ActionPower

Non – Essential is NEcessary: *Order – agnostic Multi – hop Question Generation*

Kyungho kim, Seongmin Park, Junseo Lee, Jihwa Lee

1. Background
2. Framework
3. Experiment
4. Result
5. Analysis
6. Conclusion

$NENE$
$Order-agnostic\ Multi-hop$
$Question\ Generation$

Background - QG

- Question Generation
- Generating Question Q based on the given document D and answer A
- By utilizing Language Model (LM) or LLM (Large LM)

$$Q = f_{qg}(a, \mathbb{D})$$

Background – Multihop QG

- Multihop Question Generation
- Answering complex questions by finding scattering clues and reasoning across the documents.
- Usually needs multiple documents (text segments) $\mathbb{D}$

$$Q = f_{qg}(a, \mathbb{D})$$

Background – Multihop QG

- Usually consists of 2-step : Ranker & Generator

- Ranker
  - To find only answer-relevant documents from given documents
  - Treating answer-irrelevant documents as non-essential and remove them as impurities.

- Generator
  - To generate the question from selected documents and given answer

Background – Multihop QG's problems

- The order of given documents may affect the performance in both ranker and generator
- The ranker cannot completely removes impurities, which can lead to a decrease in model performance by training-inference discrepancy
- Complexly designed framework

Framework – NENE

- To overcome those problems, we propose an auxiliary task, called ***order-agnostic***, which leverages non-essential data in the training phase to create a robust model and extract the consistent embeddings in real-world inference environments

- Single language model (LM) performs both ranker and generator through a prompt-based approach without applying additional external modules.

- The order-agnostic Encoder is designed for Ranker and Generator

Framework – NENE

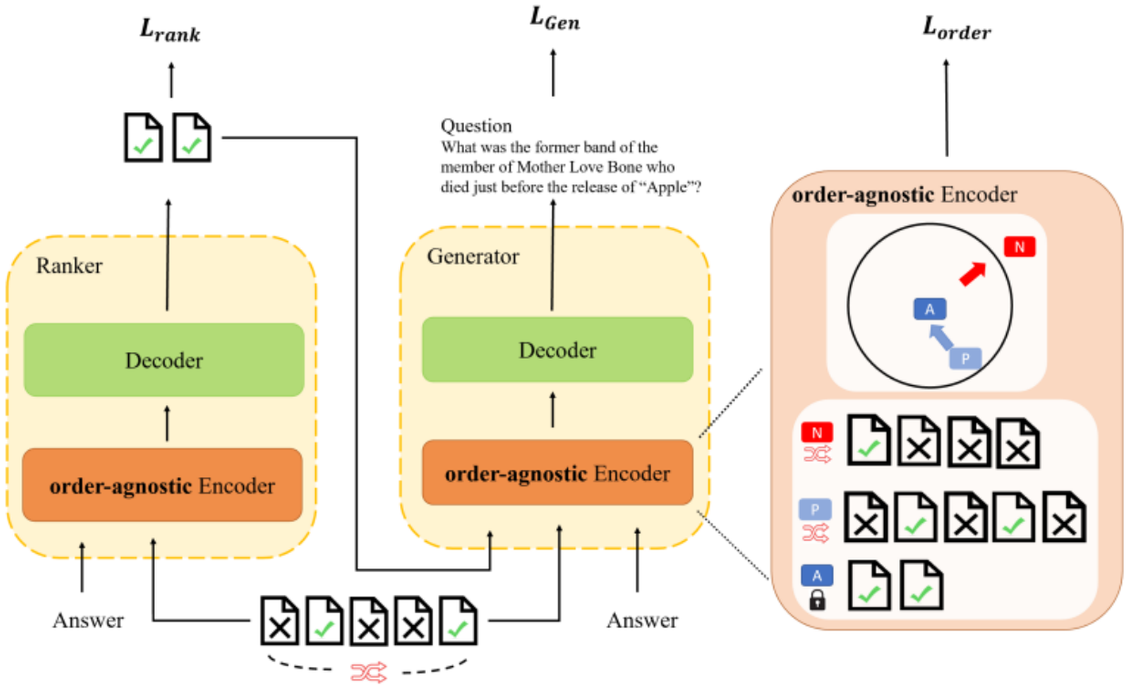


Figure 1: Overall structure of NENE. A, P, and N respectively denote the anchor, positive, and negative.

Framework - NENE

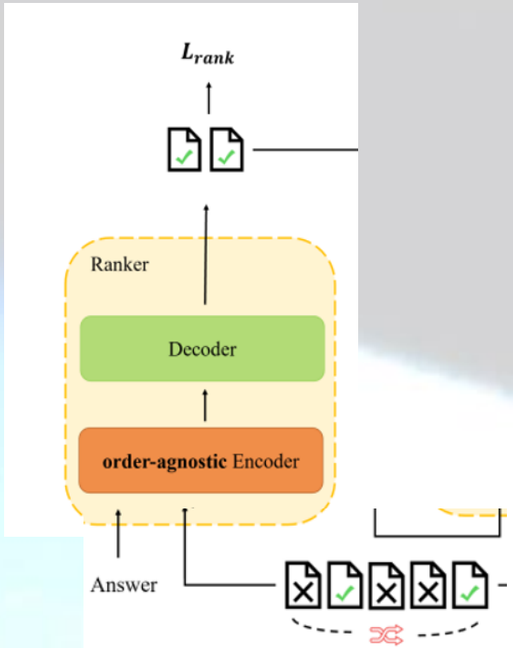
- 3 Loss
  - Subtask : Ranker
  - Auxiliary Task: Order-agnostic Encoder
  - Main Task: Generator

- Final loss

$$L_{final} = L_{gen} + \alpha L_{rank} + \beta L_{order} \qquad (7)$$

Framework - Ranker

- Subtask : Ranker
  - Select answer-relevant documents
  - Oder-agnostic Encoder is utilized
    - Trained by Auxiliary Task loss
  - The order of input documents are changed in every training step



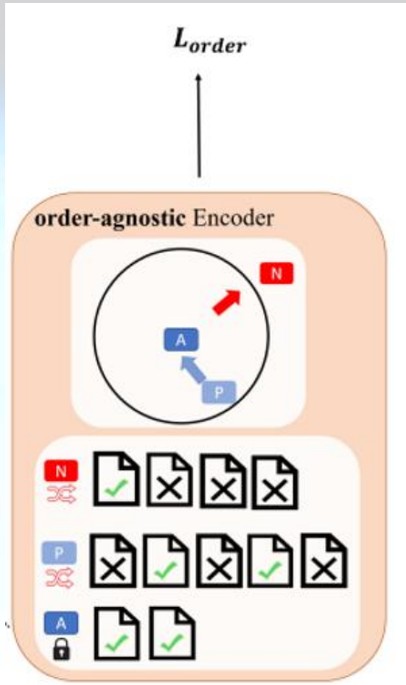
$$S = f_{rank}(a, \mathbb{D})$$

$$L_{rank} = -\Sigma s \log(\hat{s})$$

Framework – Order-agnostic Encoder

- Auxiliary Task: Order-agnostic Encoder
  - Designed for generating same encoder embedding when all answer-relevant documents are included in the input.
  - Regardless of the order and number of unrelated documents (non-essential)

- Triplet margin loss is utilized for training loss.



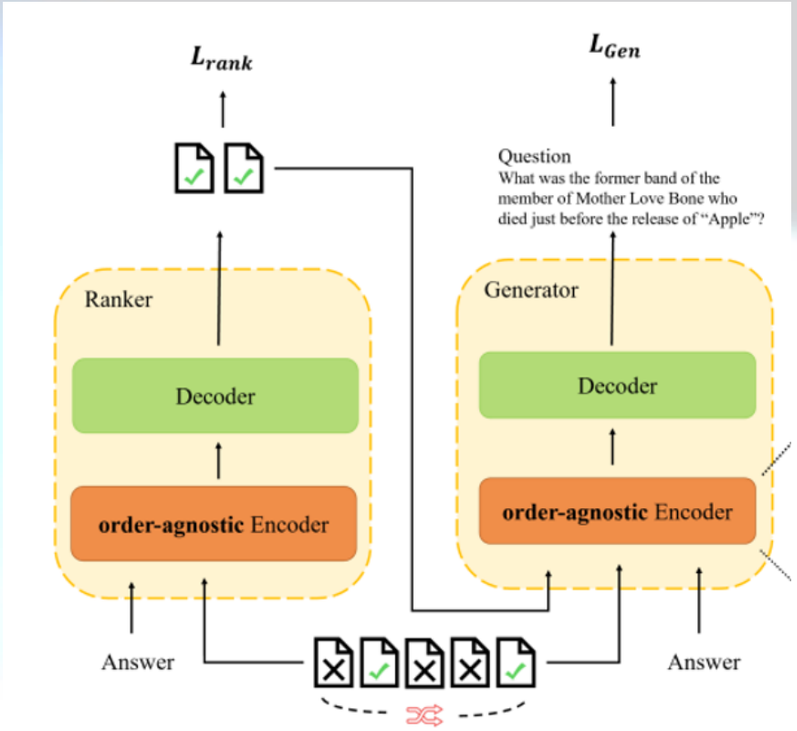
$$L_{order} = \max(0, d(A, P) - d(A, N) + \gamma) \quad (4)$$

Framework - Generator

- Main Task: Generator
  - Generating final question with selected documents

$$Q = f_{gen}(a, S, \mathbb{D})$$

$$L_{gen} = -\Sigma q \log(\hat{q})$$



Framework - NENE

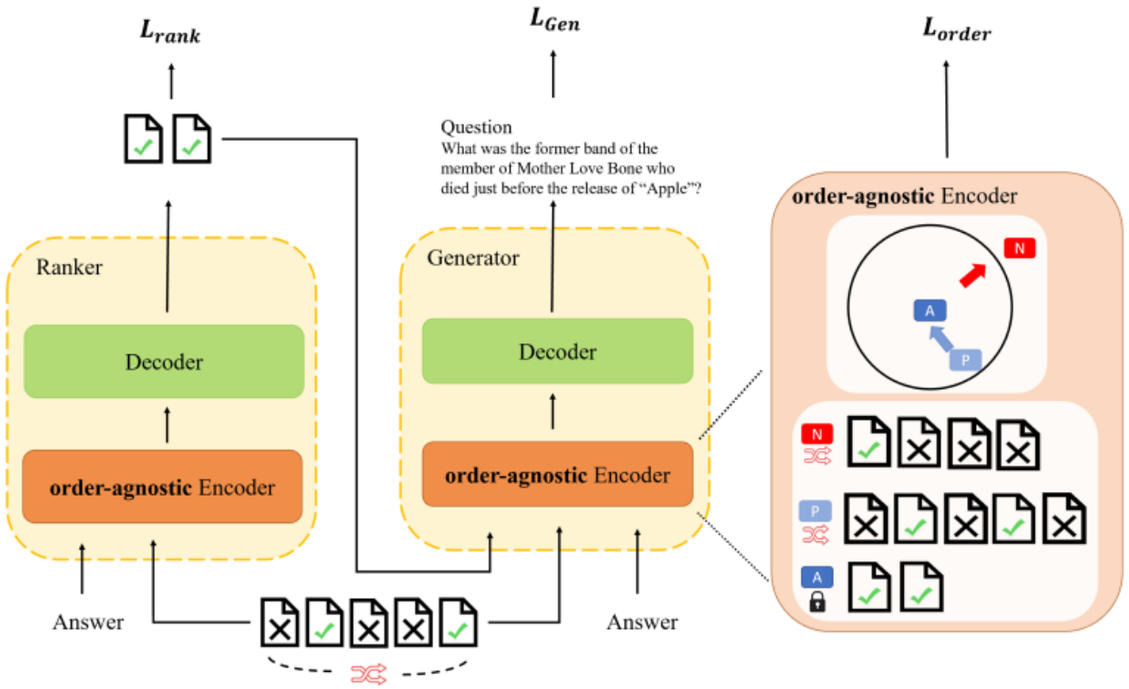


Figure 1: Overall structure of NENE. A, P, and N respectively denote the anchor, positive, and negative.

Experiment - Settings

- Dataset : HotpotQA
- Backbone LM : LMQG (based on T5-base)

- Margin Value : $\gamma = 1.0$
- Loss balance factor : $\alpha = 1.0$ & $\beta = 0.1$

- Epoch: 15
- Batch size: 16
- Learning rate : 1e-4
- Optimizer: AdamW

Experiment - Result

- With oracle answer-relevant documents, NENE shows best performance
- Even if NENE is BASE size model , it shows comparable performance compared with LARGE model.

Models	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L
with Golden Supporting Facts Sentences						
ASs2s-a (Kim et al., 2018)	37.67	23.79	17.21	12.59	17.45	33.21
SemQG (Zhang and Bansal, 2019)	39.92	26.73	18.73	14.71	19.29	35.63
F+R+A (Xie et al., 2020)	37.97	-	-	15.41	19.61	35.12
SGGDQ (DP) (Pan et al., 2020)	40.55	27.21	20.13	15.53	20.15	36.94
ADDQG (Wang et al., 2020)	44.34	31.32	22.68	17.54	20.56	38.09
QA4QG (LARGE) (Su et al., 2022)	49.55	37.91	30.79	25.70	27.44	46.48
NENE (BASE)	**49.62**	**40.1**	**32.48**	**26.22**	**37.22**	**49.14**
Full Document Context (w/o Golden Supporting Facts Sentences)						
MulQG (Su et al., 2020b)	40.15	26.71	19.73	15.2	20.51	35.3
GATE$_{NLL+CT}$ (*Sachanet al., 2020*)	-	-	-	20.02	22.40	39.49
LowResourceQG (Yu et al., 2020)	-	-	-	19.07	19.16	39.41
QA4QG (BASE) (Su et al., 2022)	43.72	31.54	24.47	19.68	24.55	40.44
QA4QG (LARGE) (Su et al., 2022)	**46.45**	**33.83**	**26.35**	**21.21**	25.53	42.44
NENE (BASE)	44.40	33.46	25.14	19.01	**32.93**	**42.61**

Analysis - Ranker

- Utilizing with non-essential can improve performance
- Utilizing Order-agnostic encoder shows best performance.

Input	BLEU	METEOR	ROUGE-L
Ranker	33.47	25.21	32.55
+ non-essential	43.29	31.90	41.80
+ order-agnostic	**44.40**	**19.01**	**42.61**

Table 2: Ablation study of generator.

Analysis - Generator

- Anchor : only answer-relevant documents (essential)
- Positive : all essential documents + non-essential
- Easy Negative : non-essential only
- Hard Negative : part of essential document + non-essential

Extractor	F1	EM
Anchor	20.59	21.47
+ Positive	72.40	41.18
+ Easy Negative	74.43	**44.25**
+ Hard Negative	**74.56**	**44.25**

Table 3: Ablation study of ranker.

Conclusion

- We propose a novel framework, NENE, that utilizes a single language model as both the ranker and the generator using a prompt without any additional modules
- We design the model to learn the dependency between documents in an implicit way and tackle the training-inference gap
- We introduce an auxiliary task called order-agnostic encoder, which ensures that the LM encoder generates consistent embeddings regardless of the order and amount of answer-irrelevant documents

ActionPower

Thank you for listening