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
zhaoyuming3@jd.com

Meng Chen
JD AI Research
chenmeng20@jd.com

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

ABSTRACT

Span-level masked language modeling (MLM) has shown to be advantageous to pre-trained language models over the original single-token MLM, as entities/phrases and their dependencies are critical to language understanding. Previous works only consider span length with some discrete distributions, while the dependencies among spans are ignored, i.e., assuming that the positions of masked spans are uniformly distributed. In this paper, we present POSPAN, a general framework to allow diverse position-constrained span masking strategies via the combination of span length distribution and position constraint distribution, which unifies all existing span-level masking methods. To verify the effectiveness of POSPAN in pre-training, we evaluate it on the datasets from several NLU benchmarks. Experimental results indicate that the position constraint is capable of enhancing span-level masking broadly, and our best POSPAN setting consistently outperforms its span-length-only counterparts and vanilla MLM. We also conduct theoretical analysis for the position constraint in masked language models to shed light on the reason why POSPAN works well, demonstrating the rationality and necessity of POSPAN.

CCS CONCEPTS

• **Computing methodologies** → **Lexical semantics**; **Natural language processing**.

KEYWORDS

Masked language modeling, pre-training, span length distribution, position constraint distribution

ACM Reference Format:
Zhenyu Zhang, Lei Shen, Yuming Zhao, Meng Chen, and Xiaodong He. 2023. POSPAN: Position-Constrained Span Masking for Language Model Pre-training. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3583780.3615197

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 **PO**sition-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_Ms are different while F_Ds are derived from the same uniform distribution. To verify the effectiveness of POSPAN for language understanding, we evaluate PLMs with POSPAN on various tasks

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10…$15.00
https://doi.org/10.1145/3583780.3615197

Algorithm 1: Position-Constrained Span Masking

```
Input: $N$, $r_m$, $T$, $F_M$, $F_D$
Output: $T_m$
// masking rate: $r_m$, input tokens: $T$, input token length: $N$
// masked tokens: $T_m$
// initialize vectors: $m$, $d$, $T_m$
$M$=vector($N$), $D$=vector($N$), $T_m$=copy($T$);
for $i$=1 → $N$ do
    sample span length $len_i \sim F_M$ ;
    sample position distance $d_i \sim F_D$ ;
    $M$[i]=$len_i$, $D$[i]=$d_i$;
spans=List(), pos=1, start=0, do_mask=False;
while ($pos < N$) do
    start=pos ;
    if(do_mask==False) then pos+=$D$[pos]
    else pos+=$M$[pos] ;
    do_mask=¬do_mask ;
    if(do_mask==True) spans.add(pair<start, pos-start>) ;
while(sum(spans.second) > $r_m N$) random.pop_one(spans) ;
for span in spans do
    set_value($T_m$, span, [MASK]) ;
```

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, $S = \{S_1, S_2, ..., S_m\}$, the length and position of the i-th span are denoted as len_i and pos_i (for simplicity, we use i to represent the full notation of i-th span, S_i), we formulate these factors under two distributions:

$$len_i \sim F_M,$$
$$|pos_{i+1} - pos_i| \sim F_D,\ \ 0 \leq pos_i < pos_{i+1} \leq N,$$

where F_M is the span length distribution, and $|pos_{i+1} - pos_i|$, 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 mainly 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 $\hat{F_D}$:

$$P(|pos_i - pos_j| \leq d) = \hat{F_D}(d). \tag{1}$$

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

$$\begin{aligned}\hat{F_D}(d) = \frac{1}{c}\bigg(&\int_0^d (pos_j + d)\mathrm{d}pos_j \\ &+ \int_{N-d}^{N}\Big(N - (pos_j - d)\Big)\mathrm{d}pos_j \\ &+ \int_{d}^{N-d}\Big(pos_j + d - (pos_j - d)\Big)\mathrm{d}pos_j\bigg) \\ = &\ \frac{2Nd - d^2}{c},\end{aligned} \tag{2}$$

where $c = (1 - r_m^2)N^2$ 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 $\hat{F_D}(d)$, the position constraint of two consecutive spans follows another polynomial distribution, denoted as $polyn(r_m, N)$:

$$\begin{aligned}polyn(r_m, N) \sim F_D = P(pos_{i+1} - pos_i \leq \frac{d}{|S| - 1}) \\ \approx \hat{F_D}(d)\end{aligned} \tag{3}$$

It is clear that F_D of existing methods is not fully considered, and we will explore the combination of different F_Ms and F_Ds 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 $len_i = |S_i|$ tokens in S_i and $len_j = |S_j|$ 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**: $S_i \longrightarrow S_j$, i.e., S_i is the premise of S_j. When S_i appears, S_j will appear most of the time.
- **Case 3**: $S_j \longrightarrow S_i$, i.e., S_j is the premise of S_i.

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

$$\begin{aligned}P(S_i, S_j|R_{ij}) &= \frac{P(R_{ij}|S_i, S_j) * P(S_i, S_j)}{P(R_{ij})}, \\ \log P(S_i, S_j|R_{ij}) &\propto \underbrace{\log P(R_{ij}|S_i, S_j)}_{①} + \underbrace{\log P(S_i, S_j)}_{②},\end{aligned} \tag{4}$$

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 $P(x_i, ..., x_{i+len_i-1}, x_j, ..., x_{j+len_j-1})$. Then, to maximize the log-likelihood of M masked spans in training, we have:

$$\begin{aligned}&\sum_{i,j}\log P(S_i, S_j) = \frac{(M-1)\log P(S_1, S_2, ..., S_M)}{2} \\ &\propto \sum_{i=1}^{M}\log P(S_i). \\ &\mathcal{L}_S :\longrightarrow max(\mathbb{E}[\log P(S_i|len_i)]),\ \ where \\ &\mathbb{E}[\log P(S_i|len_i)] = \mathbb{E}_{len_i\sim F_M}(\sum_{l=0}^{len_i-1}\log P(x_{i+l})).\end{aligned} \tag{5}$$

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 $\mathcal{L}_S$, where span length $len_i \sim F_M$, and ":⟶" means the goal of $\mathcal{L}_S$.

Previous works only focus on the second term in Equation (4) with various F_Ms, 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:

$$\begin{aligned} P(S_i) &= P(x_{pos_i-1}, x_{pos_i+len_i}, pos_i) \\ &\approx P(x_{pos_i-1}, x_{pos_i+len_i}, x_{\hat{pos}_i}), \end{aligned} \tag{6}$$

where pos_i is the position of S_i. Since we mask tokens in span S_i, the masked token $x_{\hat{pos}_i}$ 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., $I_{ij} = \{x_{pos_j-d}, ..., x_{pos_j-1}\}$. Then, R_{ij} is inferred by:

$$P(R_{ij}|d) = P(R_{ij}|S_i, S_j, I_{ij}). \tag{7}$$

We assume that given R_{ij}, unmasked tokens in I_{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):

$$P(R_{ij}|S_i, S_j) \propto P(R_{ij}|d). \tag{8}$$

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

$$\begin{aligned} &\mathcal{L} = \mathcal{L}_R + \mathcal{L}_S, \\ &\mathcal{L}_R :\longrightarrow max(\mathbb{E}[\log P(R_{ij}|F_D)]), \end{aligned} \tag{9}$$

where $\mathcal{L}_S$ is span length loss from Equation (5) and $\mathcal{L}_R$ is span dependency loss. The span length and position constraint are controlled by the prior distributions F_M and F_D, respectively, i.e., $len_i \sim F_M$ 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 POSPAN Algorithm

Various masking strategies can be achieved via the combination of different F_Ms and F_Ds. To investigate the impact of different masking strategies conveniently, we illustrate the sampling algorithm of POSPAN 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 "[MASK]" (Line 17-18).

We follow previous works [5, 15] to mask 15% of tokens ($r_m = 0.15$), where 80% of them are replaced with "[MASK]", 10% are replaced with tokens randomly sampled from the vocabulary, and the rest 10% are untouched. Through POSPAN, we can re-implement all the previous masking methods and design new masking strategies easily based on various distributions, including Normal (*Norm*), Geometric (*Geo*), Uniform (*Rand*) and Poisson (*Pois*) distribution. Table 1 illustrates the hyper-parameters of distributions we investigate, where the mean 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_Ms and F_Ds, we develop several POSPAN settings to pre-train language models.

Notation	Distribution	F_M	F_D
Pois	Poisson	$\lambda = 4$	$\lambda = 5$
Norm	Normal	σ=1,μ=4	σ=1,μ=5
Geo	Geometric	p=0.2	p=0.1
Rand	Uniform	a=1,b=5	a=4,b=6

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 POSPAN 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** 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_Ms are different), but the same span position constraint ($F_D \sim polyn(r_m, N)$). 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 of *deberta-v3-xsmall* [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

Method	CoNLL	MNLI(m/mm)	MRPC	QNLI	BoolQ	COPA	ReCoRD	SQuAD	RACE
DeBERTaV3 [8]	94.9	88.1/88.3	87.0	92.4	80.1	70.3	56.5/44.6	84.8/82.0	52.0
MLM [5]	95.3	88.2/88.5	88.4	92.5	80.5	70.9	56.3/44.9	84.8/82.1	52.1
Fixed	95.3	88.2/88.6	88.2	92.8	80.6	72.9	56.5/44.9	84.7/82.2	52.2
N-gram [3]	95.3	88.2/88.5	88.6	93.0	81.2	73.5	56.7/45.2	84.9/82.2	52.4
WWM [4]	95.2	88.2/88.5	88.0	92.7	80.8	71.8	56.4/44.7	84.8/82.2	52.3
Geo [10]	95.7	88.5/88.7	88.9	93.1	81.3	73.2	56.8/45.1	85.0/82.5	52.5
Pois [13]	95.6	88.4/88.7	87.5	93.0	81.0	73.9	56.7/45.1	85.1/82.5	52.3
POSPAN(WWM-*Norm*)	95.5	88.3/88.5	88.5	93.1	80.9	73.3	56.9/45.0	84.8/82.3	52.5
POSPAN(*Geo-Pois*)	**95.9**	88.8/89.0	**89.2**	**93.4**	81.6	**75.7**	**57.3/45.6**	85.4/82.5	52.8
POSPAN(*Pois-Pois*)	95.8	**88.9/89.3**	88.2	93.2	**81.9**	75.6	57.1/45.3	**85.6/82.7**	**53.1**

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.

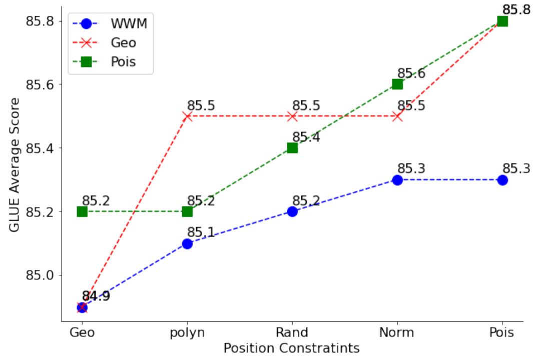

Figure 1: The model performance of POSPAN with different position constraints (x-axis).

size of 512, 5K warm-up steps, LAMB optimizer [27], peak learning rate of 1e-4 with linear scheduler. Then we follow the setups in He et al. [8, 9] to fine-tune the models with an extra classification or regression layer for each downstream dataset. Each model is fine-tuned for 10 epochs with batch size of {16,32} and learning rate of {1e-5,2e-5} with linear scheduler. We implement each model using Huggingface Transformers [24], report the average score of 10 runs on each dataset, and show the best result in bold with wilcoxon test ($p < 0.05$) [22].

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_Ds. Figure 1 shows the model performance under different F_Ds on the GLUE benchmark [21], and "*polyn*" represents the span-length-only counterparts. It shows that: (1) Position constraint distribution has a prominent impact on the model performance. For example, POSPAN(*Pois-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 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 $polyn(r_m, N)$ 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_Ms and F_Ds.

REFERENCES

[1] Stephane Aroca-Ouellette and Frank Rudzicz. 2020. On Losses for Modern Language Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, 4970–4981. https://doi.org/10.18653/v1/2020.emnlp-main.403

[2] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044* (2019).

[3] Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting Pre-Trained Models for Chinese Natural Language Processing. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020 (Findings of ACL, Vol. EMNLP 2020)*, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, 657–668. https://doi.org/10.18653/v1/2020.findings-emnlp.58

[4] Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang. 2021. Pre-Training with Whole Word Masking for Chinese BERT. *IEEE Transactions on Audio, Speech and Language Processing*. https://doi.org/10.1109/TASLP.2021.3124365

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, 4171–4186.

[6] Bill Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Third International Workshop on Paraphrasing (IWP2005)*.

[7] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 8342–8360.

[8] Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. arXiv:2111.09543 [cs.CL]

[9] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DEBERTA: DECODING-ENHANCED BERT WITH DISENTANGLED ATTENTION. In *International Conference on Learning Representations*. https://openreview.net/forum?id=XPZIaotutsD

[10] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving Pre-training by Representing and Predicting Spans. *Trans. Assoc. Comput. Linguistics* 8 (2020), 64–77. https://transacl.org/ojs/index.php/tacl/article/view/1853

[11] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding Comprehension Dataset From Examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, 785–794. https://doi.org/10.18653/v1/D17-1082

[12] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

[13] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (Eds.). Association for Computational Linguistics, 7871–7880. https://doi.org/10.18653/v1/2020.acl-main.703

[14] Yi Liao, Xin Jiang, and Qun Liu. 2020. Probabilistically Masked Language Model Capable of Autoregressive Generation in Arbitrary Word Order. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (Eds.). Association for Computational Linguistics, 263–274. https://doi.org/10.18653/v1/2020.acl-main.24

[15] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR* abs/1907.11692 (2019). arXiv:1907.11692

[16] Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. https://doi.org/10.48550/ARXIV.1806.03822

[17] Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of Plausible Alternatives: An Evaluation of Commonsense Causal Reasoning.. In *AAAI spring symposium: logical formalizations of commonsense reasoning*. 90–95.

[18] Yu Sun, Shuohuan Wang, Yu-Kun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. ERNIE 2.0: A Continual Pre-Training Framework for Language Understanding. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 8968–8975. https://aaai.org/ojs/index.php/AAAI/article/view/6428

[19] Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*. 142–147. https://www.aclweb.org/anthology/W03-0419

[20] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. *arXiv preprint arXiv:1905.00537* (2019).

[21] Alex Wang, Amanpreet 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 *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

[22] Frank Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometrics bulletin* 1, 6 (1945), 80–83.

[23] Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426* (2017).

[24] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Online, 38–45.

[25] Dongling Xiao, Yu-Kun Li, Han Zhang, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2021. ERNIE-Gram: Pre-Training with Explicitly N-Gram Masked Language Modeling for Natural Language Understanding. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (Eds.). Association for Computational Linguistics, 1702–1715. https://doi.org/10.18653/v1/2021.naacl-main.136

[26] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. 5754–5764.

[27] Yang You, Jing Li, Jonathan Hseu, Xiaodan Song, James Demmel, and Cho-Jui Hsieh. 2019. Reducing BERT Pre-Training Time from 3 Days to 76 Minutes. *CoRR* abs/1904.00962 (2019).

[28] Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. Record: Bridging the gap between human and machine commonsense reading comprehension. *arXiv preprint arXiv:1810.12885* (2018).

[29] Zhenyu Zhang, Tao Guo, and Meng Chen. 2021. DialogueBERT: A Self-Supervised Learning based Dialogue Pre-training Encoder. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, Gianluca Demartini, Guido Zuccon, J. Shane Culpepper, Zi Huang, and Hanghang Tong (Eds.). ACM, 3647–3651. https://doi.org/10.1145/3459637.3482085

[30] Zhenyu Zhang, Lei Shen, Yuming Zhao, Meng Chen, and Xiaodong He. 2023. Dialog-Post: Multi-Level Self-Supervised Objectives and Hierarchical Model for Dialogue Post-Training. *The 61st Annual Meeting of the Association for Computational Linguistics (ACL 2023)* (2023).