Pre-Training a Graph Recurrent Network for Language Representation

Yile Wang[1], **Linyi Yang**[1], **Zhiyang Teng**[1], **Ming Zhou**[2], **Yue Zhang**[1]
[1]Westlake University
[2]Langboat Technology, Beijing, China
{wangyile,yanglinyi,tengzhiyang,zhangyue}@westlake.edu.cn
zhouming@chuangxin.com

Abstract

Transformer-based models have gained much advance in recent years, becoming one of the most important backbones in natural language processing. Recent work shows that the attention mechanism in Transformer may not be necessary, both convolutional neural networks and multi-layer perceptron based models have been investigated as Transformer alternatives. In this paper, we consider a graph recurrent network for language model pre-training, which builds a graph structure for each sequence with local token-level communications, together with a sentence-level representation decoupled from other tokens. We find such architecture can give comparable results against Transformer-based ones in both English and Chinese language benchmarks. Moreover, instead of the quadratic complexity, our model has linear complexity and performs more efficiently during inference.[1]

1 Introduction

Pre-trained models (PTMs) [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] have been widely used in natural language processing (NLP), benefiting a range of tasks including language understanding [11, 12], question answering [13, 14], and dialogue [15, 16]. The dominant methods take the Transformer [17] architecture, a heavily engineered model based on a self-attention network (SAN), it also showing competitive performance in computation vision [18, 19, 20], speech [21], and biological [22] tasks.

Despite its success, Transformer typically suffers from quadratic time complexity [17], along with the requirement of large computational resources and associated financial and environmental costs [23]. In addition, recent studies show that the attention mechanism, which is the key ingredient of Transformer, may not be necessary [24, 25, 26]. For example, Tay et al. [26] find that models learning synthetic attention weights without token-token interactions also achieve competitive performance for certain tasks. Therefore, investigation of Transformer alternatives is of both theoretical and practical interest. To this end, various non-Transformer PTMs have recently been proposed [27, 28, 29, 30].

In this paper, we consider a graph neural network (GNN) [31] for language model pre-training. GNN and its variants have been widely used in NLP tasks, including machine translation [32], information extraction [33], and sentiment analysis [34]. For GNN language modeling, a key problem is how to represent a sentence in a graph structure. From this perspective, ConvSeq2seq [35] can be regarded as a graph convolutional network (GCN) [36] with node connections inside a local kernel. Transformer-based models can be regarded as a graph attention network (GAT) [37] with a full node connection. However, graph recurrent network (GRN) [38, 39] models have been relatively little considered.

We follow the structure of sentence-state LSTM (S-LSTM) [38], which represents a sentence using a graph structure by treating each word as a node, together with a sentence state node. State transitions

[1]We release the code at `https://github.com/ylwangy/slstm_pytorch`.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Type	Models	Basic Unit	Complexity	Parallel	Parameter Sharing
LSTM-based	Context2Vec [40]	RNN	$\mathcal{O}(n)$	✗	✗
	ELMo [41]			✗	✗
Transformer-based	GPT2 [1]	SAN	$\mathcal{O}(n^2)$	✓	✗
	BERT [2]			✓	✗
	RoBERTa [3]			✓	✗
	XLNet [4]			✓	✗
	ALBERT [5]			✓	✓
	BART [6]			✓	✗
	T5 [7]			✓	✗
	DeBERTa [8]			✓	✗
Others	DynamicConv [27]	CNN	$\mathcal{O}(n)$	✓	✗
	gMLP [28]	MLP	$\mathcal{O}(n)$	✓	✗
	Ours	GRN	$\mathcal{O}(n)$	✓	✓

Table 1: Overview of existing types of pre-trained models and our proposed model.

are performed recurrently to allow token nodes to exchange information with their neighbors and the sentence-level node. Such architecture has shown advantages over vanilla bidirectional LSTM in supervised text classification tasks. However, its potential for general-purpose language model pre-training has not been fully exploited. We optimize the model by exploring the suitable architecture design for pre-training, a comparison of our model and typical existing PTMs is shown in Table 1.

Experimental results show that our model can give a comparable performance on general language understanding tasks for both English and Chinese languages. During inference, our model can gain 2∼3 times speedup or more for extra long sentences against Transformer-based models. To our knowledge, we are the first to investigate a graph recurrent network for language model pre-training.

2 Model

The overall structure of our model is shown in Figure 1(a). Following S-LSTM [38], we treat each sentence as a graph with token nodes and an external sentence state node. The node state is updated in parallel according to the information received in each layer (or recurrent step).

We first transform each token w_i into token embedding using the trainable lookup table E and the position embedding lookup table P, the model input x_i is constructed by $x_i = E(w_i) + P(w_i)$. Then we initialize hidden states and hidden cells for each token node, the sentence-level node with $h_1^0, h_2^0, ..., h_n^0, g^0$ and $c_1^0, c_2^0, ..., c_n^0, c_g^0$, respectively. In each layer t ($t = 1, 2, ..., L$), the token node states h_i^t is calculated using gating mechanism similar with LSTM:

$$\begin{aligned}
&\xi_i^{t-1} = h_{i-1}^{t-1} \parallel h_i^{t-1} \parallel h_{i+1}^{t-1} \\
&\hat{i}_i^t = \sigma(\mathrm{LayerNorm}(W_i\xi_i^{t-1} + U_i x_i + V_i g^{t-1} + b_i)) \\
&\hat{l}_i^t = \sigma(\mathrm{LayerNorm}(W_l\xi_i^{t-1} + U_l x_i + V_l g^{t-1} + b_l)) \\
&\hat{r}_i^t = \sigma(\mathrm{LayerNorm}(W_r\xi_i^{t-1} + U_r x_i + V_r g^{t-1} + b_r)) \\
&\hat{f}_i^t = \sigma(\mathrm{LayerNorm}(W_f\xi_i^{t-1} + U_f x_i + V_f g^{t-1} + b_f)) \\
&\hat{s}_i^t = \sigma(\mathrm{LayerNorm}(W_s\xi_i^{t-1} + U_s x_i + V_s g^{t-1} + b_s)) \\
&o_i^t = \sigma(\mathrm{LayerNorm}(W_o\xi_i^{t-1} + U_o x_i + V_o g^{t-1} + b_o)) \\
&u_i^t = \tanh(\mathrm{LayerNorm}(W_u\xi_i^{t-1} + U_u x_i + V_u g^{t-1} + b_u)) \\
&i_i^t, l_i^t, r_i^t, f_i^t, s_i^t = \mathrm{softmax}(\hat{i}_i^t, \hat{l}_i^t, \hat{r}_i^t, \hat{f}_i^t, \hat{s}_i^t) \\
&c_i^t = l_i^t \odot c_{i-1}^{t-1} + f_i^t \odot c_i^{t-1} + r_i^t \odot c_{i+1}^{t-1} + s_i^t \odot c_g^{t-1} + i_i^t \odot u_i^t \\
&h_i^t = o_i^t \odot \tanh(c_i^t)
\end{aligned} \tag{1}$$

where $\parallel$ is concatenation operation, ξ_i^{t-1}, x_i, g^{t-1} represent the inputs from previous local states, token embedding and previous global states, respectively. In Eq. 1, we calculate multiple LSTM-style gates to control the corresponding information flow. $\hat{l}_i^t, \hat{r}_i^t, \hat{f}_i^t, \hat{s}_i^t$ are the forget gates with respect to the left token cell c_{i-1}^{t-1}, right token cell c_{i+1}^{t-1}, current token cell c_i^{t-1}, and sentence cell c_g^{t-1}. $\hat{i}_i^t$, o_i^t are

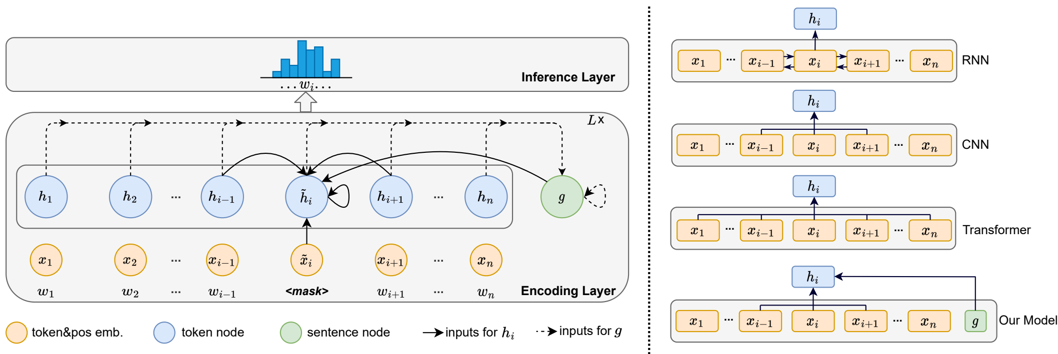

Figure 1: Left: (a) Architecture of our model. We only show the update of token node h_i and sentence-level node g for brevity. Right: (b) Comparison with different model architectures.

the input gate and output gate, respectively. Layer normalization is used to control the distributions of neurons in each gate. W_x, U_x, V_x, and b_x ($x \in \{i, l, r, f, s, o, u\}$) are model parameters.

The sentence-level node g^t takes the previous token state as inputs and is calculated by :

$$
\begin{aligned}
\overline{h} &= \mathrm{avg}(h_1^{t-1}, h_2^{t-1}, ..., h_n^{t-1}) \\
\hat{f}_i^t &= \sigma(\mathrm{LayerNorm}(W_f g^{t-1} + U_f h_i^{t-1} + b_f)) \\
\hat{f}_g^t &= \sigma(\mathrm{LayerNorm}(W_g g^{t-1} + U_g \overline{h} + b_g)) \\
o^t &= \sigma(\mathrm{LayerNorm}(W_o g^{t-1} + U_o \overline{h} + b_o)) \\
f_1^t, ..., f_n^t, f_g^t &= \mathrm{softmax}(\hat{f}_1^t, ..., \hat{f}_n^t, \hat{f}_g^t) \\
c_g^t &= f_g^t \odot c_g^{t-1} + \sum f_i^t \odot c_i^{t-1} \\
g^t &= o^t \odot \tanh(c_g^t)
\end{aligned}
\tag{2}
$$

where $\hat{f}_1^t, ..., \hat{f}_n^t, \hat{f}_g^t$ are the forget gates with respect to token cells $c_1^{t-1}, ..., c_n^{t-1}$ and sentence cell c_g^{t-1}, o^t is the output gate. W_x, U_x, and b_x ($x \in \{g, f, o\}$) are model parameters. The generated hidden states h_i^t and g^t are sent to the next layer, together with the memory state c_i^t and c_g^t.

Figure 1(b) shows the ways of hidden states generations of our model and other architectures. Different from CNN, we explicitly model sentence-level information as a feature for each token, which provides global information. Compared with Transformer, the sentence-level node representation is designed to be separated from other tokens. We make all the trainable parameters in Eq. 1 and Eq. 2 shared across GNN layers, which is similar to the parameters in LSTM along the sequence direction. In our model, we update each token using its fixed local context together with a sentence-level representation in each layer, making our model has $\mathcal{O}(n)$ complexity.

3 Experiments

3.1 Pre-training

Dataset. English models are trained using the latest Wikipedia and BookCorpus [42]. Chinese models are trained using Wikipedia. The total amount of training data is on par with BERT [2] for both languages.

Baselines. Strictly comparing the PTMs is difficult because of the different dataset processing, training strategies, and environmental settings. As shown in Table 2, we consider the most related and popular models with similar training corpus for comparison. For English, we use the published RNN-based model (ELMo), compact version of BERT (DistilBERT), BERT, and recurrent version of BERT (ALBERT). For Chinese, we add some BERT variants which use Chinese word segmentor for whole word masking [44] or modify the masked token prediction as a correction target [46].

Settings. We pre-train our model with a batch size of 128 and a maximum length of 512 for 300k steps, using Adam optimizer with learning rate lr=0.003, β_1=0.9, β_2=0.98, learning rate warmup

Models (English)	Pre-training Data	Objective
ELMo [41]	1 Billion Word	CLM
DistilBERT [43]	Wiki+BooksCorpus	BERT+KD
BERT-base [2]	Wiki+BooksCorpus	MLM+NSP
ALBERT-base [5]	Wiki+BooksCorpus	MLM+SOP
Models (Chinese)	**Data**	**Objective**
BERT-base [2]	Wiki	MLM+NSP
BERT-wwm [44]	Wiki	MLM+NSP
BERT-wwm-ext [44]	Wiki+EXT	MLM+NSP
RoBERTa-wwm [45]	Wiki+CLUECorpus	MLM
RoBERTa-wwm-ext [44]	Wiki+EXT	MLM
ALBERT-large [5]	Wiki+EXT	MLM+SOP
MacBERT [46]	Wiki+EXT	Mac+SOP

Table 2: Baseline models. CLM: casual language modeling. KD: knowledge distillation. SOP: sentence order prediction. Mac: MLM as correction. wwm: whole word masking. ext: external training data.

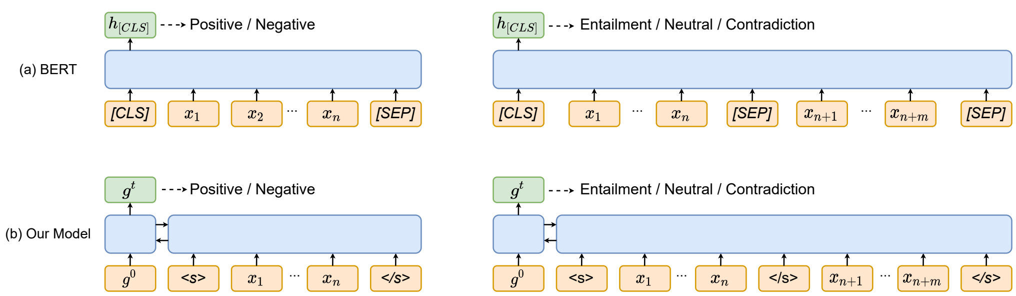

Figure 2: Illustrations of fine-tuning BERT and our model on different tasks. Left: single sentence tasks (e.g., sentiment analysis); Right: sentence pair tasks (e.g., natural language inference).

over the first 3,000 steps, and linear decay with 0.03. Both English and Chinese model has 10 layers, and 1792 hidden size, where the total parameter size is 186M. We use 8 NVIDIA GeForce RTX 3090 GPUs for pre-training and it takes around 10 days. All the pre-training implementations are based on FairSeq [47] framework.

3.2 Fine-tuning

Evaluating Benchmarks. For English tasks, we evaluate our pre-trained models on tasks in GLUE [11], including linguistic acceptability (CoLA), sentiment analysis (SST), sentence pair similarity (MRPC, QQP), and natural language inference (MNLI, QNLI, RTE).

For Chinese tasks, we evaluate on short and long text classification (TNEWS, IFLYTEK), keywords matching (CSL) [48], question matching (LCQMC) [49], document classification (THUCNews) [50] and sentiment analysis (ChnSentiCorp) [51].

Settings. Although our model architecture is different from most baselines, the fine-tuning strategy for each task can be the same as BERT-style models. As shown in Figure 2, the output of the sentence node g can be treated as the representation of [CLS] in BERT, which can be used for single sentence classification tasks directly. For sentence pair classification tasks, we concatenated two sentences and the target label is still predicted using the sentence node g in the last layer.

We use the official code from Huggingface [52] and CLUE [48] for reproducing the baseline and our results without external data augmentation, we mainly tune the parameters with training epochs in {2, 3, 5, 10}, learning rate in {2e-5, 3e-5, 5e-5} and batch size in {16, 32, 64}.

3.3 Results

The main results on English and Chinese language understanding tasks are shown in Table 3. For English tasks, our model gives an average score of 78.67, which is higher than the ELMo (69.65) and

Model (English)	**CoLA**	**SST2**	**MRPC**	**QQP**	**MNLI**	**QNLI**	**RTE**	*Avg.*
ELMo [41]	44.1	91.5	70.8	88.0	68.6	71.2	53.4	69.65
DistilBERT [43]	51.3	91.3	82.7	88.5	82.2	89.2	59.9	77.87
BERT-base [2]	**56.3**	**91.7**	83.5	**89.6**	**84.0**	**90.9**	65.3	**80.18**
ALBERT-base [5]	48.2	90.7	**87.2**	88.2	82.3	90.1	**69.7**	79.48
Ours	55.3	90.3	81.0	88.8	81.4	89.6	64.3	78.67
Model (Chinese)	**TNEWS**	**IFLYTEK**	**CSL**	**LCQMC**	**THUCNews**	**ChnSentiCorp**		*Avg.*
BERT-base [2]	56.14	59.67	81.40	87.89	95.35	92.58		78.83
BERT-wwm [44]	56.47	59.71	81.23	87.93	95.28	93.00		78.93
BERT-wwm-ext† [44]	57.35	59.90	80.86	88.05	95.43	93.00		79.09
RoBERTa-wwm† [45]	57.29	59.29	81.16	88.41	95.19	93.25		79.09
RoBERTa-wwm-ext† [44]	57.09	**60.71**	81.80	88.68	95.69	**93.33**		**79.55**
ALBERT-large [5]	55.69	58.36	80.46	88.27	93.52	91.25		77.92
MacBERT† [46]	57.50	59.36	**81.83**	**89.18**	**95.74**	**93.33**		79.49
Ours	**57.56**	60.10	80.73	86.06	95.17	93.08		78.78

Table 3: Results on GLUE and CLUE benchmark dev sets. Results are reported by matthews correlation (for CoLA) or accuracy (for others).

Model	**Settings**	**#Param.**	**Len=64**	**Len=256**	**Len=384**	**Len=512**
ELMo	2 Bi-LSTM layer	93M	0.109	0.381	0.564	0.745
DistilBERT	6 encoder layers	66M	0.016	0.021	0.025	0.034
RoBERTa-base	12 encoder layers	125M	0.017	0.026	0.042	0.051
BART-base	6 encoder & decoder layers	140M	0.018	0.033	0.047	0.063
Ours	6 layers, 1280 hidden size	107M	0.010 (1.7×)	0.010 (2.6×)	0.010 (4.2×)	0.011 (4.6×)
	12 layers, 1280 hidden size	107M	0.018 (0.9×)	0.018 (1.4×)	0.019 (2.2×)	0.019 (2.7×)
	6 layers, 2048 hidden size	238M	0.011 (1.5×)	0.011 (2.4×)	0.012 (3.5×)	0.013 (3.9×)
	12 layers, 2048 hidden size	238M	0.020 (0.9×)	0.020 (1.3×)	0.021 (2.0×)	0.021 (2.4×)

Table 4: Time cost (seconds) during inference for different architectures. Numbers in the parentheses denote the speedup compared to RoBERT-base.

on par with Transformer-based baselines (77.87∼80.18). Compared with Transformer-based models, our results on tasks such as CoLA, QQP, QNLI, and RTE exceeds DistilBERT, being close to BERT (within average 1.0 point). Overall, our model compares well to ALBERT and BERT, retaining 99% and 98% of the performance, respectively. For Chinese tasks, our model gives comparable results with BERT (within 0.05 points of accuracy) and slightly better than ALBERT (78.78 vs. 77.92), which uses the same amount of training corpus. Compared with other models which apply more pre-training data, our model also performs well in tasks such as TNEWS and IFLYTEK.

3.4 Analysis of Efficiency

We compare the inference speed of our model with different architectures in Table 4. ELMo gives the lowest results as the sequential nature of RNN structure. For Transformer-based models, DistilBERT shows the minimum time cost because of the lightweight architecture, BART is slower than RoBERTa due to the nonparallel computation in the decoder. All the models take much more time when the sequence becomes much longer. For example, sequences with a length of 512 need about 3 times more computational time than sequences with a length of 64. For our model, adding the recurrent layer and the hidden size will both leads to more inference time. However, by increasing the sequence length, the inference cost grows much slower than the baselines when the sequence length reaches 256 or more, our model can give a 2∼3 times speedup than DistilBERT or RoBERTa, even for large model settings.

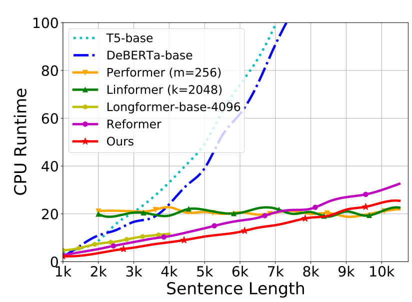

Figure 3: Comparison between models for computing long sequences.

We compared our model with Transformer variants for an extra long sequence in Figure 3. Longformer [53] and Reformer [54] give almost linear growth of runtime w.r.t sequence length. Our model is the fastest when the sequence length is below 8.5k. Linformer [55] and Performer [56] give slightly faster speed when the size reaches 10k. However, the models are particularly designed for long sequences. For example, Linformer project the full self-attention and find the low-rank representation, reducing the complexity from $\mathcal{O}(n^2)$ to $\mathcal{O}(nk)$, thus the projection dimension k should be pre-defined and less than the sequence length n. Similarly, Performer pre-defined kernel feature numbers m and reduce the complexity from $\mathcal{O}(n^2)$ to $\mathcal{O}(nm)$, the most computational efficiency is achieved only when n is relatively large. Overall, our model can handle both short and long sequences friendly.

4 Conclusion

We investigated a graph recurrent network for large-scale language model pre-training. Our model does not rely on the self-attention mechanism and retaining linear computational complexity with respect to the sequence length. Results show that the inference cost can be largely reduced while without much accuracy loss. For future work, we will study our model for seq2seq-style pre-training as in BART or T5, exploring the applications to generation tasks such as machine translation.

Acknowledgments

We would like to thank reviewers for their valuable insights and helpful advice. The work is funded by the Zhejiang Province Key Project 2022SDXHDX0003. Yue Zhang is the corresponding author.

References

[1] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL*, 2019.

[3] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv e-prints*, page arXiv:1907.11692, July 2019.

[4] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in NeurIPS*, 2019.

[5] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite BERT for self-supervised learning of language representations. In *Proc. of ICLR*, 2020.

[6] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proc. of ACL*, 2020.

[7] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[8] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: decoding-enhanced bert with disentangled attention. In *Proc. of ICLR*, 2021.

[9] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *Proc. of ICLR*, 2020.

[10] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott

Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in NeurIPS*, 2020.

[11] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proc. of EMNLP Workshop*, 2018.

[12] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in NeurIPS*, 2019.

[13] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proc. of EMNLP*, 2016.

[14] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *Proc. of ACL*, 2018.

[15] Siqi Bao, Huang He, Fan Wang, Hua Wu, and Haifeng Wang. PLATO: Pre-trained dialogue generation model with discrete latent variable. In *Proc. of ACLs*, 2020.

[16] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in NeurIPS*, 2017.

[18] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proc. of ECCV*, 2020.

[19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *Proc. of ICLR*, 2021.

[20] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In *Proc. of ICML*, 2021.

[21] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems*, 2020.

[22] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A A Kohl, Andrew J Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021.

[23] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in NLP. In *Proc. of ACL*, 2019.

[24] Felix Wu, Angela Fan, Alexei Baevski, Yann Dauphin, and Michael Auli. Pay less attention with lightweight and dynamic convolutions. In *Proc. of ICLR*, 2019.

[25] Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. Attention is not all you need: pure attention loses rank doubly exponentially with depth. In Marina Meila and Tong Zhang, editors, *Proc. of ICML*, 2021.

[26] Yi Tay, Dara Bahri, Donald Metzler, Da-Cheng Juan, Zhe Zhao, and Che Zheng. Synthesizer: Rethinking self-attention for transformer models. In Marina Meila and Tong Zhang, editors, *Proc. of ICML*, 2021.

[27] Yi Tay, Mostafa Dehghani, Jai Prakash Gupta, Vamsi Aribandi, Dara Bahri, Zhen Qin, and Donald Metzler. Are pretrained convolutions better than pretrained transformers? In *Proc. of ACL-IJCNLP*, 2021.

[28] Hanxiao Liu, Zihang Dai, David R. So, and Quoc V. Le. Pay Attention to MLPs. *arXiv e-prints*, page arXiv:2105.08050, May 2021.

[29] Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Gautier Izacard, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, and Hervé Jégou. ResMLP: Feedforward networks for image classification with data-efficient training. *arXiv e-prints*, page arXiv:2105.03404, May 2021.

[30] Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. MLP-Mixer: An all-MLP Architecture for Vision. *arXiv e-prints*, page arXiv:2105.01601, May 2021.

[31] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in NeurIPS*, 2016.

[32] Jasmijn Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an. Graph convolutional encoders for syntax-aware neural machine translation. In *Proc. of EMNLP*, 2017.

[33] Xiao Liu, Zhunchen Luo, and Heyan Huang. Jointly multiple events extraction via attention-based graph information aggregation. In *Proc. of EMNLP*, 2018.

[34] Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. Relational graph attention network for aspect-based sentiment analysis. In *Proc. of ACL*, 2020.

[35] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. In *Proc. of ICML*, 2017.

[36] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proc. of ICLR*, 2017.

[37] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *Proc. of ICLR*, 2018.

[38] Yue Zhang, Qi Liu, and Linfeng Song. Sentence-state LSTM for text representation. In *Proc. of ACL*, 2018.

[39] Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. A graph-to-sequence model for AMR-to-text generation. In *Proc. of ACL*, 2018.

[40] Oren Melamud, Jacob Goldberger, and Ido Dagan. context2vec: Learning generic context embedding with bidirectional LSTM. In *Proc. of CoNLL*, 2016.

[41] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

[42] Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *IEEE International Conference on Computer Vision*, pages 19–27, 2015.

[43] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108, 2019.

[44] Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu. Pre-Training with Whole Word Masking for Chinese BERT. *arXiv e-prints*, page arXiv:1906.08101, June 2019.

[45] Liang Xu, Xuanwei Zhang, and Qianqian Dong. CLUECorpus2020: A Large-scale Chinese Corpus for Pre-training Language Model. *arXiv e-prints*, page arXiv:2003.01355, March 2020.

[46] Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. Revisiting pre-trained models for Chinese natural language processing. In *Findings of EMNLP*, 2020.

[47] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proc. of NAACL: Demonstrations*, 2019.

[48] Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, Yin Tian, Qianqian Dong, Weitang Liu, Bo Shi, Yiming Cui, Junyi Li, Jun Zeng, Rongzhao Wang, Weijian Xie, Yanting Li, Yina Patterson, Zuoyu Tian, Yiwen Zhang, He Zhou, Shaoweihua Liu, Zhe Zhao, Qipeng Zhao, Cong Yue, Xinrui Zhang, Zhengliang Yang, Kyle Richardson, and Zhenzhong Lan. CLUE: A Chinese language understanding evaluation benchmark. In *Proc. of COLING*, 2020.

[49] Xin Liu, Qingcai Chen, Chong Deng, Huajun Zeng, Jing Chen, Dongfang Li, and Buzhou Tang. LCQMC:a large-scale Chinese question matching corpus. In *Proc. of COLING*, 2018.

[50] Jingyang Li and Maosong Sun. Scalable term selection for text categorization. In *Proc. of EMNLP-CoNLL*, 2007.

[51] Songbo Tan and Jin Zhang. An empirical study of sentiment analysis for chinese documents. *Expert Systems with Applications*, 34(4):2622–2629, 2008.

[52] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi 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 Rush. Transformers: State-of-the-art natural language processing. In *Proc. of EMNLP: System Demonstrations*, 2020.

[53] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv:2004.05150*, 2020.

[54] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *Proc. of ICLR*, 2020.

[55] Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-Attention with Linear Complexity. *arXiv e-prints*, page arXiv:2006.04768, June 2020.

[56] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. Rethinking Attention with Performers. *arXiv e-prints*, page arXiv:2009.14794, September 2020.