SpanBERT: Improving Pre-training by Representing and Predicting Spans

AUTHORS: Mandar Joshi* Danqi Chen*

Yinhan Liu Daniel S. Weld Luke Zettlemoyer Omer Levy

Presented by:
Jesse Annan

Department of Mathematics and Statistics
Georgia State University

Outline

Background

The Rise of BERT - Bidirectional Encoder Representations from Transformers

What is **BERT**? [1]

- **Bidirectional Context Understanding** - considers the context from the left and right sides of a word when predicting it representation
- **Transformer Architecture** - uses self-attention mechanisms to weigh the importance of different words in a sentence when encoding their representations.

Background

pre-training BERT

BERT optimizes two training objectives:

- Masked Language Model (MLM)
- Next Sentence Prediction (NSP)

BERT is pre-trained on a large corpus of text data in an unsupervised manner.

Background

Pre-training BERT - Masked Language Model

MLM is the task of predicting missing tokens in a sequence from their placeholders.

Implementation: Given a sequence of word or sub-word tokens $\mathbb{X} = (x_1, x_2, \cdots, x_n)$

0. $\mathbb{Y} := 15\% \ \mathbb{X}$
1. replace 80% (of $\mathbb{Y}$) with *[MASK]*
2. replace 10% with a random token
3. 10% unchanged

Background

pre-training BERT - Next Sentence Prediction

NSP is the task of predicting weather a sequence is a direct continuation of another.

Implementation: [2] Sample two sequences $(\mathbb{X}_A, \mathbb{X}_B)$.
$\mathbb{X}_B$ is:

1. 50% of the time, the actual next sentence.
2. 50% of the time, a random sentence from the corpus

Background

BERT - Summary and Limitation

BERT optimizes the **MLM** and **NSP** objectives by masking word pieces uniformly at random in data generated by the bi-sequence sampling procedure.

**Falls short in understanding spans of text.*

SpanBERT

Improving Pre-training by Representing and Predicting Spans

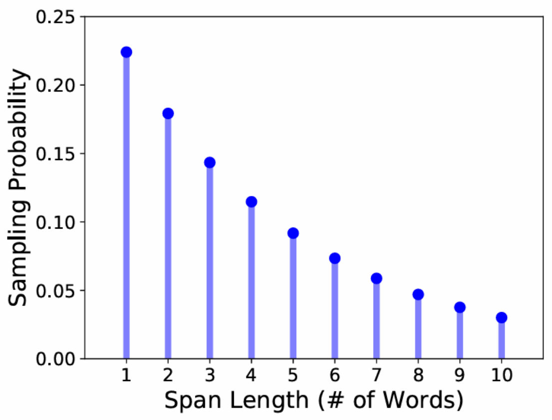
SpanBERT [3] is a pre-training method that is designed to better represent and predict spans of text.

- mask spans of token, rather than individual ones
- *predict spans using representations from span boundaries
- ~~NSP~~, samples a single segment of text for each training example

SpanBERT

Span Masking Objective (similar to MLM)

Sample a number of (complete) words from a geometric distribution, $l \sim Geo(p = 0.2)$, clipped at $l_{max} = 10$



mean($l = 3.8$); experimented with p ={*0.1, 0.2, 0.4*}

SpanBERT

Span Boundary Objective (SBO)

SBO involves predicting each token of a masked span using only the representations of the observed tokens at the boundaries.

Given a masked span of tokens $(x_s, ..., x_e) \in \mathbb{Y}$

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

where x_i is the ith token in the masked span, $\mathbf{x}_i$ is the output of the transformer encoder for ith token in the sequence, and f is a 2-layer feed-forward network with GeLU actiation function [4] and Layer Normalization [5].

SpanBERT

Span Boundary Objective (SBO) Cont'd

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

$\mathbf{y}_i$ to predict the token x_i and compute the cross-entropy loss.

$$
\begin{aligned}
\mathcal{L} &= \mathcal{L}_{MLM}(x_i) + \mathcal{L}_{SBO}(x_i) \\
&= -logP(x_i|\mathbf{x}_i) - logP(x_i|\mathbf{y}_i)
\end{aligned}
$$

GeLU Activation Function

GeLU v.s. ReLU v.s. ELU

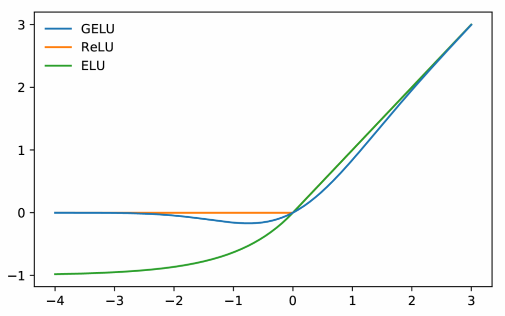


Figure: The GELU ($\mu = 0, \sigma = 1$), ReLU, and ELU ($\alpha = 1$)

GeLU has shown good empirical performance in various deep learning tasks, including NLP. It is a popular choice in transformer-based models.

SpanBERT

Single-sequence v.s. Bi-sequence training

- **SpanBERT** samples a single contiguous segment of up to $n = 512$ instead of using NSP which uses two segments for pretraining, as in **BERT**.
- It is conjectured that **single-sequence training is superior to bi-sequence training** with NSP because
  1. the model benefits from longer full-length contexts, or
  2. conditioning on, often unrelated, context from another document adds noise to the MLM.

SpanBERT

SpanBERT Framework

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

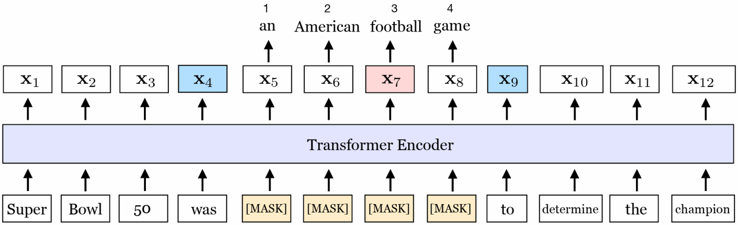


Figure: The span an American football game is masked. The SBO uses $\mathbf{x}_4$ and $\mathbf{x}_9$, to predict each token in the masked span. The equation shows the loss terms for predicting the token, football, marked by the position embedding $\mathbf{p}_3$.

SpanBERT: Experiments

Span Related Tasks

- Extractive Question Answering
- Coreference Resolution
- Relation Extraction

SpanBERT: Experiments

Baselines

- **Google BERT** - original BERT (results)
- **Our BERT** - reimplementation of BERT; used different mask at each epoch
- **Our BERT-1seq** - reimplementation of BERT; used single full-length sequences (~~NSP~~)

SpanBERT: Experiments

Extractive Question Answering - SQuAD 1.1/2.0 benchmark

	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**

Figure: **SpanBERT** exceeds the **our BERT** baseline by 2.0% and 2.8% F1, respectively. Also 3.3% and 5.4% over **Google BERT**

SpanBERT: Experiments

Extractive Question Answering - MRQA benchmark

	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**

Figure: Performance (F1) on the five MRQA extractive question answering tasks. On average, we see a 2.9% F1 improvement from reimplementation of **BERT**

SpanBERT: Experiments

Coreference Resolution - OntoNotes benchmark

	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**

Figure: **SpanBERT** achieves a new state of the art (SotA) of 79.6% F1

SpanBERT: Experiments

Relation Extraction - TACRED benchmark

	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**

Figure: **SpanBERT** exceeds the reimplementation of **BERT** by 3.3% F1 and achieves close to the current SotA

GLUE

The General Language Understanding Evaluation (GLUE) benchmark

	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**

Figure: Test set performance on GLUE tasks. MRPC: F1/accuracy, STS-B: Pearson/Spearman correlation, QQP: F1/accuracy, MNLI: matched/mismatched accuracies and accuracy for all the other tasks

SpanBERT: Experiments

Ablation Studies on masking schemes

	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

Figure: The effect of replacing **BERT**'s original masking scheme (subword tokens). **SpanBERT** geometric spans outperforms other span variants (F1 scores). All the models are based on bi-sequence training with NSP

SpanBERT: Experiments

Ablation Studies on auxiliary objectives

	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**

Figure: The effect of different auxiliary objectives. Single-sequence training typically improves performance, adding SBO further improves performance.

In Summary, **SpanBERT**:

1. masking spans of full words using a geometric distribution based masking scheme.
2. optimizing an auxiliary span-boundary objective in addition to MLM using a single-sequence data pipeline.
3. better at extractive question answering.
4. single-sequence training works considerably better than bi-sequence training with NSP.

Resources I

[1] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," CoRR, vol. abs/1810.04805, 2018. arXiv: 1810.04805. [Online]. Available: http://arxiv.org/abs/1810.04805.

[2] S.-H. Tsang, "Review - spanbert: Improving pre-training by representing and predicting spans," (2022), [Online]. Available: https://sh-tsang.medium.com/review-spanbert-improving-pre-training-by-representing-and-predicting-spans-da61f8a3e7b1.

Resources II

[3] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, "Spanbert: Improving pre-training by representing and predicting spans," CoRR, vol. abs/1907.10529, 2019. arXiv: 1907.10529. [Online]. Available: http://arxiv.org/abs/1907.10529.

[4] D. Hendrycks and K. Gimpel, "Bridging nonlinearities and stochastic regularizers with gaussian error linear units," CoRR, vol. abs/1606.08415, 2016. arXiv: 1606.08415. [Online]. Available: http://arxiv.org/abs/1606.08415.

[5] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," arXiv preprint arXiv:1607.06450, 2016.

Q & A

Thank You!