of external knowledge, and it is important to have a mechanism to incorporate such knowledge. A promising idea in this direction is KnowBERT (Peters et al., 2019), which injects Wikipedia knowledge when pre-training BERT.

Global Consistency For structure prediction tasks, such as sequence segmentation and labeling, e.g., named entity recognition, shallow parsing, and relation extraction, it is important to model the dependency between the output labels. This can be done by adding a CRF on top of BERT, but it would be nice to have this as part of the general model. More generally, for many text generation tasks, it is essential to model global consistency of the output text, e.g., avoid constant repetitions. This important in machine translation, summarization, chat bots, dialog systems, etc.

Symbolic vs. Distributed Representation Transformers are inherently based on distributed representations for words and tokens. This can have limitations, e.g., we have seen that BERT cannot pay attention to specific symbols in the input such as specific punctuation symbols like quotation marks. Having a hybrid symbolic-distributed representation might help resolve such issues. It might also make it easier to model external knowledge such as in gazetteers.

8 Conclusion and Future Work

We have shed some light on some important theoretical limitations of pre-trained BERT-style models that are inherent in the general Transformer architecture. In particular, we demonstrated in practice on two different tasks—one on segmentation, and one on segment labeling—and four datasets that these limitations are indeed harmful and that addressing them, even in some very simple and naïve ways, can yield sizable improvements over vanilla BERT and RoBERTa models. Then, we offered a more general discussion on desiderata for future additions to the Transformer architecture in order to increase its expressiveness, which we hope could help in the design of the next generation of deep NLP architectures.

In future work, we plan to analyze more BERT-style architectures, especially such requiring text generation, as here we did not touch the generation component of the Transformer. We further want to experiment with a preformulation of the task as span enumeration instead of sequence labeling with BIO tags. Moreover, we plan to explore a wider range of NLP problems, again with a focus on such involving text generation, e.g., Machine Translation, Text Summarization, and Dialog Systems. Here can also be identified similar limitations, such as problems with the output length, lack of global consistency (generation of contradictory text after some time) or external knowledge (for example, a model may not know facts from a gazetteer, as “London is the capital of Great Britain”), etc.

Ethics and Broader Impact

We would like to point out that work on computational propaganda detection could potentially be misused by malicious actors, e.g., to restrict freedom of speech. However, we should note, that the original intent is to promote media literacy by making people aware of how they are being manipulated using various propaganda techniques. We believe that the benefit of this kind of research outweighs the potential downsides. Note also that the dataset we use studies the use of propaganda techniques in news articles, i.e., we are not profiling users, and we do not work with social media at all.

We would also like to warn that the use of large-scale Transformers requires a lot of computations and the use of GPUs/TPUs for training, which contributes to global warming. This is a bit less of an issue in our case, as we do not train such models from scratch; rather, we fine-tune them on relatively small datasets. Moreover, running on a CPU for inference, once the model is fine-tuned, is perfectly feasible, and CPUs contribute much less to global warming.

References

Mikhail Arkhipov, Maria Trofimova, Yuri Kuratov, and Alexey Sorokin. 2019. Tuning multilingual transformers for language-specific named entity recognition. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, pages 89–93, Florence, Italy. Association for Computational Linguistics.

Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017. SemEval 2017 task 10: ScienceIE - extracting keyphrases and relations from scientific publications. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 546–555, Vancouver, Canada. Association for Computational Linguistics.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *ArXiv*.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamás Sarlós, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. 2020. Rethinking attention with performers. *CoRR*, abs/2009.14794.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of bert's attention. *CoRR*.

Giovanni Da San Martino, Alberto Barrón-Cedeño, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020. SemEval-2020 task 11: Detection of propaganda techniques in news articles. In *Proceedings of the 14th International Workshop on Semantic Evaluation*, SemEval 2020, Barcelona, Spain.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019. Fine-grained analysis of propaganda in news article. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5636–5646, Hong Kong, China. Association for Computational Linguistics.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

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.

Nadir Durrani, Fahim Dalvi, Hassan Sajjad, Yonatan Belinkov, and Preslav Nakov. 2019. One size does not fit all: Comparing NMT representations of different granularities. 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 1504–1516, Minneapolis, Minnesota. Association for Computational Linguistics.

Allyson Ettinger. 2020. What bert is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Yoav Goldberg. 2019. Assessing bert's syntactic abilities.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2019. Is bert really robust? a strong baseline for natural language attack on text classification and entailment.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. 2020. Transformers are rnns: Fast autoregressive transformers with linear attention. *CoRR*, abs/2006.16236.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4365–4374, Hong Kong, China. Association for Computational Linguistics.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

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.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. Linguistic knowledge and transferability of contextual representations. *Proceedings of the 2019 Conference of the North*.

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

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Martin Popel and Ondřej Bojar. 2018. Training tips for the transformer model. *The Prague Bulletin of Mathematical Linguistics*, 110(1):43–70.

Martin F Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.

Lev-Arie Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *CoNLL*.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how bert works.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. 2019. Portuguese named entity recognition using bert-crf.

Lichao Sun, Kazuma Hashimoto, Wenpeng Yin, Akari Asai, Jia Li, Philip Yu, and Caiming Xiong. 2020. Adv-bert: Bert is not robust on misspellings! generating nature adversarial samples on bert.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? probing for sentence structure in contextualized word representations.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*.

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do nlp models know numbers? probing numeracy in embeddings. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-attention with linear complexity. *CoRR*, abs/2006.04768.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32*, pages 5753–5763. Curran Associates, Inc.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. *CoRR*, abs/2007.14062.