两个贡献：
1. 提出了预测span text的概念，就是连续的mask,这个百度早就提过了
2. 提出了利用span边界token来预测span中每个token的概念，感觉这想法还是可以的，这里可以看出，其实各个被mask的token在预测的时候还是独立的，貌似有篇论文指出了这个问题，它强调了被mask的token之间也该参与彼此的预测
3.指出了NSP任务worse的原因first: 句长受限，second：NSP任务的不相关，给MLM任务增加了噪音
有关NSP任务：针对sing-sentence任务来说，确实没有train这个的必要，因为你学的是模式，而不是强调sentence order正确效果多好多好，因此你在单句子任务中加入NSP,就类似引入了噪音

SpanBERT: Improving Pre-training by Representing and Predicting Spans

Mandar Joshi[*†] **Danqi Chen**[*‡§] **Yinhan Liu**[§]
Daniel S. Weld[†ϵ] **Luke Zettlemoyer**[†§] **Omer Levy**[§]

[†] Allen School of Computer Science & Engineering, University of Washington, Seattle, WA
{mandar90,weld,lsz}@cs.washington.edu

[‡] Computer Science Department, Princeton University, Princeton, NJ
danqic@cs.princeton.edu

[ϵ] Allen Institute of Artificial Intelligence, Seattle
{danw}@allenai.org

[§] Facebook AI Research, Seattle
{danqi,yinhanliu,lsz,omerlevy}@fb.com

Abstract

We present SpanBERT, a pre-training method that is designed to better represent and predict spans of text. Our approach extends BERT by (1) masking contiguous random spans, rather than random tokens, and (2) training the span boundary representations to predict the entire content of the masked span, without relying on the individual token representations within it. SpanBERT consistently outperforms BERT and our better-tuned baselines, with substantial gains on span selection tasks such as question answering and coreference resolution. In particular, with the same training data and model size as $\text{BERT}_{\text{large}}$, our single model obtains 94.6% and 88.7% F1 on SQuAD 1.1 and 2.0 respectively. We also achieve a new state of the art on the OntoNotes coreference resolution task (79.6% F1), strong performance on the TACRED relation extraction benchmark, and even gains on GLUE.[1]

1 Introduction

Pre-training methods like BERT (Devlin et al., 2019) have shown strong performance gains using self-supervised training that masks individual words or subword units. However, many NLP tasks involve reasoning about relationships between two or more spans of text. For example, in extractive question answering (Rajpurkar et al., 2016), determining that the "Denver Broncos" is a type of "NFL team" is critical for answering the question "Which NFL team won Super Bowl 50?" Such spans provide a more challenging target for self supervision tasks, for example predicting "Denver Broncos" is much harder than predicting only "Denver" when you know the next word is "Broncos". In this paper, we introduce a span-level pretraining approach that consistently outperforms BERT, with the largest gains on span selection tasks such as question answering and coreference resolution.

We present SpanBERT, a pre-training method that is designed to better represent and predict spans of text. Our method differs from BERT in both the masking scheme and the training objectives. First, we mask random contiguous spans, rather than random individual tokens. Second, we introduce a novel *span-boundary objective* (SBO) so the model learns to predict the entire masked span from the observed tokens at its boundary. Span-based masking forces the model to predict entire spans solely using the context in which they appear. Furthermore, the span-boundary objective encourages the model to store this span-level information at the boundary tokens, which can be easily accessed during the fine-tuning stage. Figure 1 illustrates our approach.

To implement SpanBERT, we build on a well-tuned replica of BERT, which itself substantially outperforms the original BERT. While building on our baseline, we find that pre-training on single segments, instead of two half-length segments with the next sentence prediction (NSP) objective, considerably improves performance on most downstream tasks. Therefore, we add our modifications on top of the tuned single-sequence BERT baseline.

Together, our pre-training process yields models that outperform all BERT baselines on a wide

[*] Equal contribution.

[1] Our code and pre-trained models are available at https://github.com/facebookresearch/SpanBERT.

$$\mathcal{L}(\text{football}) = \mathcal{L}_{\text{MLM}}(\text{football}) + \mathcal{L}_{\text{SBO}}(\text{football})$$
$$= -\log P(\text{football} \mid \mathbf{x}_7) - \log P(\text{football} \mid \mathbf{x}_4, \mathbf{x}_9, \mathbf{p}_3)$$

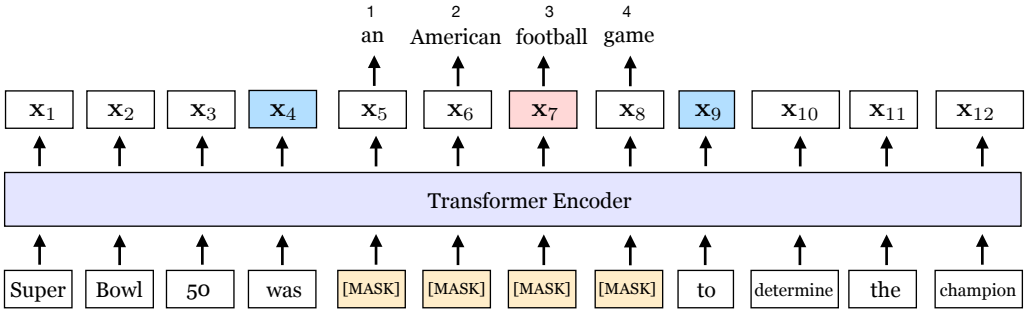

Figure 1: An illustration of SpanBERT training. The span *an American football game* is masked. The span boundary objective (SBO) uses the output representations of the boundary tokens, $\mathbf{x}_4$ and $\mathbf{x}_9$ (in blue), to predict each token in the masked span. The equation shows the MLM and SBO loss terms for predicting the token, *football* (in pink), which as marked by the position embedding $\mathbf{p}_3$, is the *third* token from x_4.

p3是相对x4的第三个位置

variety of tasks, and reach substantially better performance on span selection tasks in particular. Specifically, our method reaches 94.6% and 88.7% F1 on SQuAD 1.1 and 2.0 (Rajpurkar et al., 2016, 2018), respectively — reducing error by as much as 27% compared to our tuned BERT replica. We also observe similar gains on five additional extractive question answering benchmarks (NewsQA, TriviaQA, SearchQA, HotpotQA, and Natural Questions).[2]

SpanBERT also arrives at a new state of the art on the challenging CoNLL-2012 ("OntoNotes") shared task for document-level coreference resolution, where we reach 79.6% F1, exceeding the previous top model by 6.6% absolute. Finally, we demonstrate that SpanBERT also helps on tasks that do not explicitly involve span selection, and show that our approach even improves performance on TACRED (Zhang et al., 2017) and GLUE (Wang et al., 2019).

While others show the benefits of adding more data (Yang et al., 2019) and increasing model size (Lample and Conneau, 2019), this work demonstrates the importance of designing good pre-training tasks and objectives, which can also have a remarkable impact.

2 Background: BERT

BERT (Devlin et al., 2019) is a self-supervised approach for pre-training a deep transformer encoder (Vaswani et al., 2017), before fine-tuning it for a particular downstream task. BERT optimizes two training objectives – masked language model (MLM) and next sentence prediction (NSP) – which only require a large collection of unlabeled text.

Notation Given a sequence of word or sub-word tokens $X = (x_1, x_2, \ldots, x_n)$, BERT trains an encoder that produces a contextualized vector representation for each token: $\text{enc}(x_1, x_2, \ldots, x_n) = \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$.

Masked Language Model (MLM) Also known as a *cloze test*, MLM is the task of predicting missing tokens in a sequence from their placeholders. Specifically, a subset of tokens $Y \subseteq X$ is sampled and substituted with a different set of tokens. In BERT's implementation, Y accounts for 15% of the tokens in X; of those, 80% are replaced with `[MASK]`, 10% are replaced with a random token (according to the unigram distribution), and 10% are kept unchanged. The task is to predict the *original* tokens in Y from the modified input.

BERT selects each token in Y independently by randomly selecting a subset. In SpanBERT, we define Y by randomly selecting *contiguous spans* (Section 3.1).

Next Sentence Prediction (NSP) The NSP task takes two sequences (X_A, X_B) as input, and predicts whether X_B is the direct continuation of X_A. This is implemented in BERT by first reading X_A from the corpus, and then (1) either reading X_B from the point where X_A ended, or (2) randomly sampling X_B from a different point in the cor-

[2] We use the modified MRQA version of these datasets. See more details in Section 4.1.

pus. The two sequences are separated by a special [SEP] token. Additionally, a special [CLS] token is added to X_A, X_B to form the input, where the target of [CLS] is whether X_B indeed follows X_A in the corpus.

In summary, BERT optimizes the MLM and the NSP objectives by masking word pieces uniformly at random in data generated by the bi-sequence sampling procedure. In the next section, we will present our modifications to the data pipeline, masking, and pre-training objectives.

3 Model

We present SpanBERT, a self-supervised pre-training method designed to better represent and predict spans of text. Our approach is inspired by BERT (Devlin et al., 2019), but deviates from its bi-text classification framework in three ways. First, we use a different random process to mask *spans* of tokens, rather than individual ones. We also introduce a novel auxiliary objective – the span boundary objective (SBO) – which tries to predict the entire masked span using only the representations of the tokens at the span's boundary. Finally, SpanBERT samples a single contiguous segment of text for each training example (instead of two), and thus does not use BERT's next sentence prediction objective, which we omit.

3.1 Span Masking

Given a sequence of tokens $X = (x_1, x_2, \ldots, x_n)$, we select a subset of tokens $Y \subseteq X$ by iteratively sampling spans of text until the masking budget (e.g. 15% of X) has been spent. At each iteration, we first sample a span length (number of words) from a geometric distribution $\ell \sim \text{Geo}(p)$, which is skewed towards shorter spans. We then randomly (uniformly) select the starting point for the span to be masked. We always sample a sequence of complete words (instead of subword tokens) and the starting point must be the beginning of one word. Following preliminary trials[3], we set $p = 0.2$, and also clip ℓ at $\ell_{max} = 10$. This yields a mean span length of $\text{mean}(\ell) = 3.8$. Figure 2 shows the distribution of span mask lengths.

As in BERT, we also mask 15% of the tokens in total: replacing 80% of the masked tokens with [MASK], 10% with random tokens and 10% with the original tokens. However, we perform this replacement at the span level and not for each token individually; i.e. all the tokens in a span are replaced with [MASK] or sampled tokens.

[3] We experimented with $p = \{0.1, 0.2, 0.4\}$ and found 0.2 to perform the best.

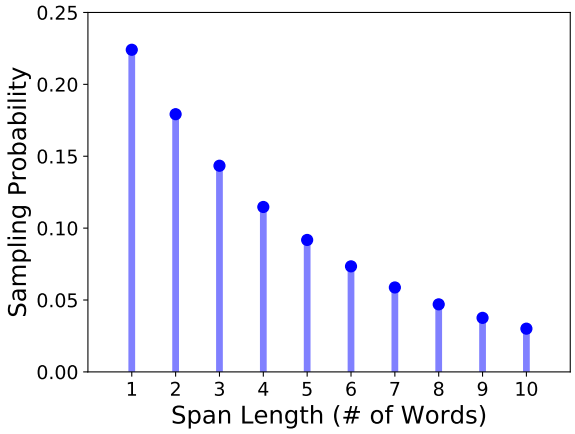

Figure 2: We sample random span lengths from a geometric distribution $\ell \sim \text{Geo}(p = 0.2)$ clipped at $\ell_{max} = 10$.

3.2 Span Boundary Objective (SBO)

Span selection models (Lee et al., 2016, 2017; He et al., 2018) typically create a fixed-length representation of a span using its boundary tokens (start and end). To support such models, we would ideally like the representations for the end of the span to summarize as much of the internal span content as possible. We do so by introducing a span boundary objective that involves predicting each token of a masked span using only the representations of the observed tokens at the boundaries (Figure 1).

Formally, we denote the output of the transformer encoder for each token in the sequence by $\mathbf{x}_1, \ldots, \mathbf{x}_n$. Given a masked span of tokens $(x_s, \ldots, x_e) \in Y$, where (s, e) indicates its start and end positions, we represent each token x_i in the span using the output encodings of the *external* boundary tokens $\mathbf{x}_{s-1}$ and $\mathbf{x}_{e+1}$, as well as the position embedding of the target token $\mathbf{p}_{i-s+1}$:

$$\mathbf{y}_i = f(\mathbf{x}_{s-1}, \mathbf{x}_{e+1}, \mathbf{p}_{i-s+1})$$

where position embeddings $\mathbf{p}_1, \mathbf{p}_2, \ldots$ mark relative positions of the masked tokens with respect to the left boundary token x_{s-1}. We implement the representation function $f(\cdot)$ as a 2-layer feed-forward network with GeLU activations (Hendrycks and Gimpel, 2016) and layer normal-

ization (Ba et al., 2016):

$$\begin{aligned}\mathbf{h}_0 &= [\mathbf{x}_{s-1}; \mathbf{x}_{e+1}; \mathbf{p}_{i-s+1}] \\ \mathbf{h}_1 &= \text{LayerNorm}(\text{GeLU}(\mathbf{W}_1\mathbf{h}_0)) \\ \mathbf{y}_i &= \text{LayerNorm}(\text{GeLU}(\mathbf{W}_2\mathbf{h}_1))\end{aligned}$$

We then use the vector representation $\mathbf{y}_i$ to predict the token x_i and compute the cross-entropy loss exactly like the MLM objective.

SpanBERT sums the loss from both the span boundary and the regular masked language model objectives for each token x_i in the masked span $(x_s, ..., x_e)$, while reusing the input embedding (Press and Wolf, 2017) for the target tokens in both MLM and SBO:

$$\begin{aligned}\mathcal{L}(x_i) &= \mathcal{L}_{\text{MLM}}(x_i) + \mathcal{L}_{\text{SBO}}(x_i) \\ &= -\log P\left(x_i \mid \mathbf{x}_i\right) - \log P\left(x_i \mid \mathbf{y}_i\right)\end{aligned}$$

3.3 Single-Sequence Training

As described in Section 2, BERT's examples contain two sequences of text (X_A, X_B), and an objective that trains the model to predict whether they are connected (NSP). We find that this setting is almost always worse than simply using a single sequence without the NSP objective (see Section 5 for further details). We conjecture that 猜测推测 single-sequence training is superior to bi-sequence training with NSP because (a) the model benefits from longer full-length contexts, or (b) conditioning on, often unrelated, context from another document adds noise to the masked language model. Therefore, in our approach, we remove both the NSP objective and the two-segment sampling procedure, and simply sample a single contiguous segment of up to $n = 512$ tokens, rather than two half-segments that sum up to n tokens together.

NSP任务不好的原因
1.句长的受限,
2.前后两个句子不相关
会飞MLM任务带来噪音

In summary, SpanBERT pre-trains span representations by: (1) masking spans of full words using a geometric distribution based masking scheme (Section 3.1), (2) optimizing an auxiliary span-boundary objective (Section 3.2) in addition to MLM using a single-sequence data pipeline (Section 3.3). A procedural description can be found in Appendix A.

4 Experimental Setup

4.1 Tasks

We evaluate on a comprehensive suite of tasks, including seven question answering tasks, coreference resolution, nine tasks in the GLUE benchmark (Wang et al., 2019), and relation extraction. We expect that the span selection tasks, question answering and coreference resolution, will particularly benefit from our span-based pre-training.

Extractive Question Answering Given a short passage of text and a question as input, the task of extractive question answering is to select a contiguous span of text in the passage as the answer.

We first evaluate on SQuAD 1.1 and 2.0 (Rajpurkar et al., 2016, 2018), which have served as major question answering benchmarks, particularly for pre-trained models (Peters et al., 2018; Devlin et al., 2019; Yang et al., 2019). We also evaluate on five more datasets from the MRQA shared task (Fisch et al., 2019)[4]: NewsQA (Trischler et al., 2017), SearchQA (Dunn et al., 2017), TriviaQA (Joshi et al., 2017), HotpotQA (Yang et al., 2018) and Natural Questions (Kwiatkowski et al., 2019). Because the MRQA shared task does not have a public test set, we split the development set in half to make new development and test sets. The datasets vary in both domain and collection methodology, making this collection a good testbed for evaluating whether our pre-trained models can generalize well across different data distributions.

Following BERT (Devlin et al., 2019), we use the same QA model architecture for all the datasets. We first convert the passage $P = (p_1, p_2, \ldots, p_l)$ and question $Q = (q_1, q_2, \ldots, q_{l'})$ into a single sequence $X =$ `[CLS]` $p_1 p_2 \ldots p_l$ `[SEP]` $q_1 q_2 \ldots q_{l'}$ `[SEP]`, pass it to the pre-trained transformer encoder, and train two linear classifiers independently on top of it for predicting the answer span boundary (start and end). For the unanswerable questions in SQuAD 2.0, we simply set the answer span to be the special token `[CLS]` for both training and testing.

Coreference Resolution Coreference resolution is the task of clustering mentions in text which refer to the same real-world entities. We evaluate on the CoNLL-2012 shared task (Pradhan et al., 2012) for document-level coreference resolution. We use the *independent* version of the Joshi et al. (2019b) implementation of the higher-order coref-

[4] https://github.com/mrqa/MRQA-Shared-Task-2019. MRQA changed the original datasets to unify them into the same format, e.g. all the contexts are truncated to a maximum of 800 tokens and only answerable questions are kept.

erence model (Lee et al., 2018). The document is divided into non-overlapping segments of a predefined length.[5] Each segment is encoded independently by the pre-trained transformer encoder, which replaces the original LSTM-based encoder. For each mention span x, the model learns a distribution $P(\cdot)$ over possible antecedent spans Y:

$$P(y) = \frac{e^{s(x,y)}}{\sum_{y' \in Y} e^{s(x,y')}}$$

The span pair scoring function $s(x, y)$ is a feedforward neural network over fixed-length span representations and hand-engineered features over x and y:

$$\begin{aligned} s(x, y) &= s_m(x) + s_m(y) + s_c(x, y) \\ s_m(x) &= \text{FFNN}_m(\mathbf{g_x}) \\ s_c(x, y) &= \text{FFNN}_c(\mathbf{g_x}, \mathbf{g_y}, \phi(x, y)) \end{aligned}$$

Here $\mathbf{g_x}$ and $\mathbf{g_y}$ denote the span representations, which are a concatenation of the two transformer output states of the span endpoints and an attention vector computed over the output representations of the token in the span. FFNN_m and FFNN_c represent two feedforward neural networks with one hidden layer, and $\phi(x, y)$ represents the hand-engineered features (e.g. speaker and genre information). A more detailed description of the model can be found in Joshi et al. (2019b).

Relation Extraction TACRED (Zhang et al., 2017) is a challenging relation extraction dataset. Given one sentence and two spans within it – subject and object – the task is to predict the relation between the spans from 42 pre-defined relation types, including *no_relation*. We follow the entity masking schema from Zhang et al. (2017) and replace the subject and object entities by their NER tags such as "`[CLS][SUBJ-PER]` was born in `[OBJ-LOC]` , Michigan, ...", and finally add a linear classifier on top of the `[CLS]` token to predict the relation type.

GLUE The General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2019) consists of 9 sentence-level classification tasks:

- Two *sentence-level classification* tasks including CoLA (Warstadt et al., 2018) for evaluating linguistic acceptability and SST-2 (Socher et al., 2013) for sentiment classification.
- Three *sentence-pair similarity* tasks including MRPC (Dolan and Brockett, 2005), a binary paraphrasing task sentence pairs from news sources, STS-B (Cer et al., 2017), a graded similarity task for news headlines, and QQP[6], a binary paraphrasing tasking between Quora question pairs.
- Four *natural language inference* tasks including MNLI (Williams et al., 2018), QNLI (Rajpurkar et al., 2016), RTE (Dagan et al., 2005; Bar-Haim et al., 2006; Giampiccolo et al., 2007) and WNLI (Levesque et al., 2011).

Unlike question answering, coreference resolution, and relation extraction, these sentence-level tasks do not require *explicit* modeling of span-level semantics. However, they might still benefit from implicit span-based reasoning (e.g., *the Prime Minister* is *the head of the government*). Following previous work (Devlin et al., 2019; Radford et al., 2018)[7], we exclude WNLI from the results to enable a fair comparison. While recent work Liu et al. (2019a) has applied several task-specific strategies to increase performance on the individual GLUE tasks, we follow BERT's single-task setting and only add a linear classifier on top of the `[CLS]` token for these classification tasks.

4.2 Implementation

We reimplemented BERT's model and pre-training method in *fairseq* (Ott et al., 2019). We used the model configuration of $\text{BERT}_{\text{large}}$ as in Devlin et al. (2019) and also pre-trained all our models on the same corpus: BooksCorpus and English Wikipedia using *cased* Wordpiece tokens.

Compared to the original BERT implementation, the main differences in our implementation include: (a) We use different masks at each epoch while BERT samples 10 different masks for each sequence during data processing. (b) We remove all the short-sequence strategies used before (they sampled shorter sequences with a small probability 0.1; they also first pre-trained with smaller se-

[5] The length was chosen from {128, 256, 384, 512}. See more details in Appendix B.

[6] https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs

[7] Previous work has excluded WNLI on account of construction issues outlined on the GLUE website – `https://gluebenchmark.com/faq`

quence length of 128 for 90% of the steps). Instead, we always take sequences of up to 512 tokens until it reaches a document boundary. We refer readers to Liu et al. (2019b) for further discussion on these modifications and their effects.

As in BERT, the learning rate is warmed up over the first 10,000 steps to a peak value of 1e-4, and then linearly decayed. We retain β hyperparameters ($\beta_1 = 0.9$, $\beta_2 = 0.999$) and a decoupled weight decay (Loshchilov and Hutter, 2019) of 0.1. We also keep a dropout of 0.1 on all layers and attention weights, and a GeLU activation function (Hendrycks and Gimpel, 2016). We deviate from the optimization by running for 2.4M steps and using an epsilon of 1e-8 for AdamW (Kingma and Ba, 2015), which converges to a better set of model parameters. Our implementation uses a batch size of 256 sequences with a maximum of 512 tokens.[8] For the SBO, we use 200 dimension position embeddings $\mathbf{p}_1, \mathbf{p}_2, \ldots$ to mark positions relative to the left boundary token. The pre-training was done on 32 Volta V100 GPUs and took 15 days to complete.

Fine-tuning is implemented based on HuggingFace's codebase (Wolf et al., 2019) and more details are given in Appendix B.

4.3 Baselines

We compare SpanBERT to three baselines:

Google BERT The pre-trained models released by Devlin et al. (2019).[9]

Our BERT Our reimplementation of BERT with improved data preprocessing and optimization (Section 4.2).

Our BERT-1seq Our reimplementation of BERT trained on single full-length sequences without NSP (Section 3.3).

5 Results

We compare SpanBERT to the baselines per task, and draw conclusions based on the overall trends.

5.1 Per-Task Results

Extractive Question Answering Table 1 shows the performance on both SQuAD 1.1 and 2.0. SpanBERT exceeds our BERT baseline by 2.0% and 2.8% F1 respectively (3.3% and 5.4% over Google BERT). In SQuAD 1.1, this result accounts for over 27% error reduction, reaching 3.4% F1 *above* human performance.

	SQuAD 1.1		SQuAD 2.0	
	EM	F1	EM	F1
Human Perf.	82.3	91.2	86.8	89.4
Google BERT	84.3	91.3	80.0	83.3
Our BERT	86.5	92.6	82.8	85.9
Our BERT-1seq	87.5	93.3	83.8	86.6
SpanBERT	**88.8**	**94.6**	**85.7**	**88.7**

Table 1: Test results on SQuAD 1.1 and SQuAD 2.0.

Table 2 demonstrates that this trend goes beyond SQuAD, and is consistent in every MRQA dataset. On average, we see a 2.9% F1 improvement from our reimplementation of BERT. Although some gains are coming from single-sequence training (+1.1%), most of the improvement stems from span masking and the span boundary objective (+1.8%), with particularly large gains on TriviaQA (+3.2%) and HotpotQA (+2.7%).

Coreference Resolution Table 3 shows the performance on the OntoNotes coreference resolution benchmark. Our BERT reimplementation improves the Google BERT model by 1.2% on the average F1 metric and single-sequence training brings another 0.5% gain. Finally, SpanBERT improves considerably on top of that, achieving a new state of the art of 79.6% F1 (previous best result is 73.0%).

Relation Extraction Table 4 shows the performance on TACRED. SpanBERT exceeds our reimplementation of BERT by 3.3% F1 and achieves close to the current state of the art (Soares et al., 2019) — Our model performs better than their BERT_{EM} but is 0.7 point behind BERT_{EM} + MTB which used entity-linked text for additional pre-training. Most of this gain (+2.6%) stems from single-sequence training although the contribution of span masking and the span boundary objective is still a considerable 0.7%, resulting largely from higher recall.

GLUE Table 5 shows the performance on GLUE. For most tasks, the different models appear to perform similarly. Moving to single-sequence training without the NSP objective substantially improves CoLA, and yields smaller

[8] On the average, this is approximately 390 sequences since some documents have fewer than 512 tokens

[9] https://github.com/google-research/bert.

	NewsQA	TriviaQA	SearchQA	HotpotQA	Natural Questions	Avg.
Google BERT	68.8	77.5	81.7	78.3	79.9	77.3
Our BERT	71.0	79.0	81.8	80.5	80.5	78.6
Our BERT-1seq	71.9	80.4	84.0	80.3	81.8	79.7
SpanBERT	**73.6**	**83.6**	**84.8**	**83.0**	**82.5**	**81.5**

Table 2: Performance (F1) on the five MRQA extractive question answering tasks.

	MUC			B^3			$CEAF_{\phi_4}$			
	P	R	F1	P	R	F1	P	R	F1	Avg. F1
Prev. SotA: (Lee et al., 2018)	81.4	79.5	80.4	72.2	69.5	70.8	68.2	67.1	67.6	73.0
Google BERT	84.9	82.5	83.7	76.7	74.2	75.4	74.6	70.1	72.3	77.1
Our BERT	85.1	83.5	84.3	77.3	75.5	76.4	75.0	71.9	73.9	78.3
Our BERT-1seq	85.5	84.1	84.8	77.8	76.7	77.2	75.3	73.5	74.4	78.8
SpanBERT	**85.8**	**84.8**	**85.3**	**78.3**	**77.9**	**78.1**	**76.4**	**74.2**	**75.3**	**79.6**

Table 3: Performance on the OntoNotes coreference resolution benchmark. The main evaluation is the average F1 of three metrics: MUC, B^3, and $CEAF_{\phi_4}$ on the test set.

	P	R	F1
$BERT_{EM}$ (Soares et al., 2019)	-	-	70.1
$BERT_{EM}$+MTB*	-	-	**71.5**
Google BERT	69.1	63.9	66.4
Our BERT	67.8	67.2	67.5
Our BERT-1seq	**72.4**	67.9	70.1
SpanBERT	70.8	**70.9**	**70.8**

Table 4: Test performance on the TACRED relation extraction benchmark. $BERT_{EM}$ and $BERT_{EM}$+MTB from Soares et al. (2019) are the current state-of-the-art. *: $BERT_{EM}$+MTB incorporated an intermediate "matching the blanks" pre-training on the entity-linked text based on English Wikipedia, which is not a direct comparison to ours trained only from raw text.

(but considerable) improvements on MRPC and MNLI. The main gains from SpanBERT are in the SQuAD-based QNLI dataset (+1.3%) and in RTE (+6.9%), the latter accounting for most of the rise in SpanBERT's GLUE average.

5.2 Overall Trends

We compared our approach to three BERT baselines on 17 benchmarks, and found that **SpanBERT outperforms BERT on almost every task**. In 14 tasks, SpanBERT performed better than all baselines. In 2 tasks (MRPC and QQP), it performed on-par in terms of accuracy with single-sequence trained BERT, but still outperformed the other baselines. In one task (SST-2), Google's BERT baseline performed better than SpanBERT by 0.4% accuracy.

When considering the magnitude of the gains, it appears that **SpanBERT is especially better at extractive question answering**. In SQuAD 1.1, for example, we observe a solid gain of 2.0% F1 even though the baseline is already well above human performance. On MRQA, SpanBERT improves between 2.0% (Natural Questions) and 4.6% (TriviaQA) F1 on top of our BERT baseline.

Finally, we observe that **single-sequence training works considerably better than bi-sequence training with next sentence prediction (NSP)** with BERT's choice of sequence lengths for a wide variety of tasks . This is surprising because BERT's ablations showed gains from the NSP objective (Devlin et al., 2019). However, the ablation studies still involved bi-sequence data processing, i.e. the pre-training stage only controlled for the NSP objective while still sampling two half-length sequences. We hypothesize that bi-sequence training, as it is implemented in BERT (see Section 2), impedes the model from learning longer-range features, and consequently hurts performance on many downstream tasks.

6 Ablation Studies

We compare our random span masking scheme with linguistically-informed masking schemes, and find that masking random spans is a competitive and often better approach. We then study the impact of the span boundary objective (SBO), and contrast it with BERT's next sentence prediction

	CoLA	SST-2	MRPC	STS-B	QQP	MNLI	QNLI	RTE	(Avg)
Google BERT	59.3	**95.2**	88.5/84.3	86.4/88.0	71.2/89.0	86.1/85.7	93.0	71.1	80.4
Our BERT	58.6	93.9	90.1/86.6	88.4/89.1	71.8/89.3	87.2/86.6	93.0	74.7	81.1
Our BERT-1seq	63.5	94.8	**91.2**/87.8	89.0/88.4	**72.1/89.5**	88.0/87.4	93.0	72.1	81.7
SpanBERT	**64.3**	94.8	90.9/**87.9**	**89.9/89.1**	71.9/**89.5**	**88.1/87.7**	**94.3**	**79.0**	**82.8**

Table 5: Test set performance on GLUE tasks. MRPC: F1/accuracy, STS-B: Pearson/Spearmanr correlation, QQP: F1/accuracy, MNLI: matched/mistached accuracies and accuracy for all the other tasks. WNLI (not shown) is always set to majority class (65.1% accuracy) and included in the average.

	SQuAD 2.0	NewsQA	TriviaQA	Coreference	MNLI-m	QNLI	GLUE (Avg)
Subword Tokens	83.8	72.0	76.3	**77.7**	86.7	92.5	83.2
Whole Words	84.3	72.8	77.1	76.6	86.3	92.8	82.9
Named Entities	84.8	72.7	78.7	75.6	86.0	93.1	83.2
Noun Phrases	85.0	**73.0**	77.7	76.7	86.5	93.2	**83.5**
Geometric Spans	**85.4**	**73.0**	**78.8**	76.4	**87.0**	**93.3**	83.4

Table 6: The effect of replacing BERT's original masking scheme (Subword Tokens) with different masking schemes. Results are F1 scores for QA tasks and accuracy for MNLI and QNLI on the development sets. All the models are based on bi-sequence training with NSP.

	SQuAD 2.0	NewsQA	TriviaQA	Coref	MNLI-m	QNLI	GLUE (Avg)
Span Masking (2seq) + NSP	85.4	73.0	78.8	76.4	87.0	93.3	83.4
Span Masking (1seq)	86.7	73.4	80.0	76.3	87.3	93.8	83.8
Span Masking (1seq) + SBO	**86.8**	**74.1**	**80.3**	**79.0**	**87.6**	**93.9**	**84.0**

Table 7: The effects of different auxiliary objectives, given MLM over random spans as the primary objective.

(NSP) objective.[10]

6.1 Masking Schemes

Previous work (Sun et al., 2019) has shown improvements in downstream task performance by masking linguistically-informed spans during pre-training for Chinese data. We compare our random span masking scheme with masking of linguistically-informed spans. Specifically, we train the following five baseline models differing only in the way tokens are masked.

Subword Tokens We sample random Wordpiece tokens, as in the original BERT.

Whole Words We sample random words, and then mask all of the subword tokens in those words. The total number of masked subtokens is around 15%.

Named Entities At 50% of the time, we sample from named entities in the text, and sample random whole words for the other 50%. The total number of masked subtokens is 15%. Specifically, we run spaCy's named entity recognizer[11] on the corpus and select all the non-numerical named entity mentions as candidates.

Noun Phrases Similar as *Named Entities*, we sample from noun phrases at 50% of the time. The noun phrases are extracted by running spaCy's constituency parser.

Geometric Spans We sample random spans from a geometric distribution, as in our SpanBERT (see Section 3.1).

Table 6 shows how different pre-training masking schemes affect performance on the development set of a selection of tasks. All the models are evaluated on the development sets and are based on the default BERT setup of bi-sequence training with NSP; the results are not directly comparable to the main evaluation. With the exception of coreference resolution, masking random spans is preferable to other strategies. Although linguistic masking schemes (named entities and

[10] To save time and resources, we use the checkpoints at 1.2M steps for all the ablation experiments.

[11] https://spacy.io/

noun phrases) are often competitive with random spans, their performance is not consistent; for instance, masking noun phrases achieves parity with random spans on NewsQA, but underperforms on TriviaQA (-1.1% F1).

On coreference resolution, we see that masking random subword tokens is preferable to any form of span masking. Nevertheless, we shall see in the following experiment that combining random span masking with the span boundary objective can improve upon this result considerably.

6.2 Auxiliary Objectives

In Section 5, we saw that bi-sequence training with the next sentence prediction (NSP) objective can hurt performance on downstream tasks, when compared to single-sequence training. We test whether this holds true for models pre-trained with span masking, and also evaluate the effect of replacing the NSP objective with the span boundary objective (SBO).

Table 7 confirms that single-sequence training typically improves performance. Adding SBO further improves performance, with a substantial gain on coreference resolution (+2.7% F1) over span masking alone. Unlike the NSP objective, SBO does not appear to have any adverse effects.

7 Related Work

Pre-trained contextualized word representations that can be trained from unlabeled text (Dai and Le, 2015; Melamud et al., 2016; Peters et al., 2018) have had immense impact on NLP lately, particularly as methods for initializing a large model before fine-tuning it for a specific task (Howard and Ruder, 2018; Radford et al., 2018; Devlin et al., 2019). Beyond differences in model hyperparameters and corpora, these methods mainly differ in their pre-training tasks and loss functions, with a considerable amount of contemporary literature proposing augmentations of BERT's masked language modeling (MLM) objective.

While previous and concurrent work has looked at masking (Sun et al., 2019) or dropping (Song et al., 2019; Chan et al., 2019) multiple words from the input – particularly as pretraining for language generation tasks – SpanBERT pretrains span representations (Lee et al., 2016), which are widely used for question answering, coreference resolution and a variety of other tasks. ERNIE (Sun et al., 2019) shows improvements on Chinese NLP tasks using phrase and named entity masking. MASS (Song et al., 2019) focuses on language generation tasks, and adopts the encoder-decoder framework to reconstruct a sentence fragment given the remaining part of the sentence. We attempt to more explicitly model spans using the SBO objective, and show that (geometrically distributed) random span masking works as well, and sometimes better than, masking linguistically-coherent spans. We evaluate on English benchmarks for question answering, relation extraction, and coreference resolution in addition to GLUE.

A different ERNIE (Zhang et al., 2019) focuses on integrating structured knowledge bases with contextualized representations with an eye on knowledge-driven tasks like entity typing and relation classification. UNILM (Dong et al., 2019) uses multiple language modeling objectives – unidirectional (both left-to-right and right-to-left), bidirectional, and sequence-to-sequence prediction – to aid generation tasks like summarization and question generation. XLM (Lample and Conneau, 2019) explores cross-lingual pre-training for multilingual tasks such as translation and cross-lingual classification. Kermit (Chan et al., 2019), an insertion based approach, fills in missing tokens (instead of predicting masked ones) during pretraining; they show improvements on machine translation and zero-shot question answering.

Concurrent with our work, RoBERTa (Liu et al., 2019b) presents a replication study of BERT pre-training that measures the impact of many key hyperparameters and training data size. Also concurrent, XLNet (Yang et al., 2019) combines an autoregressive loss and the Transformer-XL (Dai et al., 2019) architecture with a more than an eight-fold increase in data to achieve current state-of-the-art results on multiple benchmarks. XLNet also masks spans (of 1-5 tokens) during pre-training, but predicts them autoregressively. Our model focuses on incorporating span-based pre-training, and as a side effect, we present a stronger BERT baseline while controlling for the corpus, architecture, and the number of parameters.

Related to our SBO objective, *pair2vec* (Joshi et al., 2019a) encodes word-pair relations using a negative sampling-based multivariate objective during pre-training. Later, the word-pair representations are injected into the attention-layer of downstream tasks, and thus encode limited down-

stream context. Unlike pair2vec, our SBO objective yields "pair" (start and end tokens of spans) representations which more fully encode the context during both pre-training and finetuning, and are thus more appropriately viewed as *span* representations. Stern et al. (2018) focus on improving language generation speed using a block-wise parallel decoding scheme; they make predictions for multiple time steps in parallel and then back off to the longest prefix validated by a scoring model. Also related are sentence representation methods (Kiros et al., 2015; Logeswaran and Lee, 2018) which focus on predicting surrounding contexts from sentence embeddings.

8 Conclusion

We presented a new method for span-based pre-training which extends BERT by (1) masking contiguous random spans, rather than random tokens, and (2) training the span boundary representations to predict the entire content of the masked span, without relying on the individual token representations within it. Together, our pre-training process yields models that outperform all BERT baselines on a variety of tasks, and reach substantially better performance on span selection tasks in particular.

Acknowledgements

We would like to thank Pranav Rajpurkar and Robin Jia for patiently helping us evaluate SpanBERT on SQuAD. We thank the anonymous reviewers, the action editor, and our colleagues at Facebook AI Research and the University of Washington for their insightful feedback that helped improve the paper.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second PASCAL recognising textual entailment challenge. In *Proceedings of the second PASCAL challenges workshop on recognising textual entailment*, pages 6–4.

Daniel Cer, Mona Diab, Eneko Agirre, IÃśigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *International Workshop on Semantic Evaluation (SemEval)*, pages 1–14, Vancouver, Canada.

William Chan, Nikita Kitaev, Kelvin Guu, Mitchell Stern, and Jakob Uszkoreit. 2019. KERMIT: Generative insertion-based modeling for sequences. *arXiv preprint arXiv:1906.01604*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer.

Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3079–3087.

Zihang Dai, Zhilin Yang, Yiming Yang, William W Cohen, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Association for Computational Linguistics (ACL)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *North American Association for Computational Linguistics (NAACL)*.

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

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems (NIPS)*.

Matthew Dunn, Levent Sagun, Mike Higgins, V Ugur Guney, Volkan Cirik, and Kyunghyun Cho. 2017. SearchQA: A new Q&A dataset augmented with context from a search engine. *arXiv preprint arXiv:1704.05179*.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In *Proceedings of 2nd Machine Reading for Reading Comprehension (MRQA) Workshop at EMNLP*.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pages 1–9.

Luheng He, Kenton Lee, Omer Levy, and Luke Zettlemoyer. 2018. Jointly predicting predicates and arguments in neural semantic role labeling. In *Association for Computational Linguistics (ACL)*, pages 364–369.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Mandar Joshi, Eunsol Choi, Omer Levy, Daniel Weld, and Luke Zettlemoyer. 2019a. pair2vec: Compositional word-pair embeddings for cross-sentence inference. In *North American Association for Computational Linguistics (NAACL)*, pages 3597–3608.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Association for Computational Linguistics (ACL)*, pages 1601–1611.

Mandar Joshi, Omer Levy, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2019b. BERT for coreference resolution: Baselines and analysis. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Ryan Kiros, Yukun Zhu, Ruslan R. Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in Neural Information Processing Systems (NIPS)*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics (TACL)*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems (NIPS)*.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 188–197.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *North American Association for Computational Linguistics (NAACL)*, pages 687–692.

Kenton Lee, Shimi Salant, Tom Kwiatkowski, Ankur Parikh, Dipanjan Das, and Jonathan Berant. 2016. Learning recurrent span representations for extractive question answering. *arXiv preprint arXiv:1611.01436*.

Hector J Levesque, Ernest Davis, and Leora Morgenstern. 2011. The Winograd schema challenge. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*, volume 46, page 47.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics (ACL).

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. *arxiv preprint arXiv:1907.11692*.

Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. *arxiv preprint arXiv:1803.02893*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*.

Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning generic context embedding with bidirectional LSTM. In *Computational Natural Language Learning (CoNLL)*, pages 51–61.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *North American Association for Computational Linguistics (NAACL)*, pages 48–53.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *North American Association for Computational Linguistics (NAACL)*, pages 2227–2237.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *Joint Conference on EMNLP and CoNLL-Shared Task*, pages 1–40.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163. Association for Computational Linguistics (ACL).

Alec Radford, Karthik Narasimhan, Time Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. Technical report, Technical report, OpenAI.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Association for Computational Linguistics (ACL)*, pages 784–789.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 2383–2392.

Livio Baldini Soares, Nicholas Arthur FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Association for Computational Linguistics (ACL)*, pages 2895–2905.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1631–1642.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning (ICML)*, pages 5926–5936.

Mitchell Stern, Noam Shazeer, and Jakob Uszkoreit. 2018. Blockwise parallel decoding for deep autoregressive models. In *Advances in Neural Information Processing Systems (NIPS)*.

Yu Stephanie Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xinlun Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. ERNIE: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A machine comprehension dataset. In *2nd Workshop on Representation Learning for NLP*, pages 191–200.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez,

Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS)*.

Alex Wang, Amapreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations (ICLR)*.

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

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *North American Association for Computational Linguistics (NAACL)*, pages 1112–1122.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 2369–2380.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 35–45.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Association for Computational Linguistics (ACL)*, pages 1441–1451.

Appendices

A Pre-training Procedure

We describe our pre-training procedure as follows:

1. Divide the corpus into single contiguous blocks of up to 512 tokens.
2. At each step of pre-training:
   (a) Sample a batch of blocks uniformly at random.
   (b) Mask 15% of word pieces in each block in the batch using the span masking scheme (Section 3.1).
   (c) For each masked token x_i, optimize $\mathcal{L}(x_i) = \mathcal{L}_{\text{MLM}}(x_i) + \mathcal{L}_{\text{SBO}}(x_i)$ (Section 3.2).

B Fine-tuning Hyperparameters

We apply the following fine-tuning hyperparameters to all methods, including the baselines.

Extractive Question Answering For all the question answering tasks, we use `max_seq_length` = 512 and a sliding window of size 128 if the lengths are longer than 512. We choose learning rates from {5e-6, 1e-5, 2e-5, 3e-5, 5e-5} and batch sizes from {16, 32} and fine-tune 4 epochs for all the datasets.

Coreference Resolution We divide the documents into multiple chunks of lengths up to `max_seq_length` and encode each chunk independently. We choose `max_seq_length` from {128, 256, 384, 512}, BERT learning rates from {1e-5, 2e-5}, task-specific learning rates from {1e-4, 2e-4, 3e-4} and fine-tune 20 epochs for all the datasets. We use batch size $= 1$ (one document) for all the experiments.

TACRED/GLUE We use `max_seq_length` = 128 and choose learning rates from {5e-6, 1e-5, 2e-5, 3e-5, 5e-5} and batch sizes from {16, 32} and fine-tuning 10 epochs for all the datasets. The only exception is CoLA, where we used 4 epochs (following Devlin et al. (2019)), because 10 epochs lead to severe overfitting.