Research Summary of Xiaocong

May 1, 2023

Past Experiences and Publications

- 2022.4-2022.9: Visiting Researcher at USC, advised by Prof. Muhao Chen
- 2021.7-2022.3, 2022.10-now: Research Intern at Tsinghua University, advised by Prof. Zhilin Yang
- 2021.6-2021.8: Visiting Researcher at NUS (remote), advised by Dr. Lizi Liao
- 2021.1-2021.6: Research Intern at Tsinghua University, advised by Prof. Minlie Huang
- 2021.5-2021.6: Survey study at Tsinghua University, coauthor with Leixian Shen

Title of Publication	Status	Link
Parameter-Efficient Tuning with Special Token Adaptation	EACL 2023	https://arxiv.org/pdf/2210.04382.pdf
Nlp from scratch without large-scale pretraining: A simple and efficient framework	published at ICML 2022	https://proceedings.mlr.press/v162/yao22c.html
Towards natural language interfaces for data visualization: A survey	published at IEEE TVCG 2022	https://arxiv.org/abs/2109.03506
Eva: An open-domain chinese dialogue system with large-scale generative pre-training	preprint at arXiv	https://arxiv.org/abs/2108.01547

https://scholar.google.com/citations?user=s0K3OecAAAAJ&hl=en

Project at USC: PASTA

- At USC, I led the research project Parameter-Efficient Tuning with Special Token Adaptation (PASTA), a parameter-efficient tuning method with up to 0.029% trainable parameters on downstream tasks.
- As a member of parameter-efficient tuning families, it shares the merit that minimal parameters are needed to be modified on downstream tasks --> require much less storage space for model deployment in multi-task settings.
- Compared to other parameter-efficient tuning methods, PASTA has a lower parameter complexity and a consistent set of introduced parameters for different tasks. Both properties benefit the deployment of models.

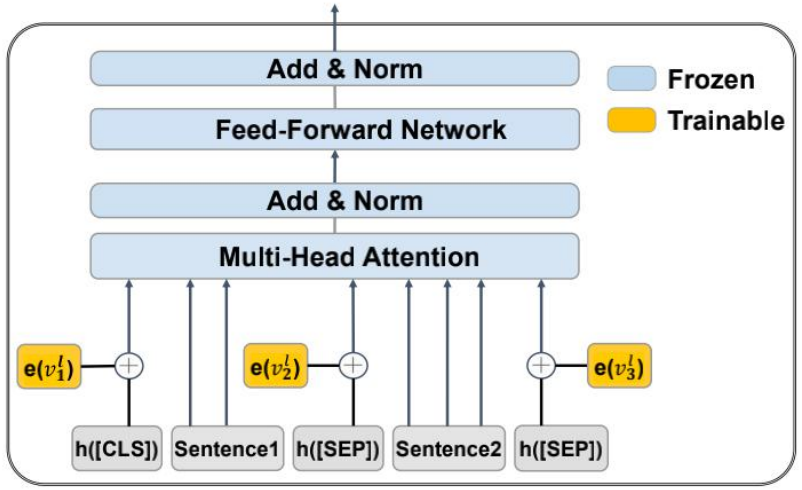


Fig. a Model structure of proposed PASTA. Trainable vectors are added to the special tokens at each layer, while the parameters of PLMs are kept frozen.

	# Param	Parameter Consistency
Adapter	$\mathcal{O}(L \times d \times r)$	✓
P-tuning v2	$\mathcal{O}(L \times d \times T)$	✓
BitFit	$\mathcal{O}(L \times (d+m))$	✓
Diff-Prune	-	✗
PASTA	$\mathcal{O}(L \times d)$	✓

Fig. b Parameter complexity and consistency of PASTA and other baselines. The number of special tokens are invariant to the scale of models, and the set of parameters introduced by PASTA is consistent across different tasks.

Project at USC: PASTA

- We are inspired by a special kind of attention heads widespread in PLMs, named “vertical heads”.
- Vertical heads copy information from special tokens to all of other tokens. Therefore, updates to special tokens also indirectly modify the value of other tokens during forward pass.
- We can adapt PLMs by only updating the value of special tokens on both sentence-level and token-level downstream tasks.

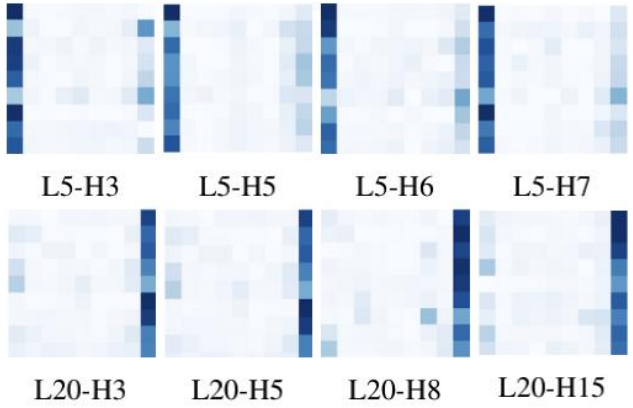
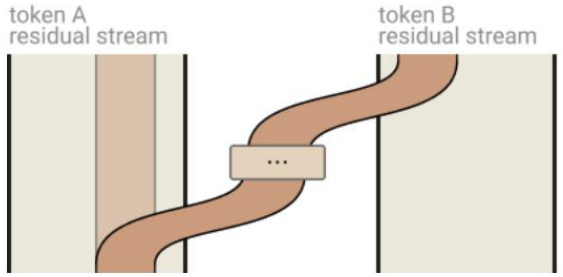


Fig. a Examples of vertical heads in BERT-large. An attention head is regarded as vertical if at least 90% tokens assign maximal attention scores to either [CLS] or [SEP].



Attention heads copy information from the residual stream of one token to the residual stream of another. They typically write to a different subspace than they read from.

Fig. b Information is copied between tokens through the attention machenism. In vertical heads, information are disseminated from special tokens to all of other tokens.

https://transformer-circuits.pub/2021/framework/index.html

Project at USC: PASTA

- Experiment results on GLUE benchmark and CoNLL03 verify the effectiveness of PASTA on both sentence and token level tasks.
- By tuning only 0.015%-0.029% parameters, PASTA can match the performance of full finetuning.

	%Param	RTE acc.	CoLA mcc.	STS-B Spearman	MRPC F1	SST-2 acc.	QNLI acc.	MNLI(m/mm) acc.	QQP F1	Avg.
Full Finetuning*	100%	70.1	60.5	86.5	89.3	**94.9**	92.7	**86.7/85.9**	**72.1**	**81.6**
Adapter**	3.6%	71.5	59.5	**86.9**	89.5	94.0	90.7	84.9/85.1	71.8	81.1
Diff-Prune†	0.5%	70.6	61.1	86.0	**89.7**	94.1	**93.3**	86.4/86.0	71.1	81.5
P-tuning v2	0.29%	70.1	60.1	86.8	88.0	94.6	92.3	85.3/84.9	70.6	81.0
BitFit‡	0.08%	**72.0**	59.7	85.5	88.9	94.2	92.0	84.5/84.8	70.5	80.9
PASTA	**0.015%-0.022%**	70.8	**62.3**	86.6	87.9	94.4	92.8	83.4/83.4	68.6	80.9

Table 2: BERT-large model performance on GLUE benchmark test set. Lines with * and ** are results from Devlin et al. (2019) and Houlsby et al. (2019), and lines with † and ‡ are from Guo et al. (2021) and Zaken et al. (2022) respectively. We reimplement experiments of P-tuning v2 on GLUE benchmark with a prompt length of 20.

	%Param	RTE acc.	CoLA mcc.	STS-B Pearson	MRPC acc.	SST-2 acc.	QNLI acc.	MNLI(overall) acc.	QQP acc.	Avg.
Full Finetuning*	100%	86.6	68.0	92.4	**90.9**	96.4	94.7	90.2	**92.2**	88.9
LoRA†	0.24%	**87.4**	68.2	**92.6**	**90.9**	96.2	94.9	**90.6**	91.6	**89.0**
PASTA	**0.015%-0.029%**	86.6	**69.7**	91.8	**90.9**	**96.8**	**95.1**	90.4	89.9	88.9

Table 3: RoBERTa-large model performance on GLUE benchmark. Lines with * are results from Liu et al. (2019), and lines with † are from Hu et al. (2021). We follow the metric settings of baselines and also report results on GLUE development set for the convenience of direct comparison.

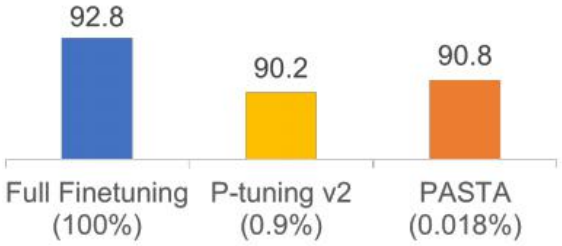


Figure 3: NER results on CoNLL03 (F1). Each method is labeled with the percentage of trainable parameter sizes with regard to full tuning.

Fig. a Experiment results on GLUE and CoNLL03. To facilitate comparison with baseline works, we take most of the experimental results from their original papers which are reported with either BERT-large or RoBERTa-large.

Project at USC: PASTA

- PASTA performs slightly worse with smaller PLMs, similar to other works in parameter-efficient tuning families.
- This is consistent with the research on intrinsic dimensionality of finetuning. Pretraining implicitly minimizes intrinsic dimension of downstream tasks and larger models tend to have lower intrinsic dimensions (“low dimensional task representations”).
- PASTA gives a substitutional method to approximate the intrinsic dimensionality of PLMs, since it needs the least trainable parameters among parameter-efficient tuning methods in existence.

	%Param	RTE	CoLA	STS-B	MRPC	SST-2	QNLI	MNLI(m/mm)	QQP	Avg.
Full Finetuning *	100%	66.4	**62.1**	**89.8**	90.9	91.6	90.0	**83.2/ -**	**87.4**	82.7
Adapter*	0.81%	71.8	61.5	88.6	89.9	91.9	**90.6**	83.1/ -	86.8	**83.0**
BitFit†	0.8%	72.3	58.8	89.2	90.4	**92.1**	90.2	81.4/ -	84.0	82.3
PASTA	**0.015%-0.022%**	**73.6**	57.9	88.7	**91.5**	91.2	89.7	77.8/78.8	80.8	81.4

Fig. a Experiment results with BERT-base for PASTA and other baselines on GLUE.

Project at USC: PASTA

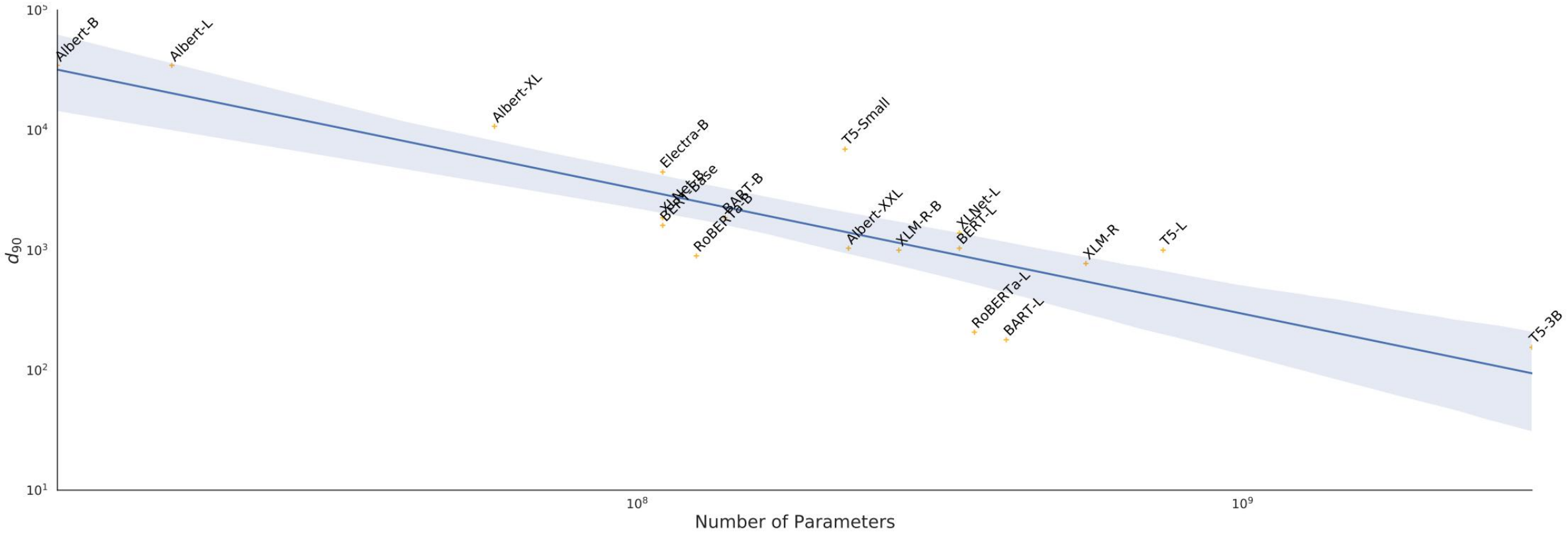


Fig. b Intrinsic dimensionality of different PLMs on MRPC. It is defined as minimum number of trainable parameters needed to reach satisfactory solutions (here is 90% accuracy of full-finetuning).

https://arxiv.org/abs/2012.13255

Project at USC: PASTA

- Further questions to be explored:

✓ We've found special tokens have a strong influence on global representations, which is related to the topic controllable text generation. Could we take similar approach to treat special tokens as global controllers in such tasks?

✓ Parameter-efficient tuning methods can be viewed as a case of modular networks, where frozen PLMs and trainable modules encode generic linguistic information (learned during pretraining) and task-specific information respectively. What could we do to increase the modularity so that each module is functionally specialized to encode more fine-grained information?

Project at IIIS, Tsinghua: Task-driven Language Models

- Earlier I worked on Task-driven Language Models (TLM), where we improved the time efficiency by more than 100 times to pretrain a task-specific language model from scratch.
- It is motivated by the trade-offs between generalizability and efficiency in Complex System theories. We sacrifice model generalizability by pretraining models on task-relevant data from scratch, instead of a domain-agnostic corpus in most pretraining frameworks.

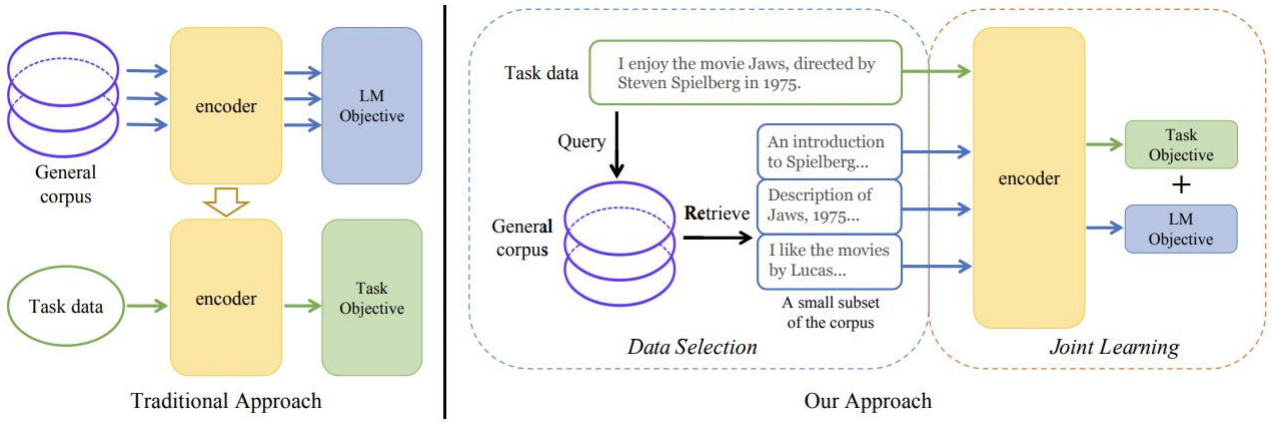


Fig. a Pipeline of proposed Task-driven Language Models. We first retrieve data that are similar to the target task data from general corpus, and combine the task data and recalled (unlabeled) data to pretrain a language model.

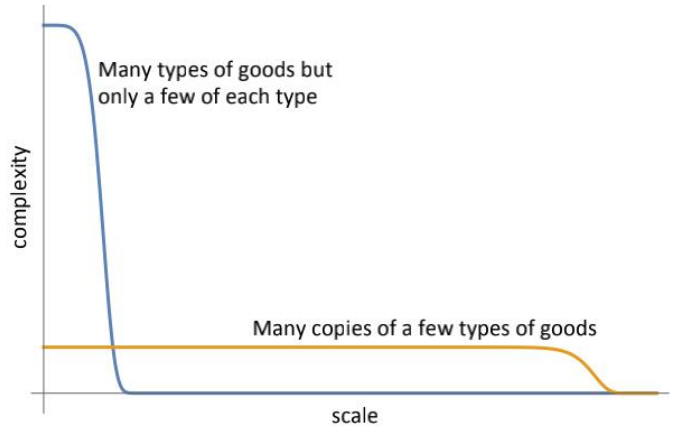


Fig.b The tradeoff between generalizability (adativeness for various market demand) and efficiency (production scale) . This is the famous conclusion -- Economies of Scale -- in complex system theories.

https://arxiv.org/abs/1912.05088

Project at IIIS, Tsinghua: Task-driven Language Models

- Experiments show our proposed method achieves comparable or better performance on 8 target tasks, with up to 100 times of training FLOPs and the size of corpus reduced.
- Ablation study shows that our method is especially effective on low-resource downstream tasks.

Model	#Param	FLOPs[1]	Data[2]	AGNews	Hyp.	Help.	IMDB	ACL.	SciERC	Chem.	RCT	Avg.
BERT-Base[3]	109M	2.79E19	16GB	93.50 ±0.15	91.93 ±1.74	69.11 ±0.17	93.77 ±0.22	69.45 ±2.90	80.98 ±1.07	81.94 ±0.38	87.00 ±0.06	83.46
BERT-Large[3]	355M	9.07E19	16GB	93.51 ±0.40	91.62 ±0.69	69.39 ±1.14	**94.76** ±0.09	69.13 ±2.93	**81.37** ±1.35	**83.64** ±0.41	**87.13** ±0.09	**83.82**
TLM *(small-scale)*	109M	2.74E18	0.91GB	**93.74** ±0.20	**93.53** ±1.61	**70.54** ±0.39	93.08 ±0.17	**69.84** ±3.69	80.51 ±1.53	81.99 ±0.42	86.99 ±0.03	83.78
RoBERTa-Base[3]	125M	1.54E21	160GB	**94.02** ±0.15	93.53 ±1.61	70.45 ±0.24	**95.43** ±0.16	68.34 ±7.27	81.35 ±0.63	82.60 ±0.53	87.23 ±0.09	84.12
TLM *(medium-scale)*	109M	8.30E18	1.21GB	93.96 ±0.18	**94.05** ±0.96	**70.90** ±0.73	93.97 ±0.10	**72.37** ±2.11	**81.88** ±1.92	**83.24** ±0.36	**87.28** ±0.10	**84.71**
RoBERTa-Large[3]	355M	4.36E21	160GB	94.30 ±0.23	**95.16** ±0.00	70.73 ±0.62	**96.20** ±0.19	**72.80** ±0.62	82.62 ±0.68	84.62 ±0.50	**87.53** ±0.13	85.50
TLM *(large-scale)*	355M	7.59E19	3.64GB	**94.34** ±0.12	**95.16** ±0.00	**72.49** ±0.33	95.77 ±0.24	72.19 ±1.72	**83.29** ±0.95	**85.12** ±0.85	87.50 ±0.12	**85.74**

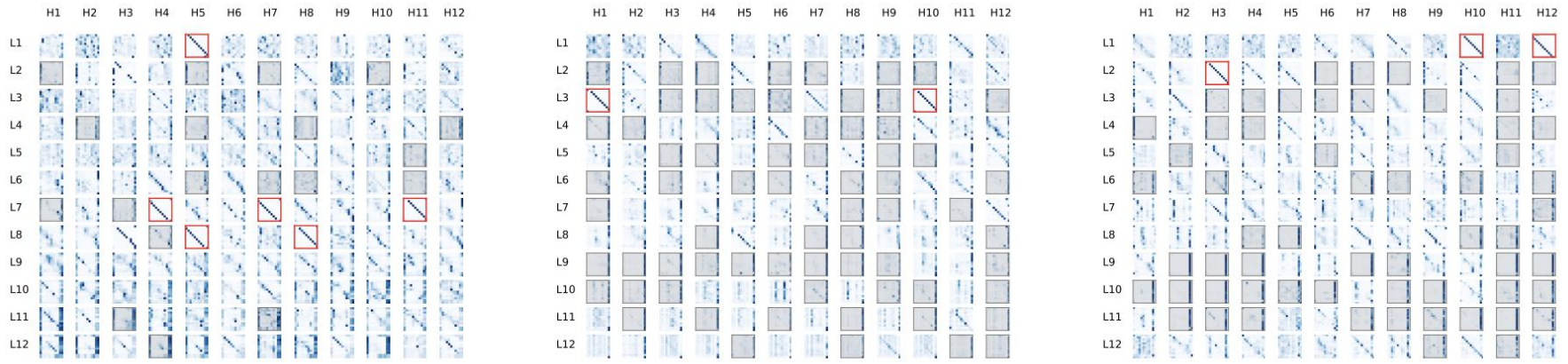
Fig. a Experiment results of Task-driven Language models and baselines.

	AGNews	SciERC	ChemProt
Only Task Data	93.41±0.10	51.23±1.13	55.05±0.18
Top-50	**94.51**±0.15	77.61±1.75	77.21±0.47
Top-500	94.32±0.05	82.39±0.55	81.44±0.50
Top-5000	94.42±0.10	**86.07**±0.48	**83.64**±0.26

Fig. b Ablation study on the number of retrieved data. AGNews is a high-resource task, while SciREC and ChemProt are low-resource ones.

Project at IIIS, Tsinghua: Task-driven Language Models

- By comparing the attention distribution between TLMs and PLMs, we observed a significant difference in model behavior: less vertical heads and more heterogeneous heads are found in TLMs.
- Previous studies show that heterogeneous heads encode specific linguistic information in input text. We attribute the training efficiency of TLMs to the fact that more model capacity are allocated to process linguistic information.



(a) TLM (*Medium scale*) (b) BERT-Base (c) RoBERTa-Base

Fig. a Attention patterns in TLMs and PLMs. Vertical heads are masked in grey, diagonal heads are boxed in red, and other heads show heterogeneous patterns. The distribution difference is consistent across all tasks and backbone models.

Project at IIIS, Tsinghua: Task-driven Language Models

- Further questions to be explored:

✓ We related training efficiency to heterogeneous heads in TLM, and in PASTA we generalize on downstream tasks by exploiting vertical heads. Could we find proper methods to quantitively study the functions of vertical and heterogeneous heads in model efficiency and generalizability?

✓ The specialization of TLMs on targeted task (and limited linguistic information) brings the high efficiency, and this is similar to the idea of modular networks. Could we extend this framework to the training of modular networks? --> Sparse training and local loss

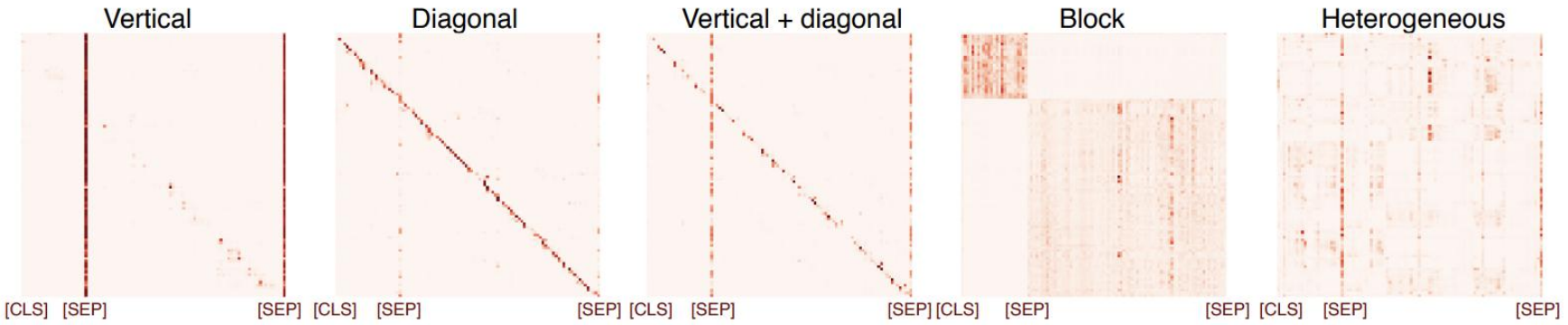


Fig. a Typical self-attention patterns in pretrained Transformers. The first three types are most likely associated with language model pre-training, while the last two encode specific semantic and syntactic information.

https://aclanthology.org/D19-1445/

Survey Project at Tsinghua: V-NLI

- I colaborated with multiple graduate students on the survey of Visualization-oriented Natural Language Interfaces (V-NLI) which accept natural language queries as input and output appropriate visualizations automatically.
- We thoroughly studied the seven stages of typical V-NLI systems, the representative research, relevant toolkits and commercial systems.

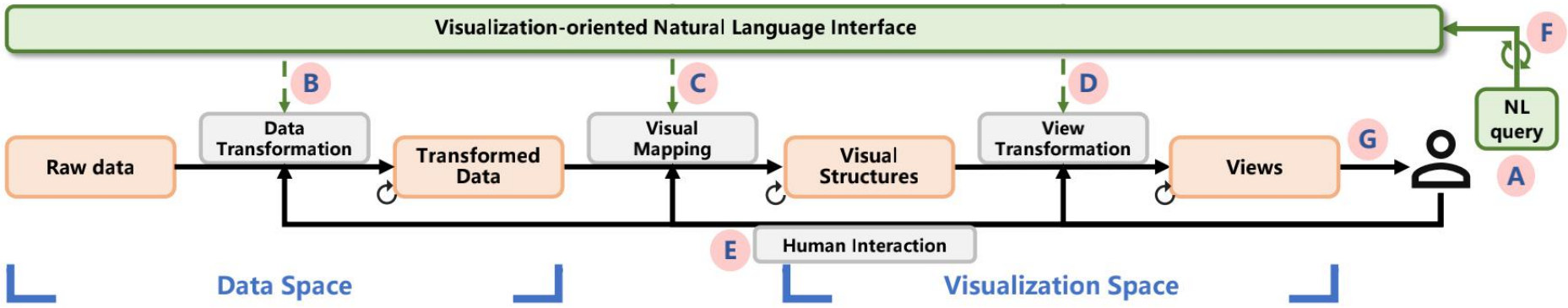


Fig. a Extension of classic information visualization pipeline with V-NLI. It depicts how V-NLI works to construct visualizations, which consists of the following seven stages: (A) Query Interpretation, (B) Data Transformation, (C) Visual Mapping, (D) View Transformation, (E) Human Interaction, (F) Dialogue Management, and (G) Presentation.

Survey Project at Tsinghua: V-NLI

- We found that, as a rapidly growing field, V-NLI faces many thorny challenges. In particular, most of the existing V-NLI systems are unable to integrate large PLMs and just apply hand-crafted grammar rules or typical NLP toolkits, which brings consequent problems including the lack of domain knowledge and failure on long inputs.

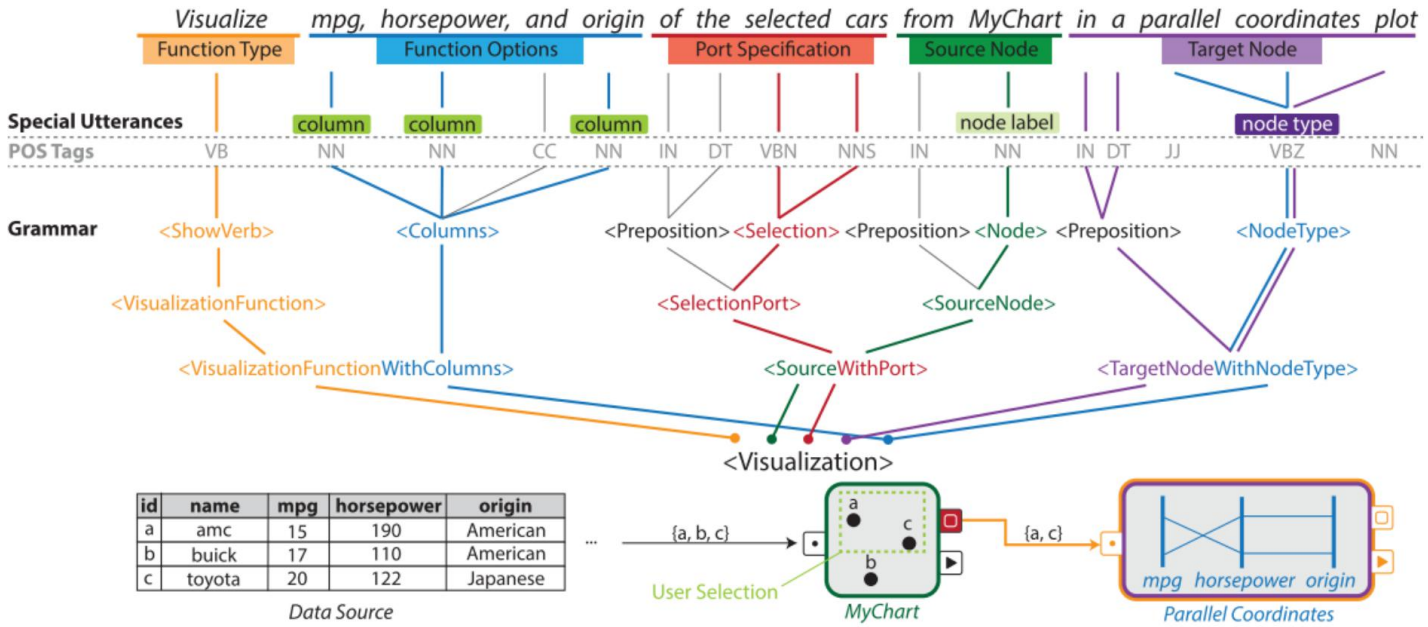


Fig. a Query intepretation stage in FlowSense. Most current V-NLI systems still use similar rule-based NLP toolkits to process user inputs.

Survey Project at Tsinghua: V-NLI

- Previous end-to-end research treat V-NLI as a machine translation problem, with natural language as input and machine-readable text as output (a json or SQL code). Though none of them has been used in commercial V-NLI systems.
- We believe LLMs with good zero-shot performance (e.g. CodeX and ChatGPT) could improve this situation, and highlight the importance of interpretability and robustness of models to mitigate the problem that the performance of current V-NLI is sensitive to the form and wording of user input.

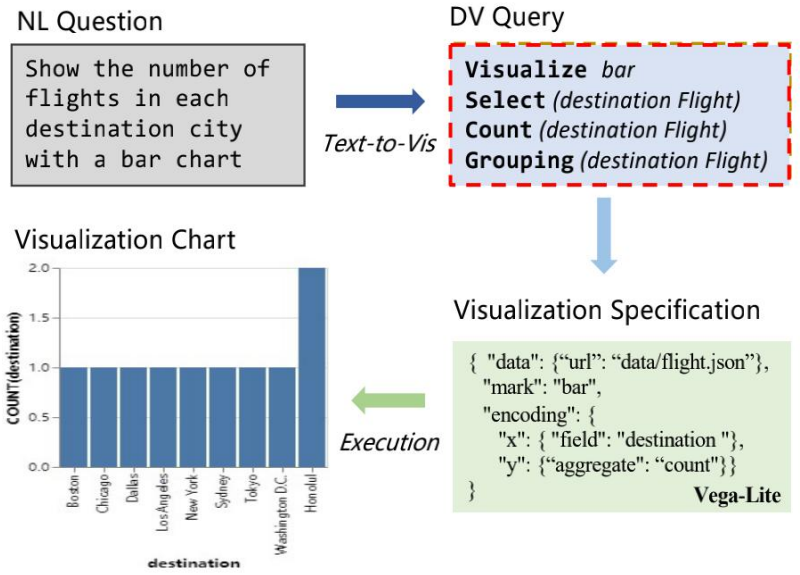


Fig. a An example of end-to-end V-NLI system. Text-to-Vis stage is performed by pretrained language models or pretrained embeddings.

https://dl.acm.org/doi/abs/10.1145/3534678.3539330

Project at CoAI, Tsinghua: EVA

- As my first research project in NLP, we developed a dialogue system EVA with pretraining a Chinese dialogue model that was the largest in existence at that time using 2.8B parameters and dialogue data collected from social media.
- I was working on the client-server communication framework and model quality test with common user inputs.
- Additional post-processing modules were used to filter out toxic content in the generated text.

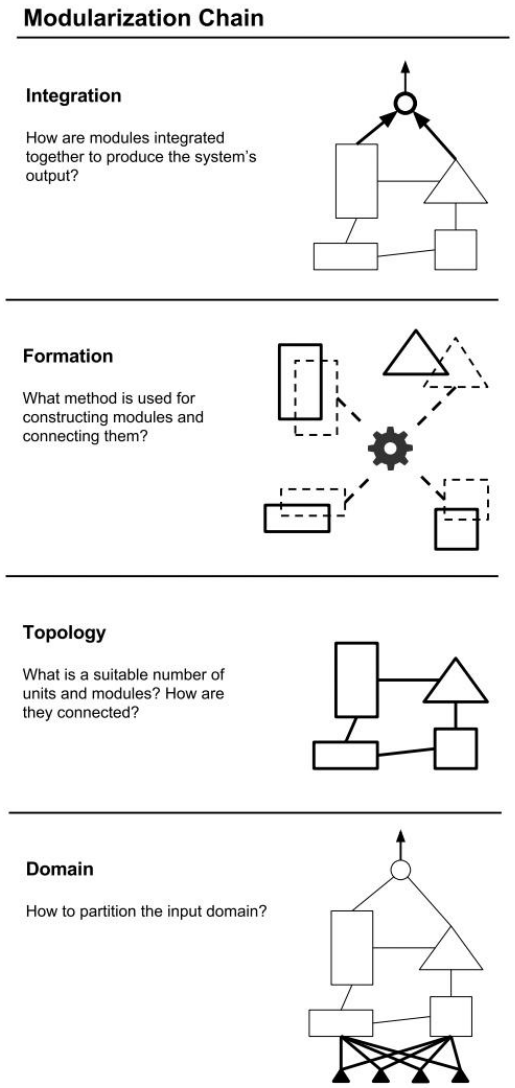
Fig. a The user interface with interactive demonstration of EVA.

Long-Term Interests:

Modularity and Representation Disentanglement

Steps of Building a Modular Network

- A modularization chain starts with partitioning the input domain. Then a modular topological structure is selected for the model. After that, formation and integration techniques are selected to build the model and integrate the different modules, respectively.
- We use representation disentanglement when talking about data and modularity when talking about model structure.



https://arxiv.org/abs/1904.12770

Motivation of Modularity

- In a Modular Network, each subsystem or module can be regarded as targeting an isolated subproblem that can be handled separately from other subproblems. This facilitates:

✓ Specialization and Collaboration: Neoclassical Economics proved that the pareto efficiency of a system improves when agents within are functionally specialized based on their comparative advantage. This is true for most efficient complex systems (e.g. human brains).

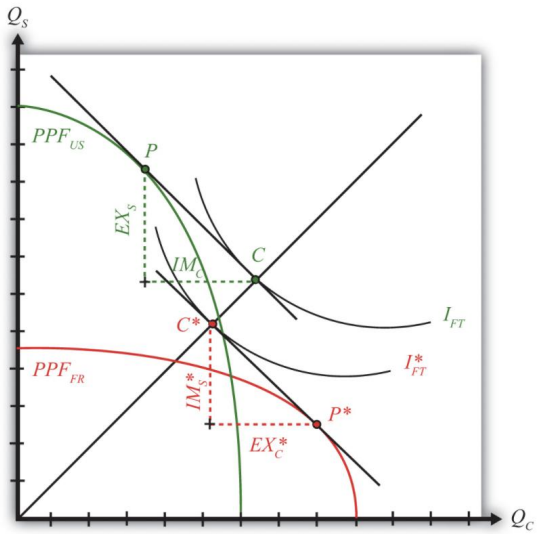


Fig. a Graphic demonstration of PPF before and after specialization and trade. After specialization of jobs the efficiency of the system improves. The “trade” of information in Transformers is the concatenation operation $\boldsymbol{H = W^o[h_1, h_2, \ldots, h_h]^T}$.

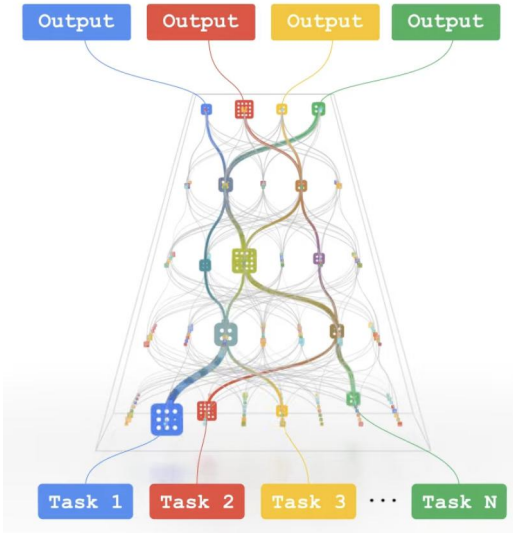


Fig. b The Pathways network (from Jeff Dean's blog, different from the version in their paper). Specialized modules are sparsely activated according to different tasks, with collaborations based on the topological structure.

https://blog.google/technology/ai/introducing-pathways-next-generation-ai-architecture/

Motivation of Modularity

- A successful implementation of module specialization is Mixture-of-Experts and its variants.
- Our work TLM is similar to the idea of Task MoE with # tasks = 1.
- A problem in these MoE works is the poor ability to generalize compositionally --> The choice of the specialization level matters!

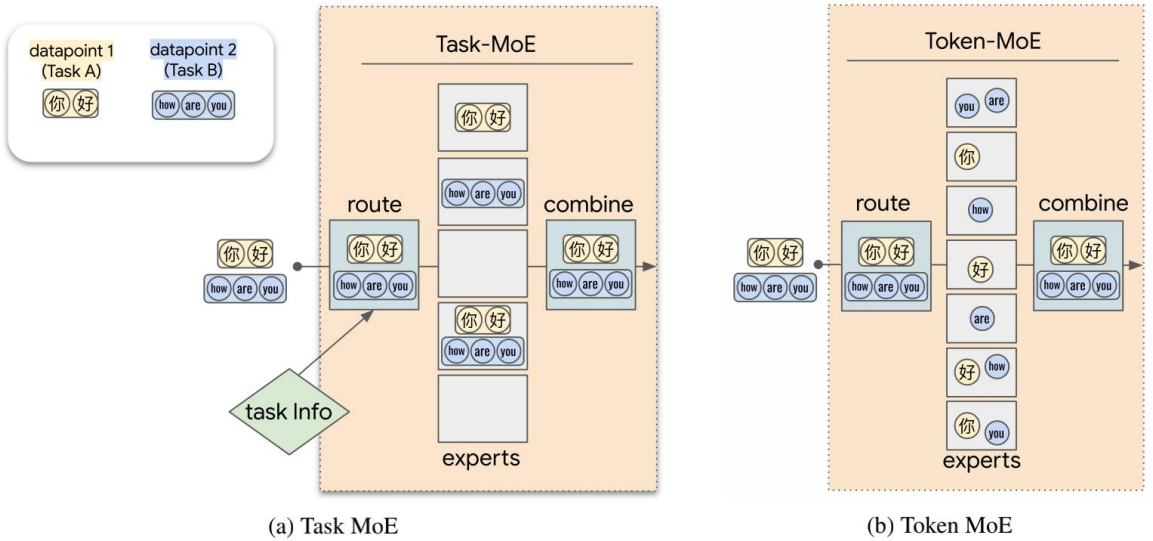


Fig. a Archtechture of vanilla MoE (Token MoE) and Task MoE.

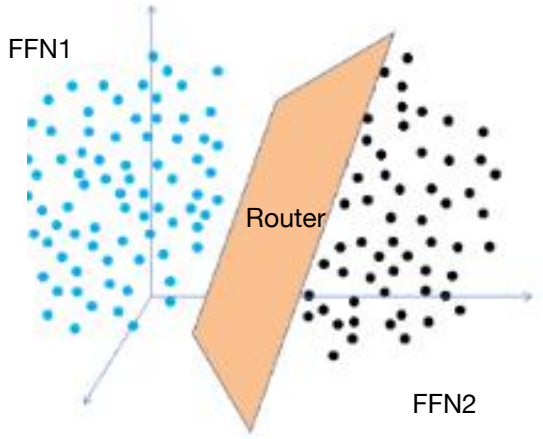


Fig. b FFNs are functionally specialized within a subset of embedding space / task space.

https://arxiv.org/abs/2110.03742

Motivation of Modularity

- When the modular specialization is at the linguistic signal level, the model could show a better compositionality.
- Even applying rule-based mapping between linguistic signals and downstream tasks brings significant preformance improvement for model pretraining.

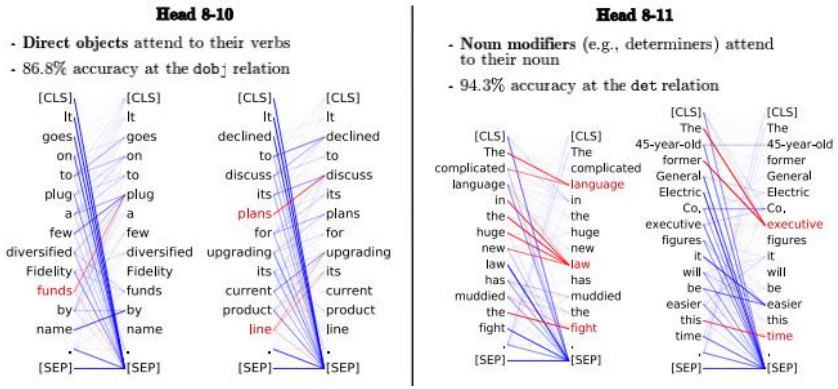


Fig. a In a pretrained Transformer model, attention heads are highly specialized to process specific liguistic signals (e.g. noun modifiers and direct objects in the input).

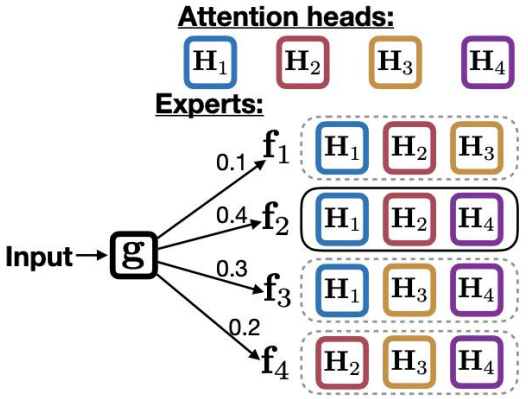


Fig. b The weighted composition of attention heads in PLMs show powerful expressiveness.

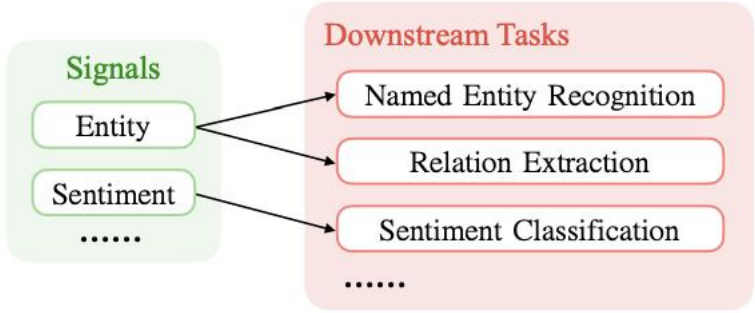


Fig. c The mapping relationship between signals and downstream tasks in reStructured pretraining.

https://arxiv.org/abs/1905.05950
https://arxiv.org/pdf/2005.06537.pdf
https://arxiv.org/abs/1912.12333

Motivation of Modularity

- Besides specialization, a modular network also facilitates:

✓ Compositional Generalization: A modular network makes it easier to scale and add more functionality without disrupting existing functions (a.k.a. catastrophic forgetting) or the need for redesigning the whole system.

✓ Error Localization: As modules correspond to different functions and are loosely coupled, error fixing could be conducted locally.

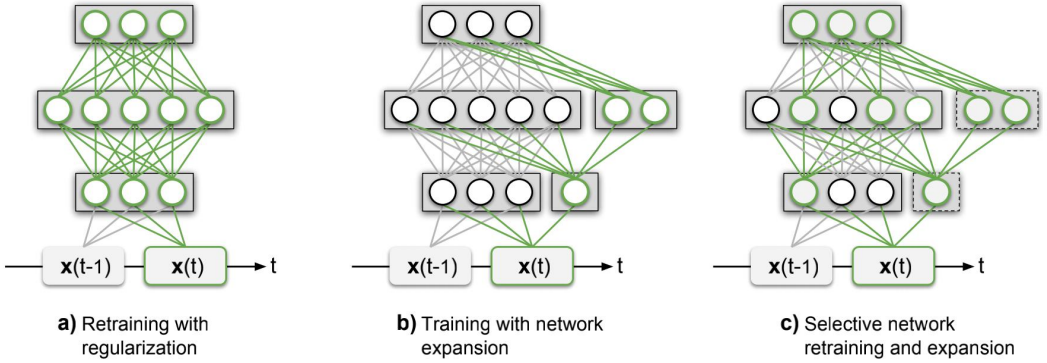


Fig. a Three schemes of model structure in continual learning. Ideally we want to obtain a model like c) with “cells” in figure representing functionally specialized modules.

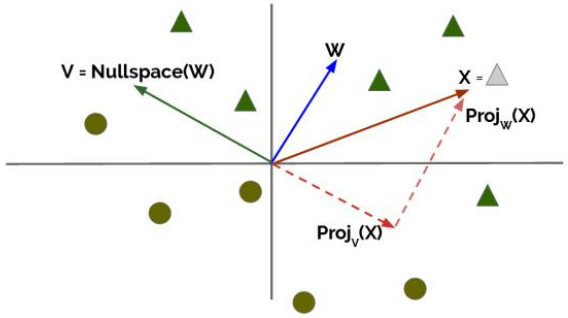


Fig. b Many works on fairness guard attributes by projecting text representations to the direction orthogonal to the decision boundary of protected attributes. The global information of text representation is needed to fix this error.

https://www.sciencedirect.com/science/article/pii/S0893608019300231
https://arxiv.org/abs/2004.07667

Theoratical Evidence of Representation Disentanglement

- Under certain conditions, any signals can be decomposed into a set of trigonometric functions along with their integrals (Fourier Transform) .
- One type of Representation Disentanglement in language is that, different syntactic/semantic information (e.g. sentiment) corresponds to signals with different frenquencies and can be disentangled.

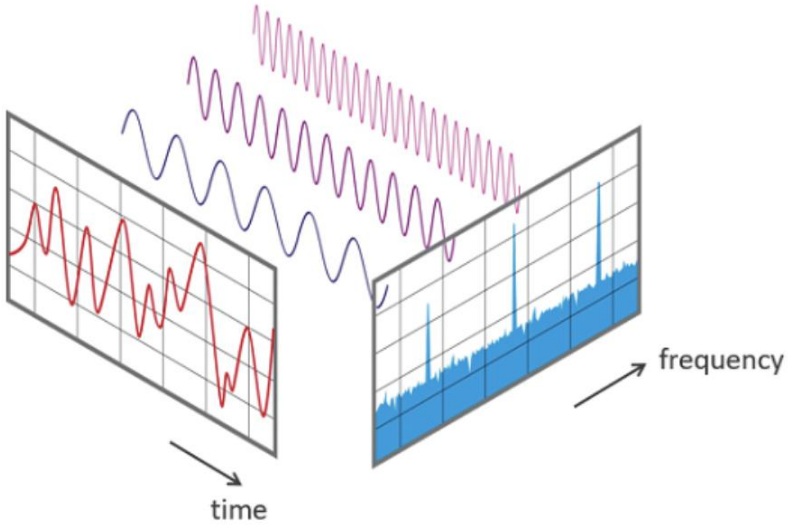


Fig. a Fourier Transform. When a sequence of language tokens is seen as a discrete series in time domain, different linguistic signals can be extracted and represented in frequency domain.

Theoratical Evidence of Representation Disentanglement

- On the contrary, the linearly additive property in Transformers makes signals from different specialized modules interactable and coupled.

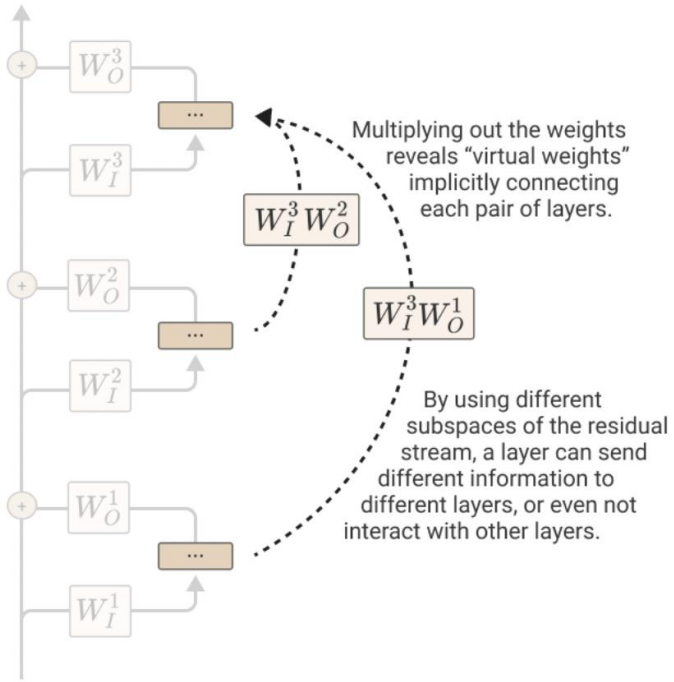


Fig. a In Transformers, modules at different layers interact with each other through residual connections, potentially causing representation entanglement.

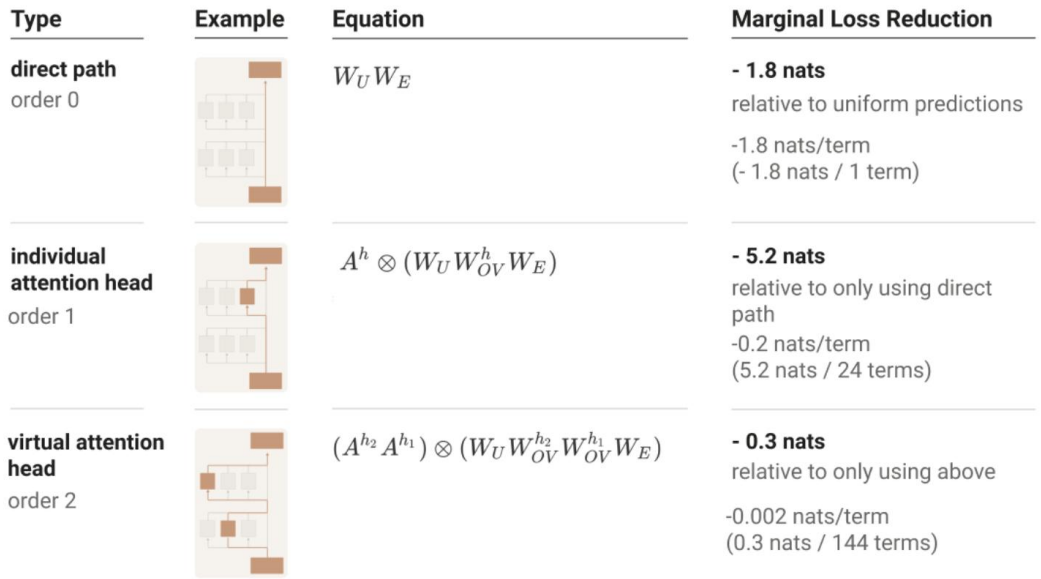


Fig. b Different type of branches show diffrent behavior processing a specific kind of information.

https://transformer-circuits.pub/2021/framework/index.html

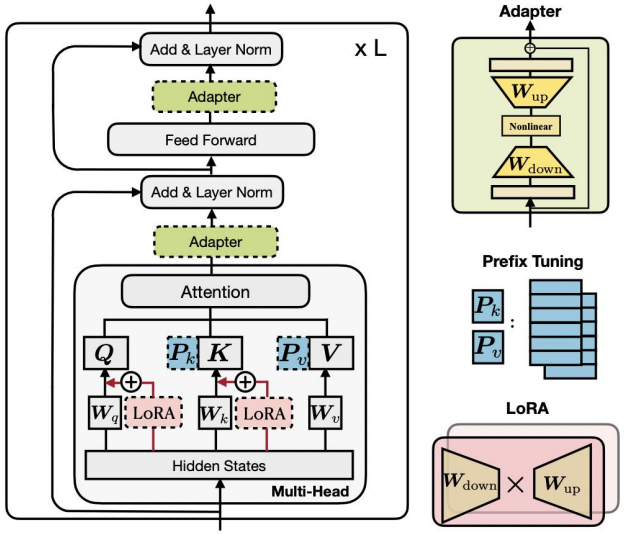
Empirical Evidence of Representation Disentanglement

- With manual design, Representation Disentanglement can be obtained by language models.

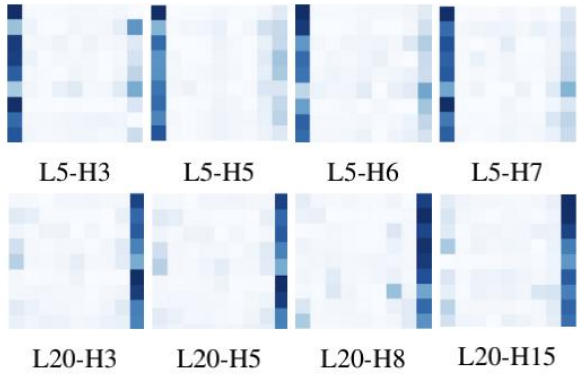
✓ Prompt-based / Adapter-based methods freeze generic signals extracted by PLM and encode task-specific signals with additional modules. PLM filters signals that were processed during pretraining.

✓ Complex-valued embeddings encode pos./token signals into non-interfering spaces.

✓ PASTA use special tokens in PLMs as information disseminators that effectively adapt PLMs to downstream tasks.



$$[r_{j,1}e^{i(\omega_{j,1}\text{pos}+\theta_{j,1})}, ..., r_{j,2}e^{i(\omega_{j,2}\text{pos}+\theta_{j,2})}, \cdots, r_{j,D}e^{i(\omega_{j,D}\text{pos}+\theta_{j,D})}]$$



https://openreview.net/forum?id=0RDcd5Axok
https://arxiv.org/abs/2206.11147

Empirical Evidence of Representation Disentanglement

- Disentanglement of signals also exists in PLMs even without purposeful design.

✓ BERT heads are highly specialized to process specific kinds of linguistic information.

✓ Across layers BERT “imitates traditional NLP pipeline”, with shallow layer mainly processing syntactic signals and deep layer processing semantic signals.

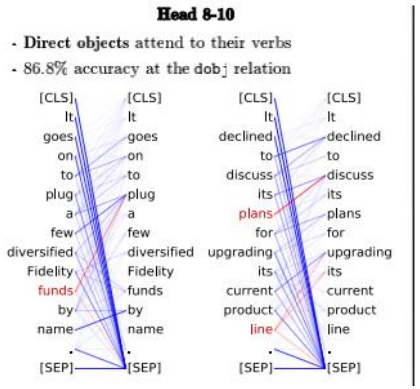


Fig. a The attention weights of different BERT heads perform quite well on specific probing tasks without any finetune.

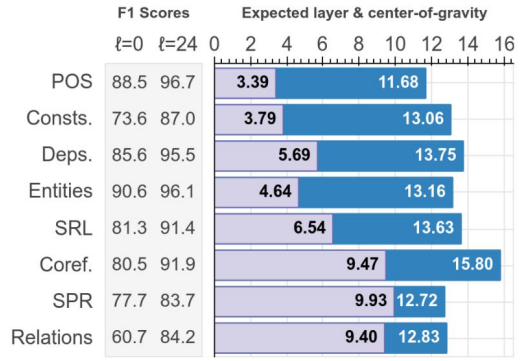


Fig. b The “expected layer” that processes a certain kind of signal in BERT-large.

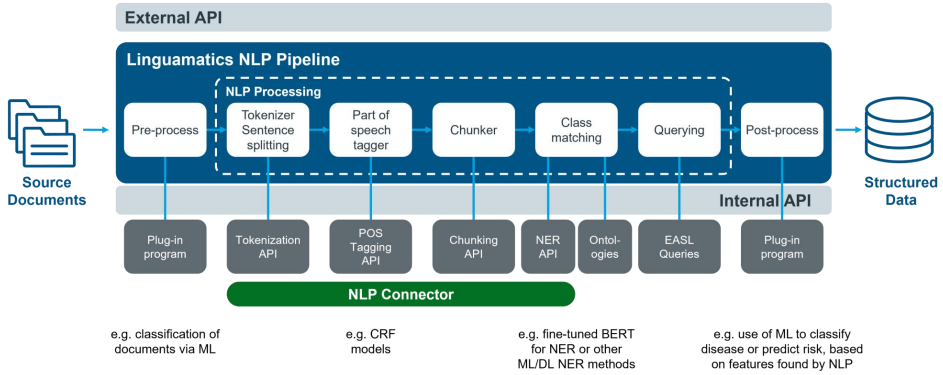


Fig. c A kind of traditional NLP pipeline.

https://arxiv.org/abs/1905.05950
https://arxiv.org/abs/1906.04341

Modular Design Still Matters

- Spontaneously formed modularity could partly fail, and explicit modular design helps improve the model performance.
- And it inherits all the merits from an efficient complex system.

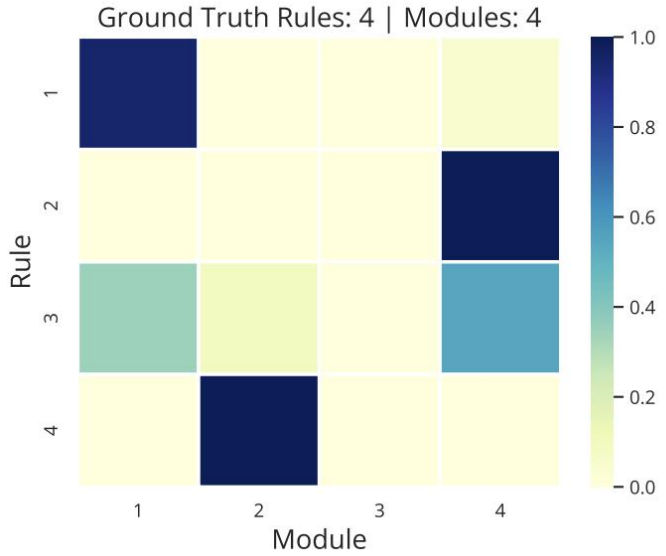


Fig. a Specializing model structure with ground truth skill rules (like reStructured Pre-training does) and corresponding modules, expecting rules and modules match.

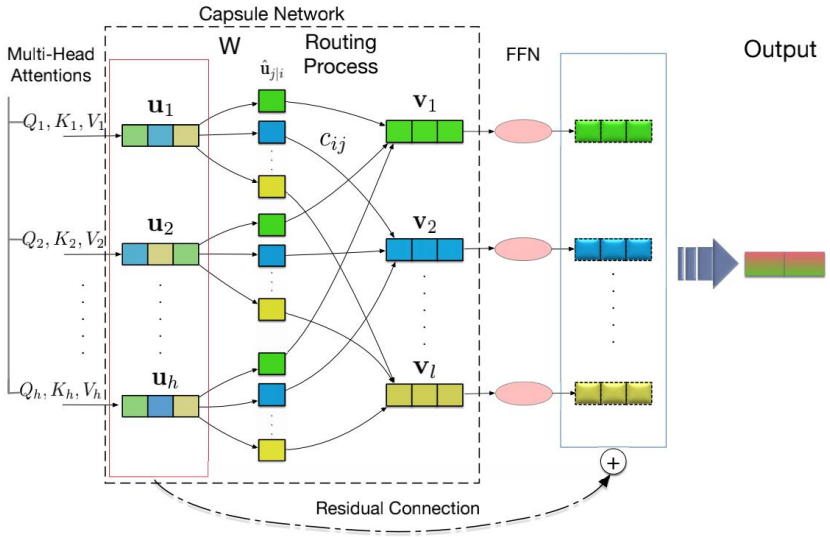


Fig. b Specializing model structure with capsule network. Ideally, each output capsule represents a distinct property of the input and carry all the deserved information when they are combined.

https://arxiv.org/abs/1909.00188
https://arxiv.org/abs/2206.02713

Successful Cases of Modular Networks

- Many successful works focus on selecting a proper topological structure for the models (except for MoE).

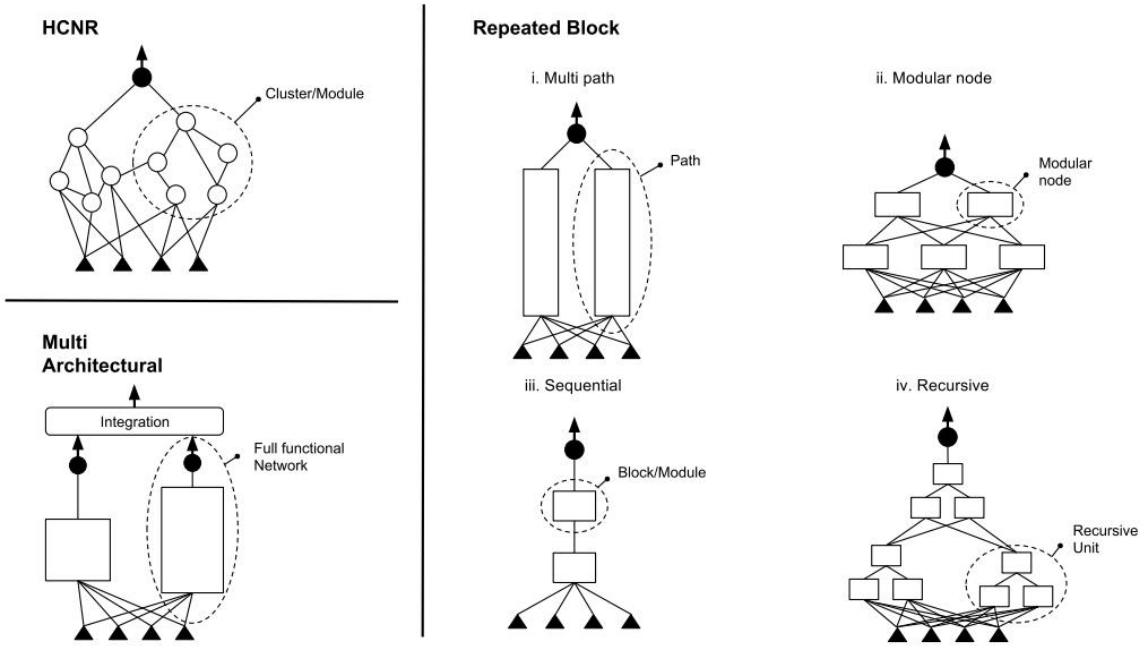


Fig. a Typical topological structures of modular networks.

https://arxiv.org/abs/1904.12770

Successful Cases of Modular Networks

- ✓ Recursive: *Standing on the Shoulders of Giant Frozen Language Models*
- ✓ Multi Path: *Analysis of Branch Specialization and its Application in Image Decomposition; Pathway Network*
- ✓ Multi Architecture: Emsemble learning

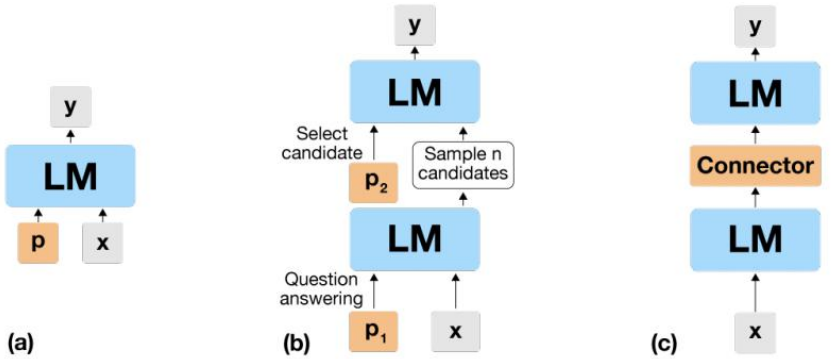


Fig. a Recursive use of frozen LMs for Question Answering. LMs are repeatedly used while performing different functions.

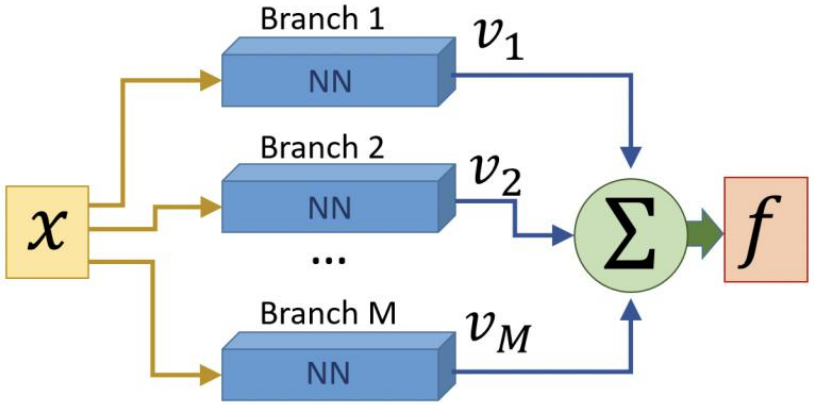


Fig. b Parallel branches with same structure, different initialization in Branch Specialization. After traininig, each branch performs a specific function in image processing.

https://arxiv.org/abs/2204.10019
https://arxiv.org/abs/2206.05810

Discussion Time

- Extremely large PLMs (e.g. T5, PaLM, ChatGPT) have shown excellent generalizability and zero-shot performance with a monolithic architecture. Is it still meaningful to research on the working machenisms of models? (Frankly, in universities we don't have enough GPUs to train these large models, but would these topics themselves be meaningful)
- I firmly believe in the value of interdisciplinary research. But ideas from other domains are usually difficult to be integrated and implemented in NLP research. What could we do to make them more feasible?

Appendix: Complex Systems Theory

- The complex systems theories emphasize the importance to treat system as a whole and the existence of emergence in a system: These system-level characteristics cannot be represented in isolation from their components, but are shaped by the interactions, dependencies, or relationships they form together in the system.

- Neural Network almost gratifies all features of a complex system. Take Transformers as a good example:

✓Networks: The interacting parts of a complex system form a network. Each layer of BERT is a fully-connected directed graph with each token as a node.

✓Nonlinearity: The activation function segment the space into a highly nonlinear one.

✓Hierarchies: The pipeline structure in pretrained language models.

✓Emergence: The semantically meaningful attention patterns are seen in head-level but not in node-level.

✓Self-organization: Without explicit regulization, at least part of signals are decomposed and captured with expert heads.

https://arxiv.org/abs/1912.05088

Short-Term Ideas

1. Modularity Degree Improvement on Heads

- ✓ From scratch? No, in most modularity works from Bengio, only smaller models like LSTM can be used for experiments.
- ✓ Instead, we could start from existing PLMs to continue pretraining with encouragement on sparse connection between inner modules. Specifically, a possible improvement is on the interaction between the output of attention heads (proven as funtional unit) .
- ✓ Different heads use the same 768-dim channel to convey information, causing competition for capacity and information loss.

$$W_O^H \begin{bmatrix} r^{h_1} \\ r^{h_2} \\ \dots \end{bmatrix} = \left[W_O^{h_1}, W_O^{h_2}, \dots \right] \cdot \begin{bmatrix} r^{h_1} \\ r^{h_2} \\ \dots \end{bmatrix} = \sum_i W_O^{h_i} r^{h_i}$$

Fig. a The output of different attention heads are connected with pointwise addition. Note that the output of a single attention head is also 768-dim (BERT-base), but the intrinsic dimension is 64-dim.

https://transformer-circuits.pub/2021/framework/index.html
https://arxiv.org/abs/2006.16362#:~:text=This%20work%20aims%20to%20enhance,heads%20to%20learn%20shared%20projections.

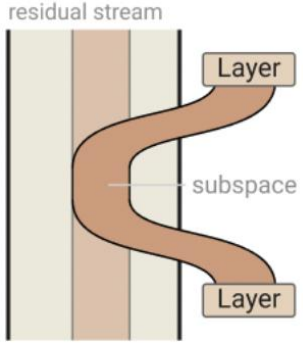


Fig. b The intrinsic dimension of the output of a single head is 64-dim in different subspace, but they are projected into the same 768-dim and pointwisely added and connected.

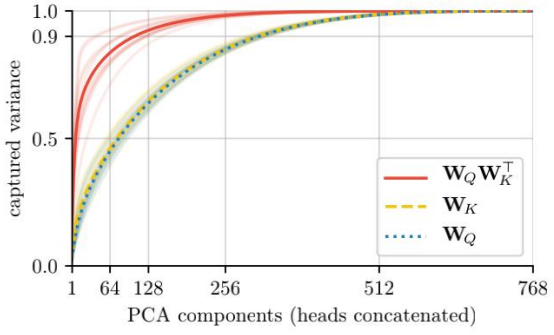


Fig. c Low rank property of $[W_Q^1 W_K^1, W_Q^2 W_K^2, \dots W_Q^h W_K^h]$. It indicates that the rank (i.e. effective data dimensions) of a 768-dim tensor is far from full-rank given an Identity Matrix (full-rank) as input, and therefore it causes information loss.

1. Modularity Degree Improvement on Heads

- ✓ The success of Mixture of attentive experts (MAE) also serves as proof of information loss during head combination. Theoratically, if the 768-dim tensor after head output combination carries all information contained in those head outputs before combination, then MAE won't give additional information more than standard head combination.
- ✓ MAE groups select the least important head, free this information capacity for other important heads. <-- Will this expert be assigned the largest weight?

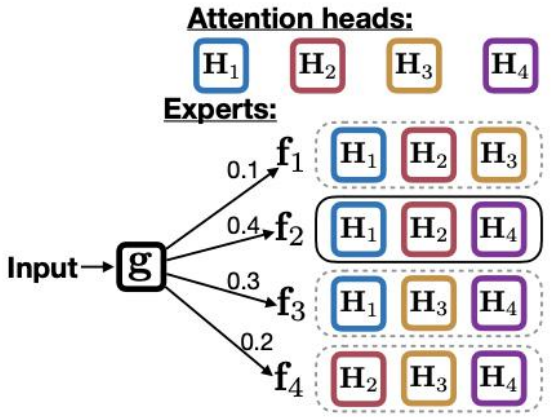


Fig. a MAE groups h-1 head outputs as experts and the combination of experts results in better performance.

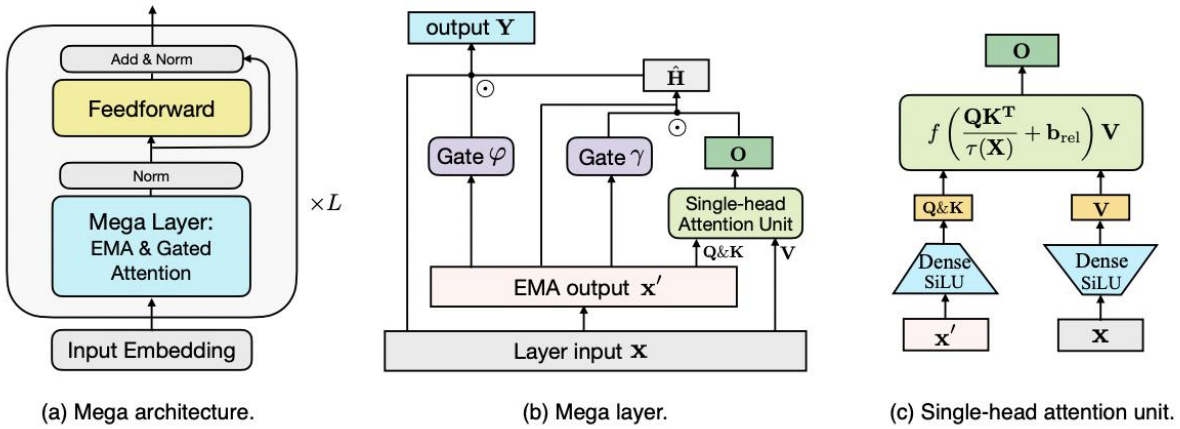
https://aclanthology.org/2020.acl-main.587.pdf

1. Modularity Degree Improvement on Heads

- ✓ Given the fact that pointwise addition in the 768-dim space among head outputs results in information loss, a possible improvement is to sparsify each W_o^{hi} so that different head outputs are encouraged to be projected into different subspace, mitigating competition for capacity. This operation reduces connection between heads and improves the modularity degree of attention heads.
- ✓ The difference from model pruning? --> We treat attention heads as functional units and W_o as connections between these units. The target is different: pruning aims at removing redundant parameters to reduce inference cost, while our target is to reduce connections between heads to reduce capacity competition and improve performance.
- ✓ More techenical methods need be discussed to implement this idea.

2. Sequence Modeling with Stochastic Process

- ✓ A recent work, MEGA, introduced moving average of inputs before current token as gates in the attention operation, and improved long-range sequence tasks significantly.
- ✓ Using specific stochastic process (e.g. Gaussian process), we can extend this method.



(a) Mega architecture. (b) Mega layer. (c) Single-head attention unit.

Fig. a MEGA uses moving average as gates to introduce additional inductive bias of the attention operation.

Table 2: (**Long Range Arena**) Accuracy on the full suite of long range arena (LRA) tasks, together with training speed and peak memory consumption comparison on the Text task with input length of 4K. ‡ indicates results replicated by us.

Models	ListOps	Text	Retrieval	Image	Pathfinder	Path-X	Avg.	Speed	Mem.
XFM	36.37	64.27	57.46	42.44	71.40	✗	54.39	–	–
XFM‡	37.11	65.21	79.14	42.94	71.83	✗	59.24	1×	1×
Reformer	37.27	56.10	53.40	38.07	68.50	✗	50.67	0.8×	0.24×
Linformer	35.70	53.94	52.27	38.56	76.34	✗	51.36	5.5×	0.10×
BigBird	36.05	64.02	59.29	40.83	74.87	✗	55.01	1.1×	0.30×
Performer	18.01	65.40	53.82	42.77	77.05	✗	51.41	**5.7**×	**0.11**×
Luna-256	37.98	65.78	79.56	47.86	78.55	✗	61.95	4.9×	0.16×
S4-v1	58.35	76.02	87.09	87.26	86.05	88.10	80.48	–	–
S4-v2	59.60	86.82	90.90	88.65	94.20	96.35	86.09	–	–
S4-v2‡	59.10	86.53	90.94	88.48	94.01	96.07	85.86	4.8×	0.14×
MEGA	**63.14**	**90.43**	**91.25**	**90.44**	**96.01**	**97.98**	**88.21**	2.9×	0.31×
MEGA-chunk	58.76	90.19	90.97	85.80	94.41	93.81	85.66	5.5×	0.13×

Fig. b This inductive bias is especially useful for long-range sequence problems.

https://arxiv.org/pdf/2209.10655.pdf

2. Sequence Modeling with Stochastic Process

- ✓ A sequence (language, speech) can be formalized as a discrete time series and a sample from a stochastic process.
- ✓ A major difference between autoregressive modeling ($P_{\boldsymbol{\theta}}(Y|X) = \prod_{i=1}^{n} P_{\boldsymbol{\theta}}(y_i|Y_{<i}, X)$) /moving average ($\mathbf{y}_t = \boldsymbol{\alpha} \odot \mathbf{x}_t + (1 - \boldsymbol{\alpha} \odot \boldsymbol{\delta}) \odot \mathbf{y}_{t-1}$) and stochastic process modeling: current value depends on past values v.s. current value depends on stochastic process learned from global information.

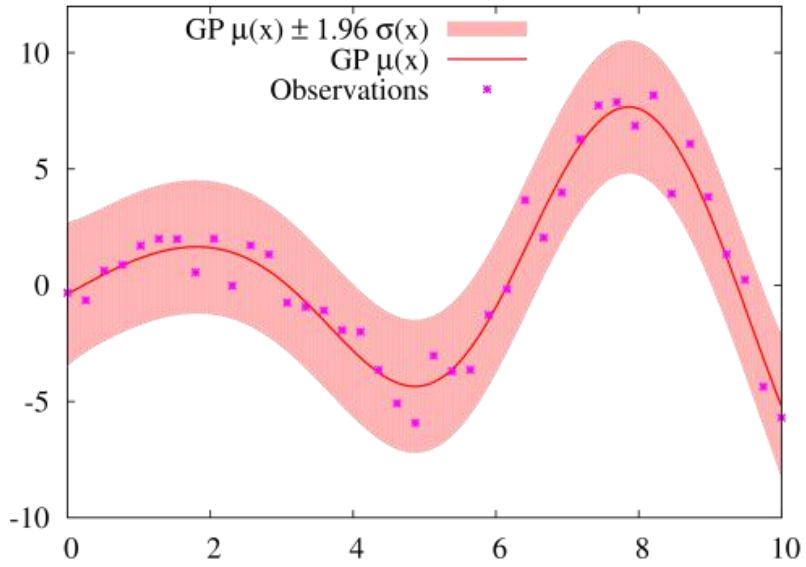


Fig. a A case of Guassian process.

https://arxiv.org/pdf/2209.10655.pdf