Huge Pre-trained Language Models: Research and Applications
巨型预训练语言模型：研究与应用

LIU Qun 刘群

Huawei Noah’s Ark Lab 华为诺亚方舟实验室

A Talk to Thai AI Engineer Group
Online，2022-02-28

NOAH'S ARK LAB

HUAWEI

Content

Content

Natural Language Representation Learning

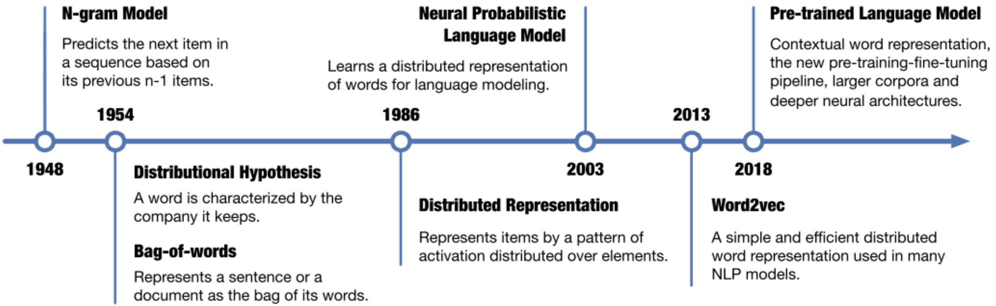
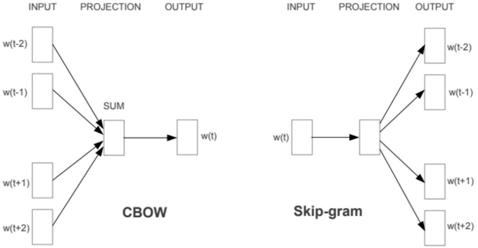


Fig. 1.3 The timeline for the development of representation learning in NLP. With the growing computing power and large-scale text data, distributed representation trained with neural networks and large corpora has become the mainstream

Liu et al., Representation Learning for Natural Language Processing, Springer, 2020

1st Generation Pre-trained NLP Models: — Word Embeddings

- Typical Models: CBOW, Skip-gram, Glove, Fasttext
- Static Representation: the presentation does not change in different contexts



CBOW: predicts the current word based on the context

Skip-gram: predicts surrounding words given the current word

(Mikolov et al., 2013)

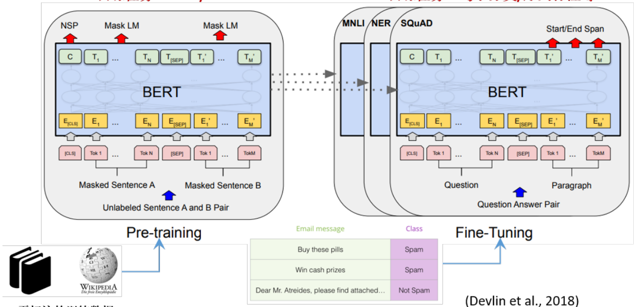
Semantically similar words are close in the space

2nd Generation Pre-trained NLP Models:
— Pre-trained Language Models (Contextualized WEs)

- Typical Models: ELMo, BERT, GPT
- A new NLP reseaerch paradigm: Pre-training then fine-tuning
- The knowledge learned in the pre-training stage is transfered to downstream tasks

训练任务：MLM/NSP

训练任务：句子分类/序列标注等

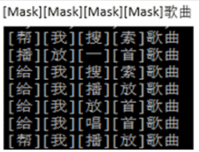


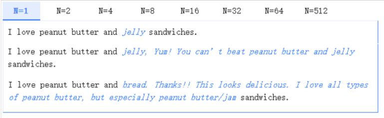
无标注的训练数据

有标注的训练数据

(Devlin et al., 2018)

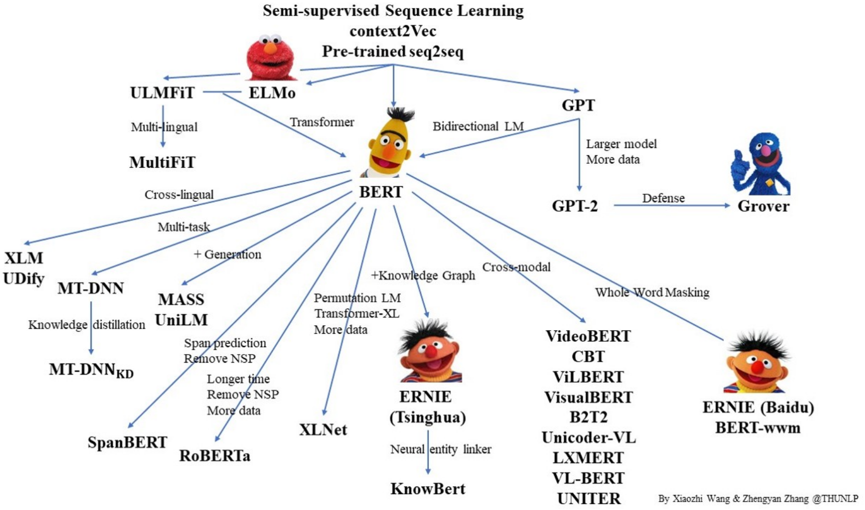
Pre-traing得到精确有效的语言表达





HUAWEI NOAH'S ARK LAB

Family of Pre-trained Language Models



https://github.com/thunlp/PLMpapers

What are Huge Pre-trained Language Models?

- A new paradigm for NLP (or AI) research
  - Deep Neural Models
  - Huge model size & huge training data
  - Non-task-specific pre-training, can be adapted to various downstream tasks
- Also called Foundation (Language) Models

What are Huge Pre-trained Language Models?

- A new paradigm for NLP (or AI) research
  - Deep Neural Models
  - Huge model size & huge training data
  - Non-task-specific pre-training, can be adapted to various downstream tasks
- Also called Foundation (Language) Models

arXiv.org > cs > arXiv:2108.07258

Search...

Help | Advanced

Computer Science > Machine Learning

[Submitted on 16 Aug 2021 (v1), last revised 18 Aug 2021 (this version, v2)]

On the Opportunities and Risks of Foundation Models

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Kohd, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang , Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, Percy Liang (collapse list)

AI is undergoing a paradigm shift with the rise of models (e.g., BERT, DALL-E, GPT-3) that are trained on broad data at scale and are adaptable to a wide range of downstream tasks. We call these models foundation models to underscore their critically central yet incomplete character. This report provides a thorough account of the opportunities and risks of foundation models, ranging from their capabilities (e.g., language, vision, robotics, reasoning, human interaction) and technical principles(e.g., model architectures, training procedures, data, systems, security, evaluation, theory) to their applications (e.g., law, healthcare, education) and societal impact (e.g., inequity, misuse, economic and environmental impact, legal and ethical considerations). Though foundation models are based on standard deep learning and transfer learning, their scale results in new emergent capabilities,and their effectiveness across so many tasks incentivizes homogenization. Homogenization provides powerful leverage but demands caution, as the defects of the foundation model are inherited by all the adapted models downstream. Despite the impending widespread deployment of foundation models, we currently lack a clear understanding of how they work, when they fail, and what they are even capable of due to their emergent properties. To tackle these questions, we believe much of the critical research on foundation models will require deep interdisciplinary collaboration commensurate with their fundamentally sociotechnical nature.

Bommasani et al., On the Opportunities and Risks of Foundation Models, arXiv:2108.07258 [cs.LG]

HUAWEI NOAH'S ARK LAB

Emergence and homogenization

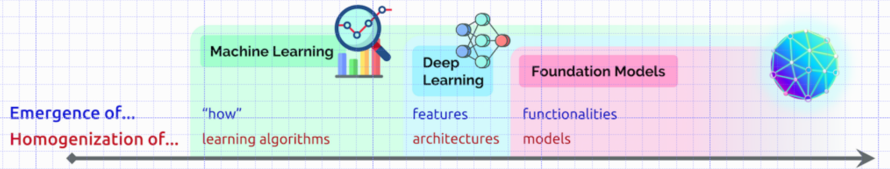
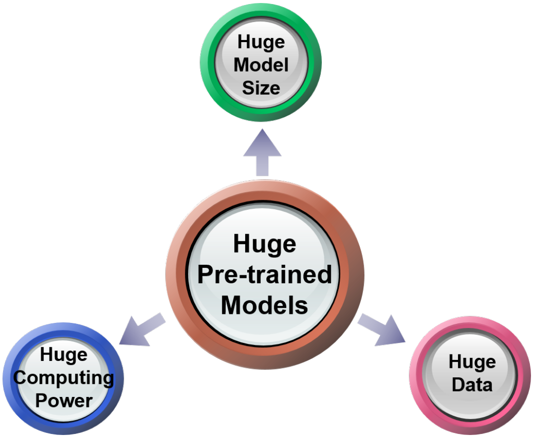


Fig. 1. The story of AI has been one of increasing *emergence* and *homogenization*. With the introduction of machine learning, *how* a task is performed emerges (is inferred automatically) from examples; with deep learning, the high-level features used for prediction emerge; and with foundation models, even advanced functionalities such as in-context learning emerge. At the same time, machine learning homogenizes learning algorithms (e.g., logistic regression), deep learning homogenizes model architectures (e.g., Convolutional Neural Networks), and foundation models homogenizes the model itself (e.g., GPT-3).

Bommasani et al., On the Opportunities and Risks of Foundation Models, arXiv:2108.07258 [cs.LG]

Characteristics of huge pre-trained langauge models



Characteristics of huge pre-trained langauge models

- Data:
  - More data: unannotated data is almost unlimited
  - More diverse: multi-lingual, multi-domain, multi-modal
- Model size:
  - Model size means model capacity
  - There are still big space to increase the model size
- Computing power:
  - The computing power used to training a centralized model gradually reaches the limit of the current hardware
  - To increase the model size (aka. model capacity) without consuming more computing power, spare or distributed model architectures should be adopted (e.g. Mixure-of-Experts or MoE).

Huge Model Size

Released in 2021-10-11

NVIDIA. DEVELOPER HOME BLOG FORUMS DOCS DOWNLOADS TRAINING

DEVELOPER BLOG

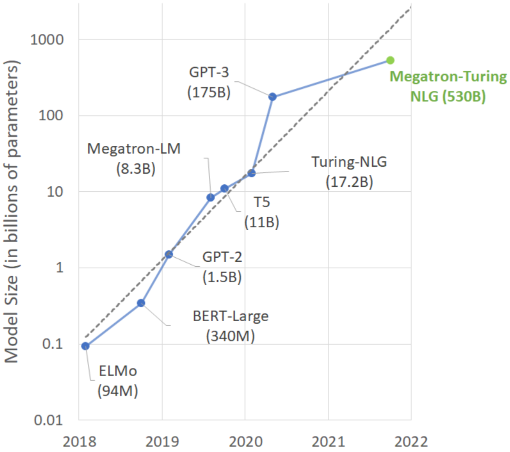
SUBSCRIBE

TECHNICAL WALKTHROUGH

Oct 11, 2021

Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, the World's Largest and Most Powerful Generative Language Model

By Paresh Kharya and Ali Alvi



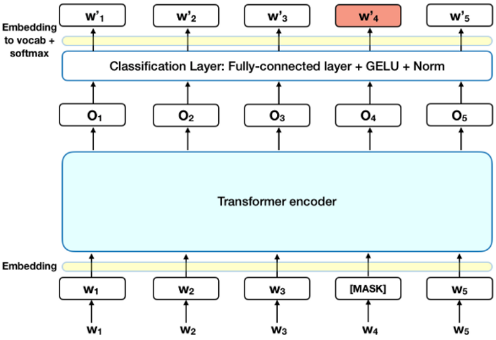
HUAWEI NOAH'S ARK LAB

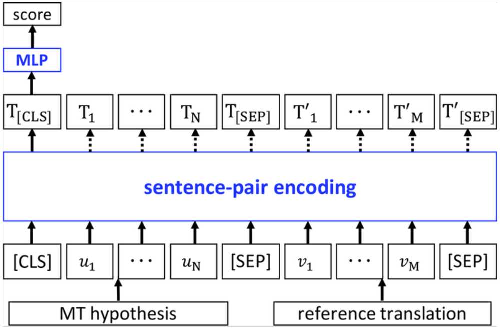
Content

Benefits brought by huge pre-trained models

- Leverage of huge unannotated or weakly-annotated data
  - self-supervised Learning
- Pre-training + fine-tuning framework
  - Simplification of model structures for downstream tasks
  - Significant performance improvements on downstream tasks
- Few-shot learning or zero-shot learning
- Multilingual representation
- Multimodal interaction
- New business model

Self-supervised Learning



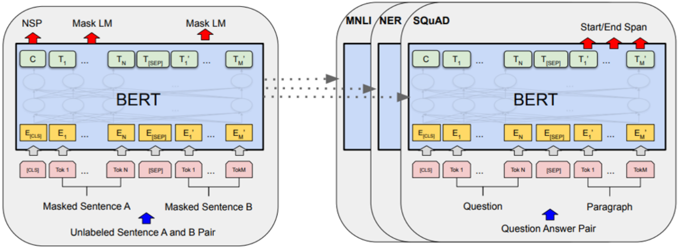


Devlin et al., BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, arXiv:1810.04805, 2018

Pre-training and fine-tuning framework

训练任务：MLM/NSP

训练任务：句子分类/序列标注等



Pre-training

无标注的训练数据

Email message	Class
Buy these pills	Spam
Win cash prizes	Spam
Dear Mr. Atreides, please find attached...	Not Spam

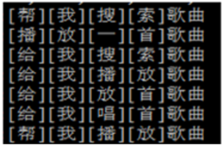
有标注的训练数据

Fine-Tuning

(Devlin et al., 2018)

Pre-traing得到精确有效的语言表达

[Mask][Mask][Mask][Mask]歌曲



N=1 N=2 N=4 N=8 N=16 N=32 N=64 N=512

I love peanut butter and *jelly* sandwiches.

I love peanut butter and *jelly, Yum! You can' t beat peanut butter and jelly* sandwiches.

I love peanut butter and *bread. Thanks!! This looks delicious. I love all types of peanut butter, but especially peanut butter/jam* sandwiches.

Devlin et al., BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, arXiv:1810.04805, 2018

Few-shot and zero-shot learning

Zero-shot

The model predicts the answer given only a natural language discription of the task. No gradient updates are performed.

```
Translate English to French:        ← task description
cheese =>                           ← prompt
```

One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

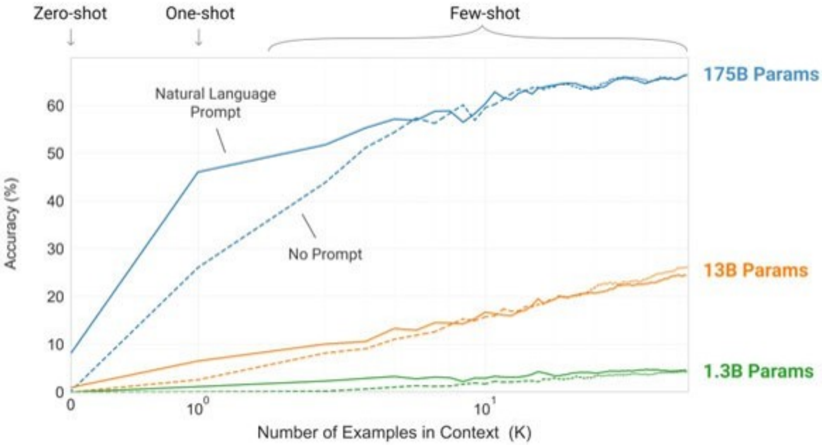
```
Translate English to French:        ← task description
sea otter => loutre de mer          ← example
cheese =>                           ← prompt
```

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
Translate English to French:        ← task description
sea otter => loutre de mer          ← examples
peppermint => menthe poivrée
plush girafe => girafe peluche
cheese =>                           ← prompt
```

Few-shot and zero-shot learning



Brown et al., Language Models are Few-Shot Learners,

arXiv:2005.14165, 2021

Multilingual representation

Multilingual representation

Models

There are two multilingual models currently available. We do not plan to release more single-language models, but we may release `BERT-Large` versions of these two in the future:

- `BERT-Base, Multilingual Cased (New, recommended)` : 104 languages, 12-layer, 768-hidden, 12-heads, 110M parameters
- `BERT-Base, Multilingual Uncased (Orig, not recommended)` : 102 languages, 12-layer, 768-hidden, 12-heads, 110M parameters
- `BERT-Base, Chinese` : Chinese Simplified and Traditional, 12-layer, 768-hidden, 12-heads, 110M parameters

Data Source and Sampling

The languages chosen were the top 100 languages with the largest Wikipedias. The entire Wikipedia dump for each language (excluding user and talk pages) was taken as the training data for each language

Model	D	#M	#lg	en	fr	es	de	el	bg	ru	tr	ar	vi	th	zh	hi	sw	ur	Avg
Fine-tune multilingual model on English training set (Cross-lingual Transfer)																			
mBERT	Wiki	N	102	82.1	73.8	74.3	71.1	66.4	68.9	69.0	61.6	64.9	69.5	55.8	69.3	60.0	50.4	58.0	66.3
XLM (MLM+TLM)	Wiki+MT	N	15	85.0	78.7	78.9	77.8	76.6	77.4	75.3	72.5	73.1	76.1	73.2	76.5	69.6	68.4	67.3	75.1
XLM-R	CC	1	100	**88.8**	**83.6**	**84.2**	**82.7**	**82.3**	**83.1**	**80.1**	**79.0**	**78.8**	**79.7**	**78.6**	**80.2**	**75.8**	**72.0**	**71.7**	**80.1**
Translate everything to English and use English-only model (TRANSLATE-TEST)																			
BERT-en	Wiki	1	1	88.8	81.4	82.3	80.1	80.3	80.9	76.2	76.0	75.4	72.0	71.9	75.6	70.0	65.8	65.8	76.2
RoBERTa	CC	1	1	**91.3**	82.9	84.3	81.2	81.7	83.1	78.3	76.8	76.6	74.2	74.1	77.5	70.9	66.7	66.8	77.8
Fine-tune multilingual model on each training set (TRANSLATE-TRAIN)																			
XLM (MLM)	Wiki	N	100	82.9	77.6	77.9	77.9	77.1	75.7	75.5	72.6	71.2	75.8	73.1	76.2	70.4	66.5	62.4	74.2
Fine-tune multilingual model on all training sets (TRANSLATE-TRAIN-ALL)																			
XLM (MLM+TLM)	Wiki+MT	1	15	85.0	80.8	81.3	80.3	79.1	80.9	78.3	75.6	77.6	78.5	76.0	79.5	72.9	72.8	68.5	77.8
XLM (MLM)	Wiki	1	100	84.5	80.1	81.3	79.3	78.6	79.4	77.5	75.2	75.6	78.3	75.7	78.3	72.1	69.2	67.7	76.9
XLM-R	CC	1	100	**88.7**	**85.2**	**85.6**	**84.6**	**83.6**	**85.5**	**82.4**	**81.6**	**80.9**	**83.4**	**80.9**	**83.3**	**79.8**	**75.9**	**74.3**	**82.4**

https://github.com/google-research/bert/blob/master/multilingual.md

Multimodal interaction

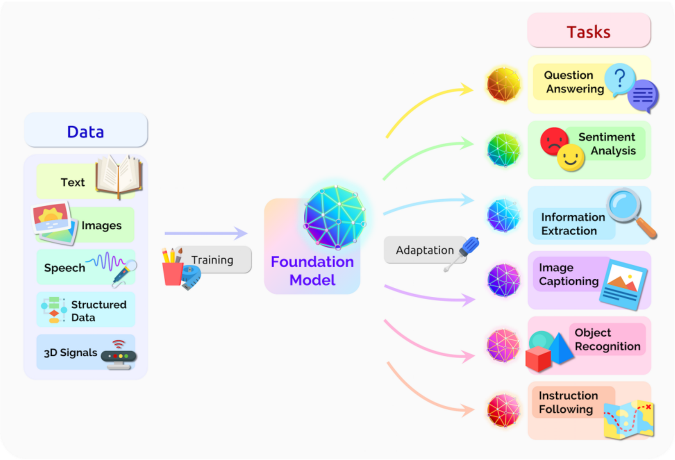
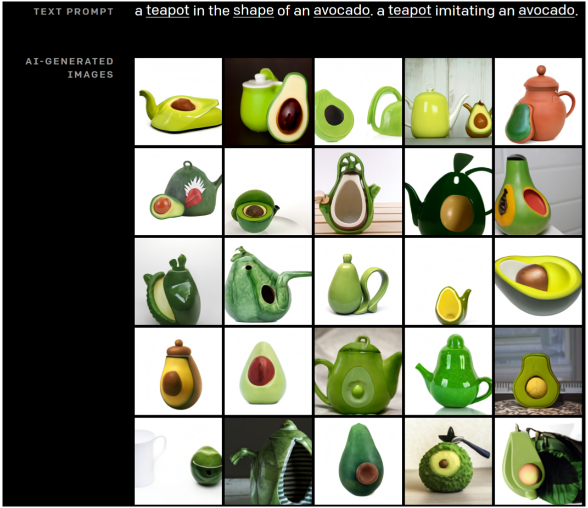


Fig. 2. A foundation model can centralize the information from all the data from various modalities. This one model can then be adapted to a wide range of downstream tasks.

Bommasani et al., On the Opportunities and Risks of Foundation Models, arXiv:2108.07258 [cs.LG]

Multimodal interaction



OpenAI DALL-E demo, source: https://openai.com/blog/dall-e/

New business model

- Distributed $\rightarrow$ Centralized， similar to what happened in the history：
  - Enterprise search, library retrival system$\rightarrow$general search engine (like Google, Baidu)
  - Enterprise IT system$\rightarrow$Cloud computing
- AI service provider:
  - Provide centrealized AI ability
  - Models for various domains, application types will flourish
- AI service customers:
  - Medium/small size entreprises or persons can order customized AI ability
    - For example, refridgarator producers can order dialog services for the refridgarators they produce, without develop such systems by themselves
  - Users can also order small models by using the model compression services provided by the AI service providers

Content

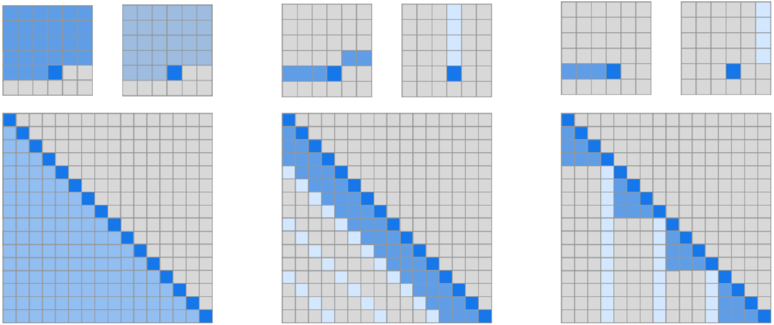
Content

Challenges of Huge PLMs and Potential Solutions

- Model size challenge: How can we increase the model size?
- Competence challenge: doing more and doing better
- Training challenge: how can we train models more efficiently?
- Fine-tuning challenge: how can we fune-tune a huge model?
- Challenge of safety and trustworthiness

Sparse Transformers

- Sparse factorizations of the attention matrix which reduce this to $O(n\sqrt{n})$:



(a) Transformer (b) Sparse Transformer (strided) (c) Sparse Transformer (fixed)

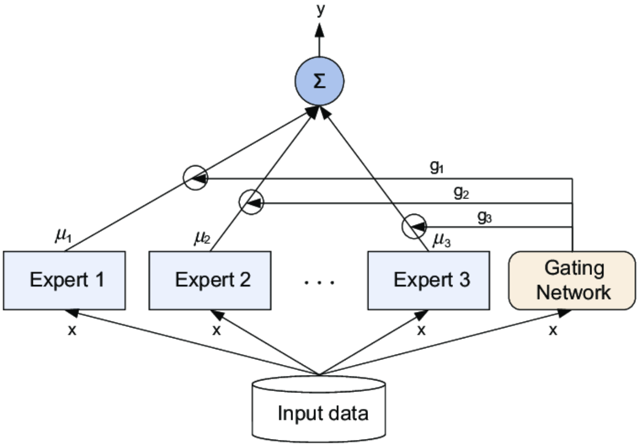
Child et al., Generating Long Sequences with Sparse Transformers, arXiv:1904.10509

Sparse Transformers

- Related work:
  - Big Bird (Zaheer et al. 2020, NeurIPS),
  - Longformer (Beltagy et al. 2020),
  - Reformer (Kitaev et al. 2020, ICLR),
  - Routing Transformer (Roy et al. 2021, ACL),

MoE Transformers

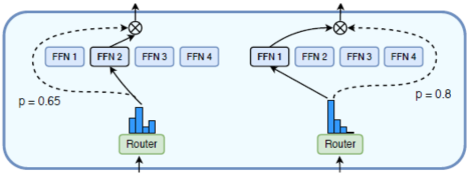
- Introduces Mixture-of-Experts (MoE) in Transformer components

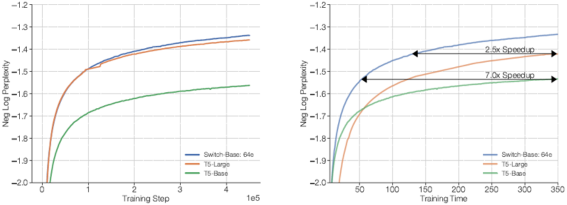


Jason Brownlee, A Gentle Introduction to Mixture of Experts Ensembles (blog)

MoE Transformers

- Switch Transformers (Google, 2021.01)
  - Backbone: T5
  - Parameters: 1571B, 15 layers, 2048 experts
  - Dataset: C4 (180B tokens)
  - Router: switch routing (top-1)





Fedus et al., Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. arXiv:2101.03961, 2021

MoE Transformers

- Top-1 routing (Google)
  - Select only one expert for each token. Greatly reduce the communication cost and speed the training.
  - Fedus et al., Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. arXiv:2101.03961, 2021
- Top-K routing (Google)
  - Select top-k experts for each token. Usually k=2.
  - Shazeer et al., Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. arXiv:1701.06538, 2017
- Hash routing (Facebook)
  - The token-expert mapping is predefined by a hash function, without training.
  - Roller et al., Hash Layers For Large Sparse Models. arXiv:2106.04426, 2021
- Domain routing (AI2 & Facebook)
  - The experts are selected according the domain of the input data.
  - Gururangan et al., DEMix Layers: Disentangling Domains for Modular Language Modeling. arXiv:2108.05036. 2021

Content

Challenges of Huge PLMs and Potential Solutions

- Model size challenge: How can we increase the model size?
- Competence challenge: doing more and doing better
- Training challenge: how can we train models more efficiently?
- Fine-tuning challenge: how can we fune-tune a huge model?
- Challenge of safety and trustworthiness

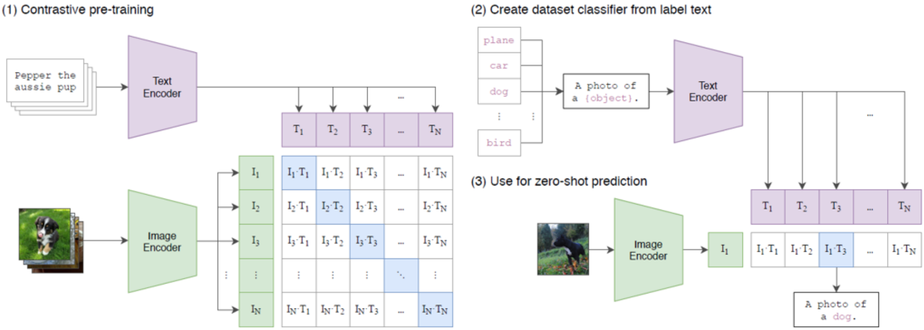
Heterogeneous data training

- Huge pre-training models are able to modeling data with high diversity
- Different types of data can mutual enhanced to make a stronger model
- Heterogeneous data:
  - Multimodal
  - Knowledge
  - External text (enhanced with retrieval)
  - Codes

Image-text pre-training: How different modalities interact?

- Two-tower model: features from the two modalities are encoded seperately and then interacted with constrastive loss
  - CLIP, ALIGN, WENLAN
- One-tower model: concatenate the features from images and texts to a single sequence, and feed it to a Transformre model
  - Encoder: VILT, SOHO
  - Decoder: DALL-E, Frozen
  - Mix: M6, OPT
- Others: modality interaction happens in the cross-attention of the decoder
  - ALBEF

Image-text pre-training: CLIP: a typical two-tower model



Connecting Text and Images by Contrastive Language-Image Pre-training, OpenAI 2021

Image-text pre-training: CLIP: a typical two-tower model

- Novelty: multi-model contrastive learning
  - constrastive learning using global features from images and texts
- large-scale dataset construction:
  - OpenAI 400M monolingual data with images
- large-scale training:
  - CLIP_SMALL: VIT-B/32 + GPT(12L-8head-emb512)
  - CLIP_LARGE: VIT-L/14 + GPT-BASE(12L-12head-emb768)
- Multiple downstream tasks: inlcuding zero-shot image classification, image-text retrieval, etc.

- Zero-shot Image Classification

		Food101	CIFAR10	CIFAR100	Birdsnap	SUN397	Stanford Cars	FGVC Aircraft	VOC2007	DTD	Oxford Pets	Caltech101	Flowers102	MNIST	FER2013	STL10	EuroSAT	RESISC45	GTSRB	KITTI	Country211	PCam	UCF101	Kinetics700	CLEVR	HatefulMemes	Rendered SST2	ImageNet
CLIP-ResNet	RN50	81.1	75.6	41.6	32.6	59.6	55.8	19.3	82.1	41.7	85.4	82.1	65.9	66.6	42.2	94.3	41.1	54.2	35.2	42.2	16.1	57.6	63.6	43.5	20.3	59.7	56.9	59.6
	RN101	83.9	81.0	49.0	37.2	59.9	62.3	19.5	82.4	43.9	86.2	85.1	65.7	59.3	45.6	96.7	33.1	58.5	38.3	33.3	16.9	55.2	62.2	46.7	28.1	61.1	64.2	62.2
	RN50x4	86.8	79.2	48.9	41.6	62.7	67.9	24.6	83.0	49.3	88.1	86.0	68.0	75.2	51.1	96.4	35.0	59.2	35.7	26.0	20.2	57.5	65.5	49.0	17.0	58.3	66.6	65.8
	RN50x16	90.5	82.2	54.2	45.9	65.0	72.3	30.3	82.9	52.8	89.7	87.6	71.9	80.0	56.0	97.8	40.3	64.4	39.6	33.9	24.0	62.5	68.7	53.4	17.6	58.9	67.6	70.5
	RN50x64	91.8	86.8	61.3	48.9	66.9	76.0	35.6	83.8	53.4	93.4	90.6	77.3	90.8	61.0	98.3	59.4	69.7	47.9	33.2	29.6	65.0	74.1	56.8	27.5	62.1	70.7	73.6
CLIP-ViT	B/32	84.4	91.3	65.1	37.8	63.2	59.4	21.2	83.1	44.5	87.0	87.9	66.7	51.9	47.3	97.2	49.4	60.3	32.2	39.4	17.8	58.4	64.5	47.8	24.8	57.6	59.6	63.2
	B/16	89.2	91.6	68.7	39.1	65.2	65.6	27.1	83.9	46.0	88.9	89.3	70.4	56.0	52.7	98.2	54.1	65.5	43.3	44.0	23.3	48.1	69.8	52.4	23.4	61.7	59.8	68.6
	L/14	92.9	96.2	77.9	48.3	67.7	77.3	36.1	84.1	55.3	93.5	92.6	78.7	87.2	57.5	99.3	59.9	71.6	50.3	23.1	32.7	58.8	76.2	60.3	24.3	63.3	64.0	75.3
	L/14-336px	93.8	95.7	77.5	49.5	68.4	78.8	37.2	84.3	55.7	93.5	92.8	78.3	88.3	57.7	99.4	59.6	71.7	52.3	21.9	34.9	63.0	76.9	61.3	24.8	63.3	67.9	76.2

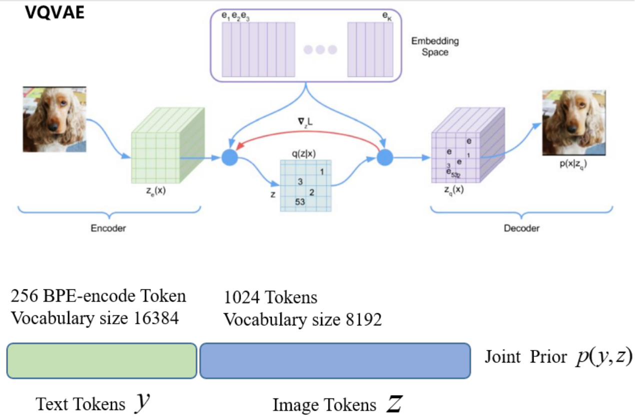
- Image-text retrieval

		Text Retrieval						Image Retrieval					
		Flickr30k			MSCOCO			Flickr30k			MSCOCO		
		R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
Finetune	Unicoder-VL[a]	86.2	96.3	99.0	62.3	87.1	92.8	71.5	90.9	94.9	46.7	76.0	85.3
	Uniter[b]	87.3	98.0	99.2	65.7	88.6	93.8	75.6	94.1	**96.8**	52.9	79.9	88.0
	VILLA[c]	87.9	97.5	98.8	-	-	-	76.3	**94.2**	**96.8**	-	-	-
	Oscar[d]	-	-	-	**73.5**	**92.2**	**96.0**	-	-	-	**57.5**	**82.8**	**89.8**
	ERNIE-ViL[e]	**88.7**	98.0	99.2	-	-	-	**76.7**	93.6	96.4	-	-	-

Connecting Text and Images by Contrastive Language-Image Pre-training, OpenAI 2021

Image-text pre-training: Dall-E: a typical one-tower model

- Image (numeric data): be converted to discrete tokens using a model like VQVAE, which is also used to generate data in the original modality.
- Text (symbolic data): discrete tokens already.
- Concatenate the image tokens and text tokens to a sequence. Train an autoregressive LM (like GPT).



Zero-Shot Text-to-Image Generation. OpenAI, 2021

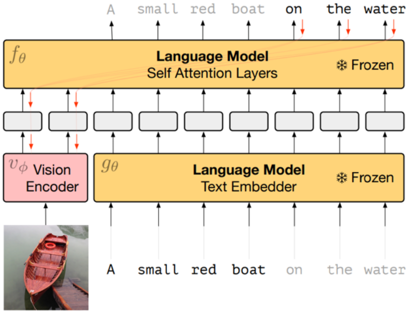
Image-text pre-training: Dall-E: a typical one-tower model

- Text-grounded Image Generation

Zero-Shot Text-to-Image Generation. OpenAI, 2021

Image-text pre-training: Frozen: a typical few-shot model

- Pre-training:
  - Model:
    - Fix the 7B pre-trained text GPT, and fine-tune the vision prefix (prompt)
  - Objective:
    - Use the image caption objective to fine-tune a NF-ResNet-50 model on the CC12M dataset.
  - Obtain a cross-modality few-shot (in-context) learning ability (like GPT-3)
- Comparison with prefix tuning in NLP:
  - Similarities:
    - Fix the pre-trained text model, and only fine-tune the prefix.
  - Differences:
    - Cross-modality prefix
      The prefix is sample-dependent: different image will give different prefix.



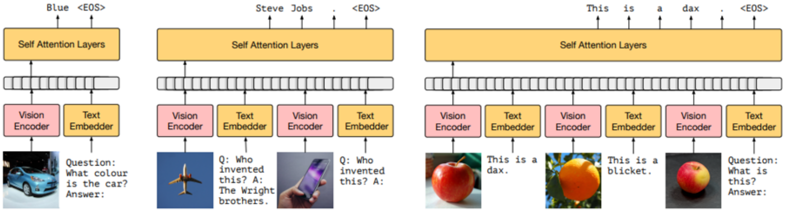
Multimodal Few-Shot Learning with Frozen Language Models, Deepmind 2021

Image-text pre-training: Frozen: a typical few-shot model

- VQA:

n-shot Acc.	n=0	n=1	n=4	$\mathcal{T}$
Frozen	29.5	35.7	38.2	✗
Frozen scratch	0.0	0.0	0.0	✗
Frozen finetuned	24.0	28.2	29.2	✗
Frozen train-blind	26.2	33.5	33.3	✗
Frozen VQA	48.4	–	–	✓
Frozen VQA-blind	39.1	–	–	✓
Oscar [23]	73.8	–	–	✓

- Inference:



(a) **0-shot VQA** (b) **1-shot outside-knowledge VQA** (c) **Few-shot image classification**

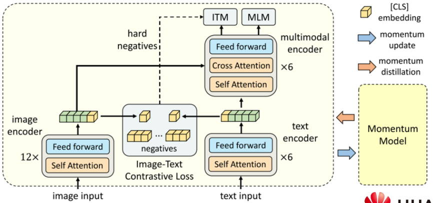
Multimodal Few-Shot Learning with Frozen Language Models, Deepmind 2021

Image-text pre-training: ALBEF: a typical cross-attention model

- Pre-training:
  - Model:
    - Image pre-trained VIT, text pre-trained BERT
  - Objective:

    Image-text contrastive learning between [CLS] of the last layer of image VIT and [CLS] of the sixth layer of text BERT.
    Mask language model training in text
    Image features are fed to the text decoder (last 6 BERT layer), followed by multi-modal cross-attention and image-text matching.
  - Obtain a cross-modality few-shot (in-context) learning ability (like GPT-3)
- Multiple downstream tasks
  - Include image-text retrieval, VQA, VE, Grounding, etc.

Align before Fuse: Vision and Language Representation Learning with Momentum Distillation, Salesforce 2021



HUAWEI NOAH'S ARK LAB

Image-text pre-training: ALBEF: a typical cross-attention model

- Image-text Retrieval:

Method	# Pre-train Images	Flickr30K (1K test set) TR R@1	TR R@5	TR R@10	IR R@1	IR R@5	IR R@10	MSCOCO (5K test set) TR R@1	TR R@5	TR R@10	IR R@1	IR R@5	IR R@10
UNITER	4M	87.3	98.0	99.2	75.6	94.1	96.8	65.7	88.6	93.8	52.9	79.9	88.0
VILLA	4M	87.9	97.5	98.8	76.3	94.2	96.8	-	-	-	-	-	-
OSCAR	4M	-	-	-	-	-	-	70.0	91.1	95.5	54.0	80.8	88.5
ALIGN	1.2B	95.3	99.8	100.0	84.9	97.4	98.6	77.0	93.5	96.9	59.9	83.3	89.8
ALBEF	4M	94.3	99.4	99.8	82.8	96.7	98.4	73.1	91.4	96.0	56.8	81.5	89.2
ALBEF	14M	**95.9**	**99.8**	**100.0**	**85.6**	**97.5**	**98.9**	**77.6**	**94.3**	**97.2**	**60.7**	**84.3**	**90.5**

Table 2: Fine-tuned image-text retrieval results on Flickr30K and COCO datasets.

- Visual Grounding:

Figure 6: Grad-CAM visualizations on the cross-attention maps corresponding to individual words.

Align before Fuse: Vision and Language Representation Learning with Momentum Distillation, Salesforce 2021

Integration with Knowledge: Triplet-Enhanced PLMs

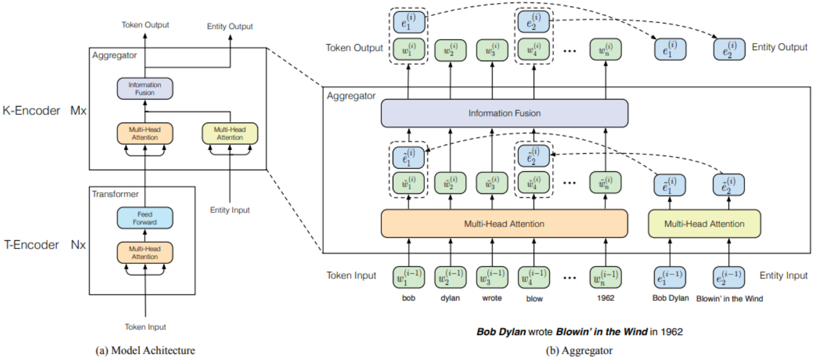


Figure 2: The left part is the architecture of ERNIE. The right part is the aggregator for the mutual integration of the input of tokens and entities. Information fusion layer takes two kinds of input: one is the token embedding, and the other one is the concatenation of the token embedding and entity embedding. After information fusion, it outputs new token embeddings and entity embeddings for the next layer.

Zhang et al., ERNIE: Enhanced Language Representation with Informative Entities, ACL 2019

Integration with Knowledge: Entity-Enhanced PLMs

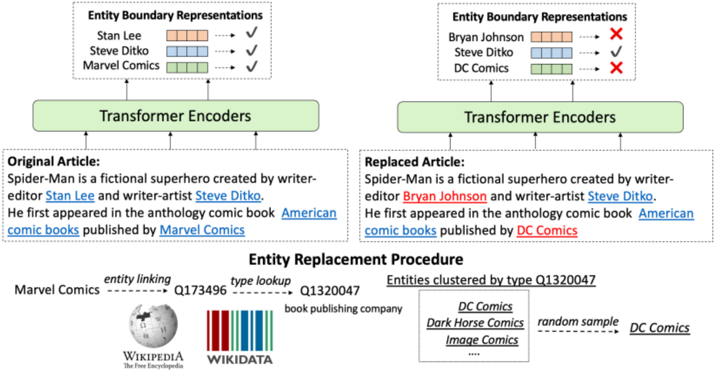
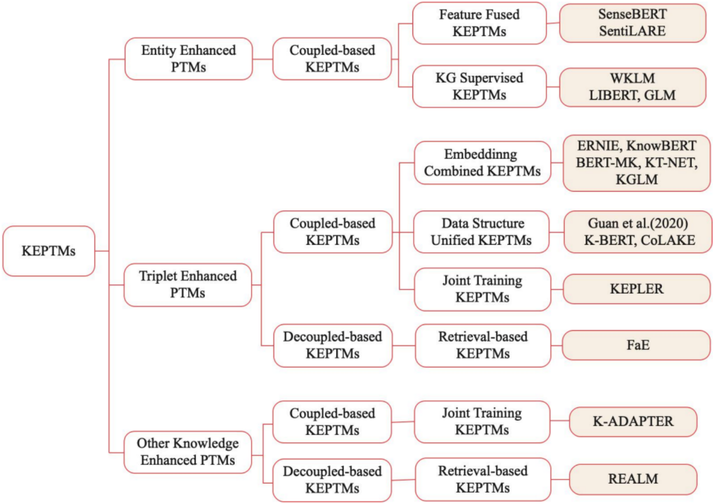


Figure 1: Type-Constrained Entity Replacements for Knowledge Learning.

Xiong et al., Pretrained Encyclopedia: Weakly Supervised Knowledge-Pretrained Language Model, ICLR 2020

Integration with Knowledge: Related Progress



Yang et al., A Survey of Knowledge Enhanced Pre-trained Models, arXiv:2110.00269

Integration with retrieval

- Why PLMs need retrieval
  - Generate text which is more faithful to facts
  - Adapt to external knowledge which is dynamically changed
- Concerns of retrieval augmented models:
  - Retrieval in pre-training or in fine-tuning?
  - How to encode the multiple docuemnts retrieved.
  - Training of retriever and generator (predictor): pipeline or end-to-end?

	Backbone model	Downstream tasks	Retrieval in pre-training	Retrieval in fine-tuning	End2End training
REALM[1]	BERT	ODQA	✔	✔	✔
RAG[2]	BART	ODQA/Generative QA/Dialogue generation	✘	✔	✔
FiD[3]	T5/BART	ODQA/Generative QA/Dialogue generation/Multi docs summarization	✘	✔	✘

Integration with retrieval: REALM(Retrieval-augmented Pre-training)

- Retrieval Augmented Pre-training
  - Train the Retriever and the Generator jointly in pre-training
  - From pattern memory (BERT) to retrieval + memory

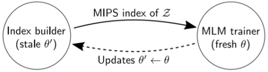
$$p(y\,|\,x) = \sum_{\underset{\text{Z=Top(K) passages}}{z \in \mathcal{Z}}} \underbrace{p_\phi(y|x,z)}_{\text{Generator}} \underbrace{p_\theta(z|x)}_{\text{Retriever}}$$

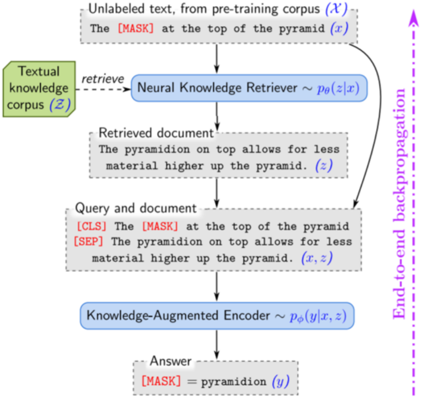
- Knowledge Retriever
  - MLM object provides distant supervision signals for Retriever

$$p(z\,|\,x) = \frac{\exp f(x,z)}{\sum_{z'} \exp f(x,z')},$$

$$f(x,z) = \texttt{Embed}_{\texttt{input}}(x)^\top \texttt{Embed}_{\texttt{doc}}(z)$$

- Biggest challenge of End2End training: Document Index update
  - Asynchronous MIPS updating





Guu, Kelvin, et al. "Realm: Retrieval-augmented language model pre-training."

HUAWEI NOAH'S ARK LAB

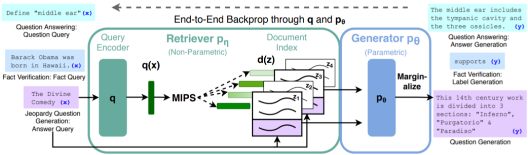
Integration with retrieval: RAG(Retrieval-augmented Generation)

- Using retriever in fine-tuning:
  - End2end training (similar to REALM), but not updating the document index
  - Concatenate the retrieved documents with query, which is limited by the max-seq-length of the encoder (similar to REALM)
- Objective:
  - RAG-Sequence Model:

$$p_{\text{RAG-Sequence}}(y|x) \approx \sum_{z\in\text{top-}k(p(\cdot|x))} p_\eta(z|x)p_\theta(y|x,z) = \sum_{z\in\text{top-}k(p(\cdot|x))} p_\eta(z|x)\prod_i^N p_\theta(y_i|x,z,y_{1:i-1})$$

  - RAG-Token Model:

$$p_{\text{RAG-Token}}(y|x) \approx \prod_i^N \sum_{z\in\text{top-}k(p(\cdot|x))} p_\eta(z|x)p_\theta(y_i|x,z,y_{1:i-1})$$



Lewis, Patrick, et al. "Retrieval-augmented generation for knowledge-intensive nlp tasks."

Integration with retrieval: FiD(Fusion in Decoder)

- FiD integrates information in decoder side
  - Documents are encoded independently in encoder side
  - The interaction between documents happen in the cross-attention in decoder side
- FiD uses the multi-document information more efficiently
  - The decouple of Generator and Retriever makes it more flexible than REALM and RAG
  - Cross Attention Scores provide explainability to some extent
  - Obtain SoTA performance in multiple downstream tasks including QA, dialog generations, etc.

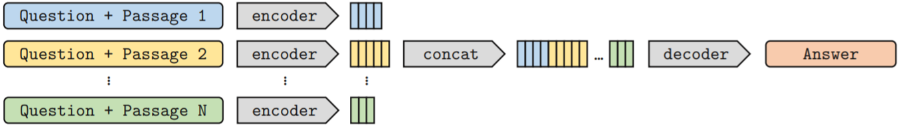


Figure 2: Architecture of the Fusion-in-Decoder method.

Izacard, Gautier, and Edouard Grave. "Leveraging passage retrieval with generative models for open domain question answering."

Content

Challenges of Huge PLMs and Potential Solutions

- Model size challenge: How can we increase the model size?
- Competence challenge: doing more and doing better
- Training challenge: how can we train models more efficiently?
- Fine-tuning challenge: how can we fune-tune a huge model?
- Challenge of safety and trustworthiness

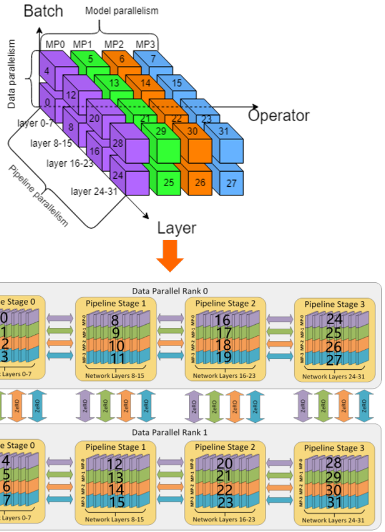
How can we train models more efficiently?

- The training of huge PLMs is expensive because of the huge model size and huge training data.
- It is critical to improve the training efficiency to reduce the cost.
  - Distributed parallel training
  - Transfer learning (reuse parameters of existing models)
  - Continuous training (incremental training, life-long training, avoiding catastrophic forgetting)

3-D parallel training

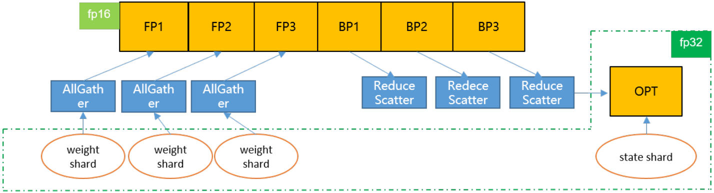
- 3-D mixture paralle: data parallel + pipeline parallel + model parallel
  - Data parallel: partition in batche dimension
  - Pipeline parallel: partition in layer dimension
  - Model parallel: partition in operator dimension
- By mapping 3-D coordinates to physical devices, we can train the huge models like GPT-3 efficiently.

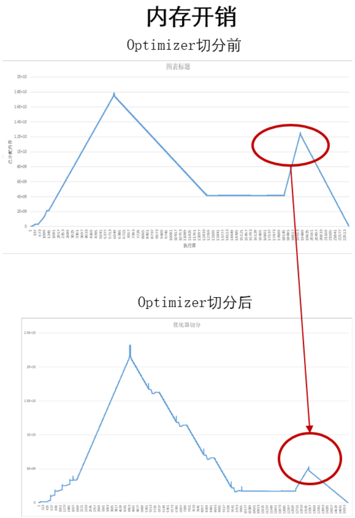
Coordinate	RANK	Coordinate	RANK	Coordinate	RANK	Coordinate	RANK
(0, 0, 0)	0	(1, 0, 0)	8	(2, 0, 0)	16	(3, 0, 0)	24
(0, 0, 1)	1	(1, 0, 1)	9	(2, 0, 1)	17	(3, 0, 1)	25
(0, 0, 2)	2	(1, 0, 2)	10	(2, 0, 2)	18	(3, 0, 2)	26
(0, 0, 3)	3	(1, 0, 3)	11	(2, 0, 3)	19	(3, 0, 3)	27
(0, 1, 0)	4	(1, 1, 0)	12	(2, 1, 0)	20	(3, 1, 0)	28
(0, 1, 1)	5	(1, 1, 1)	13	(2, 1, 1)	21	(3, 1, 1)	29
(0, 1, 2)	6	(1, 1, 2)	14	(2, 1, 2)	22	(3, 1, 2)	30
(0, 1, 3)	7	(1, 1, 3)	15	(2, 1, 3)	23	(3, 1, 3)	31



https://www.microsoft.com/en-us/research/blog/deepspeed-extreme-scale-model-training-for-everyone/

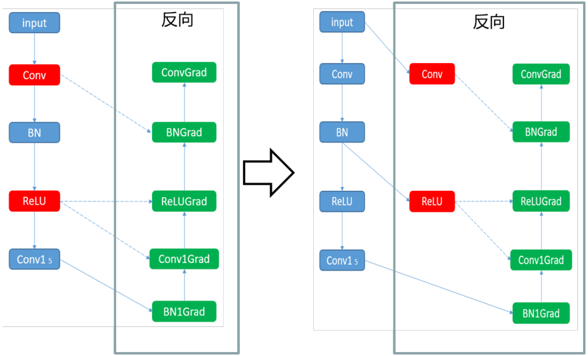
Optimizer state parallel





- Feature：
  - inner-layer partition: partition in dimensions of parameters, optimizer states and gradients
  - communication grouping parallel: allgather and reduce-catter, forward and backword computing
  - mixture precision: use fp16 for forward-backword propogation and communication, use fp32 for optimizer parameters

Re-computing

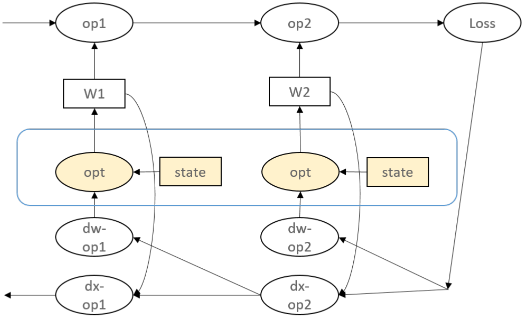


- Abandon activitions in forward computing, and re-computing them in backward propogation. Trade time for spaces.

HUAWEI NOAH'S ARK LAB

Heterogeneous computing

- In the past few years, the model sizes increased by 1000 times, while the memory of parallel computing devices only increased by 5 times (GPU memory: 16G to 80G)
- Move parts of computing of training to Host CPUs and Host memories. A typical solution is optimizer heterogeneous computing.
  - The number of Adam Optimizer states is twice of the number of model weights: A 175B GPT-3 model has 350B optimizer states
  - Move the adam optimizer computing to Host CPU, and optimizer states to Host memory.
  - This can greatly reduce the memory cost in GPU/NPUs.



Optimizer CPU execution

Content

Challenges of Huge PLMs and Potential Solutions

- Model size challenge: How can we increase the model size?
- Competence challenge: doing more and doing better
- Training challenge: how can we train models more efficiently?
- Fine-tuning challenge: how can we fune-tune a huge model?
- Challenge of safety and trustworthiness

How can we fune-tune a huge model?

- It is expensive to use the traditional full-parameter fine-tune method for huge models, because of the huge model size.
  - Prompt tuning has attracted attentions in the research community
  - Another idea is adapter-based fine-tuning
- Traditional knowledge distillation becomes almost impossible because of the huge training data.
- Other model compression methods like quantization and pruning also face new problems with huge models.

Prompting Methods for Downstream Tasks

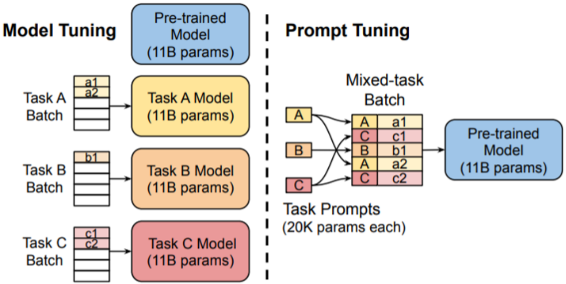
Type	Task	Input ([X])	Template	Answer ([Z])
Text CLS	Sentiment	I love this movie.	[X] The movie is [Z].	great fantastic ...
	Topics	He prompted the LM.	[X] The text is about [Z].	sports science ...
	Intention	What is taxi fare to Denver?	[X] The question is about [Z].	quantity city ...
Text-span CLS	Aspect Sentiment	Poor service but good food.	[X] What about service? [Z].	Bad Terrible ...
Text-pair CLS	NLI	[X1]: An old man with ... [X2]: A man walks ...	[X1]? [Z], [X2]	Yes No ...
Tagging	NER	[X1]: Mike went to Paris. [X2]: Paris	[X1] [X2] is a [Z] entity.	organization location ...
Text Generation	Summarization	Las Vegas police ...	[X] TL;DR: [Z]	The victim ... A woman ... ...
	Translation	Je vous aime.	French: [X] English: [Z]	I love you. I fancy you. ...

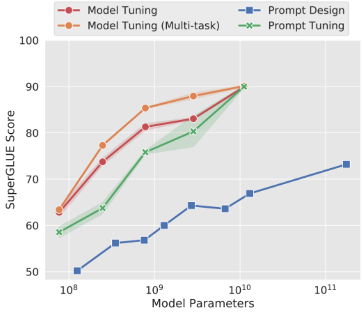
Liu et al., Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing, arXiv:2107.13586, 2021

Prompt-based Learning for Large-scale PLMs

Discrete Prompt		Dense Prompt	
Tuning-free Prompting	Fixed-prompt, LM Tuning	Prompt+LM Tuning	Fixed-LM, Prompt Tuning
Hand-crafted prompt: GPT-3	Hand-crafted prompt & Finetuning on target dataset: T5, PET	For better performance: P-Tuning	Lightweight finetuning: Prefix-Tuning
Automated prompt: AutoPrompt			The scale of PLMs is important: PromptTuning
Finetuning on untargeted datasets: instruction tuning			Better initialization for dense prompt: PPT

Prompt Tuning

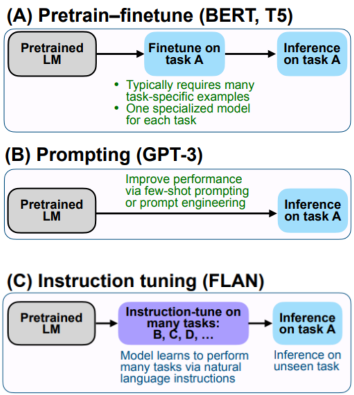


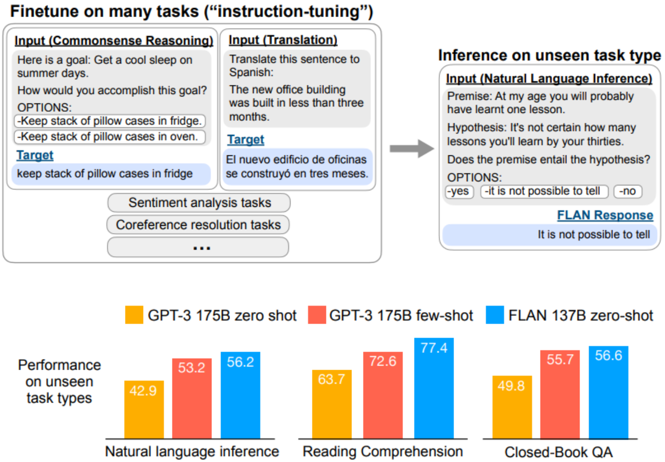


Lester et al., The Power of Scale for Parameter-Efficient Prompt Tuning, arXiv:2104.08691, 2021

Figure 1: Standard **model tuning** of T5 achieves strong performance, but requires storing separate copies of the model for each end task. Our **prompt tuning** of T5 matches the quality of model tuning as size increases, while enabling the reuse of a single frozen model for all tasks. Our approach significantly outperforms few-shot **prompt design** using GPT-3. We show mean and standard deviation across 3 runs for tuning methods.

Instruction Tuning





Wei et al., Finetuned Language Models Are Zero-Shot Learners, arXiv:2109.01652, 2021 (Google)

Content

Challenges of Huge PLMs and Potential Solutions

- Model size challenge: How can we increase the model size?
- Competence challenge: doing more and doing better
- Training challenge: how can we train models more efficiently?
- Fine-tuning challenge: how can we fune-tune a huge model?
- Challenge of safety and trustworthiness

Safty and trustworthiness: social aspects of huge pre-trained models

- Bias and equality
- Abuse and misuse
- Environmental impact
- Legality
- Economic impact
- Ethic problems

Content

Content

Our Work

- Our Models
- Efficient Training and Deployment
- Applications of PLMs

NEZHA: A Pre-trained Language Model for Chinese Language Understanding

NEZHA: NEURAL CONTEXTUALIZED REPRESENTATION FOR CHINESE LANGUAGE UNDERSTANDING

TECHNICAL REPORT

Junqiu Wei, Xiaozhe Ren, Xiaoguang Li, Wenyong Huang, Yi Liao, Yasheng Wang, Jiashu Lin*, Xin Jiang, Xiao Chen, Qun Liu
Noah's Ark Lab, *HiSilicon, Huawei Technologies
{wei.junqiu1, renxiaozhe, lixiaoguang11, wenyong.huang, liao.yi, wangyasheng, linjiashu, jiang.xin, chen.xiao2, qun.liu}@huawei.com

September 4, 2019

- Technical reports: https://arxiv.org/abs/1909.00204

NEZHA: Applications

Carrier BG， Cloud & AI BG， Consumer BG， Smart Car Solutions BU

HUAWEI HUAWEI CLOUD

华为小艺

Huawei Mobile Services

AI Algorithms	Dialog System	NLP Service Framework

TinyBERT Model Compression

NEZHA Pre-trained Language Model

NEZHA: Applications

- Basic Models and Algorithms
  - Pre-trained Language Models and Model Compression Tools
- Dialog Systems
  - Intent Detection and Slot Filling
- Search and Recommendation
  - Query analysis
  - Intent understanding in search ads
  - Fine-grained re-ranking for enterprise document retrieval
  - Document labeling and classification

Pangu-α: A Large-scale Autoregressive Pretrained Chinese Language Model

PanGu-α: Large-scale Autoregressive Pretrained Chinese Language Models with Auto-parallel Computation

Technical Report

Wei Zeng* Xiaozhe Ren* Teng Su* Hui Wang*

Yi Liao Zhiwei Wang Xin Jiang Zhenzhang Yang Kaisheng Wang Xiaoda Zhang

Chen Li Ziyan Gong Yifan Yao Xinjing Huang Jun Wang Jianfeng Yu Qi Guo

Yue Yu Yan Zhang Jin Wang Hengtao Tao Dasen Yan Zexuan Yi Fang Peng

Fangqing Jiang Han Zhang Lingfeng Deng Yehong Zhang Zhe Lin

Chao Zhang Shaojie Zhang Mingyue Guo Shanzhi Gu Gaojun Fan Yaowei Wang

Xuefeng Jin Qun Liu Yonghong Tian

PanGu-α Team

Technical report: http://arxiv.org/abs/2104.12369

Pangu-α: Model architecture

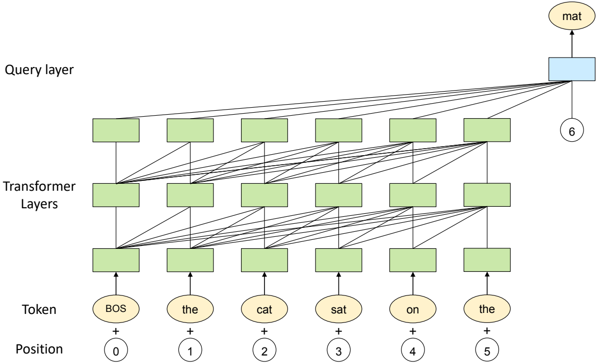


Figure 1: The architecture of PanGu-α. The model is based on a uni-directional Transformer decoder. A query layer is stacked on top of Transformer layers with the position embedding as the query in the attention mechanism to generate the token at the next position.

Pangu-α: Model sizes and data collection and filtering

Table 1: Model sizes and hyperparameters of PanGu-α models.

Model	#Parameters	#Layers (L)	Hidden size (d)	FFN size (d_{ff})	#Heads (N_h)
PanGu-α 2.6B	2.6B	32	2560	10240	40
PanGu-α 13B	13.1B	40	5120	20480	40
PanGu-α 200B	207.0B	64	16384	65536	128

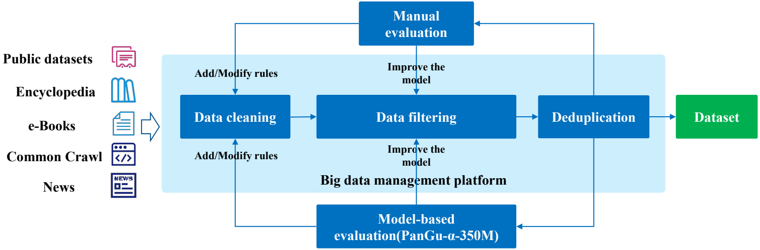


Figure 2: The data sources and the process of constructing pretraining data for PanGu-α.

Pangu-α: Data composition and sampling strategy

Table 3: Data composition of the 1.1TB Chinese text corpus.

	Size (GB)	Data source	Processing steps
Public datasets	27.9	15 public datasets including DuReader, BaiDuQA, CAIL2018, Sogou-CA, etc.	Format conversion[11] and text deduplication
Encyclopedia	22	Baidu Baike, Sogou Baike, etc.	Text deduplication
e-Books	299	e-Books on various topics (e,g., novels, history, poetry, ancient prose, etc.).	Sensitive word and model-based spam filtering
Common Crawl	714.9	Web data from January 2018 to December 2020 from Common Crawl.	All steps
News	35.5	News data from 1992 to 2011.	Text deduplication

Table 4: Sampling strategy of the corpora in training PanGu-α models.

	PanGu-α 200B			PanGu-α 2.6B&13B	
	Quantity (tokens)	Weight in training mix	Epochs elapsed when training	Quantity (tokens)	Weight in training mix
Public datasets	25.8B	10.23%	3.65	7B	27.99%
e-Books	30.9B	12.23%	0.41	5.6B	18%
Common Crawl	176.2B	62.81%	0.85	2.5B	10%
News	19.8B	7.83%	2.2	5.6B	22%
Encyclopedia data	5.8B	6.9%	3	5.8B	23%

Pangu-α: Parallelization strategy

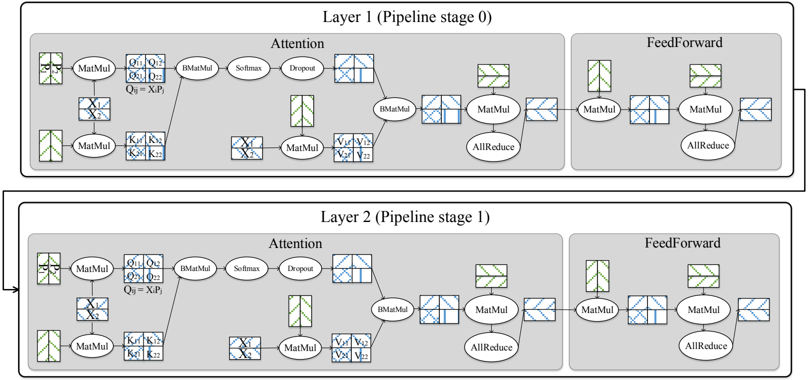


Figure 6: A simplified PanGu-α's parallelization strategy. The ellipsoids stand for the operators, blue rectangles represent tensors, and green rectangles represent trainable parameters. Parameters are partitioned along the row and column dimension respectively, and the input tensor is partitioned along the row dimension. And, two layers are assigned to different pipeline stages.

Pangu-α: Training curves

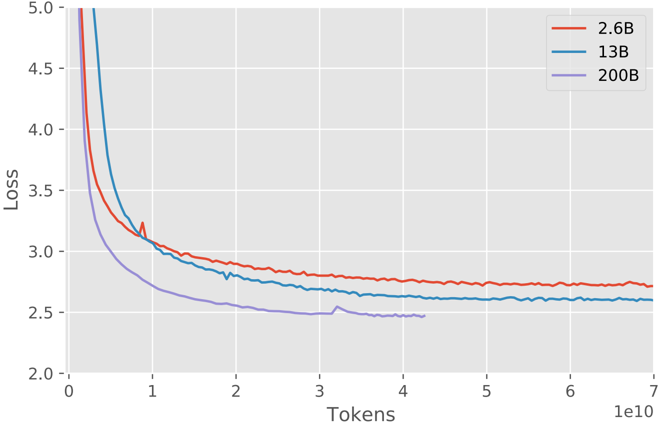


Figure 8: Training curves of three PanGu-α models with different model sizes. The x-axis denotes the number of training tokens, which is measured as $training\_steps * batch\_size * sequence\_length$. The y-axis denotes the training loss.

Pangu-α: Experimental results

Table 9: Performance comparison of CPM 2.6B v.s. PanGu-α 2.6B on few-shot NLP tasks.

Dataset	Method	Metrics	Task Types	Zero-Shot		One-Shot		Few-Shot		
				CPM 2.6B	PanGu-α 2.6B	CPM 2.6B	PanGu-α 2.6B	#Shot(K)	CPM 2.6B	PanGu-α 2.6B
CMRC2018	Generation	Em/F1	Read Comprehension	0.59/10.12	**1.21/16.647**	1.71/11.29	**2.49/18.57**	Dynamic	3.11/14.64	**5.68/23.22**
DRCD	Generation	Em/F1	Read Comprehension	0/4.62	**0.8/9.99**	0.22/5.17	**2.47/12.48**	Dynamic	0.15/7.14	**5.31/18.29**
DuReader	Generation	Rouge-1	Read Comprehension	16.63	**21.07**	16.42	**20.18**	6,6	17.85	**21.43**
WebQA	Generation	Em/f1	Closed-Book QA	6/12.59	**6/16.32**	6/11.82	**12/23.39**	8,8	4/12.23	**24/33.94**
PD-CFT	Generation	Acc	Cloze(without choices)	35.73/38.99	**38.47/42.39**	33.3/39.73	**38.8/41.61**	3,3	32.03/39.84	**39.07/42.05**
CMRC2017	Generation	Acc	Cloze(without choices)	24.60	**37.83**	25.40	**38.00**	3,3	23.50	**36.33**
CHID	PPL	Acc	Cloze(multi-choices)	68.62	**68.73**	67.91	**68.16**	3,3	**66.82**	66.56
CMRC2019	PPL	Acc	Cloze (multi-choices)	47.69	**61.93**	47.99	**61.54**	2,2	47.20	**62.42**
CMNLI	PPL	Acc	Natural Language Inference	49.10	**50.20**	47.56	**49.54**	6,12	49.29	**51.17**
OCNLI	PPL	Acc	Natural Language Inference	**44.20**	42.61	**44.30**	44.00	3,6	44.00	**46.78**
TNEWS	PPL	Acc	Text classification	**65.44**	60.95	**69.50**	57.95	6,6	**70.17**	63.62
IFLYTEK	PPL	Acc	Text classification	68.91	**74.26**	**79.84**	79.03	3,3	**83.99**	80.15
AFQMC	PPL	Acc	Sentence Pair Similarity	**66.34**	59.29	39.70	**64.62**	4,4	38.29	**69.00**
CSL	PPL	Acc	Keyword Recognition	**52.30**	50.50	**51.20**	50.90	10,10	50.50	**52.00**
CLUEWSC2020	PPL	Acc	WSC	**73.684**	73.36	73.684	**75.33**	14,14	70.065	**72.70**
C^3	PPL	Acc	Common Sense Reasoning	49.81	**53.42**	51.43	**52.82**	3,3	51.60	**53.64**

Pangu-α: Experimental results

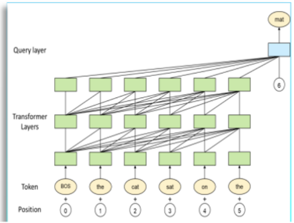
Table 10: Performance comparison of PanGu-α 2.6B v.s. PanGu-α 13B on few-shot NLP tasks.

Dataset	Method	Metrics	Task Types	Zero-Shot		One-Shot		Few-Shot		
				PanGu-α 2.6B	PanGu-α 13B	PanGu-α 2.6B	PanGu-α 13B	#Shot(K)	PanGu-α 2.6B	PanGu-α 13B
CMRC2018	Generation	Em/F1	Read Comprehension	1.21/16.65	**1.46/19.28**	2.49/18.57	**3.76/21.46**	Dynamic	5.68/23.22	**9.76/29.23**
DRCD	Generation	Em/F1	Read Comprehension	**0.8**/9.99	0.66/**10.55**	2.47/12.48	**4.22/15.01**	Dynamic	5.31/18.29	**9.09/23.46**
DuReader	Generation	Rouge-1	Read Comprehension	21.07	**24.46**	20.18	**25.99**	6,6	21.43	**27.67**
WebQA	Generation	Em/f1	Closed-Book QA	4.43/13.71	**5.13/14.47**	10.22/20.56	**13.43/24.52**	8,8	23.71/33.81	**31.18/41.21**
PD-CFT	Generation	Acc	Cloze(without choices)	38.47/42.39	**43.86/46.60**	38.8/41.61	**40.97/45.42**	3,3	39.07/42.05	**41.13/45.86**
CMRC2017	Generation	Acc	Cloze(without choices)	37.83	**38.90**	38.00	**38.40**	3,3	36.33	**37.86**
CHID	PPL	Acc	Cloze(multi-choices)	68.73	**70.64**	68.16	**70.05**	3,3	66.56	**70.91**
CMRC2019	PPL	Acc	Cloze (multi-choices)	68.22	**70.54**	68.05	**70.02**	2,2	66.26	**71.28**
CMNLI	PPL	Acc	Natural Language Inference	**50.20**	48.44	**49.54**	46.81	6,12	**51.17**	46.18
OCNLI	PPL	Acc	Natural Language Inference	**42.61**	41.53	44.00	**44.10**	3,6	**46.78**	46.44
TNEWS	PPL	Acc	Text classification	**60.95**	60.26	57.95	**63.83**	6,6	63.62	**65.17**
IFLYTEK	PPL	Acc	Text classification	**74.26**	73.80	**79.03**	78.95	3,3	80.15	**80.34**
AFQMC	PPL	Acc	Sentence Pair Similarity	59.29	**65.76**	**64.62**	63.55	4,4	**69.00**	68.91
CSL	PPL	Acc	Keyword Recognition	**50.50**	49.30	**50.90**	50.20	10,10	52.00	**55.70**
CLUEWSC2020	PPL	Acc	WSC	73.36	**75.00**	**75.33**	75.00	14,14	72.70	**78.62**
C^3	PPL	Acc	Common Sense Reasoning	53.42	**54.47**	52.82	**53.92**	3,3	53.64	**54.58**
WPLC	PPL	ppl	Chinese WPLC	16.70	**19.18**	-	-	-	-	-

Pangu-α: Release (May 2021)

联合鹏城实验室发布业界首个两千亿参数量中文预训练语言模型–盘古 α

华为AI全栈：MindSpore + CANN + ModelArts + Atlas 900 集群



PanGu-α: Large-scale Autoregressive Pretrained Chinese Language Models with Auto-parallel Computation

Technical Report

Wei Zeng* Xiaozhe Ren* Teng Su* Hui Wang*
Yi Liao Zhiwei Wang Xin Jiang Zhenzhang Yang Kaisheng Wang Xiaoda Zhang
Chen Li Ziyan Gong Yifan Yao Xinjing Huang Jun Wang Jianfeng Yu Qi Guo
Yue Yu Yan Zhang Jin Wang Hengtao Tao Dasen Yan Zexuan Yi Fang Peng
Fangqing Jiang Han Zhang Lingfeng Deng Yehong Zhang Zhe Lin
Chao Zhang Shaojie Zhang Mingyue Guo Shanzhi Gu Gaojun Fan Yaowei Wang
Xuefeng Jin Qun Liu Yonghong Tian Wen Gao

PanGu-α Team

https://arxiv.org/abs/2104.12369

HUAWEI NOAH'S ARK LAB

Pangu-α: Text summarization

Prompt	“据媒体报道，最初华为只决定做车联网。2012年，华为开始进行汽车相关研究，当时在2012实验室下成立车联网实验室，研究电动汽车技术。 2013年，华为分别展示过为汽车内部提供3GWifi热点的DA6810、应用于保险行业及车队管理的汽车在线诊断系统DA3100，以及车载通信模块产品“ME909T”。 当时，华为对外的说法是，“华为终端只会聚焦于自己擅长的车联网通信解决方案。华为愿景是做到车联网模块第一，业绩上做到50亿 60亿元的规模”。 2019年5月27日，任正非签发华为组织变动文件，批准成立智能汽车解决方案BU部门，隶属于ICT管理委员会管理。智能汽车解决方案BU之下分：智能座舱部门、自动驾驶部门、智能车云服务部门。到这个阶段，华为已经确立了自动驾驶部门。 2020年7月12日晚，华为消费者业务CEO余承东官宣了HUAWEI HiCar车机系统，HiCar系统基于鸿蒙OS。HiCar可以利用移动设备的5G网络能力加强车机系统的数据交互能力，能够接入华为在AI、语音、计算机视觉等方面的能力。同时，能够调用车速、方向盘转角、档位模式、汽车环境光传感器在内的车身数据以及空调、车窗、喇叭等车身控制部件。 直到徐直军在2021年的分析师大会透露出一个重点信号，华为将汽车服务的重心定在自动驾驶软件上，才明确了华为的自动驾驶业务。据媒体报道，华为跟百度、小鹏、文远知行、驭势科技等一样，都是选择在乘用车领域重点布局。”的新闻摘要是
Generation	“华为在自动驾驶领域的布局，是基于华为在人工智能、大数据、云计算等方面的技术积累，以及华为在汽车领域的多年经验。”

Pangu-α: Title generation

Prompt	“据媒体报道，最初华为只决定做车联网。2012年，华为开始进行汽车相关研究，当时在2012实验室下成立车联网实验室，研究电动汽车技术。 2013年，华为分别展示过为汽车内部提供3GWifi热点的DA6810、应用于保险行业及车队管理的汽车在线诊断系统DA3100，以及车载通信模块产品“ME909T”。 当时，华为对外的说法是，“华为终端只会聚焦于自己擅长的车联网通信解决方案。华为愿景是做到车联网模块第一，业绩上做到50亿 60亿元的规模”。 2019年5月27日，任正非签发华为组织变动文件，批准成立智能汽车解决方案BU部门，隶属于ICT管理委员会管理。智能汽车解决方案BU之下分：智能座舱部门、自动驾驶部门、智能车云服务部门。到这个阶段，华为已经确立了自动驾驶部门。 2020年7月12日晚，华为消费者业务CEO余承东官宣了HUAWEI HiCar车机系统，HiCar系统基于鸿蒙OS。HiCar可以利用移动设备的5G网络能力加强车机系统的数据交互能力，能够接入华为在AI、语音、计算机视觉等方面的能力。同时，能够调用车速、方向盘转角、档位模式、汽车环境光传感器在内的车身数据以及空调、车窗、喇叭等车身控制部件。 直到徐直军在2021年的分析师大会透露出一个重点信号，华为将汽车服务的重心定在自动驾驶软件上，才明确了华为的自动驾驶业务。据媒体报道，华为跟百度、小鹏、文远知行、驭势科技等一样，都是选择在乘用车领域重点布局。”的新闻标题是《
Generation	华为在车联网业务方面有深厚积累，并将全面发力自动驾驶软件业务》

Pangu-α: Open domain QA

Prompt	阅读文章：株洲北站全称广州铁路（集团）公司株洲北火车站。除站场主体，另外管辖湘潭站、湘潭东站和三个卫星站，田心站、白马垅站、十里冲站，以及原株洲车站货房。车站办理编组、客运、货运业务。车站机关地址：湖南省株洲市石峰区北站路236号，邮编412001。株洲北站位于湖南省株洲市区东北部，地处中南路网，是京广铁路、沪昆铁路两大铁路干线的交汇处，属双向纵列式三级七场路网性编组站。车站等级为特等站，按技术作业性质为编组站，按业务性质为客货运站，是株洲铁路枢纽的主要组成部分，主要办理京广、沪昆两大干线四个方向货物列车的到发、解编作业以及各方向旅客列车的通过作业。每天办理大量的中转车流作业，并有大量的本地车流产生和集散，在路网车流的组织中占有十分重要的地位，是沟通华东、华南、西南和北方的交通要道，任务艰巨，作业繁忙。此外，株洲北站还有连接石峰区喻家坪工业站的专用线。株洲北站的前身是田心车站。请问：株洲北站的前身是哪个车站？回答：
Generation	田心车站

Prompt	中国的首都是哪里?
Generation	中国的首都是北京。
Prompt	中国、美国、日本和澳大利亚的首都分别是哪里?
Generation	中国的首都是北京，美国的首都是华盛顿，日本的首都是东京，澳大利亚的首都是堪培拉。
Prompt	五一川西行程规划
Generation	五一川西旅游活动主要集中在绵阳、成都、雅安、甘孜4市（州），主要景区有大九寨、新都桥、丹巴、海螺沟、泸定、康定、成都新会展中心。

Pangu-α: Dialog generation

我和妈妈上街的时候淋雨了。
好吧, 那你要注意身体哦!
谢谢关心。
我要上班了, 再见。
再见!

我也想吃饺子, 你还有吗。
我也是, 你想吃什么馅的?
我想吃肉馅的。
那我给你推荐一家饺子馆吧, 叫做
好啊!
这家店的人均价格是多少啊?
人均价格在元左右。
地址在哪儿呢?
地址在北京市朝阳区望京街望京SOHO西区10号楼(近地铁望京站)。
评分怎么样啊?
评分. 哦哦。

我来给大家读一段吧!
"盘古开天地, 天崩地裂, 日月无光, 山河破碎。"
这是我最喜欢的一首诗。
你喜欢这首诗吗?
我还挺喜欢的。
那你知道这首诗出自谁的笔下吗?
不知道啊!是谁写的呢?
是王勃的作品。
哦!原来是王勃啊!他可是个大才子呢!他写过很多好诗呢!你知道他是哪年出生的吗?
我不清楚啊!他出生于哪一年啊?
他出生于公元756年, 逝世于公元817年。你知道他有哪些作品吗?
我只知道《滕王阁序》和《秋日登洪府滕王阁》这两首诗吧!

Pangu-α: Stylized text generation

非常简单地在prompt注入不同场景，不同人物个性化信息，即可实现场景化，个性化，风格化的对话引擎：

Prompt注入职业信息：警察
你好啊！你叫什么名字？
你好，我叫范明明。
好的，明明你是做什么的呢？
我是名警察。
平时上班的主要都干些什么呢？
平时主要是跟警犬一起工作。
警犬?我听我朋友说起过,是个什么类型的呢?
是大型犬类。
大型犬类是不是都是很凶猛的呢?
也不一定，有的就很温顺。

Prompt注入个性化信息：甄嬛体（母子高考前对话）
儿臣不日将高考，能有心仪的学府完成学业真是极好的。
今儿个是开始填志愿了，急切选择学府虽是要紧，却也不能忘了专业二字。
儿臣是真真的不知所措了。
日常暇余之间常听闻五道口有一职业技术学院倒是极不错的。
儿臣倒也听先生提及，谓之清华，其学术氛围浓郁，众学子潜心学业。
整个五道口城楼街道行人驻足皆是热议清华学子,甚是羡慕。
儿臣就想着不日便要面临那番环境,且更想着能早日成为其中一员。可以说,此乃儿臣的荣耀。

Prompt注入个性化场景信息：快递客服
你好啊！你叫什么名字？
你好，我叫范明明。
好的，我是圆通的快递员
好的，有什么事情吗
你的货到了，在这签个字。
我先看看货。
不签收不能验货。
为什么?
我们公司的规定。
好,那我签收了。

Prompt注入职业信息：老师
你好啊！你叫什么名字？
你好，我叫范明明。
好的，明明你是做什么的呢？
我是名老师。
平时上班的主要都干些什么呢？
我平时上班是教小朋友数学的。
哇,是位小朋友的数学老师,真羡慕你,你平时有什么爱好呢?
我平常喜欢看电影。

Prompt注入个性化信息：人话版（母子高考前对话）
儿子马上要高考了，要是能考到心仪的大学就太好了。
今天开始填志愿了，学校很重要，也要选好专业。
我也很迷茫。
我经常听说五道口有一所职业学校非常不错。
我也经常听到老师提到，叫清华，学术氛围很浓，大家学习很刻苦。
我知道,清华是中国的最高学府。
我也知道,清华是中国的骄傲。

Prompt注入个性化场景信息：京东客服
你好啊！你叫什么名字？
你好，我叫范明明。
好的，我是中国京东商城的客户服务代表。请问你有什么事情吗？
我买的木瓜牛奶,牛奶送来的时候,发现没有发票,这个我可以补吗?
请问是这个订单号吗
对的

Pangu-α: Gaokao essay generation

盘古α生成效果 – 2021年北京高考作文

论生逢其时

每个人都生活在特定的时代，每个人在特定时代中的人生道路各不相同。

在同一个时代，有人慨叹生不逢时，有人只愿安分随时。有很多人因为种种原因而不能实现自己的理想。他们或因生活所迫、或因性格缺陷、或因家庭变故等等原因而无法实现理想；他们也曾想过要放弃这个目标，但是又没有勇气去面对现实生活中存在着太多不可预知和不可克服困难；他们也曾想过要改变现状、改变命运。当你真正去做了以后才发现自己当初所想与实际相差太远。

每个人都是自己的主人，每个人的命运都把握在自己手里。人生的道路是由自己选择的，每个人都有自己独特的人生道路。在这个世界上，没有谁可以主宰你的命运。但是，你可以选择怎样去走这条路？如何才能使自己不迷失方向、不走错路呢？我们应该正视现实生活中存在着太多太多不可知因素和难以预料因素。只有这样才能使我们更好地把握住人生道路上每个机会。

在人生的道路上，每个人都会遇到各种各样的困难和挫折。面对这些困难和挫折，我们要学会勇敢地面对、积极地战胜它们。只有这样，才能使自己不断地走向成功之路。

在我们这个充满竞争和压力的社会中，每个人都渴望成功，但成功并不是蹴而就、一帆风顺的。当我们遇到困难时，是否能保持一颗平静之心？是否能够坦然面对失败？这一切都取决于个人对待生活、对待工作、对待他人以及自身价值实现等方面所持有怎样种态度和价值观念。

"我不相信命运"这是很多年轻人经常挂在嘴边的话之一。然而事实上命运从来不会眷顾那些没有准备之人！如果你没有足够强大到足以应对各种困难与挑战的话，那么就不要去想着改变自己了！因为改变命运只能靠自己！

Pangu-α: Advertisement generation

使用Prompt输入推广关键信息，自动写出商业文案：

Prompt注入信息：华为云相关描述

华为坚持开放，合作，共赢的云生态，作为“智能世界”的黑土地，帮助合作伙伴快速融入当地生态。华为云携手合作伙伴联合创新，持续为客户和伙伴创造价值。截至2019年底，华为云已上线210+云服务、210+解决方案，服务全球众多知名企业。

华为云已与国际领先云基础设施提供商合作，在全球数十个国家部署云服务，帮助客户连接国际网络，全球服务网络。利用华为云，全球顶尖跨国企业将平台部署在全球各地，用户能享受到低成本、低风险的云服务，同时还能随时随地安全全地访问全球信息资源。

华为云帮助世界级企业搭建云网络、云应用，通过领先的技术和丰富的业务经验，帮助他们将业务快速扩展至5G和物联网领域。我们的客户既有来自全球的顶级航空公司、汽车制造商、半导体公司等行业龙头，也有众多中国的优秀企业。在云服务领域，华为拥有业内最丰富的云服务产品线，拥有独一无二的业务体验，能够满足客户对云服务的更高期望。华为云与合作伙伴的合作有力推动了5G发展，不断促进云服务生态体系的成熟。华为致力于通过开放的云生态系统，与合作伙伴共同帮助企业客户走向成功。

JABER and SABER: Junior and Senior Arabic BERt

JABER and SABER: Junior and Senior Arabic BERt

Abbas Ghaddar[1], Yimeng Wu[1], Ahmad Rashid[1], Khalil Bibi[1], Mehdi Rezagholizadeh[1], Chao Xing[1], Yasheng Wang[1], Duan Xinyu[2], Zhefeng Wang[2], Baoxing Huai[2], Xin Jiang[1], Qun Liu[1] and Philippe Langlais[3]

[1] Huawei Technologies Co., Ltd.

[2] Huawei Cloud Computing Technologies Co., Ltd

[3] RALI/DIRO, Université de Montréal, Canada

{abbas.ghaddar,yimeng.wu,ahmad.rashid}@huawei.com

{khalil.bibi,mehdi.rezagholizadeh}@huawei.com

{xingchao.ml,duanxinyu,wangyasheng}@huawei.com

{wangzhefeng,huaibaoxing,jiang.xin,qun.liu}@huawei.com

Preprint: https://arxiv.org/pdf/2112.04329v3.pdf

JABER and SABER: ALUE Leaderboard

ALUE

Arabic Language Understanding Evaluation

ALUE Paper Cite Code Tasks Leaderboard FAQ Diagnostics Submit Login

Leaderboard

Rank	Name	Model	Details	Score	MQ2Q	MDD	SVREG	SEC	FID	OOLD	XNLI	OHSD	DIAG
1	Huawei Noah's Ark Lab MTL	SABER		77.3	93.3	66.5	79.2	38.8	86.5	93.4	76.3	84.1	26.2
2	Huawei Noah's Ark Lab MTL	JABER		73.7	93.1	64.1	70.9	31.7	85.3	91.4	73.4	79.6	24.4
3	ALUE Baseline	ARABIC-BERT		67.1	85.7	59.7	55.1	25.1	82.2	89.5	61.0	78.7	19.6
4	ALUE Baseline	BERT Multi-lingual Cased		61.0	83.2	61.3	33.9	14.0	81.6	80.3	63.1	70.5	19.0
5	ALUE Baseline	BERT Multi-lingual Uncased		58.6	75.8	58.0	32.0	13.8	81.0	79.8	57.9	70.6	15.1

ALUE Leaderboard https://www.alue.org/leaderboard

Wukong: A Large-scale Chinese Cross-modal Pre-trained Model and Dataset

Wukong: 100 Million Large-scale Chinese Cross-modal Pre-training Dataset and A Foundation Framework

Jiaxi Gu[1*], Xiaojun Meng[1*], Guansong Lu[1], Lu Hou[1], Minzhe Niu[1], Hang Xu[1†], Xiaodan Liang[2†], Wei Zhang[1], Xin Jiang[1], Chunjing Xu[1]

Technical report: https://arxiv.org/abs/2202.06767.pdf

Wukong: Dataset

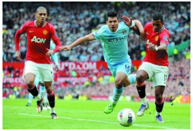
Table 1: An overview of datasets for VLP model pre-training.

Dataset	Language	Availability	Image-text pairs
Flickr30k (Young et al., 2014)	English	✓	31,783
CxC (Parekh et al., 2020)	English	✓	247,315
SBU Captions (Ordonez et al., 2011b)	English	✓	1,000,000
Product1M (Zhan et al., 2021)	Chinese	✓	1,000,000
CC12M (Changpinyo et al., 2021)	English	✓	12,000,000
YFCC100M (Thomee et al., 2016)	English	✓	99,200,000
WIT (Srinivasan et al., 2021)	multilingual	✓	11,500,000
LAION-400M (Schuhmann et al., 2021)	English	✓	400,000,000
JFT-300M (Sun et al., 2017)	English	✗	300,000,000
JFT-3B (Zhai et al., 2021a)	English	✗	3,000,000,000
IG-3.5B-17k (Mahajan et al., 2018)	English	✗	3,500,000,000
M6-Corpus (Lin et al., 2021)	Chinese	✗	60,500,000
Wukong (Ours)	Chinese	✓	101,483,885

Wukong: Data Examples

狗子示意来访人员要想进去，先过来扫码，狗子还特意下来用嘴巴对着 (*The dog signaled to the visitors to scan the code first before entrance, and the dog also deliberately came down and pointed his mouth at it.*)

你好，我们是社区工作人员，是来做接种疫苗排查工作的 (*Hello, we are community workers and are here to do vaccination screening.*)



13-14赛季 英超第5轮 曼城 vs 曼联 13.09.22 (*13-14 Premier League Round 5 Manchester City vs Manchester United 13.09.22*)

中国骄傲中国女排成功抵达东京不到6天就将在赛场上再展风采 (*China pride, the Chinese women's volleyball team, will show its style on the field in less than 6 days right after its arrival in Tokyo*)

简欧三居室酒柜装修效果图 (*Renderings of the decoration of the wine cabinet in the three bedrooms of Europe*)

【互邦工厂旗舰店】上海互邦轮椅钢管轻便手动折叠轮椅 (*【Hubang factory flagship store】 Shanghai Hubang wheelchair steel pipe lightweight manual folding wheelchair*)

Figure 2: Examples of image-text pairs in our Wukong dataset. This large-scale dataset covers a diverse range of concepts from the web, and suits vision-language pre-training.

HUAWEI NOAH'S ARK LAB

Wukong: Base model

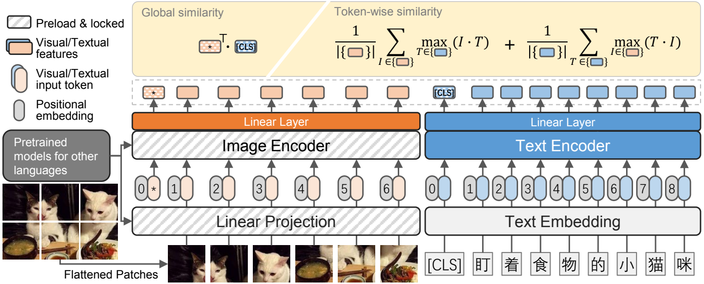


Figure 1: The base model consists of an image encoder and a text encoder with visual tokens and textual tokens as inputs. The input tokens from the two modalities are then concatenated and are added with position embeddings indicating token positions. For the image encoder, weights from an external model trained on datasets of other language are preloaded and locked. We compute the global similarity and token-wise similarity in the contrastive pre-training loss.

Wukong: Visualization of word-patch alignment

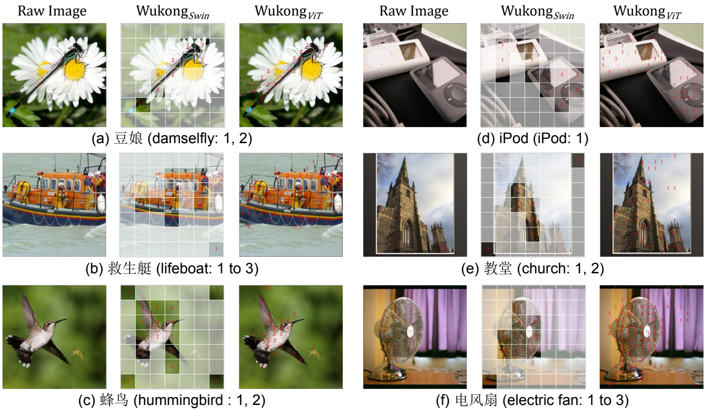


Figure 4: Visualization of word-patch alignment. We randomly choose six classes in the Chinese ImageNet dataset. Each Chinese label name is used as a prompt, whose English text is described in the parentheses. Behind which, the tail numbers indicate the location indices of this class label in the tokenized textual input. Take (a) as an example, the number 0 always represents `[CLS]`, the number 1 is the tokenized “豆” and the number 2 is “娘”. Indices of the tokenized label name are highlighted in red.

SPIRAL: Speech Pre-Training with Perturbation-Invariant Representation Learning

SPIRAL: SELF-SUPERVISED PERTURBATION-INVARIANT REPRESENTATION LEARNING FOR SPEECH PRE-TRAINING

Wenyong Huang, Zhenhe Zhang, Yu Ting Yeung, Xin Jiang, Qun Liu
Huawei Noah's Ark Lab
{wenyong.huang,zhangzhenhe1,yeung.yu.ting}@huawei.com
{jiang.xin,qun.liu}@huawei.com

Preprint: https://arxiv.org/pdf/2201.10207.pdf (Accepted by ICLR 2022)

SPIRAL: Model architecture

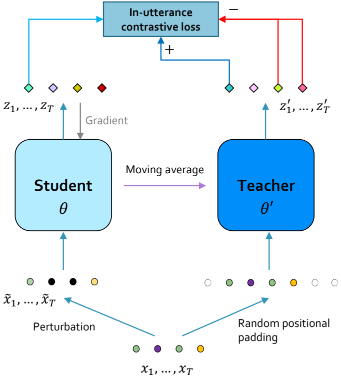


Figure 1: Illustration of SPIRAL architecture for speech pre-training.

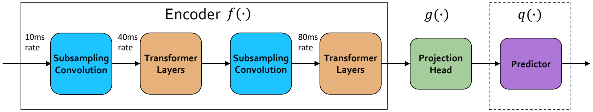


Figure 2: The architecture of the student model in SPIRAL. The frame rate of input is denoted as '10/40/80 ms'. The dashed line indicates the optional predictor which can be removed with small performance degradation. The structure of the teacher model is the same but without the predictor.

HUAWEI NOAH'S ARK LAB

Content

Our Work

Our Models

Efficient Training and Deployment

Applications of PLMs

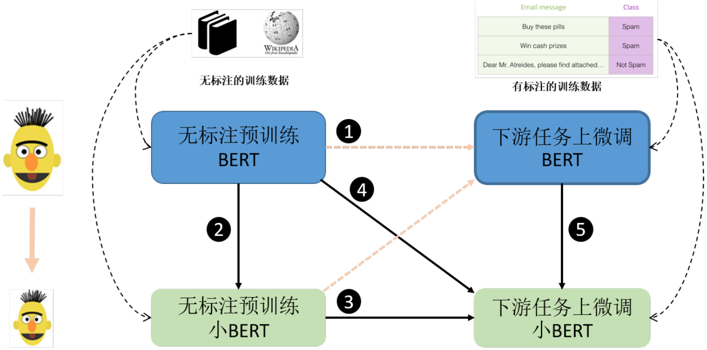
TinyBERT

TinyBERT: Distilling BERT for Natural Language Understanding

Xiaoqi Jiao[1*†], **Yichun Yin**[2*‡], **Lifeng Shang**[2‡], **Xin Jiang**[2]
Xiao Chen[2], **Linlin Li**[3], **Fang Wang**[1‡] **and Qun Liu**[2]
[1]Key Laboratory of Information Storage System, Huazhong University of Science and Technology, Wuhan National Laboratory for Optoelectronics
[2]Huawei Noah's Ark Lab
[3]Huawei Technologies Co., Ltd.
{jiaoxiaoqi,wangfang}@hust.edu.cn
{yinyichun,shang.lifeng,jiang.xin}@huawei.com
{chen.xiao2,lynn.lilinlin,qun.liu}@huawei.com

Published in: EMNLP 2020 Findings (Long paper)

TinyBERT: Overview of Knowledge Distillation for Pre-trained Language Models



迁移1：基于无标注预训练的BERT到基于下游任务微调的BERT

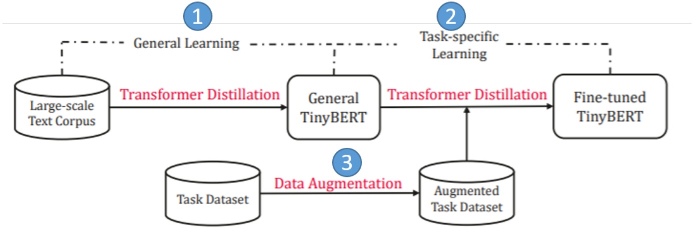
迁移2+3：通过两步，将在无标注语料学到的知识迁移到小模型

迁移4：通过一步，将无标注语料学到的知识迁移到小模型

迁移5：将下游任务上的老师迁移到小模型

HUAWEI NOAH'S ARK LAB

TinyBERT: Knowledge Distillation Procedure



(1) First Step Distillation: GD(General Distillation)
  - Transfer knowledge from pre-trained teacher BERT to general TinyBERT

(2) Second Step Distillation: TD(Task-specific Distillation)
  - Transfer knowledge from fine-tuned teacher BERT to fine-tuned TinyBERT

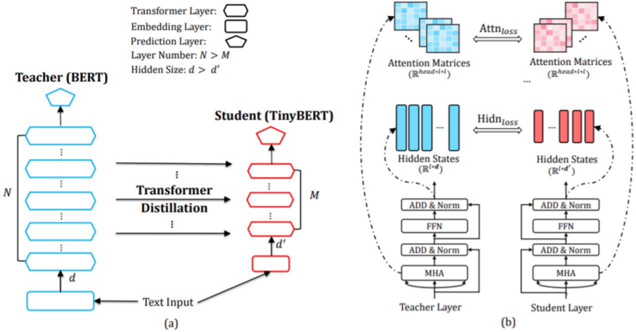
(3) DA(Data Augmentation)

- Source code: link

TinyBERT: Loss Function

$$\mathcal{L}_{model} = \sum_{m=0}^{M+1} \lambda_m \mathcal{L}_{layer}(S_m, T_{g(m)})$$

- Computing loss in embedding layer, tranformer layer and prediction layer
- In transformer lays, computing the loss in hidden states and attention matrix



$$\mathcal{L}_{layer}(S_m, T_{g(m)}) = \begin{cases} \mathcal{L}_{embd}(S_0, T_0), & m = 0 \\ \mathcal{L}_{hidn}(S_m, T_{g(m)}) + \mathcal{L}_{attn}(S_m, T_{g(m)}), & M \geq m > 0 \\ \mathcal{L}_{pred}(S_{M+1}, T_{N+1}), & m = M + 1 \end{cases}$$

TinyBERT: Data Enhancement

- We generate synthetic data in TD, because the fine-tuning data size is usually small.
  - We randomly replace some tokens in each sample in fine-tuning
  - We use BERT and Glove to choose similar words (tokens)
  - We use a threshold to control the percentage of the tokens to be replaced
  - We find that the performance achieve the best when the size of synthetic data is 20 times of the original fine-tuning data size

```
[Mask][Mask][Mask][Mask]歌曲
```

```
[帮][我][搜][索]歌曲
[播][放][一][首]歌曲
[给][我][搜][索]歌曲
[给][我][播][放]歌曲
[给][我][放][首]歌曲
[给][我][唱][首]歌曲
[帮][我][播][放]歌曲
```

TinyBERT: Experimental Result (GLUE)

System	#Params	#FLOPS	Speedup	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	Avg
$\text{BERT}_{\text{BASE}}$ (Teacher)	109M	22.5B	1.0x	83.9/83.4	71.1	90.9	93.4	52.8	85.2	87.5	67.0	79.5
$\text{BERT}_{\text{TINY}}$	14.5M	1.2B	9.4x	75.4/74.9	66.5	84.8	87.6	19.5	77.1	83.2	62.6	70.2
$\text{BERT}_{\text{SMALL}}$	29.2M	3.4B	5.7x	77.6/77.0	68.1	86.4	89.7	27.8	77.0	83.4	61.8	72.1
BERT_4-PKD	52.2M	7.6B	3.0x	79.9/79.3	70.2	85.1	89.4	24.8	79.8	82.6	62.3	72.6
DistilBERT_4	52.2M	7.6B	3.0x	78.9/78.0	68.5	85.2	91.4	32.8	76.1	82.4	54.1	71.9
$\text{MobileBERT}_{\text{TINY}}$†	15.1M	3.1B	-	81.5/81.6	68.9	**89.5**	91.7	**46.7**	80.1	**87.9**	65.1	**77.0**
TinyBERT_4 (ours)	14.5M	1.2B	9.4x	**82.5/81.8**	**71.3**	87.7	**92.6**	44.1	**80.4**	86.4	**66.6**	**77.0**
BERT_6-PKD	67.0M	11.3B	2.0x	81.5/81.0	70.7	89.0	92.0	-	-	85.0	65.5	-
DistilBERT_6	67.0M	11.3B	2.0x	82.6/81.3	70.1	88.9	92.5	49.0	81.3	86.9	58.4	76.8
TinyBERT_6 (ours)	67.0M	11.3B	2.0x	**84.6/83.2**	**71.6**	**90.4**	**93.1**	**51.1**	**83.7**	**87.3**	**70.0**	**79.4**

AutoTinyBERT

AutoTinyBERT: Automatic Hyper-parameter Optimization for Efficient Pre-trained Language Models

Yichun Yin[1], Cheng Chen[2*], Lifeng Shang[1], Xin Jiang[1], Xiao Chen[1], Qun Liu[1]
[1]Huawei Noah's Ark Lab
[2]Department of Computer Science and Technology, Tsinghua University
{yinyichun,shang.lifeng,jiang.xin,chen.xiao2,qun.liu}@huawei.com
c-chen19@mails.tsinghua.edu.cn

Accepted by ACL 2021 Proceedings (Long paper)

AutoTinyBERT: Overview

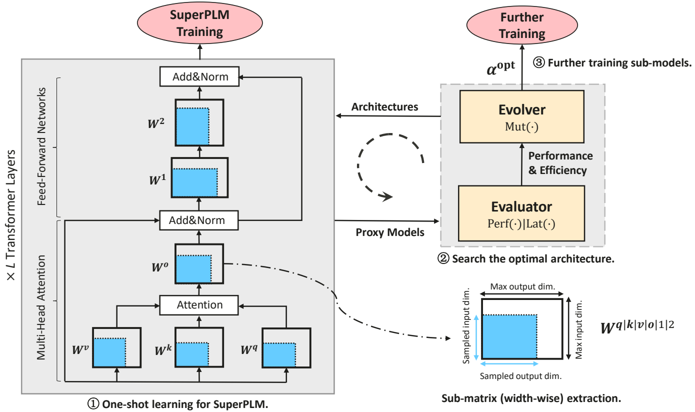


Figure 2: Overview of AutoTinyBERT. We first train an effective SuperPLM with one-shot learning, where the objectives of pre-training or task-agnostic BERT distillation are used. Then, given a specific latency constraint, we perform an evolutionary algorithm on the SuperPLM to search optimal architectures. Finally, we extract the corresponding sub-models based on the optimal architectures and further train these models.

TernaryBERT

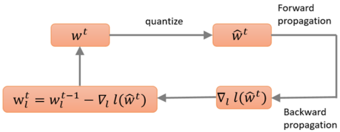
TernaryBERT: Distillation-aware Ultra-low Bit BERT

Wei Zhang,[*] Lu Hou,[*] Yichun Yin,[*] Lifeng Shang, Xiao Chen, Xin Jiang, Qun Liu
Huawei Noah's Ark Lab
{zhangwei379, houlu3, yinyichun, shang.lifeng, chen.xiao2, jiang.xin, qun.liu}@huawei.com

Accepted by EMNLP 2020 Proceedings (Long paper)

TernaryBERT: Overview

- Combine the knowledge distillaton technology with the extremely low bit (1/2 bit) quantization



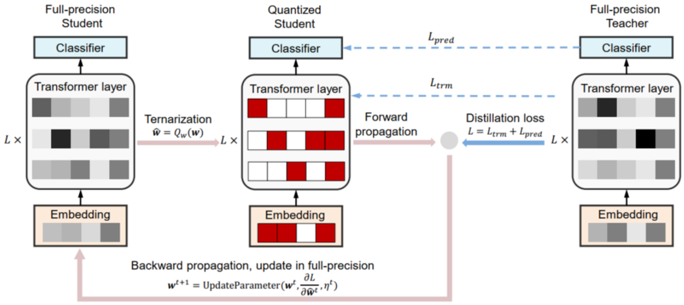


Figure 2: Depiction of the proposed distillation-aware ternatization of BERT model.

HUAWEI NOAH'S ARK LAB

TernaryBERT: Experimental results

- Model weights W: 2-bit quantization, a.k.a. -1,0,1
- Word embedding E: 2-bit quantization, a.k.a. -1,0,1
- Activation A: 8-bit quantization

Table 1: Development set results of quantized BERT and TinyBERT on the GLUE benchmark. We abbreviate the quantization bits for weights of Transformer layers as “W-bit”, word embedding as “E-bit”, activations as “A-bit”.

	W-E-A (#bits)	Size (MB)	MNLI-m/mm	QQP	QNLI	SST-2	CoLA	MRPC	STS-B	RTE
BERT	32-32-32	418 (×1)	84.5/84.9	87.5/90.9	92.0	93.1	58.1	90.6/86.5	89.8/89.4	71.1
Q-BERT	2-8-8	43 (×9.7)	76.6/77.0	-	-	84.6	-	-	-	-
Q2BERT	2-8-8	43 (×9.7)	47.2/47.3	67.0/75.9	61.3	80.6	0	81.2/68.4	4.4/4.7	52.7
TernaryBERT (TWN)	2-2-8	28 (×14.9)	83.3/83.3	86.7/90.1	91.1	92.8	**55.7**	91.2/87.5	**87.9/87.7**	**72.9**
TernaryBERT (LAT)	2-2-8	28 (×14.9)	**83.5/83.4**	86.6/90.1	**91.5**	92.5	54.3	91.1/87.0	87.9/87.6	72.2
TernaryTinyBERT (TWN)	2-2-8	18 (×23.2)	83.4/83.8	**87.2/90.5**	89.9	**93.0**	53.0	**91.5/88.0**	86.9/86.5	71.8
Q-BERT	8-8-8	106 (×3.9)	83.9/83.8	-	-	92.9	-	-	-	-
Q8BERT	8-8-8	106 (×3.9)	-/-	**88.0/-**	90.6	92.2	58.5	89.6/-	89.0/-	68.8
Ours (BERT)	8-8-8	106 (×3.9)	84.2/84.7	87.1/90.5	**91.8**	**93.7**	**60.6**	**90.8/87.3**	**89.7/89.3**	71.8
Ours (TinyBERT)	8-8-8	65 (×6.4)	**84.4/84.6**	87.9/91.0	91.0	93.3	54.7	90.0/89.4	91.2/87.5	**72.2**

TernaryBERT: Trading between precision and model size

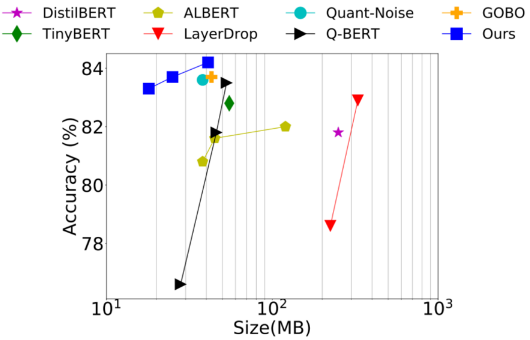


Figure 1: Model Size vs. MNLI-m Accuracy. Our proposed method outperforms other BERT compression methods. Details are in Section 4.4.

Method	W-E-A (#bits)	Size (MB)	Accuracy (%)
DistilBERT	32-32-32	250	81.6
TinyBERT	32-32-32	55	82.8
ALBERT-E64	32-32-32	38	80.8
ALBERT-E128	32-32-32	45	81.6
ALBERT-E256	32-32-32	62	81.5
ALBERT-E768	32-32-32	120	82.0
LayerDrop-6L	32-32-32	328	82.9
LayerDrop-3L	32-32-32	224	78.6
Quant-Noise	PQ	38	83.6
Q-BERT	2/4-8-8	53	83.5
Q-BERT	2/3-8-8	46	81.8
Q-BERT	2-8-8	28	76.6
GOBO	3-4-32	43	83.7
GOBO	2-2-32	28	71.0
3-bit BERT	3-3-8	41	84.2
3-bit TinyBERT	3-3-8	25	83.7
TernaryBERT	2-2-8	28	83.5
TernaryTinyBERT	2-2-8	18	83.4

BinaryBERT

BinaryBERT: Pushing the Limit of BERT Quantization

Haoli Bai[1], Wei Zhang[2], Lu Hou[2], Lifeng Shang[2],
Jing Jin[3], Xin Jiang[2], Qun Liu[2], Michael Lyu[1], Irwin King[1]
[1] The Chinese University of Hong Kong
[2]Huawei Noah's Ark Lab, [3]Huawei Technologies Co., Ltd.
{hlbai, lyu, king}@cse.cuhk.edu.hk
{zhangwei379, houlu3, shang.lifeng, jinjing12, jiang.xin, qun.liu}@huawei.com

Accepted by ACL-IJCNLP 2021 Proceedings (Long paper)

BinaryBERT: Overall workflow

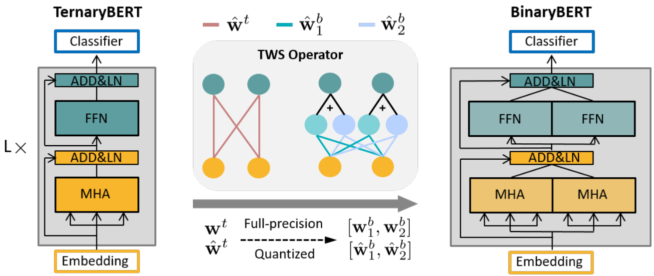


Figure 4: The overall workflow of training BinaryBERT. We first train a half-sized ternary BERT model, and then apply ternary weight splitting operator (Equations (6) and (7)) to obtain the latent full-precision and quantized weights as the initialization of the full-sized BinaryBERT. We then fine-tune BinaryBERT for further refinement.

DynaBERT

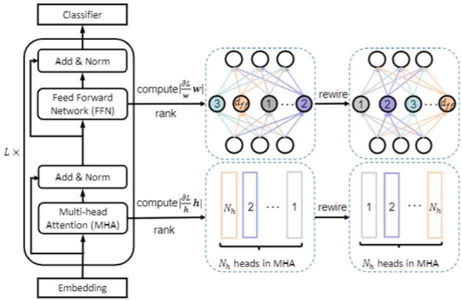
DynaBERT: Dynamic BERT with Adaptive Width and Depth

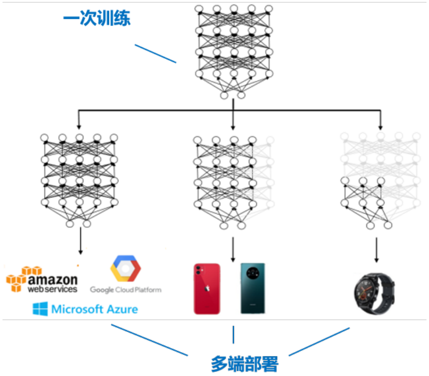
Lu Hou[1], **Zhiqi Huang**[2], **Lifeng Shang**[1], **Xin Jiang**[1], **Xiao Chen**[1], **Qun Liu**[1]
[1]Huawei Noah's Ark Lab
{houlu3,shang.lifeng,Jiang.Xin,chen.xiao2,qun.liu}@huawei.com
[2]Peking University, China
zhiqihuang@pku.edu.cn

Published in: NeurIPS 2020 Proceedings (Spotlight paper)

DynaBERT: Scalable Pre-trained Language Model

- Train once, deploy in multiple devices / scenarios
- Select different sub-networks flexibly in deployment
- Define different depths and widths for sub-networks
- Ranking the neurons / attention heads according to their importance
- The more important neurons / heads will be more shared

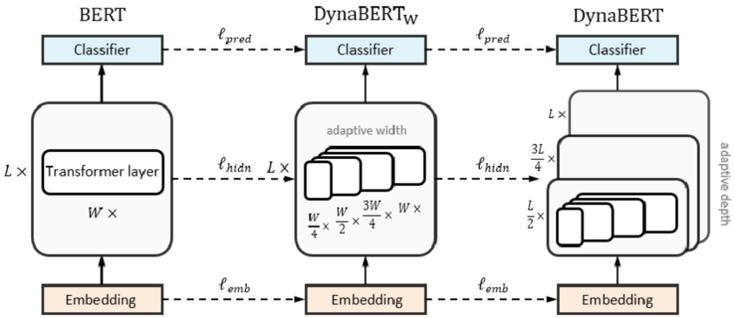




HUAWEI NOAH'S ARK LAB

DynaBERT: Training

- First, training the scalable network in different widths
- Second, training the scalable network jointly in different widths and depths
- Use knowledge distillation like TinyBERT

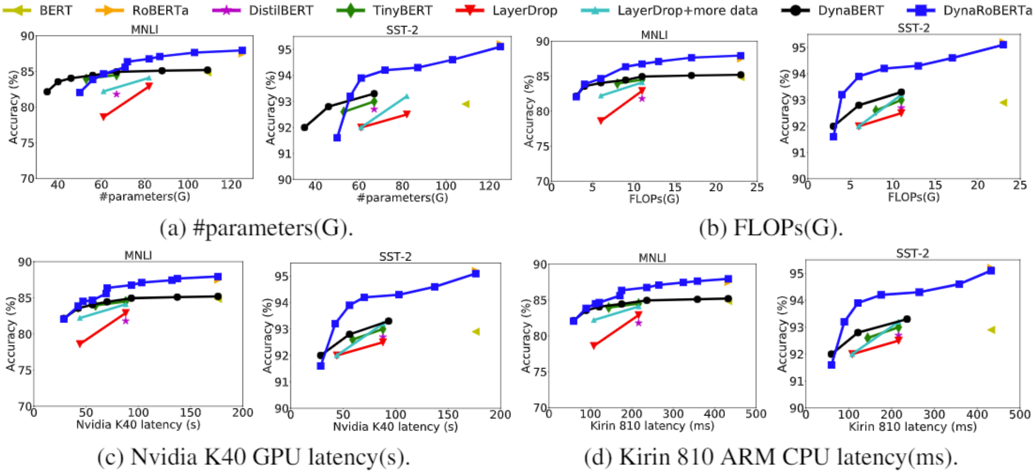


DynaBERT: Experimental results

Table 1: Development set results of the GLUE benchmark using DynaBERT and DynaRoBERTa with different width and depth multipliers (m_w, m_d).

Method		CoLA			STS-B			MRPC			RTE		
$\text{BERT}_{\text{BASE}}$		58.1			89.8			87.7			71.1		
	(m_w, m_d)	1.0x	0.75x	0.5x	1.0x	0.75x	0.5x	1.0x	0.75x	0.5x	1.0x	0.75x	0.5x
DynaBERT	1.0x	59.7	59.1	54.6	**90.1**	89.5	88.6	86.3	85.8	85.0	72.2	71.8	66.1
	0.75x	**60.8**	59.6	53.2	90.0	89.4	88.5	**86.5**	85.5	84.1	71.8	**73.3**	65.7
	0.5x	58.4	56.8	48.5	89.8	89.2	88.2	84.8	84.1	83.1	72.2	72.2	67.9
	0.25x	50.9	51.6	43.7	89.2	88.3	87.0	83.8	83.8	81.4	68.6	68.6	63.2
		MNLI-(m/mm)			QQP			QNLI			SST-2		
$\text{BERT}_{\text{BASE}}$		84.8/84.9			90.9			92.0			92.9		
	(m_w, m_d)	1.0x	0.75x	0.5x	1.0x	0.75x	0.5x	1.0x	0.75x	0.5x	1.0x	0.75x	0.5x
DynaBERT	1.0x	**84.9/85.5**	84.4/85.1	83.7/84.6	**91.4**	**91.4**	91.1	92.1	91.7	90.6	93.2	93.3	92.7
	0.75x	84.7/85.5	84.3/85.2	83.6/84.4	**91.4**	91.3	91.2	92.2	91.8	90.7	93.0	93.1	92.8
	0.5x	84.7/85.2	84.2/84.7	83.0/83.6	91.3	91.2	91.0	**92.2**	91.5	90.0	**93.3**	92.7	91.6
	0.25x	83.9/84.2	83.4/83.7	82.0/82.3	90.7	91.1	90.4	91.5	90.8	88.5	92.8	92.0	92.0

DynaBERT: Trading between precision and model size



(a) #parameters(G). (b) FLOPs(G).

(c) Nvidia K40 GPU latency(s). (d) Kirin 810 ARM CPU latency(ms).

DynaBERT: Visualization of attention matrix

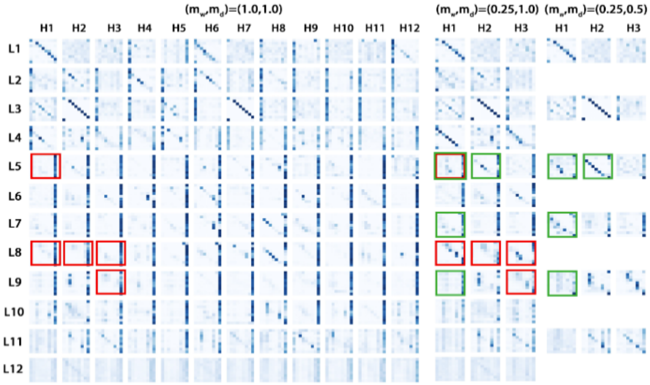


Figure 5: Attention maps in sub-networks with different widths and depths in DynaBERT trained on CoLA.

We observed that the attention patterns in small models integrate multiple patterns in large models

HUAWEI NOAH'S ARK LAB

GhostBERT

GhostBERT: Generate More Features with Cheap Operations for BERT

Zhiqi Huang[1], Lu Hou[2], Lifeng Shang[2], Xin Jiang[2], Xiao Chen[2], Qun Liu[2]
[1]Peking University, [2]Huawei Noah's Ark Lab
zhiqihuang@pku.edu.cn, {houlu3, shang.lifeng, jiang.xin, chen.xiao, qun.liu}@huawei.com

Accepted by ACL 2021 Proceedings （Long paper)

GhostBERT: Motivation

- Redundant features (**feature maps, attention pattern**) are similar

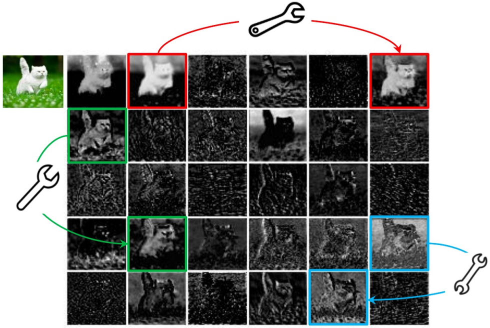


Figure1: Feature maps are similar

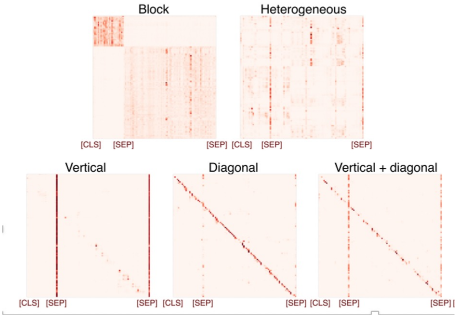
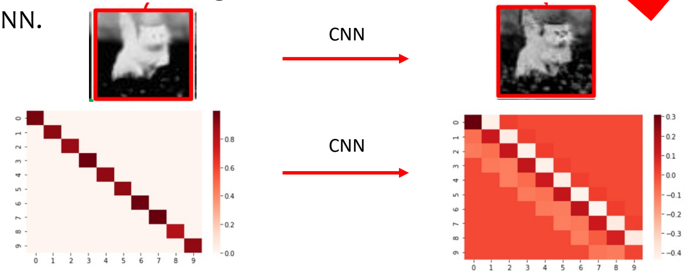


Figure2: Attention patterns in BERT

[1] Han et al., "GhostNet: More Features From Cheap Operations" CVPR 2021.
[2] Kovaleva et al., "Revealing the Dark Secrets of BERT." EMNLP 2019.

GhostBERT: Motivation

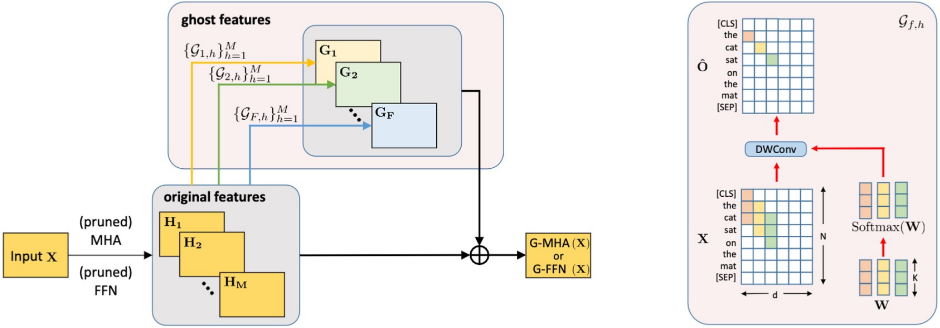
- Can we directly discarded the redundant features?
  - The theory behind why more features help can be related to how over-parameterized neural networks benefit both training [1] and generalization [2].
- Can we use other operations to generate redundant features?
  - For example, CNN.



[3] Samet et al., "Overparameterized nonlinear learning: Gradient descent takes the shortest path?" *ICML* 2019.

[4] Yuan et al., "Generalization bounds of stochastic gradient descent for wide and deep neural networks." *arXiv:1905.13210*, 2019.

GhostBERT: Methods



(a) Adding ghost modules $\{\mathcal{G}_{f,h}\}_{f=1,h=1}^{F,M}$ to MHA and FFN.

(b) Ghost Module $\mathcal{G}_{f,h}$.

Figure3: Using ghost modules to generate more features in BERT. G-MHA/FFN stands for Ghost-MHA/FFN.

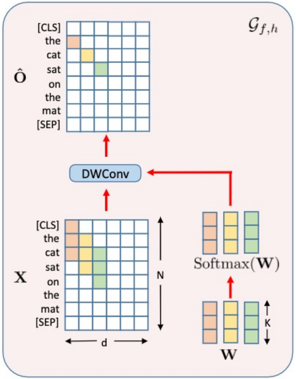
GhostBERT: Methods

- Convolution Type

$$O_{i,c} = \mathrm{DWConv}(\mathbf{X}_{:,c}, \mathbf{W}_{c,:}, i, c)$$
$$= \sum_{m=1}^{k} W_{c,m} \cdot X_{i-\lceil \frac{k+1}{2} \rceil + m, c}.$$

- Normalization

$$\hat{O}_{i,c} = \mathrm{DWConv}(\mathbf{X}_{:,c}, \mathrm{Softmax}(\mathbf{W}_{c,:}), i, c).$$



(b) Ghost Module $\mathcal{G}_{f,h}$.

GhostBERT: Experimental results

- 1. With only 55.3K more parameters (0.05% of BERT) and 14.2M more FLOPs (0.06% of BERT), adding ghost modules to pre-trained model increases the accuracy.

Model-Size	FLOPs(G)	#params(M)	MNLI	QNLI	QQP	RTE	SST-2	MRPC	CoLA	STS-B	Avg.	
BERT-base (Devlin et al., 2019)	22.5	110	84.5	92.0	90.9	71.1	92.9	87.8	58.1	89.8	83.4	
GhostBERT ($m = 12/12$)	22.5	110	84.7	92.3	91.1	71.8	93.0	88.0	63.6	89.7	**84.3**	+0.9
GhostBERT ($m = 9/12$)	16.9	88	84.8	92.1	91.2	72.6	92.6	87.5	61.1	89.8	84.0	
GhostBERT ($m = 6/12$)	11.3	67	84.7	92.2	91.2	72.2	92.9	87.3	58.1	89.2	83.5	
GhostBERT ($m = 3/12$)	5.8	46	84.3	91.6	91.4	72.9	94.6	86.5	53.9	89.2	83.1	
GhostBERT ($m = 1/12$)	2.0	32	82.8	90.0	90.5	66.1	92.8	86.0	46.1	87.8	80.3	
RoBERTa-base (Liu et al., 2019)	22.5	125	87.6	92.8	91.9	78.7	94.8	90.2	63.6	91.2	86.4	
GhostRoBERTa ($m = 12/12$)	22.5	125	88.0	93.1	91.9	80.5	95.3	90.7	65.0	91.3	**87.0**	+0.6
GhostRoBERTa ($m = 9/12$)	16.9	103	87.6	92.9	91.9	79.4	95.4	89.0	60.8	90.7	86.0	
GhostRoBERTa ($m = 6/12$)	11.3	82	86.8	92.6	91.6	77.6	94.4	89.7	57.6	90.3	85.1	
GhostRoBERTa ($m = 3/12$)	5.8	61	86.1	91.7	91.2	73.6	94.5	88.0	52.4	89.2	83.3	
GhostRoBERTa ($m = 1/12$)	2.0	47	82.1	89.2	90.5	66.1	93.7	83.3	39.8	87.4	79.0	
ELECTRA-small (Clark et al., 2020)	1.7	14	78.9	87.9	88.3	68.5	88.3	87.4	56.8	86.8	80.4	
GhostELECTRA-small ($m = 4/4$)	1.7	14	82.5	89.3	90.7	71.5	92.0	88.7	59.6	88.4	**82.8**	+2.4

(slides made by Zhiqi Huang)

GhostBERT: Experimental results

- 2. GhostBERT (m= 6/12) and GhostRoBERTa (m=9/12) get **similar** results to backbones
- 3. When the compression rate increases (i.e., **m = 3/12**, 1/12), we still achieve **99.6% performance** (resp. 96.3%) with only **25% FLOPs** (resp. 8%) of BERT-base.

Model-Size	FLOPs(G)	#params(M)	MNLI	QNLI	QQP	RTE	SST-2	MRPC	CoLA	STS-B	Avg.
BERT-base (Devlin et al., 2019)	22.5	110	84.5	92.0	90.9	71.1	92.9	87.8	58.1	89.8	83.4
GhostBERT (m = 12/12)	22.5	110	84.7	92.3	91.1	71.8	93.0	88.0	63.6	89.7	**84.3**
GhostBERT (m = 9/12)	16.9	88	84.8	92.1	91.2	72.6	92.6	87.5	61.1	89.8	84.0
GhostBERT (m = 6/12)	11.3	67	84.7	92.2	91.2	72.2	92.9	87.3	58.1	89.2	83.5
GhostBERT (m = 3/12)	5.8	46	84.3	91.6	91.4	72.9	94.6	86.5	53.9	89.2	83.1
GhostBERT (m = 1/12)	2.0	32	82.8	90.0	90.5	66.1	92.8	86.0	46.1	87.8	80.3
RoBERTa-base (Liu et al., 2019)	22.5	125	87.6	92.8	91.9	78.7	94.8	90.2	63.6	91.2	86.4
GhostRoBERTa (m = 12/12)	22.5	125	88.0	93.1	91.9	80.5	95.3	90.7	65.0	91.3	**87.0**
GhostRoBERTa (m = 9/12)	16.9	103	87.6	92.9	91.9	79.4	95.4	89.0	60.8	90.7	86.0
GhostRoBERTa (m = 6/12)	11.3	82	86.8	92.6	91.6	77.6	94.4	89.7	57.6	90.3	85.1
GhostRoBERTa (m = 3/12)	5.8	61	86.1	91.7	91.2	73.6	94.5	88.0	52.4	89.2	83.3
GhostRoBERTa (m = 1/12)	2.0	47	82.1	89.2	90.5	66.1	93.7	83.3	39.8	87.4	79.0
ELECTRA-small (Clark et al., 2020)	1.7	14	78.9	87.9	88.3	68.5	88.3	87.4	56.8	86.8	80.4
GhostELECTRA-small (m = 4/4)	1.7	14	82.5	89.3	90.7	71.5	92.0	88.7	59.6	88.4	**82.8**

(slides made by Zhiqi Huang)

GhostBERT: Experimental results

- Comparison with Other Compression Methods.

Model	FLOPs(G)	#params(M)	MNLI	QNLI	QQP	RTE	SST-2	MRPC	CoLA	STS-B	Avg.
BERT-base (Devlin et al., 2019)	22.5	110	84.6	90.5	89.2	66.4	93.5	84.8	52.1	85.8	80.9
RoBERTa-base (Liu et al., 2019)	22.5	125	86.0	92.5	88.7	73.0	94.6	86.5	50.5	88.1	82.5
ELECTRA-small (Clark et al., 2020)	1.7	14	79.7	87.7	88.0	60.8	89.1	83.7	54.6	80.3	78.0
TinyBERT_6 (Jiao et al., 2020)	11.3	67	84.6	90.4	89.1	70.0	93.1	87.3	51.1	83.7	81.2
TinyBERT_4 (Jiao et al., 2020)	1.2	15	82.5	87.7	89.2	66.6	92.6	86.4	44.1	80.4	78.7
ConvBERT-medium (Jiang et al., 2020)	4.7	17	82.1	88.7	88.4	65.3	89.2	84.6	56.4	82.9	79.7
ConvBERT-small (Jiang et al., 2020)	2.0	14	81.5	88.5	88.0	62.2	89.2	83.3	54.8	83.4	78.9
MobileBERT w/o OPT (Sun et al., 2020)	5.7	25	84.3	91.6	88.3	70.4	92.6	84.5	51.1	84.8	81.0
MobileBERT (Sun et al., 2020)	5.7	25	83.3	90.6	-	66.2	92.8	-	50.5	84.4	-
MobileBERT-tiny (Sun et al., 2020)	3.1	15	81.5	89.5	-	65.1	91.7	-	46.7	80.1	-
GhostBERT (m = 12/12)	22.5	110	84.6	91.1	89.3	70.2	93.1	86.9	54.6	83.8	81.7
GhostBERT (m = 9/12)	16.9	88	84.9	91.0	88.6	69.2	92.9	86.1	53.7	84.0	81.3
GhostBERT (m = 6/12)	11.3	67	84.2	90.8	89.1	69.6	93.1	84.0	53.4	83.1	80.9
GhostBERT (m = 3/12)	5.8	46	83.8	90.7	89	68.6	93.2	82.5	51.3	82.5	80.2
GhostBERT (m = 1/12)	2.0	32	82.5	89.3	88.7	65.0	92.9	81.0	41.3	80.0	77.6
GhostRoBERTa (m = 12/12)	22.5	125	87.9	93.0	89.6	74.6	95.1	88.0	52.4	88.3	83.6
GhostRoBERTa (m = 9/12)	16.9	103	87.7	92.6	89.5	73.0	94.5	85.7	51.9	87.1	82.8
GhostRoBERTa (m = 6/12)	11.3	82	86.3	92.1	89.5	71.5	94.5	86.8	51.2	87.0	82.4
GhostRoBERTa (m = 3/12)	5.8	61	85.5	91.2	89.1	68.5	93.4	85.3	48.9	84.7	80.8
GhostRoBERTa (m = 1/12)	2.0	47	81.3	88.6	88.5	62.8	92.1	82.8	39.7	81.8	77.2
GhostELECTRA-small (m = 4/4)	1.7	14	82.3	88.3	88.5	64.7	91.9	88.4	55.8	83.5	80.4

(slides made by Zhiqi Huang)

Conv-Transformer Transducer

Conv-Transformer Transducer: Low Latency, Low Frame Rate, Streamable End-to-End Speech Recognition

Wenyong Huang, Wenchao Hu, Yu Ting Yeung, Xiao Chen

Huawei Noah's Ark Lab

{wenyong.huang, huwenchao, yeung.yu.ting, chen.xiao2}@huawei.com

Published in Interspeech 2020

Conv-Transformer Transducer: model architecture

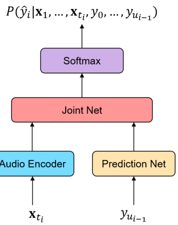


Figure 1: *Transducer model architecture*

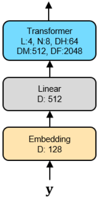


Figure 4: *Prediction net of Conv-Transformer Transducer. We denote the output dimensions of embedding layer and linear layer as **D**.*

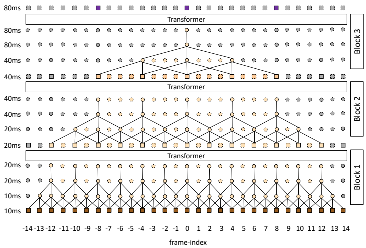


Figure 3: *Illustration of context window and frame rate change of convolution layers in audio encoder of Conv-Transformer Transducer at current input frame (frame-index = 0). Input and output of each block are represented by squares. Convolution layers are represented by circles. Only time dimension is shown. For each layer, activation with dashed outline is skipped in computation.*

Content

YueFu: GPT-based Chinese Traditional Poetry Generation

GPT-based Generation for Classical Chinese Poetry *

Yi Liao, Yasheng Wang, Qun Liu, Xin Jiang

Huawei Noah's Ark Lab

June 2019

Preprint: https://arxiv.org/pdf/1907.00151.pdf

YueFu: Model architecture and training data

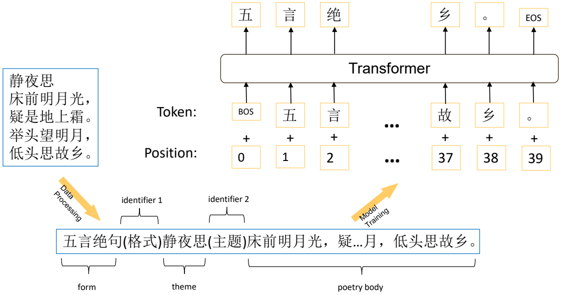


Figure 1: The process for training the poetry generation model

Training Phases	Corpus type	Corpus size
Pre-training	Chinese news	235M sentences
Fine-tuning	Jueju and Lüshi Cipai Couplet	250,000 Jueju and Lüshi 20,000 Cis 700,000 pairs of couplets

Table 2: Statistics of Training Data and Training epochs

HUAWEI NOAH'S ARK LAB

YueFu: WeChat app

微信小程序：

HUAWEI NOAH'S ARK LAB

YueFu: Examples

乐游西湖

西湖多胜事，清夜泛楼台。
月照山头树，春生水面苔。
香飘花片起，云过雨声来。
醉罢歌船发，游人晚未回。

伊妹赞

二八佳人质，娇羞不肯妆。
临风笑相语，问我意何长。
独立鸟关外，含颦花甸傍。
世间儿女多，无奈自迷茫。

闻秋虫有感

西风黄叶堕阶前，秋客愁思正可怜。
夜静子规啼滴滴，天寒乌鹊影翩翩。
一声塞雁江南去，几处家书海北连。
莫道征鸿无泪落，年年辛苦到燕然。

龙虾

江上初逢龙虾来，
满盘红锦粲成堆。
一时纵啖三千颗，
不直仙槎到酒杯。

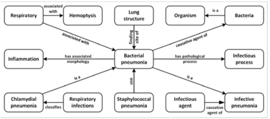
BERT-MK: Integrating Graph Contextualized Knowledge into PLMs

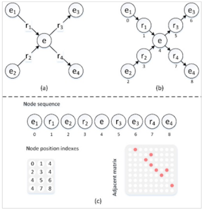
BERT-MK: Integrating Graph Contextualized Knowledge into Pre-trained Language Models

Bin He[1], Di Zhou[1], Jinghui Xiao[1], Xin Jiang[1], Qun Liu[1], Nicholas Jing Yuan[2], Tong Xu[3]
[1]Huawei Noah's Ark Lab
[2]Huawei Cloud & AI
[3]School of Computer Science, University of Science and Technology of China
{hebin.nlp, zhoudi7, xiaojinghui4, jiang.xin, qun.liu, nicholas.yuan}@huawei.com, tongxu@ustc.edu.cn

Published in: Findings of EMNLP 2020

BERT-MK: Model overview



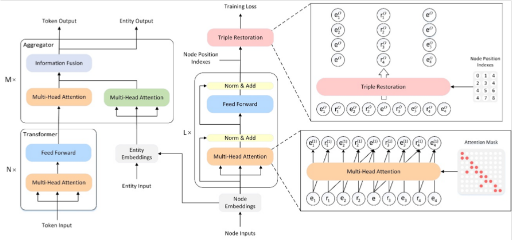


Method

- Generate subgraphs from knowledge graph;
- Learn graph contextualized knowledge;
- Integrate knowledge into the language model;

Result

- BERT-MK achieves better performance than previous biomedical pre-trained language models on entity typing and relation classification tasks



Task	Dataset	Metrics	E-SVM	CNN-M	BERT-Base	BioBERT	SCIBERT	BERT-MK
Entity Typing	2010 i2b2/VA	Acc	-	-	96.76	97.43	**97.74**	97.70
	JNLPBA	Acc	-	-	94.12	94.37	**94.60**	94.55
	BC5CDR	Acc	-	-	98.78	99.27	99.38	**99.54**
Relation Classification	2010 i2b2/VA	P	-	73.1	72.6	76.1	74.8	**77.6**
		R	-	66.7	65.7	71.3	71.6	**72.0**
		F	-	69.7	69.2	73.6	73.1	**74.7**
	GAD	P	79.21	-	74.28	76.43	77.47	**81.67**
		R	89.25	-	85.11	87.65	85.94	**92.79**
		F	83.93	-	79.33	81.66	81.45	**86.87**
	EU-ADR	P	-	-	75.45	81.05	78.42	**84.43**
		R	-	-	**96.55**	93.90	90.09	91.17
		F	-	-	84.71	87.00	85.51	**87.49**

He et al., BERT-MK: Integrating Graph Contextualized Knowledge into Pre-trained Language Models, Findings of EMNLP 2020.

SparTerm: Learning Term-based Sparse Representation for Fast Text Retrieval

SparTerm: Learning Term-based Sparse Representation for Fast Text Retrieval

Yang Bai*†
Tsinghua University

Xiaoguang Li*
Huawei Noah's Ark Lab

Gang Wang
Huawei Noah's Ark Lab

Chaoliang Zhang
Huawei Noah's Ark Lab

Lifeng Shang
Huawei Noah's Ark Lab

Jun Xu
Renmin University of China

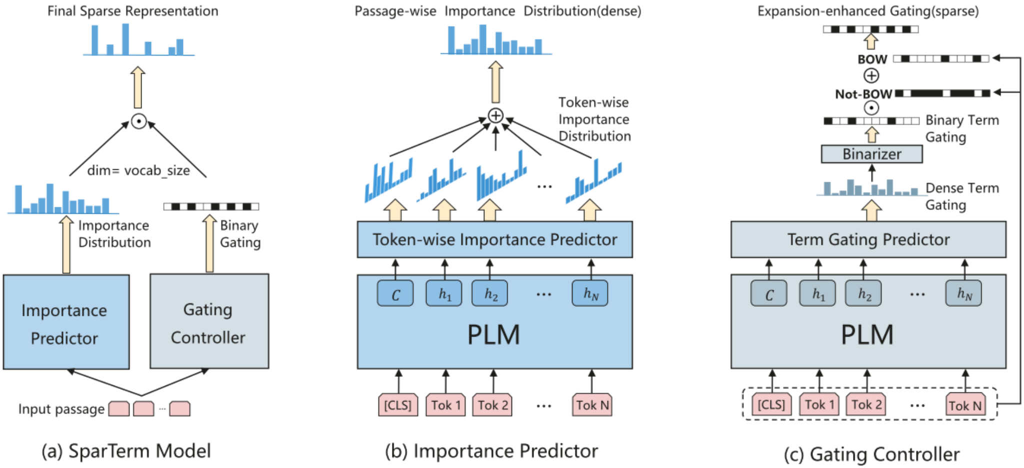
Zhaowei Wang
Huawei Noah's Ark Lab

Fangshan Wang
Huawei Technologies Co., Ltd

Qun Liu
Huawei Noah's Ark Lab

Preprint: https://arxiv.org/pdf/2010.00768.pdf

SparTerm: Model overview



Bai et al. SparTerm: Learning Term-based Sparse Representation for Fast Text Retrieval. arXiv:2010.00768

Generate and Rank: A Multi-task Framework for Math Word Problems

Generate & Rank: A Multi-task Framework for Math Word Problems

Jianhao Shen[1†], Yichun Yin[2], Lin Li[3], Lifeng Shang[2], Xin Jiang[2], Ming Zhang[1*], Qun Liu[2]
[1]Department of Computer Science, School of EECS, Peking University
[2]Huawei Noah's Ark Lab
[3]Huawei HiSilicon
{jhshen, mzhang_cs}@pku.edu.cn
{yinyichun, lilin29, shang.lifeng, jiang.xin, qun.liu}@huawei.com

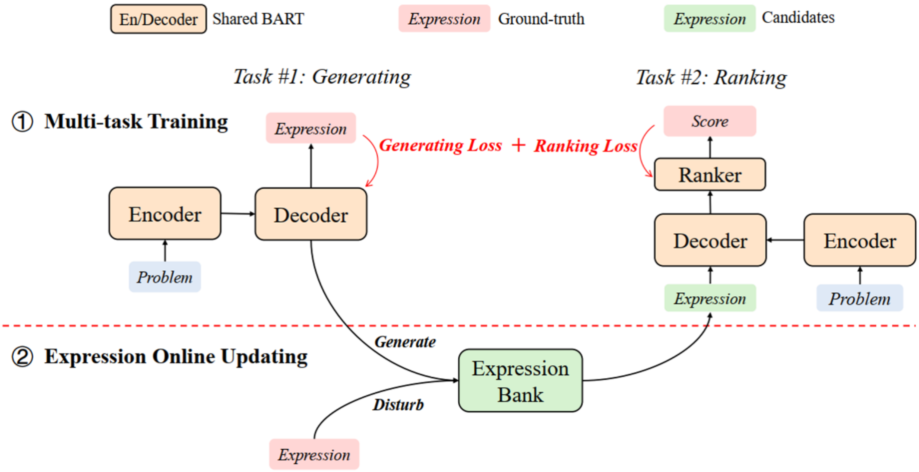
Published in: Findings of EMNLP 2021

Generate and Rank: Math Word Problem (MWP)

- Input: a math problem described in natural language, with a question about an unknown quantity
- Output: an expression that solves the problem

Original MWP	
Problem	A project is completed in 25 days by 12 workers. If it takes 20 days to complete, how many workers will it take?
Solution	*25 * 12 / 20*
Number-mapped MWP	
Problem	A project is completed in *NUM0* days by *NUM1* workers. If it takes *NUM2* days to complete, how many workers will it take?
Solution	*NUM0 * NUM1 / NUM2*

Generate and Rank: A Multi-task Framework for MWPs



Generate and Rank: Expression Bank

- Model-based Generation
  - Use beam search to produce top-K expressions
- Tree-based Disturbance
- Online updating
  - Update the expression bank at each training epoch

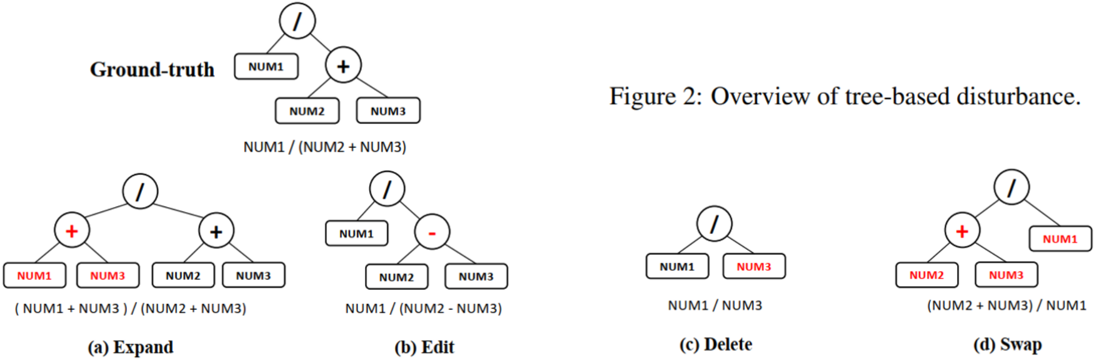


Figure 2: Overview of tree-based disturbance.

Generate and Rank: Experimental Results

Model	Math23K†	Math23K‡	MAWPS‡
DNS	-	58.1	59.5
Math-EN	66.7	-	69.2
T-RNN	66.9	-	66.8
S-Aligned	-	65.8	-
Group-ATT	69.5	66.9	76.1
AST-Dec	69.0	-	-
GTS	75.6	74.3	82.6
Graph2Tree	77.4	75.5	83.7
Multi-E/D	78.4	76.9	-
mBART	80.8	80.0	80.1
Generate & Rank	**85.4**	**84.3**	**84.0**

Table 2: Solution accuracy on MAWPS and Math23K. † refers to the result of test set and ‡ denotes the result of 5-fold cross-validation. "-" means that the results are not reported in the original papers.

#Op	Pro	AST-Dec	G2T	mBART	Ours
1	17.3	82.7	85.5	90.2	90.8 (+0.6)
2	52.2	74.5	83.7	88.1	90.2 (+2.1)
3	19.1	59.9	71.7	71.2	79.1 (+7.9)
4	6.6	42.4	51.5	53.0	63.6 (+10.6)
5	3.4	44.1	38.2	41.2	58.8 (+17.6)
6	0.9	55.6	55.6	55.6	88.8 (+33.2)

Table 5: Accuracy for increasing length of expressions. #Op is the number of operations in expressions. Pro denotes proportion of expressions with different lengths.

HUAWEI NOAH'S ARK LAB

Content

Content

Summary

- Introduction of Huge Pre-trained Language Models
- Opportunities brought by huge PLMs
- Challenges of huge PLMs and potential solutions
- Our work
  - Huge PLMs
  - Efficient PLM training and deployment
  - Applications of PLMs
- Our team

Thank you!

把数字世界带入每个人、每个家庭、
每个组织，构建万物互联的智能世界。

Bring digital to every person, home and organization for a fully connected, intelligent world.

HUAWEI