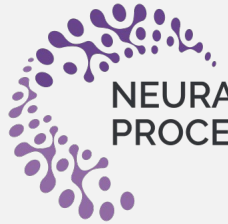
浙江大学 计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY
ZHEJIANG UNIVERSITY

蚂蚁集团
ANT GROUP



NEURAL INFORMATION
PROCESSING SYSTEMS

MKGL: Mastery of a Three-Word Language

Lingbing Guo, Zhongpu Bo, Zhuo Chen, Yichi Zhang, Jiaoyan Chen,
Yarong Lan, Mengshu Sun, Zhiqiang Zhang, Yangyifei Luo, Qian Li,
Qiang Zhang, Wen Zhang*, Huajun Chen*

NeurIPS 2024

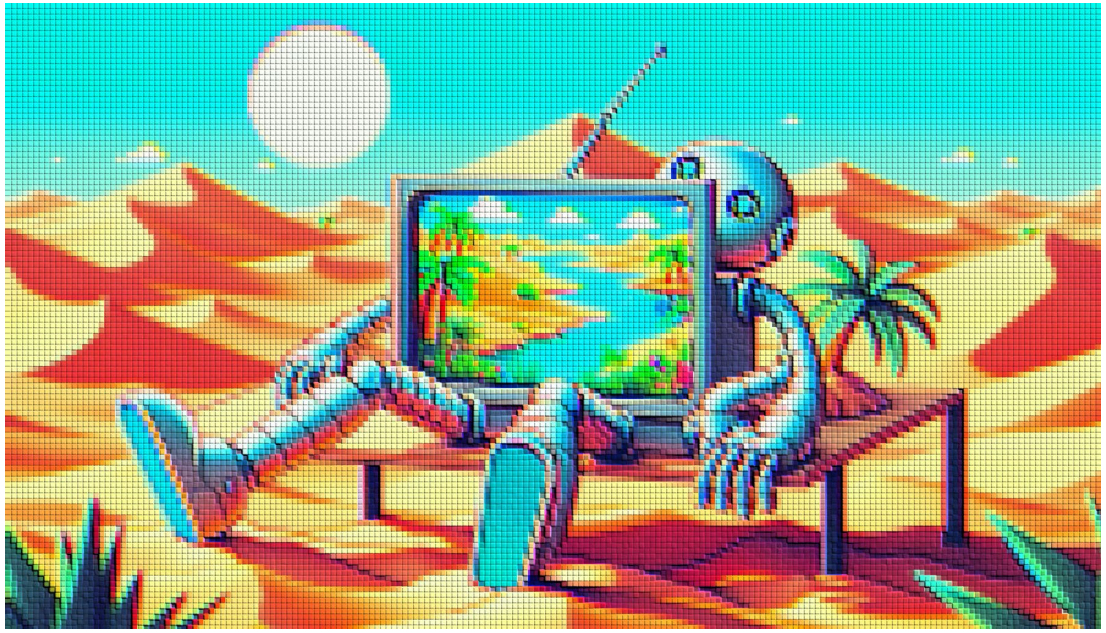
*: corresponding authors
github.com/zjukg/NCGNN

Background

Knowledge graphs (KGs) have been important resources for many data-driven applications.

Yet, the rapid advancement of large language models (LLMs) have challenged the conventional reliance on KGs.

To mitigate the “hallucination” problem, many recent studies start to resort KGs once again.



Background

In our paper, we investigate the capacity of LLMs to study a new KG language (KGL), and it has the following features:

A KGL sentence exactly consists 3 words, an entity noun, a relation verb, and ending with another entity noun.

The words in KGL are not directly readable to an LLM.

However, their embeddings are familiar to the LLM, as they are constructed from the token embeddings of the LLM.

"Wendee Lee is an actor in Mighty Morphin Power Rangers"

to KGL (i.e., triplet)

(Wendee Lee, actor of, Mighty Morphin Power Rangers)

: a KGL word cannot be further split.

Background

Why learn KGL:

With KG, entities can be identified better even they have same name.

With LLM, the model do more creative KG tasks like triplet generation.

Also, improves relevant KG tasks with the power of LLM.

Mix-generation of natural language and KGL (future work).

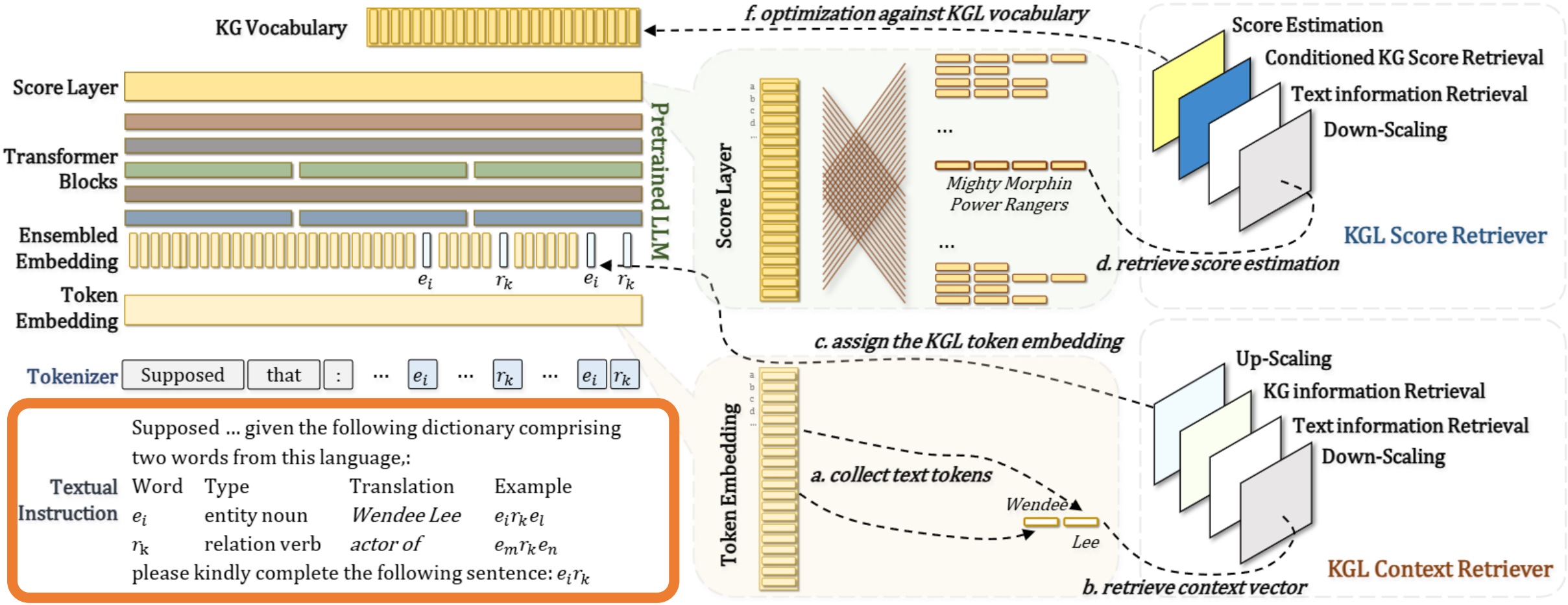
“Wendee Lee is an actor in Mighty Morphin Power Rangers”

to KGL (i.e., triplet)

(Wendee Lee, actor of, Mighty Morphin Power Rangers)

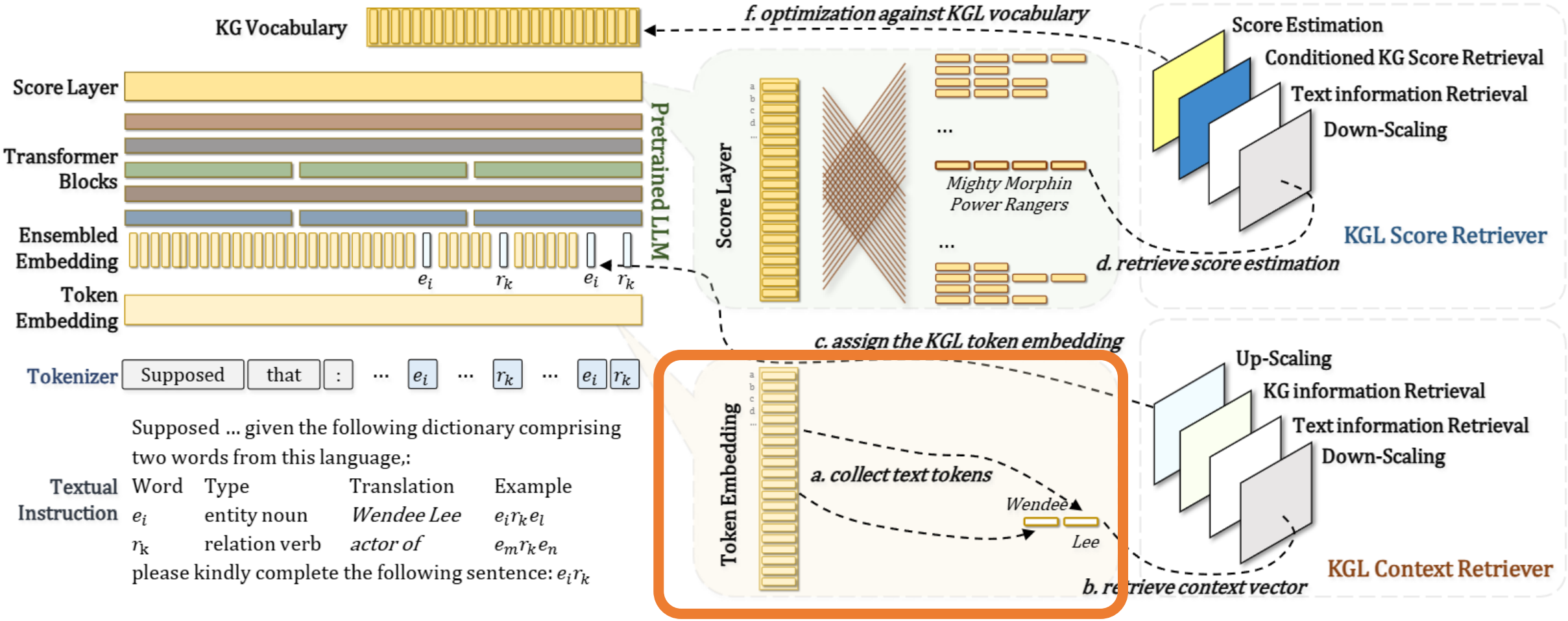
: a KGL word cannot be further split.

Our method



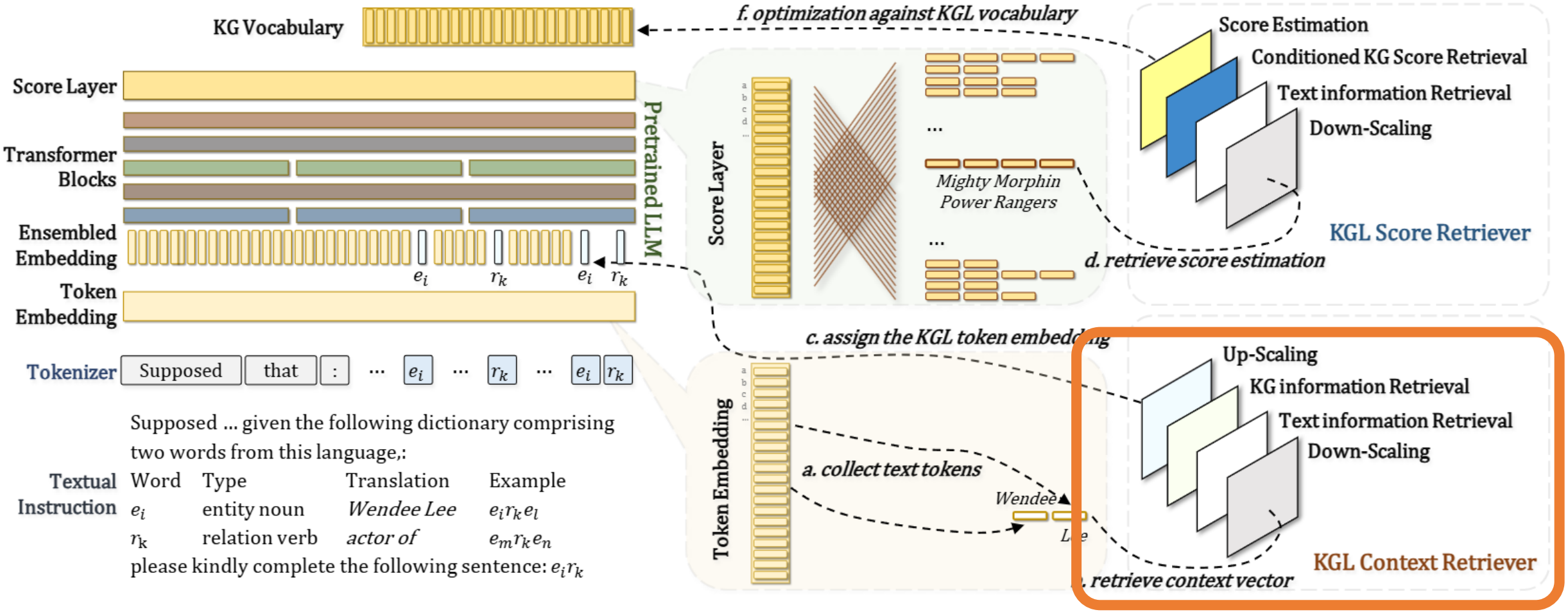
The task is to construct new KG sentences. At first, the tokenizer will tokenizes the input text, where the **entities and relations are represented as special tokens out of the original vocabulary**.

Our method



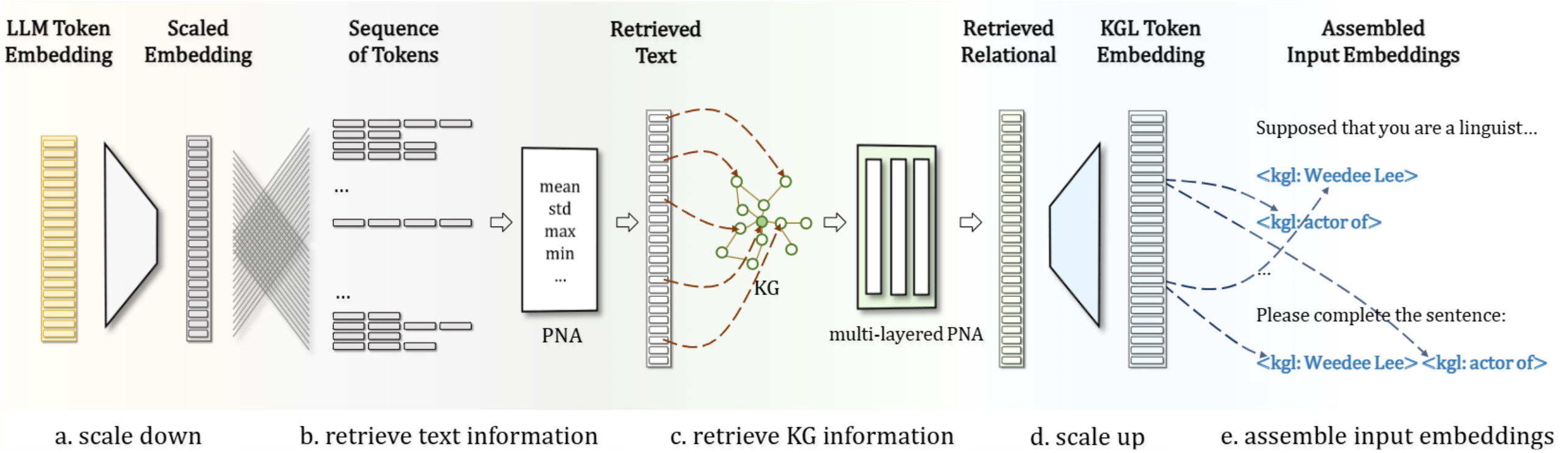
Then, for these special KGL tokens, we collect the embeddings of their constituting text tokens as feature to produce their text context vectors.

Our method



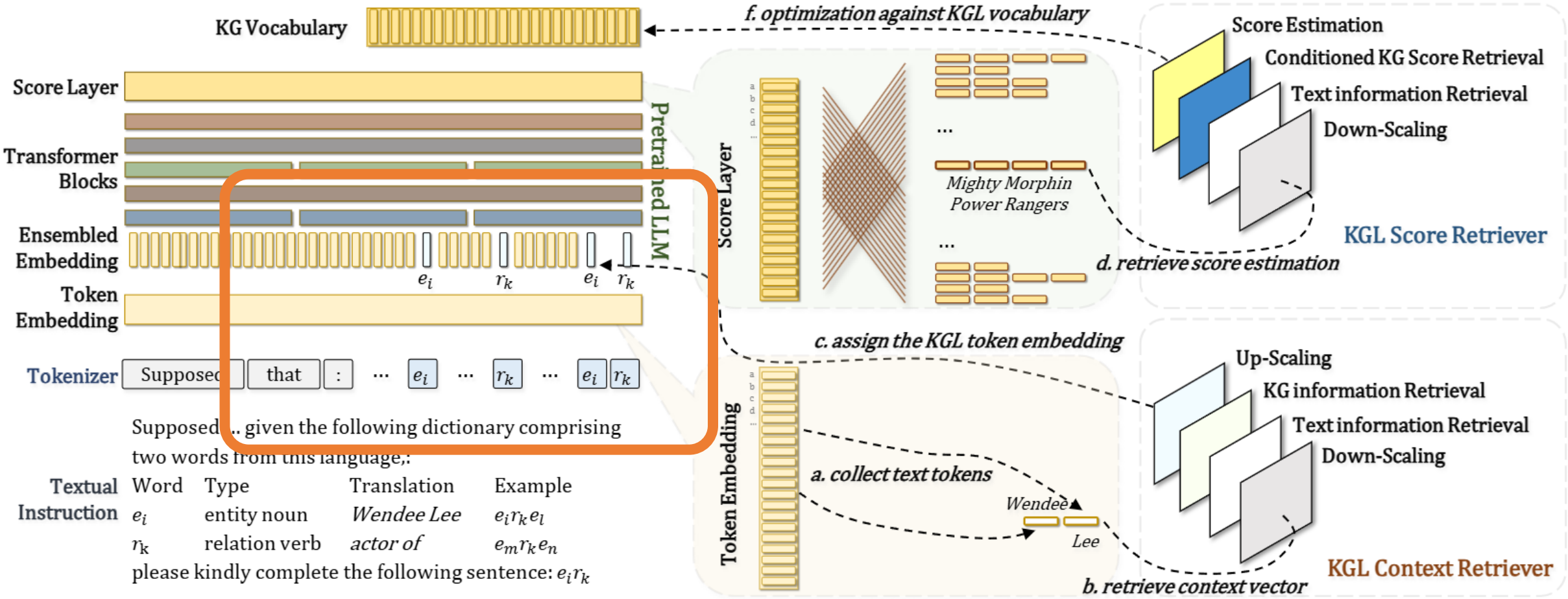
We use a 4-step retriever to encode the textual and relational information into KGL token embeddings.

Our method



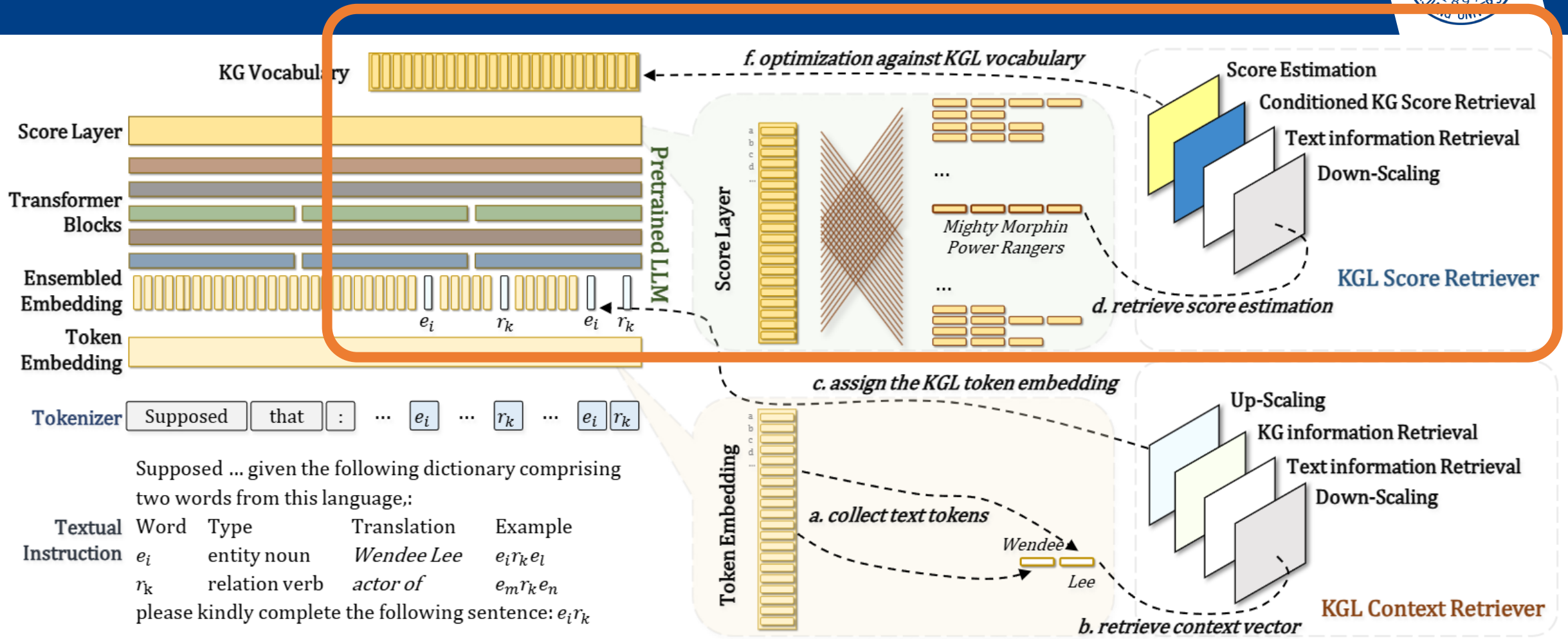
Specifically, we use PNA to aggregate the text and KG information. The first and the last steps in the retriever are LoRA-like down-scaling and up-scaling operations.

Our method



Finally, the output will be assigned as the embeddings of these special KGL tokens.

Our method



Similar to the context retriever, we also design a score retriever to retriever the score information.

Experiments

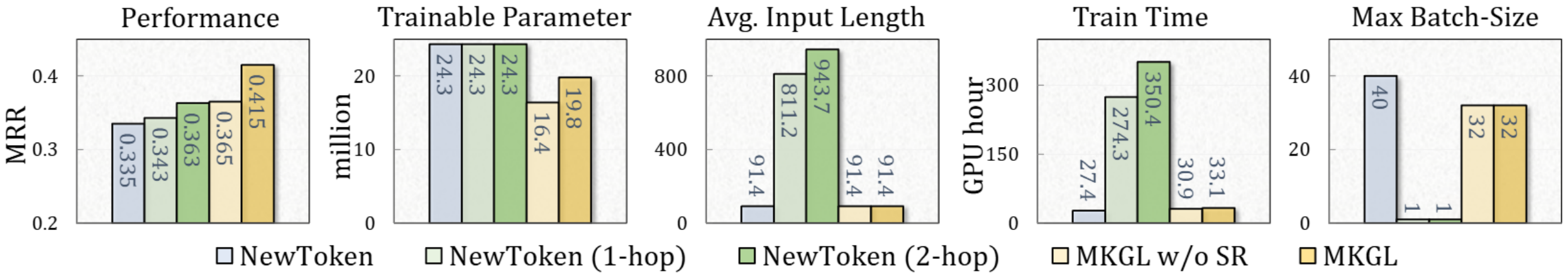
Our method MKGL has better or competitive performance to commercial LLMs.

It can evaluate the scores of all candidate entities at one-shot, instead of re-sort top-K candidates obtained from other tools.

Table 2: The KG completion results on FB15k-237 and WN18RR. The best and second-best results are **boldfaced** and underlined, respectively. ↑: higher is better; ↓: lower is better. -: unavailable entry.

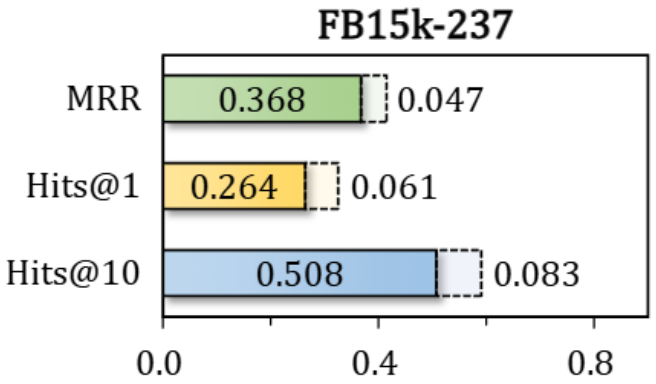
Model	FB15k-237				WN18RR			
	MRR↑	Hits@1↑	Hits@3↑	Hits@10↑	MRR↑	Hits@1↑	Hits@3↑	Hits@10↑
TransE [23]	.310	.218	.345	.495	.232	.061	.366	.522
RotatE [26]	.338	.241	.375	.533	.476	.428	.492	.571
TuckER [56]	.358	.266	.394	.544	.470	.443	.526	.526
CompGCN [28]	.355	.264	.390	.535	.479	.443	.494	.546
DAN [15]	.354	.261	-	.544	.458	.422	-	.537
CoKE [29]	.364	.272	.400	.549	.484	.450	.496	.553
KG-BERT [14]	-	-	-	.420	.216	.041	.302	.524
StAR [38]	.296	.205	.322	.482	.401	.243	.491	.709
KGLM [40]	.289	.200	.314	.468	.467	.330	.538	.741
FTL-LM [39]	.348	.253	.386	.521	.543	.452	.637	**.773**
DET [30]	.376	.281	-	.560	.507	.465	-	.585
KG-Llama-7b [42]	-	-	-	-	-	.242	-	-
GPT 3.5 Turbo [41]	-	.267	-	-	-	.212	-	-
KICGPT [10]	.412	**.327**	.448	.554	.549	.474	**.585**	.641
MKGL	**.415**	.325	**.454**	**.591**	**.552**	**.500**	.577	.656

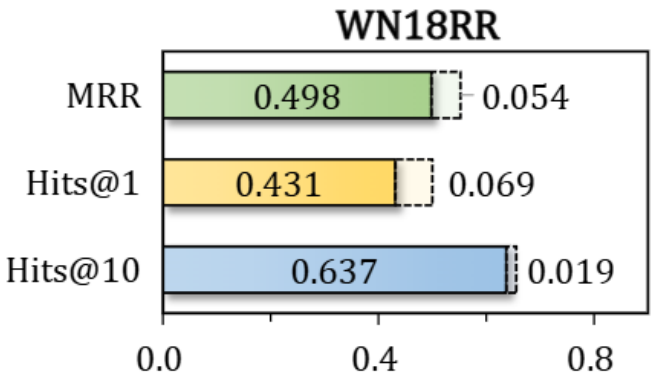
Experiments



Our method also has better performance, and higher efficiency, as well as higher speed, in comparison with fine-tuning an LLM with random-initialized KG token embeddings.

Experiments





Head entity	Relation	Tail entity	
<Primetime Emmy Award>	<nominated for>	<Temple Grandin (film)>	√
		<Backstairs at the White House>	√
		<John Adams>	√
		<Gulliver's Travels (miniseries)>	√
	<award winner>	<Karl Malden>	√
	<ceremony>	<60th Primetime Emmy Awards>	√
<play, use or move>	<verb group>	<play, put (a card or piece) into play during a game>	√
		<utilize, make work or employ for a particular purpose >	√
		<play, participate in games or sport>	
		<play, employ in a game or in a specific position>	
		<take on, contend against an opponent in a sport, game, or battle>	√

Furthermore, it can generate KGL sentences, with only small performance loss.

Conclusion

- In this paper, we propose **MKGL** to instruct the LLM in the language of KGs.
- KGL has its own vocabulary and token embeddings, enables the LLM to evaluate the KGL candidates at one shot.
- With the power of LLM, our method significantly outperforms conventional methods, and even the commercial LLM-based methods.

Thanks for your attention!

- **Code and datasets are available at github.com/zjukg/MKGL**

- This work is funded by National Natural Science Foundation of China (NSFC62306276/NSFCU23B2055/NSFCU19B2027/NSFC6240072039),
- Zhejiang Provincial Natural Science Foundation of China (No. LQ23F020017),
- Yongjiang Talent Introduction Programme (2022A-238-G),
- Fundamental Research Funds for the Central Universities (226-2023-00138),
- and supported by AntGroup