spanBERT

2021 08 04

주세준

소개

- Mandar Joshi
- Facebook AI
- TACL 2020
- span

SpanBERT: Improving Pre-training by Representing and Predicting Spans

Mandar Joshi[*†] Danqi Chen[*‡§] Yinhan Liu[§]
Daniel S. Weld[†ᵋ] 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
[ᵋ] 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 SBO 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,

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

Transactions of the Association for Computational Linguistics, vol. 8, pp. 64–77, 2020. https://doi.org/10.1162/tacl_a_00300
Action Editor: Radu Florian. Submission batch: 9/2019; Revision batch: 10/2019; Published 3/2020.

BERT

- individual word, subword 단위로 마스킹 하였음. 그러나 NLP task는 text span 간의 relation에 대한 추론을 할 수 있어야 함
- “question answering” "conference resolution“ 같은 task를 위한 span-level pre-train 방법을 제시

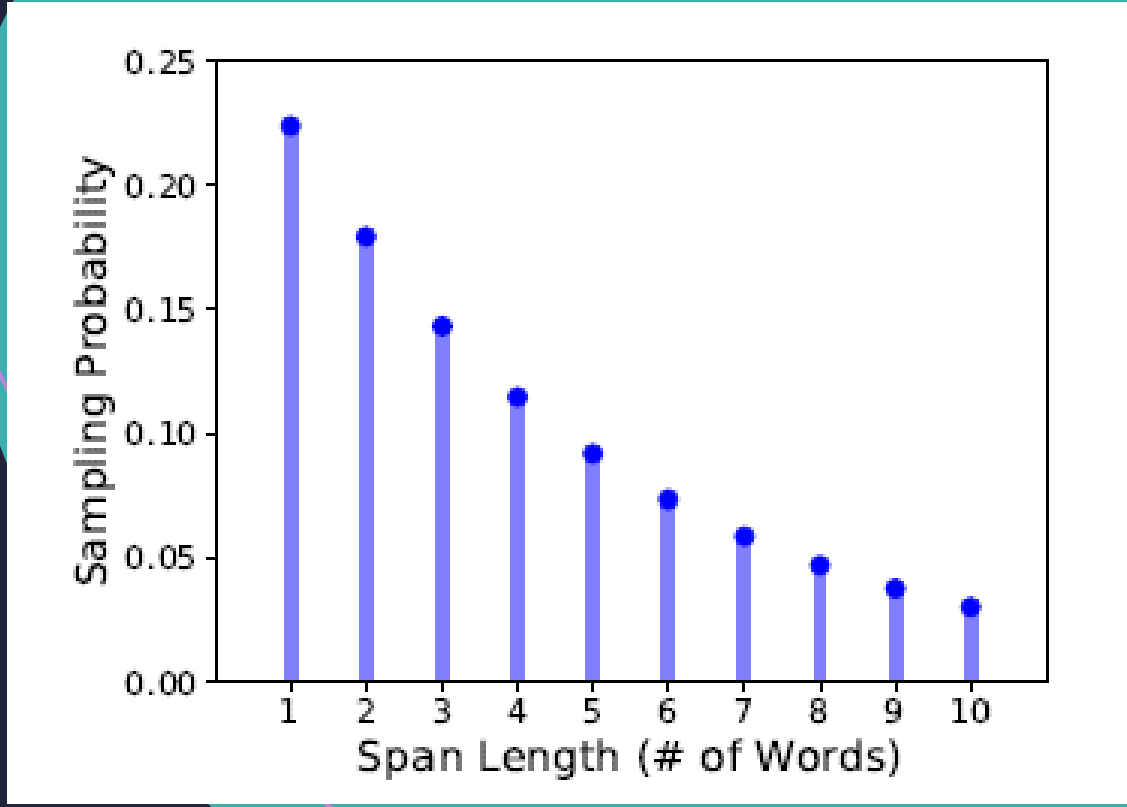
BERT

- Token -> contextualized vector representation (encoder)
- MLM
  sequence 의 15% individual token mask(0.8 [MASK] 0.1 random token 0.1 not change)
  [MASK] 예측
- NSP
  [CLS] XA [SEP] XB

Input = [CLS] the man went to [MASK] store [SEP] he bought a gallon [MASK] milk [SEP] LABEL = IsNext

1. Span Masking

- Masking budget (15%)가 채워질 때까지 text span sampling
- Geometric distribution 을 활용하여 span length sampling (left skewed)
- Span starting point random
- Complete word로 span 길이 측정



2. Span Boundary Objective (SBO)

- Span boundary 의 token representation 만 사용하여 masked span의 token을 예측하는 task
- Start End positional embedding of target token
- Span 내부 token represent

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

- 2 layer ffn with GeLU (Yi 는 vector representation)

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

- Mlm sbo loss 합해서 사용

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

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

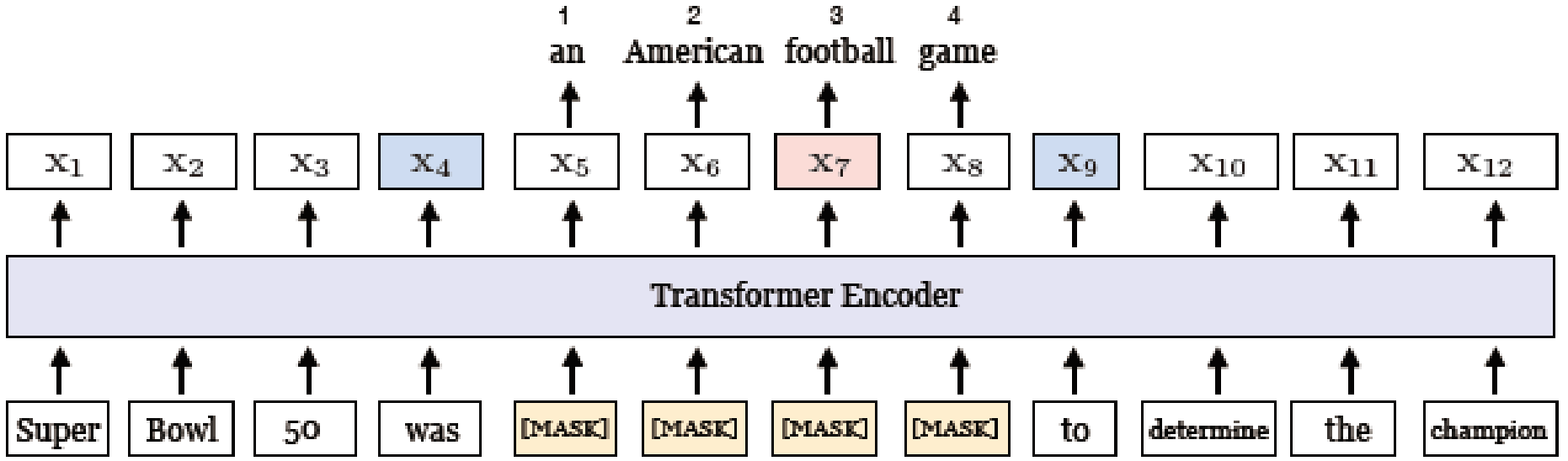


Figure 1: An illustration of SpanBERT training. The span *an American football game* is masked. The span boundary objective (SBO) uses the output representations of the boundary tokens, $\mathbf{x}_4$ and $\mathbf{x}_9$ (in blue), to predict each token in the masked span. The equation shows the MLM and 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.

3. Single-Sequence Training

- NSP를 pre-training 에 사용하는 것이 오히려 악영향
- Full-length context
- 두 문장이 mlm에 오히려 noise로 작용함
- => n = 512 의 single contiguous segment 사용

Baseline

- Different mask at each epoch vs. 10 different mask for each sequence during data processing
- doesn't use short sequence
- BERT large
- 32 Volta V100 GPUs and took 15 days

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

	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.

- SpanBERT is especially better at extractive question answering
- Relation extraction .

	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.

BERT various 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

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.

- **Subword Tokens : 오리지널 BERT방식으로 wordpiece token단위로 마스킹한다.**
- **Whole Words : word단위로 랜덤하게 마스킹한다. 마스킹된 subtokens의 수가 15%를 유지하도록 한다.**
- **Named Entities : 50%의 확률로 named entity를, 나머지 50%의 확률로 whole words를 랜덤하게 뽑아 마스킹한다. named entity는 spacy를 사용하여 뽑았다고 한다.**
- **Noun Phrases : 위와 유사하게 50%의 확률로 noun을 뽑고, spacy를 사용한다.**
- **Random Spans : geometric distribution을 따르는 random span masking으로 SpanBERT에서 사용하는 방법이다.**