

GNN-LM: Language Modeling Based on Global Contexts via GNN

Yuxian Meng, Shi Zong, Xiaoya Li, Xiaofei Sun, Tianwei Zhang, Fei Wu, Jiwei Li



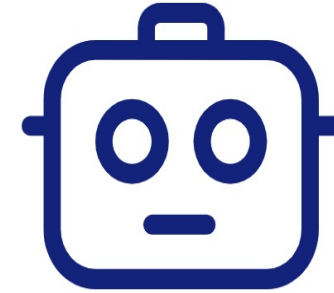
Memorization

Close-book Exam Strategy

A Large-scale Database



Memorize Data



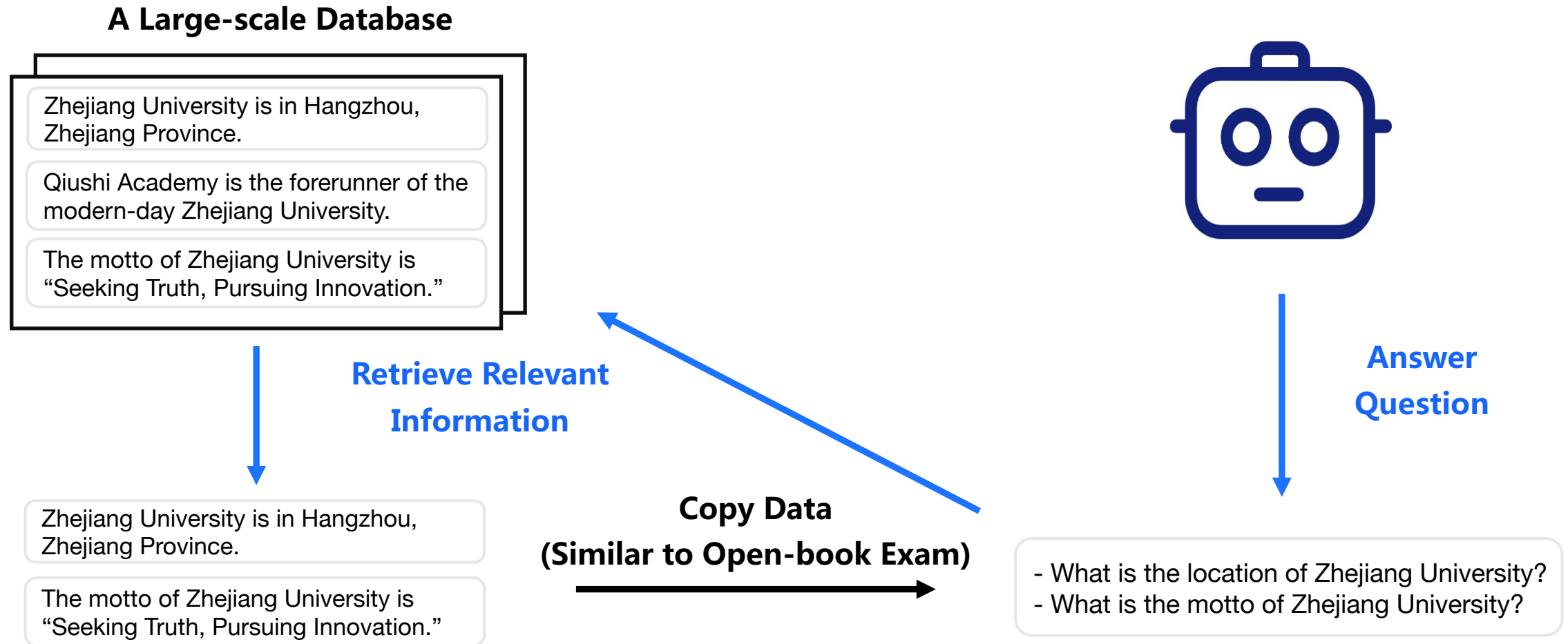
Answer
Question

Not Seen
(Similar to Close-book Exam)

- What is the location of Zhejiang University?
- What is the motto of Zhejiang University?

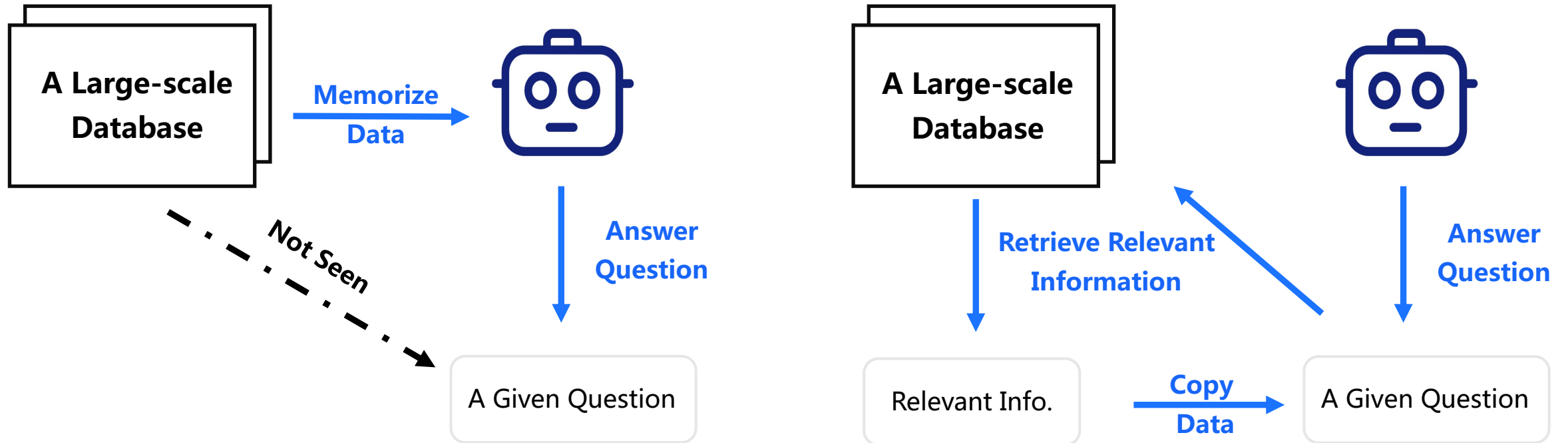
Copy

Open-book Exam Strategy



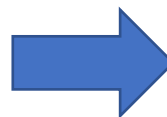
Copy, not Memorization!

Open-book Exam is Easier than Close-book



Memorization-based Method: Models have to memorize data and knowledge

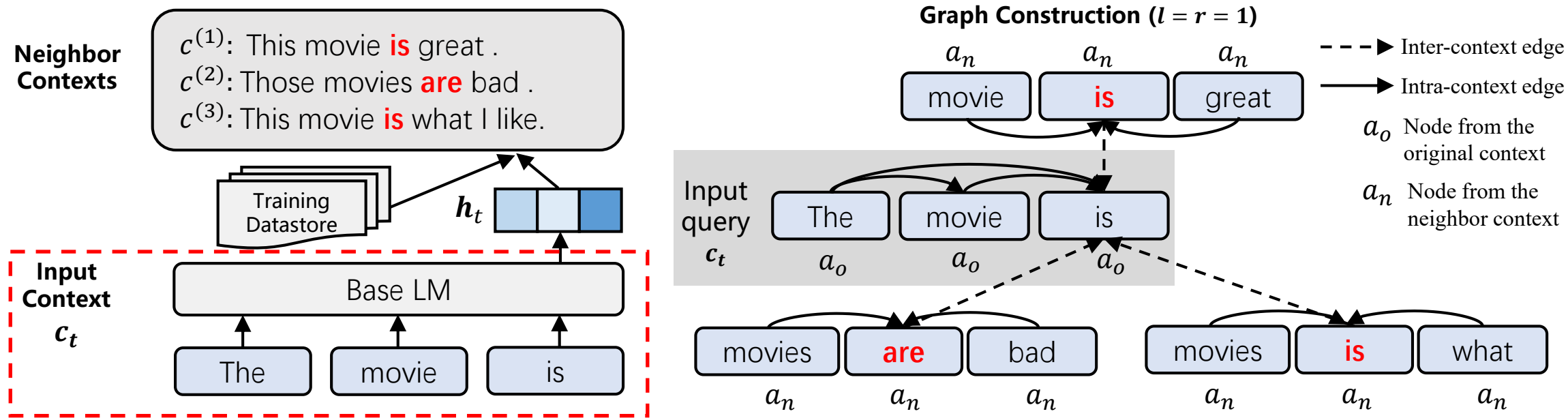
Harder



Copy-based Method: Models only need to retrieve and search for relevant information

Easier

GNN-LM

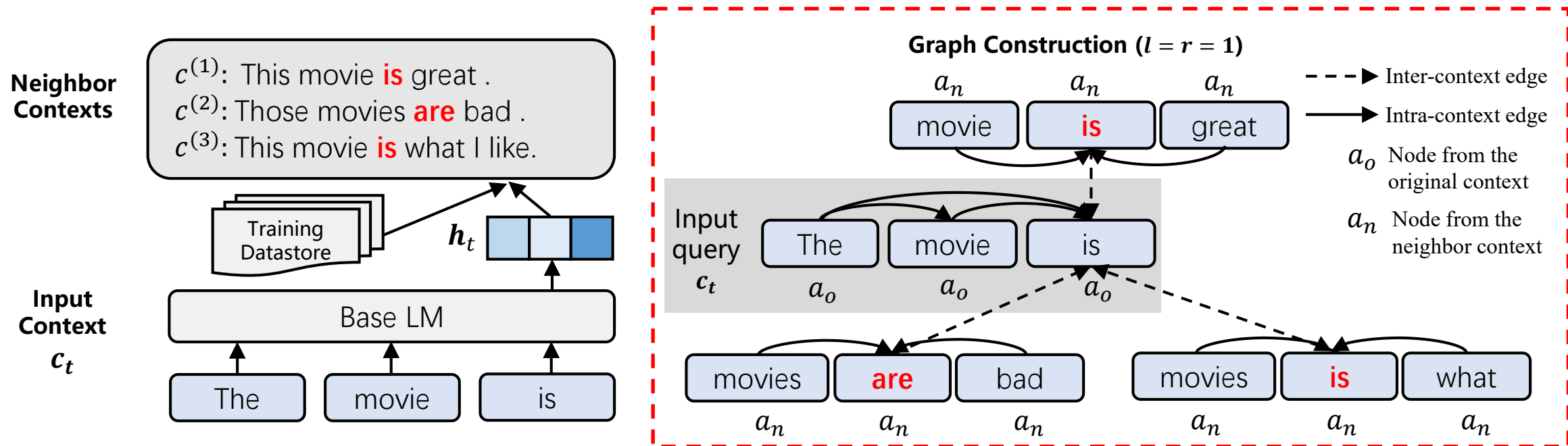


Input Context Encoding

- A vanilla language model $f(\cdot)$ is used to encode a given input sequence $c_t = (w_1, w_2, \dots, w_{t-1})$

$$h_t = f(c_t) \in \mathbb{R}^d$$

GNN-LM



Graph Construction and Message Passing

- h_t is used to query neighbor contexts and build a directed heterogeneous graph
- A self-attention augmented GNN is applied to update h_t and get an estimate for $p_{\text{LM}}(w_t | c_t)$

Additional information from reference tokens are incorporated via GNN

Next Token Prediction

Probability for the next token is estimated by the linear interpolation of $p_{\text{LM}}(w_t|\mathbf{c}_t)$ and $p_{\text{kNN}}(w_t|\mathbf{c}_t)$

$$p(w_t|\mathbf{c}_t) = \lambda p_{\text{kNN}}(w_t|\mathbf{c}_t) + (1 - \lambda) p_{\text{LM}}(w_t|\mathbf{c}_t)$$

kNN-LM (Khandelwal et al., 2019)

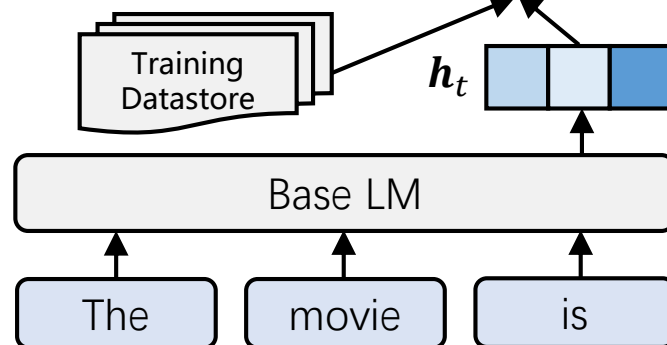
GNN-LM

GNN-LM

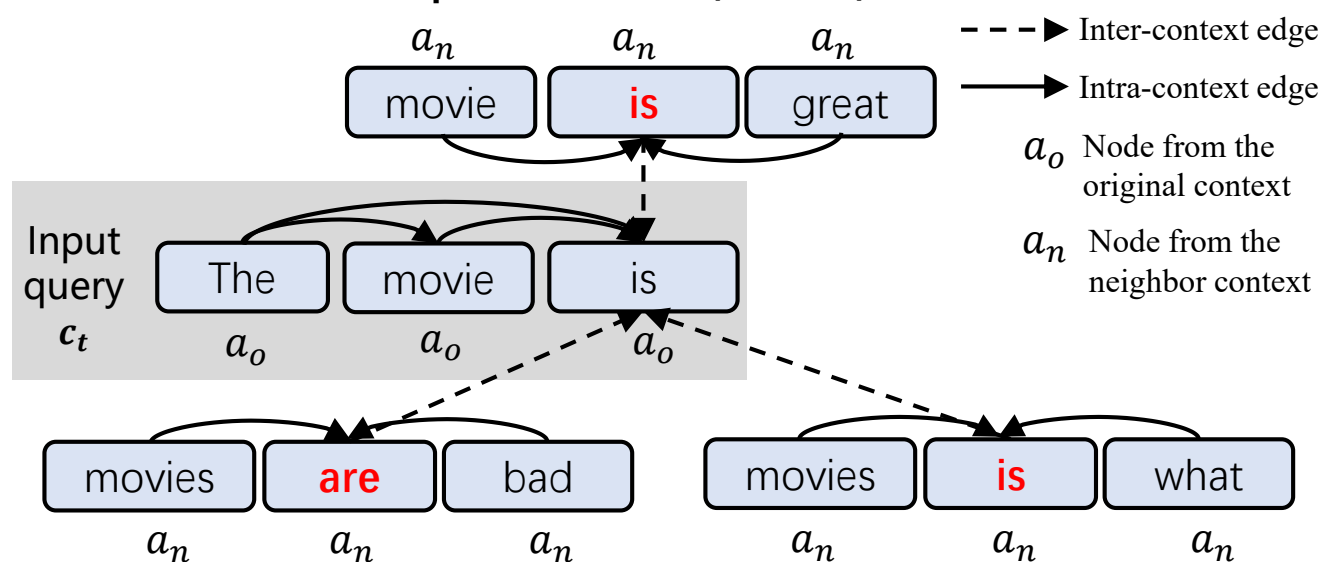
Neighbor Contexts

$c^{(1)}$: This movie **is** great .
 $c^{(2)}$: Those movies **are** bad .
 $c^{(3)}$: This movie **is** what I like.

Input Context \mathbf{c}_t



Graph Construction ($l = r = 1$)



Experimental Results

WikiText-103 Dataset


Model	# Param	Test ppl (\downarrow)
Hebbian + Cache (Rae et al., 2018)	151M	29.9
Transformer-XL (Dai et al., 2019)	257M	18.3
Transformer-XL + Dynamic Eval (Krause et al., 2019)	257M	16.4
Compressive Transformer (Rae et al., 2019)	-	17.1
KNN-LM + Cache (Khandelwal et al., 2019)	257M	15.8
Sandwich Transformer (Press et al., 2020a)	247M	18.0
Shortformer (Press et al., 2020b)	247M	18.2
SegaTransformer-XL (Bai et al., 2021)	257M	17.1
Routing Transformer (Roy et al., 2021)	-	15.8
base LM (Baevski & Auli, 2018)	247M	18.7
+GNN	274M	16.8
+GNN+ k NN	274M	14.8

- GNN-LM reduces the base LM perplexity from 18.7 to 16.8
- Combining GNN and k NN leads to a new **SOTA** result of **14.8**

Experimental Results

One Billion Word Dataset

Model	# Param	Test ppl (↓)
LSTM+CNN (Jozefowicz et al., 2016)	1.04B	30.0
High-Budget MoE (Shazeer et al., 2016)	5B	28.0
DynamicConv (Wu et al., 2018)	0.34B	26.7
Mesh-Tensorflow (Shazeer et al., 2018)	4.9B	24.0
Evolved Transformer (Shazeer et al., 2018)	-	28.6
Transformer-XL (Dai et al., 2019)	0.8B	21.8
Adaptive inputs (base) (Baevski & Auli, 2018)	0.36B	25.2
Adaptive inputs (large) (Baevski & Auli, 2018)	0.46B	23.9
base LM (Baevski & Auli, 2018)	1.03B	23.0
+ <i>k</i> NN	1.02B	22.8
+GNN	1.05B	22.7
+GNN+ <i>k</i> NN	1.05B	22.5



GNN-LM helps base LM reduce **0.5** perplexity with
only **27M** additional parameters

Experimental Results

Enwik8 Dataset

Model	# Param	BPC (↓)
64L Transformer (Al-Rfou et al., 2019)	235M	1.06
18L Transformer-XL (Dai et al., 2019)	88M	1.03
24L Transformer-XL (Dai et al., 2019)	277M	0.99
24L Transformer-XL + Dynamic Eval (Krause et al., 2019)	277M	0.94
Longformer (Beltagy et al., 2020)	102M	0.99
Adaptive Transformer (Sukhbaatar et al., 2019)	209M	0.98
Compressive Transformer (Rae et al., 2019)	277M	0.97
Sandwich Transformer (Press et al., 2020a)	209M	0.97
12L Transformer-XL (Dai et al., 2019)	41M	1.06
+ k NN	41M	1.04
+GNN	48M	1.04
+GNN+ k NN	48M	1.03

GNN- k NN-LM outperforms base LM by 0.03 Bit per Character (BPC), achieving 1.03 BPC with only **48M** parameters

Take Away

- We propose GNN-LM, a new paradigm for language modeling
 - Enable a LM model to reference similar contexts from the entire training corpus as cues for prediction
- Our method outperforms strong baselines in benchmark datasets
 - Achieve the state-of-the-art results on WikiText-103 with a test perplexity of 14.8

Code is publicly available at:
<https://github.com/ShannonAI/GNN-LM>