Knowledge-Grounded Pre-Training for Data-to-Text Generation

The goal is to solve the task of Natural language generation (NLG)

Context of study

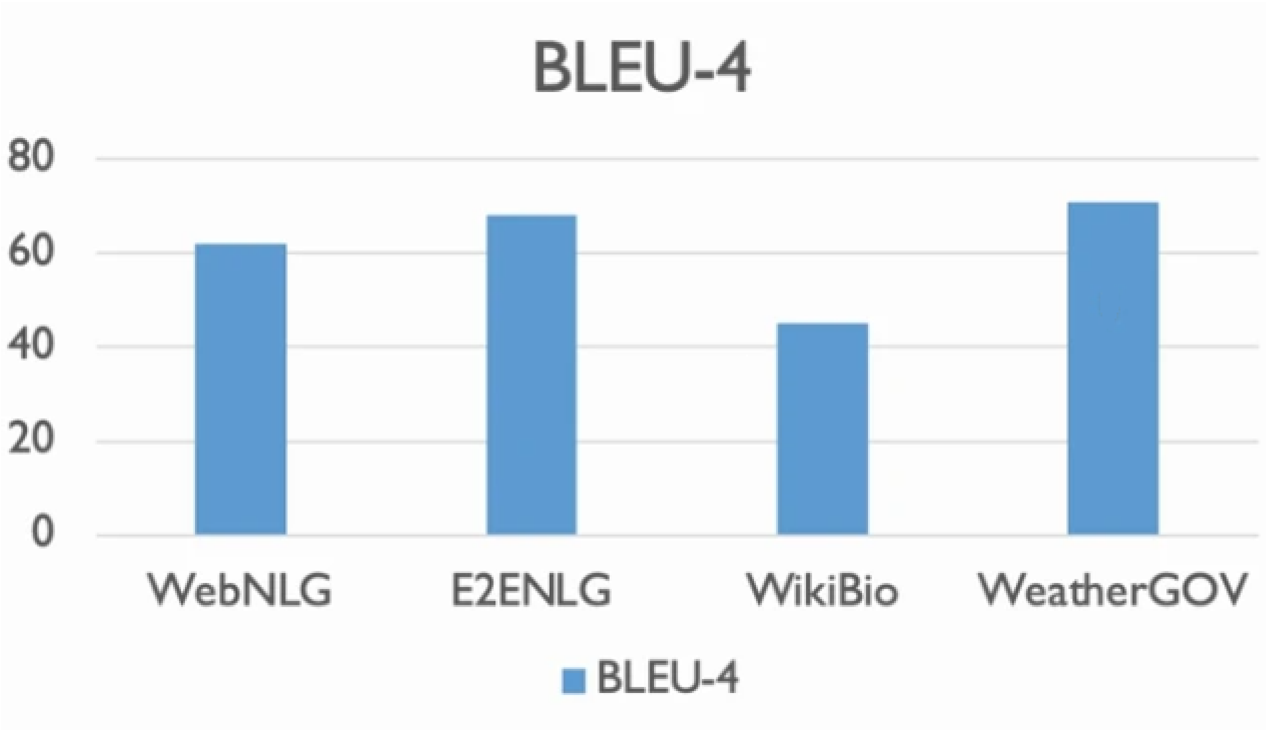
Data-to-Text Generation: Generating text from structured data/knowledge

Diverse Forms of Structured Knowledge

- E2ENLG: generate response from dialog act
- WebNLG: generate description from RDF triples
- WikiBio: generate biography from info-box
- TOTTO: generate sentences from multi-row table

Metric: *Bilingual Evaluation Understudy or BLUE Papineni et al.from IBM Watson Research Center.*

BLEU-4 for NLG



Limitations

1. no unified model to solve all these tasks (one model per task according to SOTA)
2. different models have weak generalization (out of vocabulary entities, unseen relations, etc.)
3. rely on large amount of annotation, not suitable for few-shot settings

Can we use some pre-trained model such as GPT-2, BART, T5? No.

- GPT-2 is not encoder-decoder architecture
- BART, T5's encoders are not designed to encode structured knowledge

Proposal

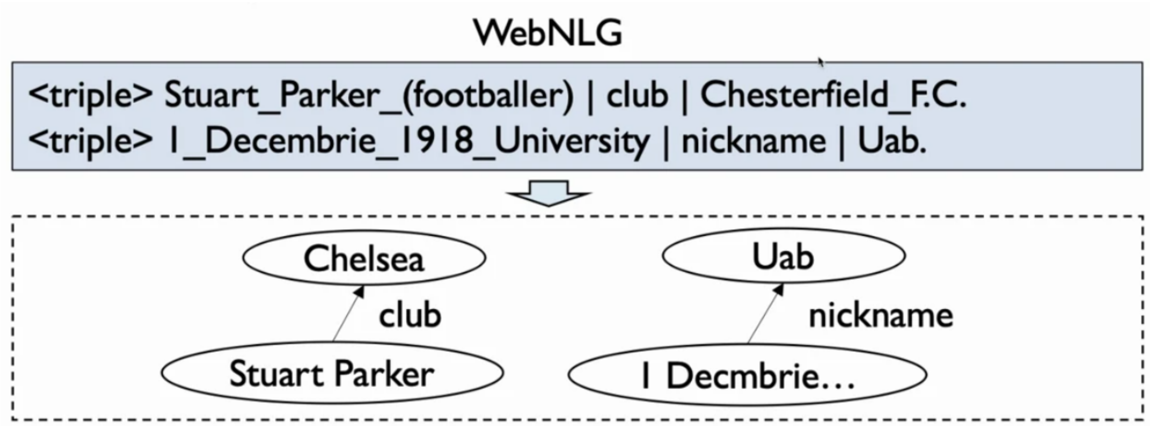
Pre-trained paradigm: **Knowledge-Grounded Language Pre-trained**

Contributions

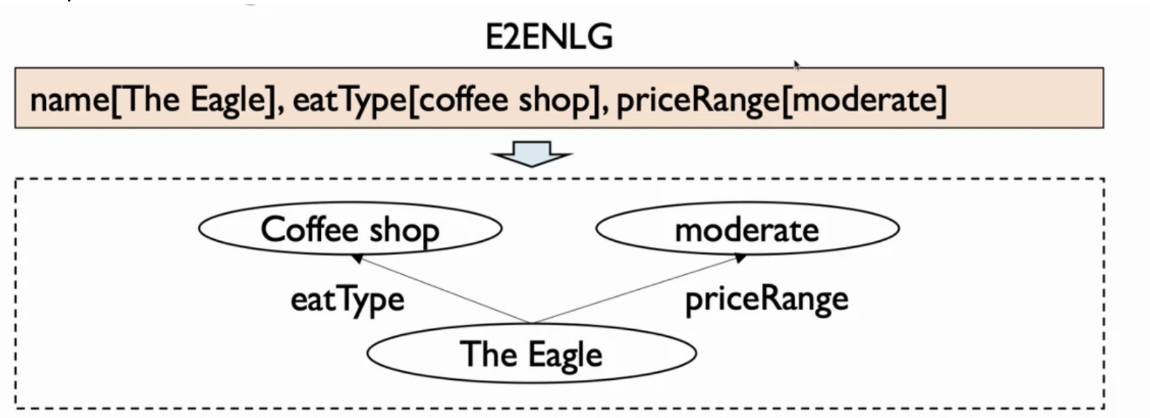
1. Universalize the knowledge representation:

    - *unify knowledge triples, attribute-value pairs, infobox, tables into unified graph representation*
      node --> entity/value/topic
      edge --> relation/attribute/table header

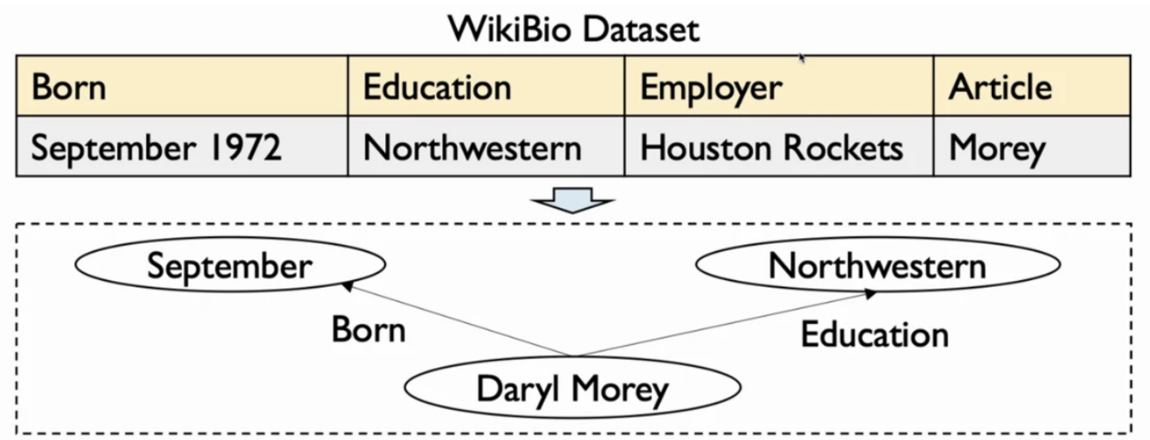
      Example WebNLG:

      Example E2ENLG:

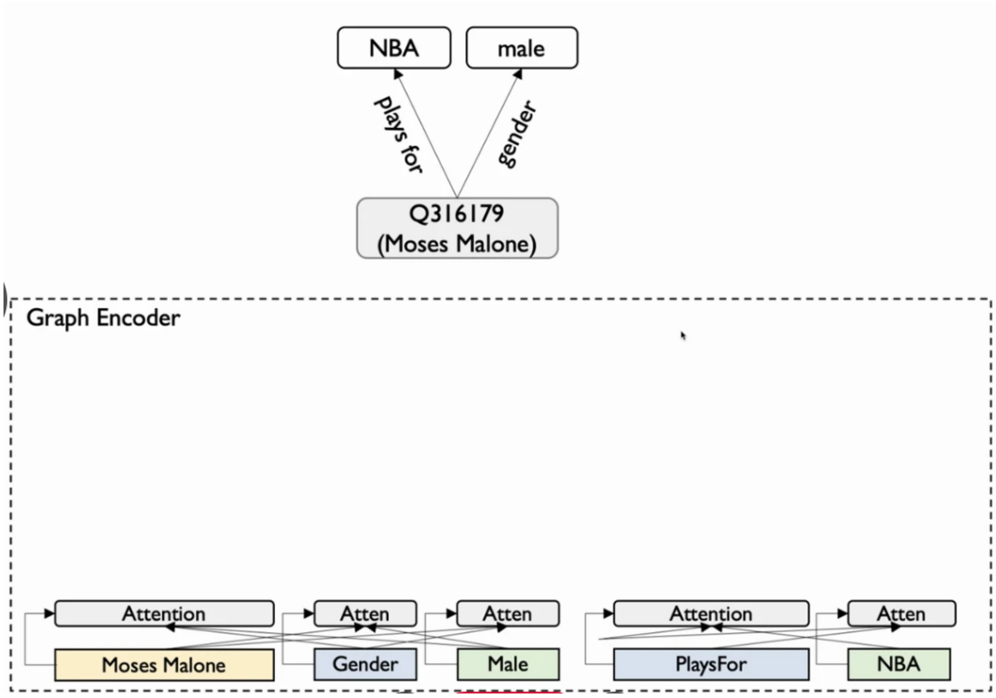
      Example WikiBio:

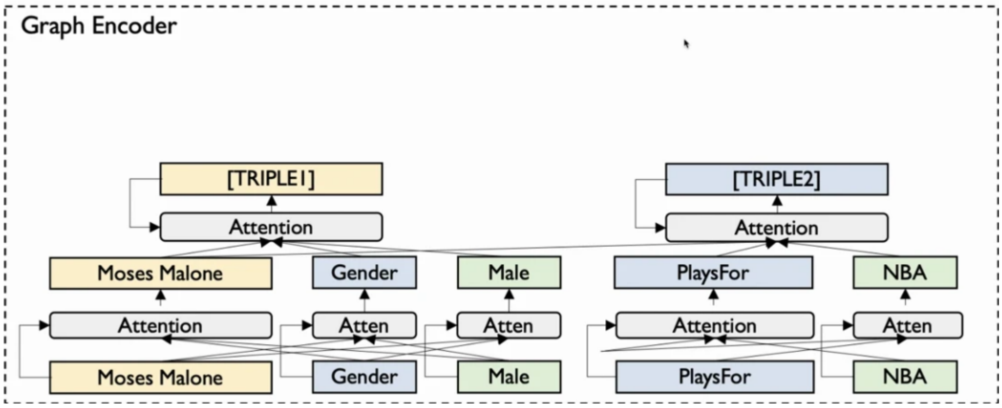
    - *data-2-text generation tasks modeled as graph-to-text problem*
      graph to text model using GNN-based Encoder + Transformer Decoder

- GNN-based Encoder

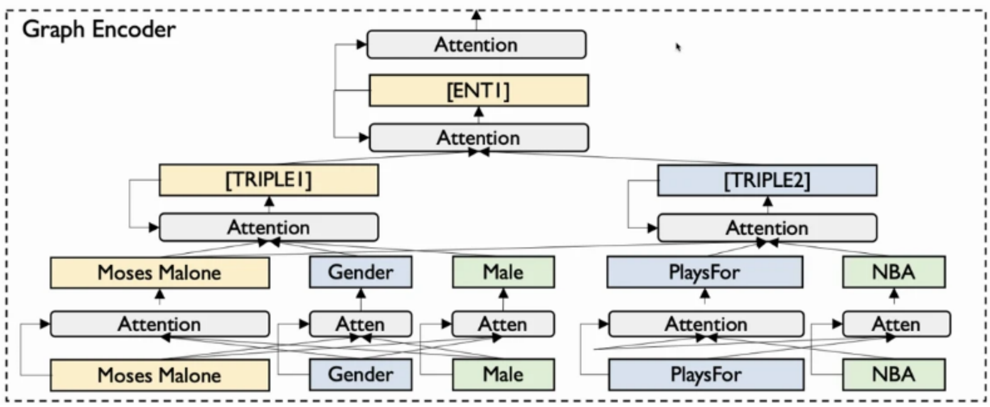
    1. Map the lexicons of all the head tailed entities and relations into their vector representation



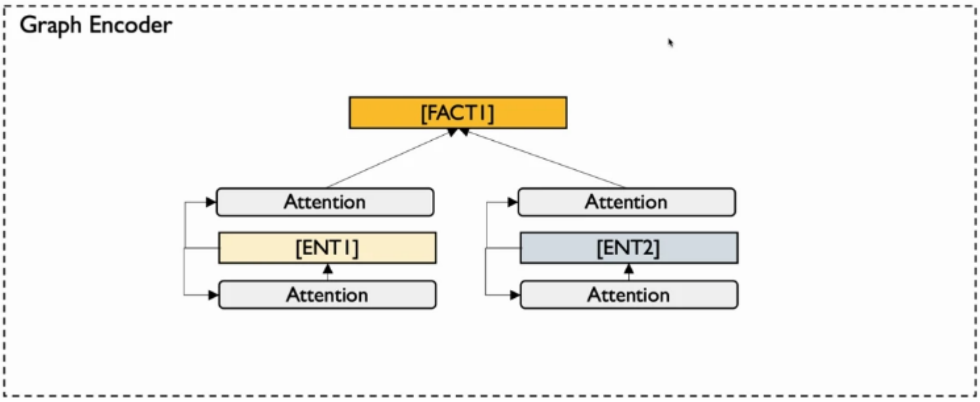
    2. head-tail entities and the relations are aggregated into a triple representation



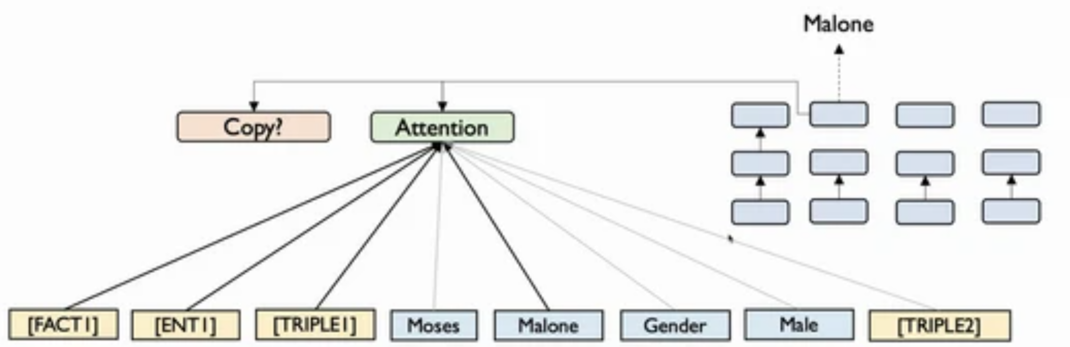
    3. different triples sharing the same head entities are aggregated into a entity representation



4. different entities are aggregated into factor representation



- Transformer Decoder
  After encoder we have sequences of representations over all the nodes and edges. Transformer decoder with copy mechanism to generate the final text token by token



2. Pretrain a large model on a knowledge-grounded text corpus

- Construct Pseudo Graph-to-Text Pairs
  - WikiData (Knowledge Graph)
  - Wikipedia (text)

- WikiData (Hyperlink<-->Wikipedia)

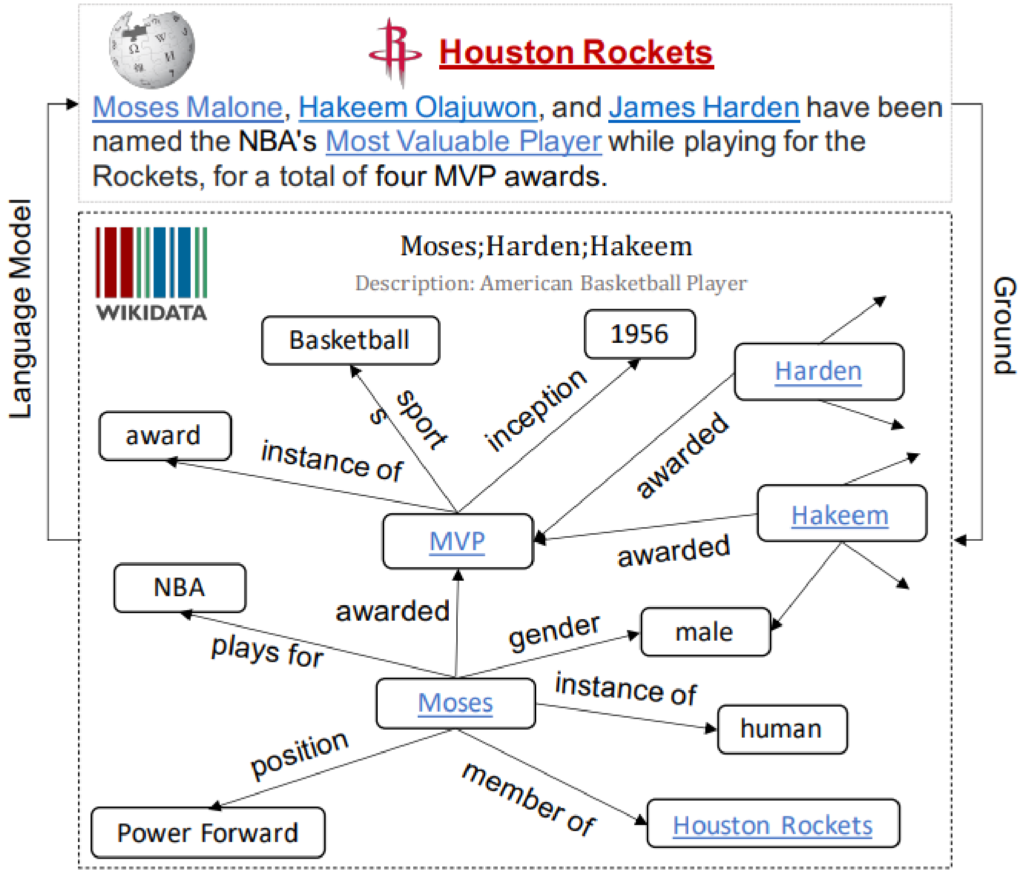


Figure 1: An example from the constructed KGTEXT, which pairs a hyperlinked sentence from Wikipedia with a knowledge subgraph from WikiData.

- Data selection
  Some sentences could be not relevant to the case.
  Lexical Overlap

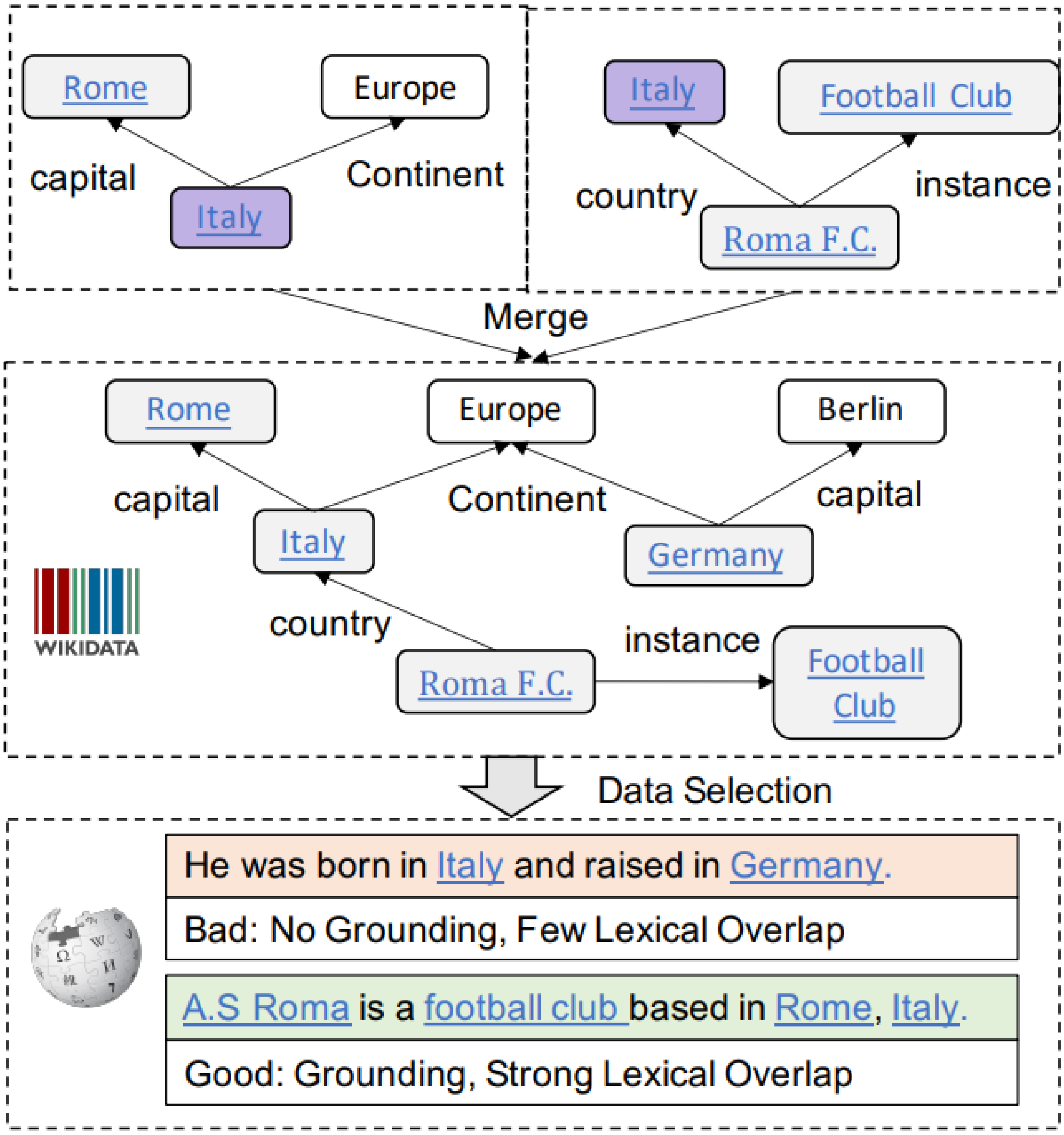


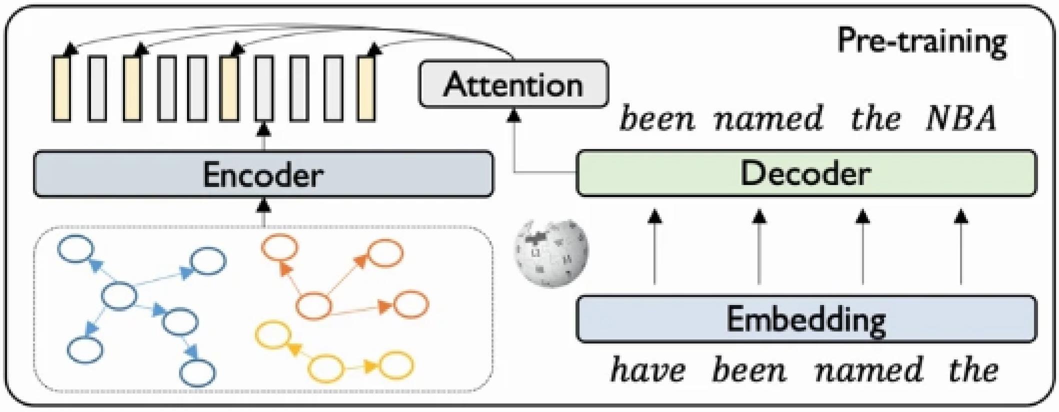
Figure 2: Data denoising procedure for the KGText.

#Sent	Length	#Ent	#Pred	#Triple	#Ent/Sent
7M	20.2	1.8M	1210	16M	3.0

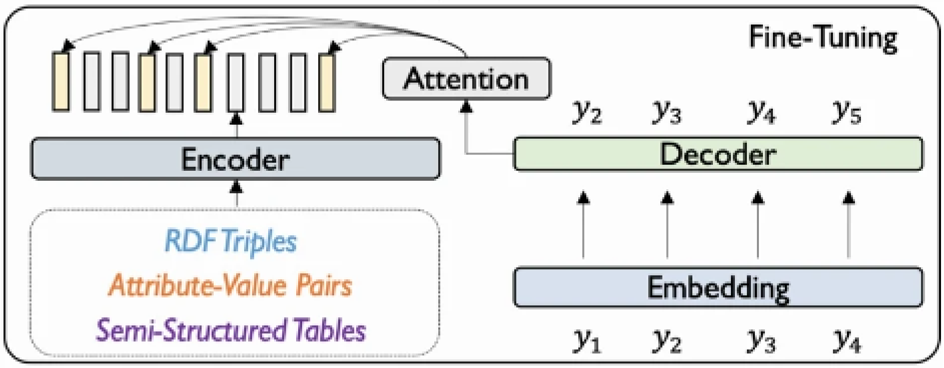
Table 1: Statistics of collected KGText dataset

3. Fine-tune on downstream data-to-text task with only a few examples

- Pre-training the model on the KGText dataset



- fine-tune on the downstream task to solve individual problems



The KGLP model can achieve nearly the same performance with as few as 100 samples.

Results

Dataset

- E2ENLG (Dialog Act)
- WebNLG (RDF Triple)
- WikiBio (Table)

Setting

- Fully-Supervised

Model	BLEU	METEOR	ROUGE
Seq2Seq†	54.0	37.0	64.0
Seq2Seq+Delex†	56.0	39.0	67.0
Seq2Seq+Copy†	61.0	42.0	71.0
GCN	60.80	42.76	71.13
KGPT-Graph w/o Pre	62.30	44.33	73.00
KGPT-Seq w/o Pre	61.79	44.39	72.97
KGPT-Graph w/ Pre	63.84	46.10	74.04
KGPT-Seq w/ Pre	**64.11**	**46.30**	**74.57**

Table 4: Experimental results on WebNLG's test set, w/ Pre refers to the model with pre-training, otherwise it refers to the model training from scratch. † results are copied from Shimorina and Gardent (2018).

Two types of encoder Graph-encoder and Sequential-encoder, both encoding schemes with a copy mechanism without copy loss.

- Few-Shot

Model	0.5%	1%	5%	10%
Seq2Seq	1.0	2.4	5.2	12.8
Seq2Seq+Delex	4.6	7.6	15.8	23.1
KGPT-Graph w/o Pre	0.6	2.1	5.9	14.4
KGPT-Seq w/o Pre	0.2	1.7	5.1	13.7
Template-GPT-2	8.5	12.1	35.3	41.6
KGPT-Graph w/ Pre	**22.3**	**25.6**	**41.2**	**47.9**
KGPT-Seq w/ Pre	21.1	24.7	40.2	46.5

Table 7: Few-shot results on WebNLG's test set.

- Zero-shot

Model	BLEU	METEOR	ROUGE
All Baselines	0	0	1.2
Template-GPT-2	0.3	0.5	3.4
KGPT-Graph w/ Pre	13.66	19.17	30.22
KGPT-Seq w/ Pre	13.86	20.15	30.23

Table 11: Zero-shot results on WebNLG's test set.

Conclusion

KGPT performs better than GPT-2 based models

- fewer compute resources for pre-training.
- pretraining boost few-shots performance task
- strong generalization to understand unseen knowledge inputs