POSPAN: Position-Constrained Span Masking for Language Model Pre-training

Zhenyu Zhang  
JD AI Research  
zhangzhenyu47@jd.com

Lei Shen  
JD AI Research  
shenlei20@jd.com

Yuming Zhao  
JD AI Research  
zhaoyming3@jd.com

Meng Chen  
JD AI Research  
chenmeng20@jd.com

Xiaodong He  
JD AI Research  
xiaodong.he@jd.com

1 INTRODUCTION

Large-scale pre-trained language models (PLMs) have achieved unmatched performance in many natural language understanding (NLU) tasks [5, 12, 18, 26]. As one of the most dominant model families, BERT-based models leverage masked language modeling (MLM) [5], a token-level denoised auto-encoding task, to facilitate the representation learning during pre-training. MLM samples tokens from the input text and replaces them with a special token “[MASK]”, then asks the model to reconstruct original tokens based on the contextual hidden representations of masked tokens, enabling the model to capture semantics of each token with bi-directional context. However, the vanilla MLM only considers masking individual tokens (words or sub-words) randomly, hence it neglects the information beyond token level, such as the compositional semantics for a phrase and the inter-token dependency for an entity, which are critical components of NLU in pre-training [25].

To learn the rich semantics in spans, e.g., phrases, entities, and n-grams, there has been an increasing body of works aiming to improve MLM [3, 10, 25, 29]. They mostly focus on span-level masking methods, where a contiguous token sequence, i.e., span, is masked according to some discrete span length distributions whose parameters are either preset [10] or adapted from external knowledge [18]. Despite being advantageous over the vanilla MLM in Joshi et al. [10], existing span-level masking methods all assume that the positions of masked spans are uniformly distributed, that is, the masked spans of a sequence are independent from each other [1]. This could cause sub-optimal selection of masked spans in pre-training, since the intrinsic features of natural languages, including semantic flows and inter-span interactions, can lead to strong dependencies of phrases and entities across context [3]. Therefore, the assumption of uniformly-distributed positions of masked spans might not be a proper inductive bias, conversely, span masking bonded with certain position distance is likely to yield a better modeling of contextual dependencies.

To this end, we propose POSPAN, a general span masking algorithm that allows diverse position-constrained span masking strategies. POSPAN employs a span length distribution $F_M$ and a position constraint distribution $F_D$ to control the length and position distance of masked spans, respectively. Moreover, all current span-level masking methods could be unified under POSPAN, where $F_M$s are different while $F_D$s are derived from the same uniform distribution. To verify the effectiveness of POSPAN for language understanding, we evaluate PLMs with POSPAN on various tasks.
from the GLUE [21] and Super GLUE [20] benchmarks. Experimental results show that POSPAN can outperform the previous span-length-only counterparts consistently when combined with suitable position constraints, which also matches the conclusion of our theoretical analysis.

2 APPROACH

2.1 Span Masking with Two Distributions

For span masking, there are two important factors: (1) the length of a span, i.e., how many consecutive tokens need to be masked, and (2) the position of a span, i.e., where to start masking. Given a sequence with \( m \) masked spans, \( S = \{S_1, S_2, \ldots, S_m\} \), the length and position of the \( i \)-th span are denoted as \( l_{en_i} \) and \( p_{os_i} \) for simplicity, where we use \( i \) to represent the full notation of \( i \)-th span, \( S_i \). We formulate these factors under two distributions:

\[
\begin{align*}
    l_{en_i} & \sim F_M, \\
    |p_{os_i+1} - p_{os_i}| & \sim F_D, \quad 0 \leq p_{os_i} < p_{os_{i+1}} \leq N,
\end{align*}
\]

where \( F_M \) is the span length distribution, and \( |p_{os_{i+1}} - p_{os_i}| \), named as position constraint, is the distance between two spans, \( S_i \) and \( S_{i+1} \), which follows the span constraint distribution \( F_D \). \( N \) is the length of input tokens.

Previous works mostly focus on the design of \( F_M \), while the span position is randomly selected. We can derive that \( F_D \) behind random selection is a polynomial distribution. We start with two random spans, \( S_i \) and \( S_j \), and the probability of their position constraint with length \( d \) is a new distribution \( F_D \):

\[
P(|p_{os_i} - p_{os_j}| \leq d) = F_D(d). \tag{1}
\]

Given the fixed masking rate \( r_m \in (0, 1) \), \( r_m \times N \) tokens will be masked, and the position constraint of \( d \) is between 0 and \((1 - r_m)N\), i.e., \( 0 \leq d \leq (1 - r_m)N \). Then, \( F_D(d) \) can be obtained by integrating the position constraint over all positions:

\[
F_D(d) = \frac{1}{c} \left( \int_0^d (p_{os_j} + d)dp_{os_j} + \sum_{j=1}^N (N - (p_{os_j} - d))dp_{os_j} + \int_d^{N-d} (p_{os_j} + d - (p_{os_j} - d))dp_{os_j} \right) - \frac{2Nd - d^2}{c},
\]

where \( c = (1 - r_m)^2 \) is added to scale the result in range (0, 1).

Since there are \( |S| \) spans uniformly sampled and the position constraint of each span pair follows \( F_D(d) \), the position constraint of two consecutive spans follows another polynomial distribution, denoted as \( poly(r_m, N) \):

\[
poly(r_m, N) \sim F_D = P(|p_{os_{i+1}} - p_{os_i}| \leq d) \leq \frac{d}{|S| - 1} \approx F_D(d)
\]

It is clear that \( F_D \) of existing methods is not fully considered, and we will explore the combination of different \( F_M \) and \( F_D \)s in the following subsections.

2.2 Theoretical Analysis of POSPAN

2.2.1 Latent Semantic Dependency. Assuming two spans of an input text are \( S_i \) and \( S_j \) whose start positions are \( i \) and \( j \), respectively. There are \( l_{en_i} = |S_i| \) tokens in \( S_i \) and \( l_{en_j} = |S_j| \) tokens in \( S_j \). The contextual dependency of spans entails rich semantics that are important for the prediction of masked spans. We use a latent variable \( R_{ij} \) to represent the semantic dependency between \( S_i \) and \( S_j \). In general, there are 3 cases for \( R_{ij} \) in the natural language:

- **Case 1**: There are barely any dependency or semantic relationship between \( S_i \) and \( S_j \), i.e. we can predict \( S_i \) and \( S_j \) independently without knowing each other.
- **Case 2**: \( S_i \rightarrow S_j \), i.e., \( S_i \) is the premise of \( S_j \). When \( S_i \) appears, \( S_j \) will appear most of the time.
- **Case 3**: \( S_j \rightarrow S_i \), i.e., \( S_j \) is the premise of \( S_i \).

With the latent variable \( R_{ij} \), we denote span masking methods as follows:

\[
P(S_i, S_j|R_{ij}) = \frac{P(R_{ij}|S_i, S_j) \times P(S_i, S_j)}{P(R_{ij})},
\]

\[
\log P(S_i, S_j|R_{ij}) \propto \log P(R_{ij}|S_i, S_j) + \log P(S_i, S_j), \tag{2}
\]

where \( P(R_{ij}) \) is the prior probability that can be estimated from the corpus, and \( P(S_i, S_j) \) is the span pair probability that is represented as \( P(x_1, \ldots, x_{i+|l_{en_i}|}, x_j, \ldots, x_{j+|l_{en_j}|}) \). Then, to maximize the log-likelihood of \( M \) masked spans in training, we have:

\[
\frac{\sum_{i=1}^M \log P(S_i, S_j)}{2} \rightarrow \sum_{i=1}^M \log P(S_i),
\]

\[
\mathbb{L}_S \rightarrow \max (\mathbb{E} [\log P(S_i|l_{en_i})]), \quad \text{where}
\]

\[
\mathbb{E} [\log P(S_i|l_{en_i})] = \mathbb{E}_{len_i, F_M} \left( \sum_{i=0}^{l_{en_i}-1} \log P(x_{is}) \right).
\]
The assumption in Equation (5) is that the probability of predicting a masked token is independent from each other. \( \sum_{i,j} P(S_i, S_j) \) represents \( M(M-1)/2 \) combinations of \( M \) unique spans. We denote the objective function as the span loss \( L_S \), where span length \( l_{en} \sim F_M \), and "\( \rightarrow \)" means the goal of \( L_S \).

Previous works only focus on the second term in Equation (4) with various \( F_M \)'s, and ignore the influence of latent semantic dependency \( R_{ij} \), denoted as the first term that is governed by \( F_D \). For Case 1, the first term in Equation (4) is negligible. However, for Case 2 and 3, the first term in Equation (4) is critical, since improper settings of \( F_D \) can harm span masking language models for natural language understanding.

2.2.2 Position Constraint as Prior Knowledge. The prediction of masked spans can be achieved via the usage of boundary tokens of a span \([10]\), i.e., \( P(S_i) \) can be estimated from the boundary tokens of \( S_i \):

\[
P(S_i) = P(x_{pos_{i-1}} \cdot x_{pos_{i}} \cdot l_{en_{i}} \cdot x_{pos_{i+1}}) \\
= P(x_{pos_{i-1}} \cdot x_{pos_{i}} \cdot l_{en_{i}} \cdot x_{pos_{i+1}}),
\]

where \( pos_i \) is the position of \( S_i \). Since we mask tokens in span \( S_i \), the masked token \( x_{pos_{i}} \) is used to represent \( pos_i \). The distance dependency between \( S_i \) and \( S_j \) is reflected by \( d \) tokens between \( S_i \) and \( S_j \), i.e., \( l_{ij} = [x_{pos_{i}} - d, \ldots, x_{pos_{j}} - 1] \). Then, \( R_{ij} \) is inferred by:

\[
P(R_{ij}|d) = P(R_{ij}|S_i, S_j, l_{ij}).
\]

We assume that given \( R_{ij} \), unmasked tokens in \( l_{ij} \) are independent from tokens in \( S_i \) and \( S_j \), then the likelihood maximization of \( P(R_{ij}|d) \) is equivalent to optimize the first term in Equation (4):

\[
P(R_{ij}|S_i, S_j) \propto P(R_{ij}|d).
\]

Finally, the pre-training with masked language modeling can be decomposed into two losses:

\[
L = L_R + L_S, \\
L_R := \max(\mathbb{E}[\log P(R_{ij}|F_D)]),
\]

where \( L_S \) is span length loss from Equation (5) and \( L_R \) is span dependency loss. The span length and position constraint are controlled by the prior distributions \( F_M \) and \( F_D \), respectively, i.e., \( l_{en} \sim F_M \) and \( d \sim F_D \). By properly setting the prior knowledge, we can improve the upper bound of pre-training for NLU tasks.

2.3 POSSPAN Algorithm

Various masking strategies can be achieved via the combination of different \( F_M \)'s and \( F_D \)'s. To investigate the impact of different masking strategies conveniently, we illustrate the sampling algorithm of POSSPAN in Algorithm 1. Given a text sequence, the algorithm first samples \( N \) span lengths \(( \sim F_M \) and \( N \) inter-span position constraints \(( \sim F_D \)) stored in vector \( M \) and \( D \), respectively (Line 3-8). Then for each token position, we iteratively negate the value of \( do_{mask} \) to obtain the span length or position constraint turn by turn until all tokens are traversed, and select all possible spans into the set \( spans \) (Line 9-15). Next, we remove spans from \( spans \) until the masked token number satisfies the masking rate requirement (Line 16). Finally, we replace the selected tokens with "\( \text{[MASK]} \)" (Line 17-18).

We follow previous works \([5, 15]\) to mask 15% of tokens \((m_0 = 0.15)\), where 80% of them are replaced with "\( \text{[MASK]} \)". 10% are replaced with tokens randomly sampled from the vocabulary, and the rest 10% are untouched. Through POSSPAN, we can re-implement all the previous masking methods and design new masking strategies easily based on various distributions, including Normal (\( \text{Norm} \)), Geometric (\( \text{Geo} \)), Uniform (\( \text{Rand} \)) and Poisson (\( \text{Pois} \)) distribution.

Table 1 illustrates the hyper-parameters of distributions we investigate, where the range of these distributions are around 4 and 5 for \( F_M \) and \( F_D \), respectively, so as to be comparable with previous methods \([3, 10]\). By combining different \( F_M \)'s and \( F_D \)'s, we develop several POSSPAN settings to pre-train language models.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Distribution</th>
<th>( F_M )</th>
<th>( F_D )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pois</td>
<td>Poisson</td>
<td>( \lambda = 4 )</td>
<td>( \lambda = 5 )</td>
</tr>
<tr>
<td>Norm</td>
<td>Normal</td>
<td>( \sigma = 1, \mu = 4 )</td>
<td>( \sigma = 1, \mu = 5 )</td>
</tr>
<tr>
<td>Geo</td>
<td>Geometric</td>
<td>( p = 0.2 )</td>
<td>( p = 0.1 )</td>
</tr>
<tr>
<td>Rand</td>
<td>Uniform</td>
<td>( a = 1, b = 5 )</td>
<td>( a = 4, b = 6 )</td>
</tr>
</tbody>
</table>

Table 1: Hyper-parameters of different distributions. We tune hyper-parameters of the distributions via grid search and find the best settings.

3 EXPERIMENTS

In this section, we first introduce the experimental setup. Then, we illustrate experimental results and conduct further discussions.

3.1 Experimental Settings

Datasets. We conduct experiments on four common types of NLU tasks, including named entity recognition (e.g., CoNLL 2003 [19]), sentence pair classification (e.g., MNLI [23], MRPC [6], QNLI [21]), question & answering (e.g., BoolQ [2], COPA [17]), and machine reading comprehension (e.g., ReCoRD [28], SQuAD v2.0 [16], RACE [11]). For space limitation, we omit the details of each dataset. Considering the computational cost and experimental efficiency, we take the popular post-training (i.e., the second-stage pre-training) strategy \([7, 30]\) with POSSPAN instead of pre-training from scratch. We collect the text and remove labels from all training sets for post-training, which is about 1.5M sentences and 250M tokens. For fine-tuning, all experiments were followed the setup in previous works \([8, 15]\).

Baselines. All experiments were conducted with the DeBERTaV3 \([8]\) backbone. The following baselines are compared: (1) DeBERTaV3 is the publicly available model checkpoint without post-training. (2) MLM \([5]\) post-trains DeBERTaV3 with sub-token masking, that is, the span length is 1. (3) Fixed \( \sim \) masks spans of length 4. (4) WWM \([4]\) masks spans of the whole word with several sub-tokens. (5) N-gram \([3]\) conducts masking where 10/20/30/40% of spans are in length of 1/2/3/4. (6) Geo \([10]\) and (7) Pois \([13]\), in which the length of spans is sampled from Geometric and Poisson distribution, respectively. These methods use diverse strategies of span length masking (\( F_M \)'s are different), but the same span position constraint \((F_D \sim \text{poly}(r_m, N))\). For fair comparison, all models are trained with the same post-training and fine-tuning corpora and then evaluated on the same test sets as mentioned above.

Implementation Details. We load the publicly released checkpoint of \texttt{deberta-v3-small} \([8]\) for model initialization. The number of hidden layers and attention heads is 12 and 6, and the hidden size, embedding size, and intermediate size is 384, 384, and 1536, respectively. We first post-train a model for 20 epochs with batch...
Table 2: Experimental results of POSPAN. POSPAN(Geo-Pois) denotes $F_M \sim Geo$ and $F_D \sim Pois$. CoNLL and SQuAD represent ConNLL 2003 and SQuAD v2.0. MNLI (m/mm) represents the two versions of MNLI, MNLI-matched and MNLI-mismatched.

<table>
<thead>
<tr>
<th>Method</th>
<th>CoNLL</th>
<th>MNLI(m/mm)</th>
<th>MRPC</th>
<th>QNLI</th>
<th>BoolQ</th>
<th>COPA</th>
<th>ReCoRD</th>
<th>SQuAD</th>
<th>RACE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeBERTaV3 [8]</td>
<td>94.9</td>
<td>88.1/88.3</td>
<td>87.0</td>
<td>92.4</td>
<td>80.1</td>
<td>70.3</td>
<td>56.5/46.6</td>
<td>84.8/82.0</td>
<td>52.0</td>
</tr>
<tr>
<td>MLM [5]</td>
<td>95.3</td>
<td>88.2/88.5</td>
<td>88.4</td>
<td>92.5</td>
<td>80.5</td>
<td>70.9</td>
<td>56.3/44.9</td>
<td>84.8/82.1</td>
<td>52.1</td>
</tr>
<tr>
<td>Fixed</td>
<td>95.3</td>
<td>88.2/88.6</td>
<td>88.2</td>
<td>92.8</td>
<td>80.6</td>
<td>72.9</td>
<td>56.5/44.9</td>
<td>84.7/82.2</td>
<td>52.2</td>
</tr>
<tr>
<td>N-gram [3]</td>
<td>95.3</td>
<td>88.2/88.5</td>
<td>88.6</td>
<td>93.0</td>
<td>81.2</td>
<td>73.5</td>
<td>56.7/45.2</td>
<td>84.9/82.2</td>
<td>52.4</td>
</tr>
<tr>
<td>WWM [4]</td>
<td>95.2</td>
<td>88.2/88.5</td>
<td>88.0</td>
<td>92.7</td>
<td>80.8</td>
<td>71.8</td>
<td>56.4/44.7</td>
<td>84.8/82.2</td>
<td>52.3</td>
</tr>
<tr>
<td>Geo [10]</td>
<td>95.7</td>
<td>88.5/88.7</td>
<td>88.9</td>
<td>93.1</td>
<td>81.3</td>
<td>73.2</td>
<td>56.8/45.1</td>
<td>85.0/82.5</td>
<td>52.5</td>
</tr>
<tr>
<td>Pois [13]</td>
<td>95.6</td>
<td>88.4/88.7</td>
<td>87.5</td>
<td>93.0</td>
<td>81.0</td>
<td>73.9</td>
<td>56.7/45.1</td>
<td>85.1/82.5</td>
<td>52.3</td>
</tr>
<tr>
<td>POSPAN(WWM-Norm)</td>
<td>95.5</td>
<td>88.3/88.5</td>
<td>88.5</td>
<td>93.1</td>
<td>80.9</td>
<td>73.3</td>
<td>56.9/45.0</td>
<td>84.8/82.3</td>
<td>52.5</td>
</tr>
<tr>
<td>POSPAN(Geo-Pois)</td>
<td>95.9</td>
<td>88.8/89.0</td>
<td>89.2</td>
<td>93.4</td>
<td>81.6</td>
<td>75.7</td>
<td>57.3/45.6</td>
<td>85.4/82.5</td>
<td>52.8</td>
</tr>
<tr>
<td>POSPAN(Pois-Pois)</td>
<td>95.8</td>
<td>88.9/89.3</td>
<td>88.2</td>
<td>93.2</td>
<td>81.9</td>
<td>75.6</td>
<td>57.1/45.3</td>
<td>85.6/82.7</td>
<td>53.1</td>
</tr>
</tbody>
</table>

Figure 1: The model performance of POSPAN with different position constraints ($x$-axis).

3.2 Main Results

Table 2 illustrates the experimental results. It’s observed that: (1) All post-training models can bring further improvements compared to the strong baseline DeBERTaV3, which shows the effectiveness of post-training. (2) Compared with single-token masking, all span-level masking methods yield substantial improvements, which indicates the advantage of span-level masking on capturing the critical semantics of language. (3) Our proposed POSPAN obtained the best performance across different tasks. Specially, POSPAN surpassed the previous best baselines by 1.8%, 0.9%, 0.6% on COPA, BoolQ, RACE respectively. It demonstrates the superiority and necessity of position constraint for span masking.

Previous research [14] has shown that the masking probability can introduce a type of prior knowledge for language models. We conjecture that POSPAN also introduces $F_M$ and $F_D$ as two kinds of prior knowledge, where $F_M$ can help the model capture n-gram-level sub-structures [3, 10, 25] and $F_D$ is capable of catching the semantic dependencies among spans. Besides, the theoretical analysis in Section 2.2 also proves the necessity of both span length distribution and position constraint distribution for language model pre-training, which explains why POSPAN works.

3.3 Discussions

To investigate the position constraint in POSPAN, we combine WWM, Geo, and Pois with various $F_D$s. Figure 1 shows the model performance under different $F_D$s on the GLUE benchmark [21], and “poly/n” represents the span-length-only counterparts. It shows that: (1) Position constraint distribution has a prominent impact on the model performance. For example, POSPAN(Geo-Pois) improves the Pois by 0.6%, while POSPAN(Geo-Geo) decreases the score of Geo by 0.6%. (2) POSPAN with $F_D \sim Geo$ consistently harms the original span masking methods, while other $F_D$s distributions boost the performance of original span masking by different extent.

The above results corroborate our hypothesis, that is, span positions are indeed not independent from each other, and utilizing $F_D$ to represent dependent span positions and model contextual dependency is beneficial to downstream tasks universally. Mathematically speaking, when sampling span positions randomly, the distribution $poly/n(r_m, N)$ promotes small inter-span distances, which prevents span dependency modeling, thus hinders the learning of contextual semantics. Similarly, $F_D \sim Geo$ also promotes small inter-span distances and results in unsatisfying performance. Empirically, the discrete distribution Pois performs relatively better than other continuous distributions. Such performance bias caused by distributions might indicate the discrete and flexible length of dependency in natural language.

4 CONCLUSION AND FUTURE WORK

In this paper, we propose POSPAN, a novel position-constrained span masking method for language model pre-training. POSPAN leverages span length and position constraint distributions to mask tokens, and works as a general framework to unify existing span-level masking methods. Extensive experiments are conducted to verify the effectiveness of POSPAN on various NLU tasks. Moreover, the theoretical analysis reveals the rationality and necessity of both span length distribution and position constraint distribution, which encourages language models to learn span-level semantics and their contextual dependencies. For the future work, we will explore more effective masking strategies by designing better $F_M$s and $F_D$s.