GiLT: Augmenting Transformer Language Models with Dependency Graphs

Tianyu Huang, Yida Zhao, Chuyan Zhou, Kewei Tu

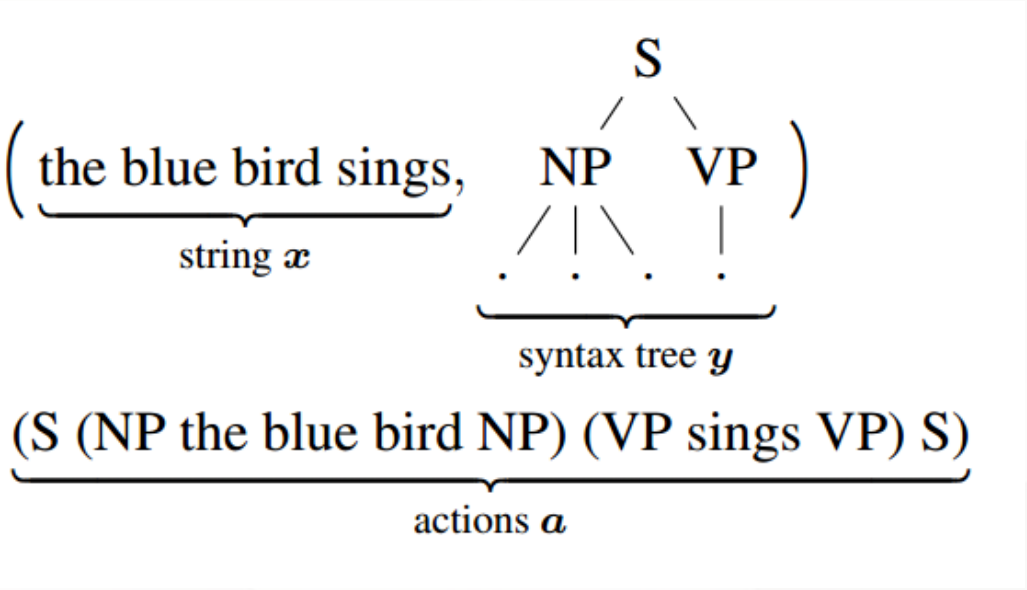
School of Information Science and Technology, ShanghaiTech University

Shanghai Engineering Research Center of Intelligent Vision and Imaging

Background

上海科技大学
ShanghaiTech University

- Transformer language models
  a) Have achieved remarkable success
  b) Do not explicitly encode linguistic structures, which have been deemed essential in traditional NLP
- Syntactic language models (SLMs)
  a) Jointly modeling words and syntactic structures
  b) Have been shown to improve syntactic generalization



Top: A pair of string x and tree y
Bottom: Linearized sequence of Transformer Grammars (Sartran et al., 2022)

Our work

上海科技大学
ShanghaiTech University

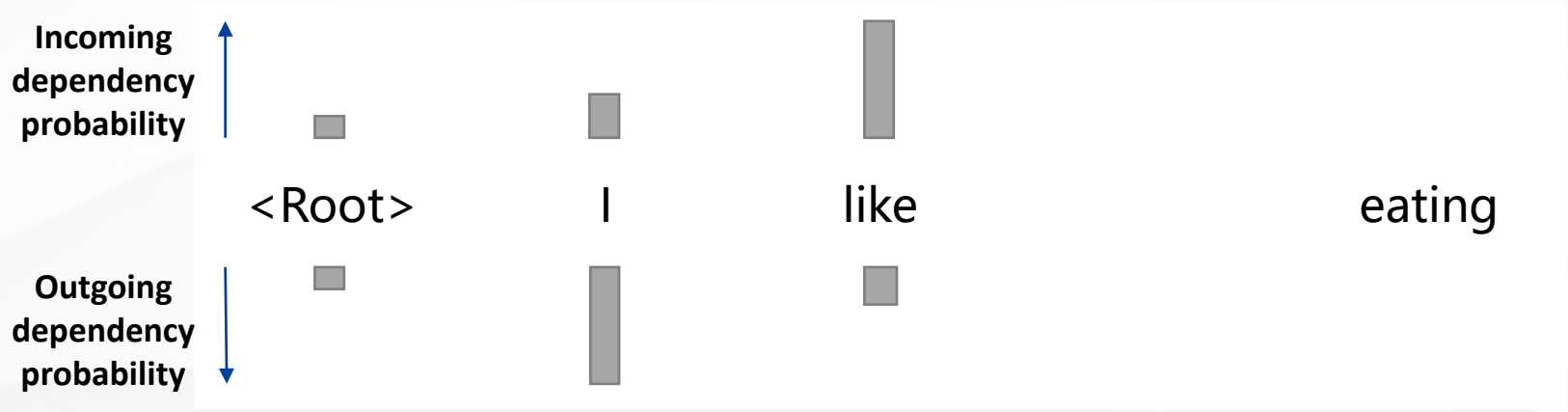
- Limitations of prior work
  a) Most rely on constituency trees
  b) Many of them insert extra tokens into the sequence
- Our work

  Propose Graph-Infused Layers, resulting Graph-Infused Layer Transformer LM (GiLT)
  a) Use dependency graphs
  b) Do not change the LM's input and output space
  c) Extract graph-based features and use them to modulate self-attention

Graph-Infused Layer

上海科技大学
ShanghaiTech University

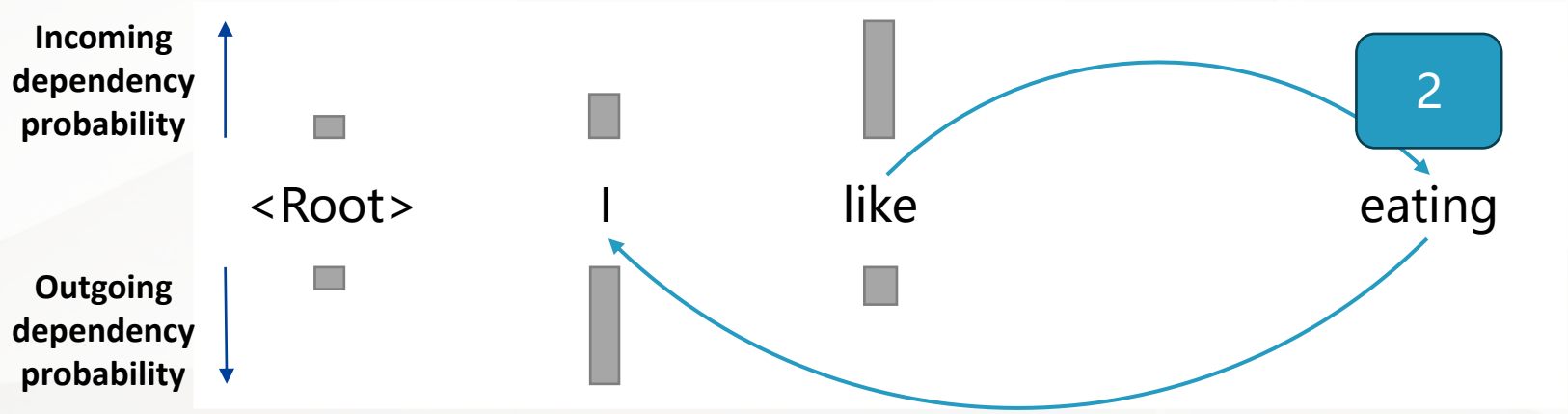
- At each step, when the LM generates a new word, GiLT does three things:
  a) **Scores all possible dependencies from and to the new word using a biaffine mechanism**
  b) Updates the dependency graph by first predicting the number of dependencies
  c) Extracts features from the partially built graph to create a feature tape



Graph-Infused Layer

上海科技大学
ShanghaiTech University

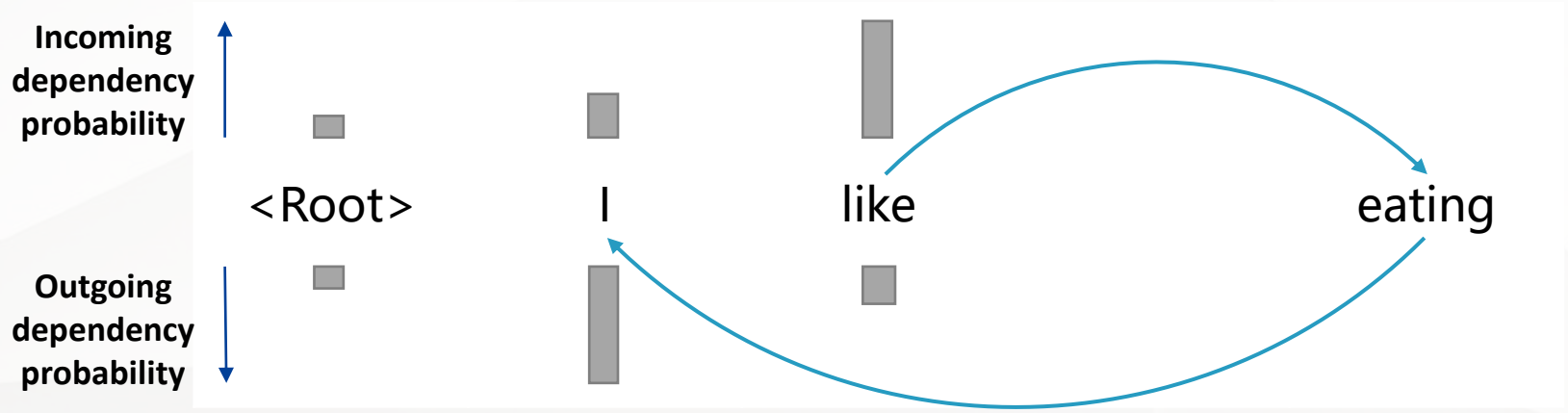
- At each step, when the LM generates a new word, GiLT does three things:
  a) Scores all possible dependencies from and to the new word using a biaffine mechanism
  b) Updates the dependency graph by first predicting the number of dependencies
  c) Extracts features from the partially built graph to create a feature tape



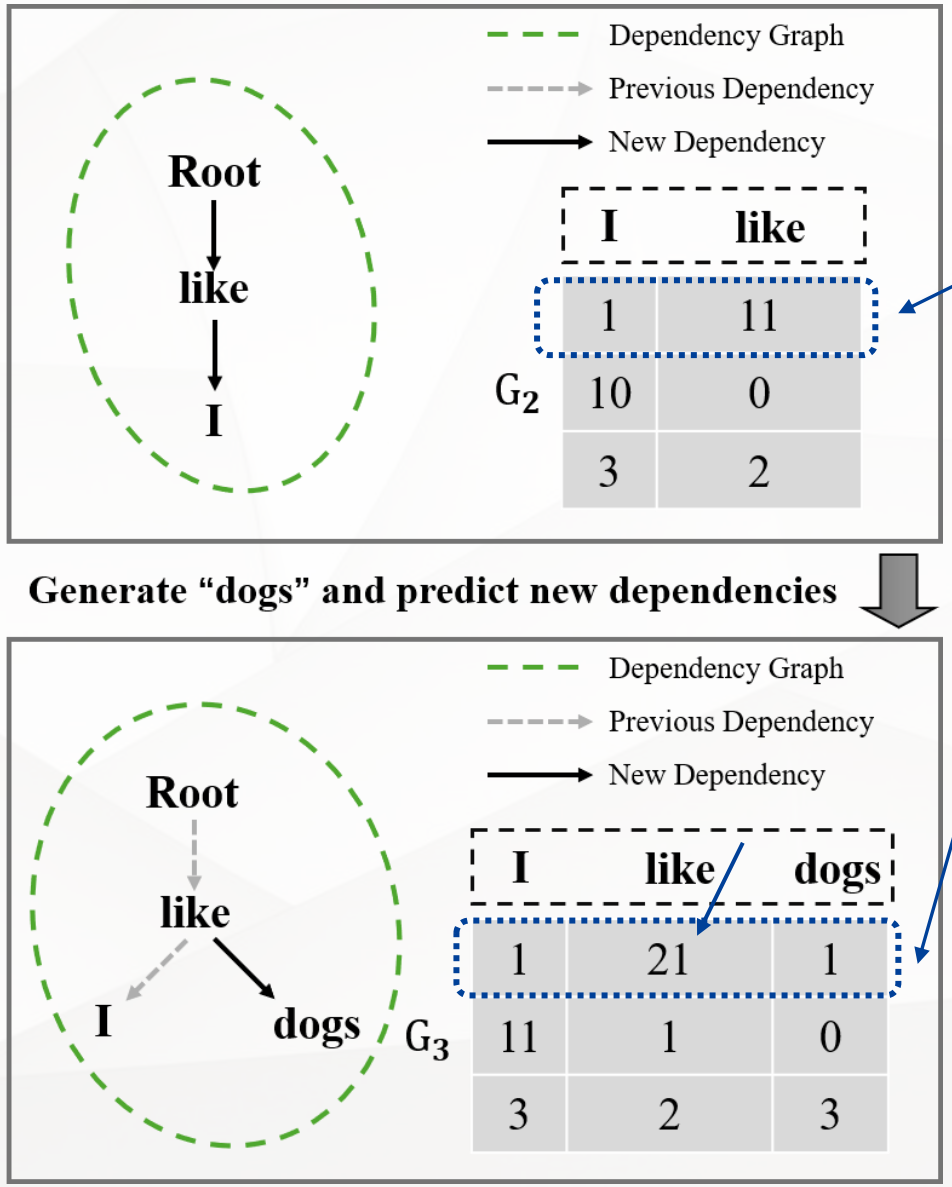
Graph-Infused Layer

上海科技大学
ShanghaiTech University

- At each step, when the LM generates a new word, GiLT does three things:
  a) Scores all possible dependencies from and to the new word using a biaffine mechanism
  b) Updates the dependency graph by first predicting the number of dependencies
  c) Extracts features from the partially built graph to create a feature tape



Graph-Based Feature Tapes



- Contains three types of information
  - a) **Degree**

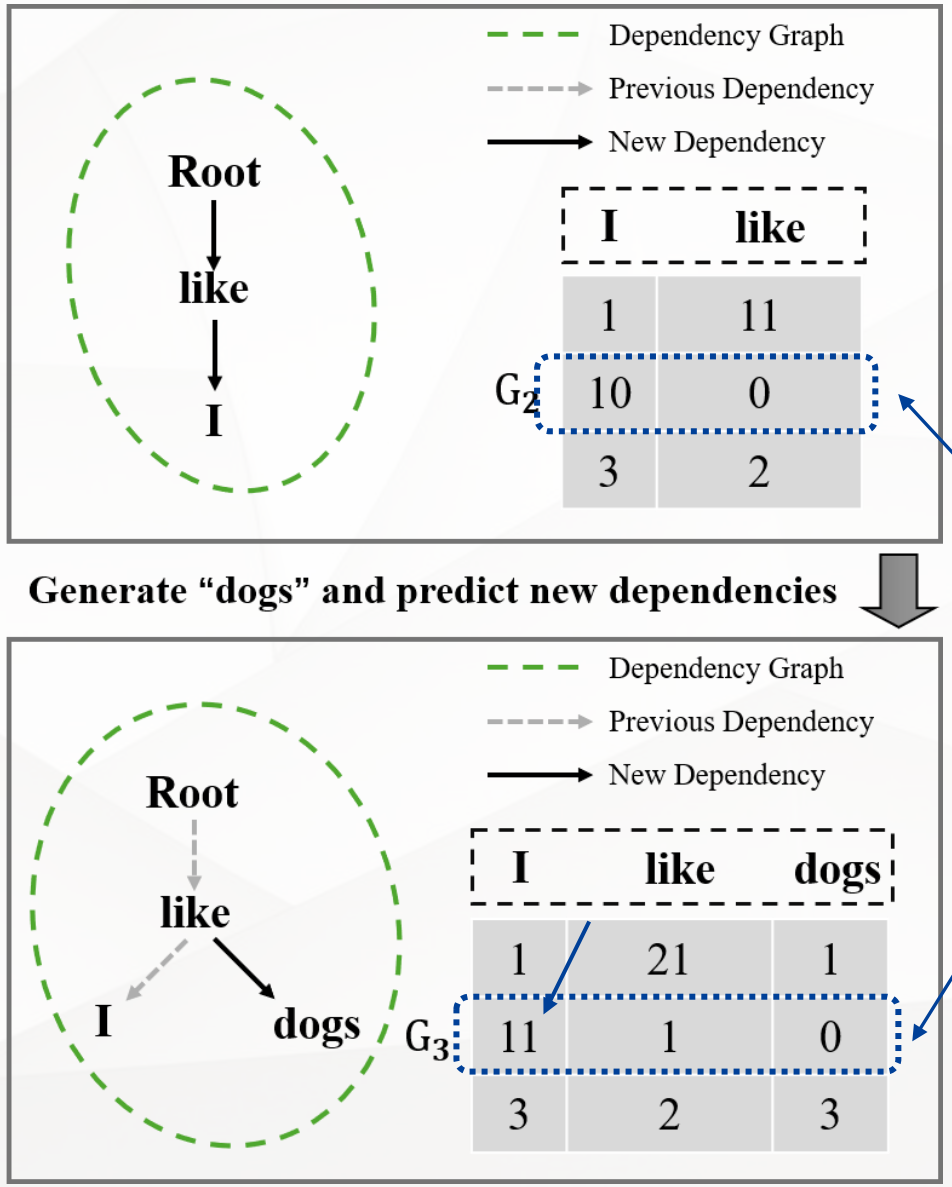
    weighted sum of in-degree c_{in} and out-degree c_{out}:

    $$m_{out}c_{out} + m_{in}c_{in}$$

    $m_{out} = 10, m_{in} = 1$ in this figure
  - b) **Distance**

    shortest path length on the graph: When traversing a dependency along its direction, we weight it by m_{out}; against the direction, we use m_{in}

Graph-Based Feature Tapes



- Contains three types of information
  - a) **Degree**

    weighted sum of in-degree c_{in} and out-degree c_{out}:

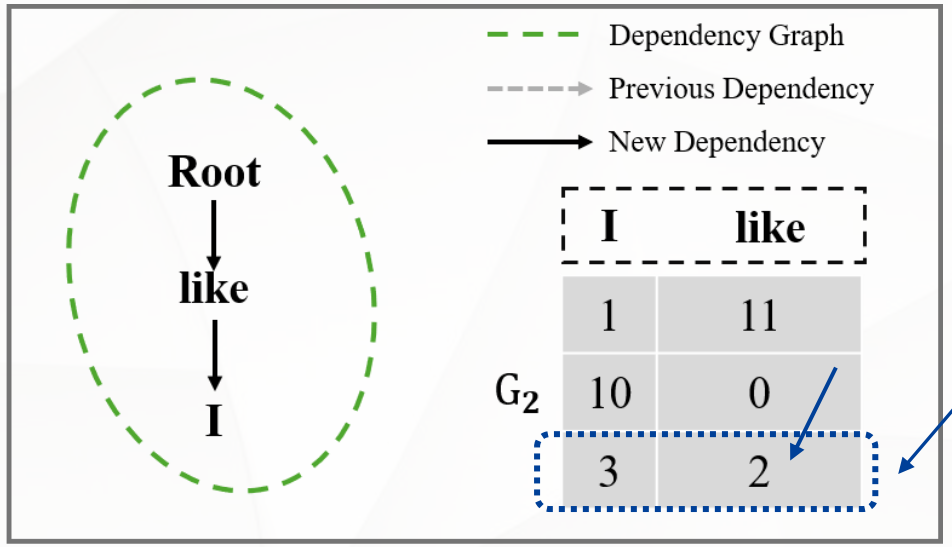
    $$m_{out}c_{out} + m_{in}c_{in}$$

    $m_{out} = 10, m_{in} = 1$ in this figure
  - b) **Distance**

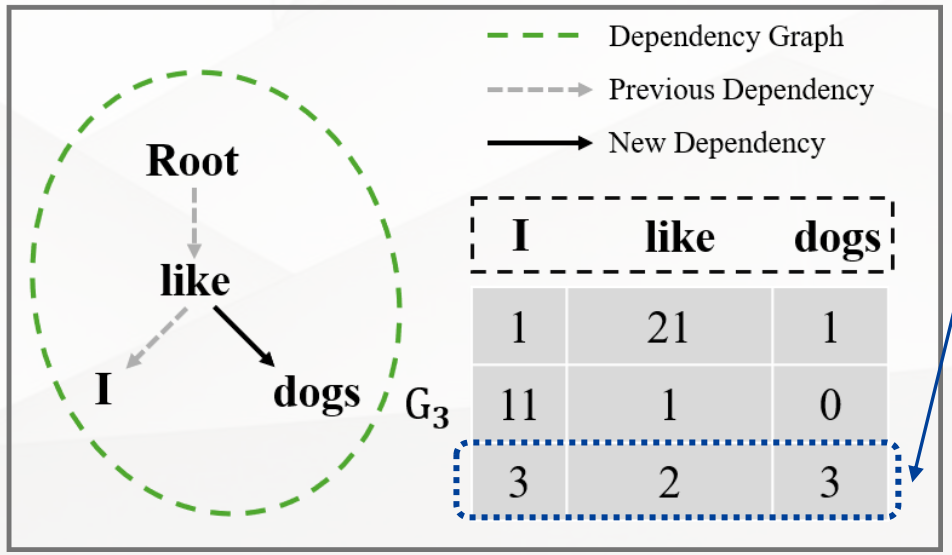
    shortest path length on the graph:

    When traversing a dependency along its direction, we weight it by m_{out}; against the direction, we use m_{in}

Graph-Based Feature Tapes



Generate "dogs" and predict new dependencies



- Contains three types of information
  - c) **Depth**

    the value equals to the distance ($m_{out} = m_{in} = 1$) to the **Root** + 1
- Computing attention scores:

$$G_3 \in \mathbb{N}^{(3*3)}$$

↓ embedding layer

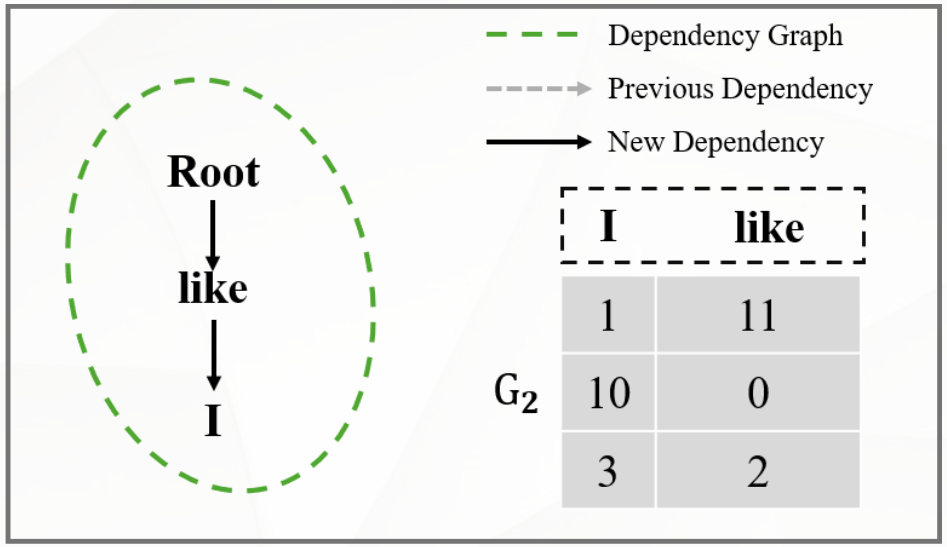
$$\tilde{e}_3 \in \mathbb{R}^{(3*3*\tilde{d})}$$

↓ linear projection f_l

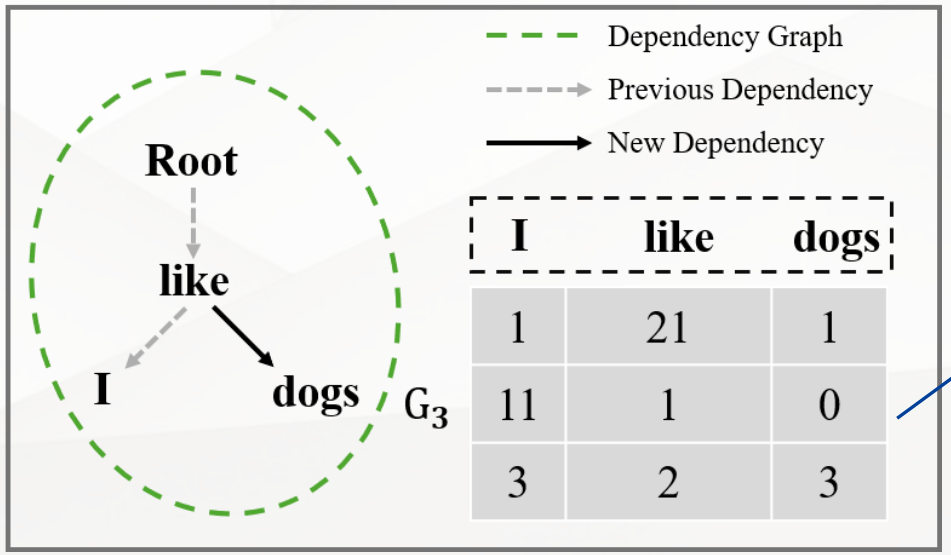
$$e_3^l \in \mathbb{R}^{(3*d)}$$

$$\tilde{\alpha}_{kj}^l = \left[h_j^l + \boldsymbol{e_{kj}^l}\right]^T W_k^T W_q h_k^l$$

Graph-Based Feature Tapes



Generate "dogs" and predict new dependencies



- Contains three types of information
  - c) **Depth**

    the value equals to the distance ($m_{out} = m_{in} = 1$) to the **Root** + 1
- Computing attention scores:

$$G_3 \in \mathbb{N}^{(3*3)}$$

↓ embedding layer

$$\tilde{e}_3 \in \mathbb{R}^{(3*3*\tilde{d})}$$

↓ linear projection f_l

$$e_3^l \in \mathbb{R}^{(3*d)}$$

$$\tilde{\alpha}_{kj}^l = \left[h_j^l + \boldsymbol{e_{kj}^l}\right]^T W_k^T W_q h_k^l$$

Experiment

上海科技大学 ShanghaiTech University

- Language modeling & Syntactic generalization

Model	PPL ↓	10% BLiMP ↑	SG ↑
Models that add structural tokens to inputs			
PLM	29.8♠	75.1♣	76.9
PLM-Mask	49.1♠	75.3♣	74.7
TG	18.4	73.5♣	79.6
DTG	14.9	76.1♣	81.2
Models that do not add extra tokens to inputs			
TXL (baseline)	14.8	75.1	72.1
TXL-Large (334M)	14.7	75.4	71.8
Pushdown-LM	**14.6**	74.0	74.6
Ours GiLT-DP	15.5	**76.1**	72.3
Ours GiLT-PAS	15.0	73.0	75.7
Ours GiLT-DM	14.9	74.9	76.3
Ours GiLT-PSD	14.9	75.7	**79.7**

Benchmark on syntactic generalization

Trained on different dependency graphs

Table 1: Results on language modeling and syntactic generalization. The best values over models that do not add extra tokens are **bold**. Overall best values are underlined. All PPL results except for that of TXL are approximated upper bounds. SG scores are computed without the "other" suite. ♠ denotes that the PPL is taken from the original paper. ♣ denotes that the result is evaluated with the full BLiMP dataset reported in the original paper.

Experiment

上海科技大学 ShanghaiTech University

- Finetuning on pretrained LM

24 layers pretrain model

Continue training on BLLIP-LG

replace its last 12 layers with Graph-infused layers and continue training on BLLIP-LG

Model	10%BLiMP ↑	SG ↑
pretrained GPT2	82.8	79.4
Post-GPT2	83.1	84.6
GiLT-GPT2	**83.2**	**85.5**

Finetune these 2 models separately on downstream tasks

Model	RTE ↑	SST2 ↑	MRPC ↑	STS-B ↑
Post-GPT2	64.1	94.84	80.75/86.29	84.2/83.6
GiLT-GPT2	**65.3**	**95.11**	**81.39/86.97**	**85.2/84.3**

Experiment

上海科技大学 ShanghaiTech University

- Finetuning on pretrained LM

24 layers pretrain model

Continue training on BLLIP-LG

replace its last 12 layers with Graph-infused layers and continue training on BLLIP-LG

Model	10%BLiMP ↑	SG ↑
pretrained GPT2	82.8	79.4
Post-GPT2	83.1	84.6
GiLT-GPT2	**83.2**	**85.5**

Finetune these 2 models separately on downstream tasks

Model	RTE ↑	SST2 ↑	MRPC ↑	STS-B ↑
Post-GPT2	64.1	94.84	80.75/86.29	84.2/83.6
GiLT-GPT2	**65.3**	**95.11**	**81.39/86.97**	**85.2/84.3**

Other Experiments

上海科技大学
ShanghaiTech University

- See our paper for more experiments
  - Ablation study
  - Efficiency comparison

Thanks

Paper

Code