A Practical Toolkit for Multilingual Question and Answer Generation

Asahi Ushio, *Fernando Alva-Manchego and Jose Camacho-Collados*

Computer Science & Informatics, Cardiff University, Cardiff NLP

ACL 2023 System Demonstration

Question & Answer Generation

Paragraph

Dante Gabriel Rossetti, was an English poet, painter, and member of the Rossetti family. He founded the Pre-Raphaelite Brotherhood in 1848 with William Holman Hunt and John Everett Millais. Rossetti was later to be the main inspiration for a second generation of artists and writers influenced by the movement, most notably William Morris and Edward Burne-Jones.

Question & Answer

Q: What was Dante Gabriel Rossetti's career?
A: poet, painter, and member of the Rossetti family.

Q: What group did Dante Gabriel Rossetti found in 1848?
A: Pre-Raphaelite Brotherhood

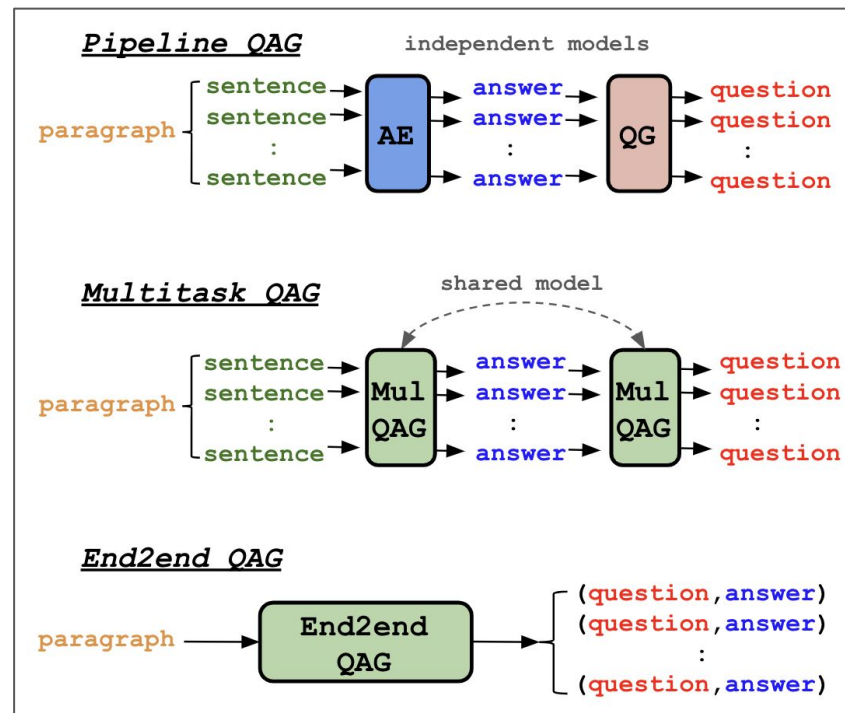Q: Along with Edward Burne-Jones, who was influenced by the Pre-Raphaelite Brotherhood?
A: William Morris

A Practical Toolkit for Multilingual Question and Answer Generation
Asahi Ushio, Fernando Alva-Manchego, Jose Camacho-Collados

2

Three QAG Approaches

- ***Pipeline QAG***
  - QG + AE (Separate Model)
  - Prediction per sentence

- ***Multitask QAG***
  - QG + AE (Shared Model)
  - Prediction per sentence

- ***End2end QAG***
  - QAG (Single Model)
  - Prediction per paragraph

*QG: Question Generation

*AE: Answer Extraction

A Practical Toolkit for Multilingual Question and Answer Generation
Asahi Ushio, Fernando Alva-Manchego, Jose Camacho-Collados

Three QAG Approaches

- ***Pipeline QAG***
  - QG + AE (Separate Model)
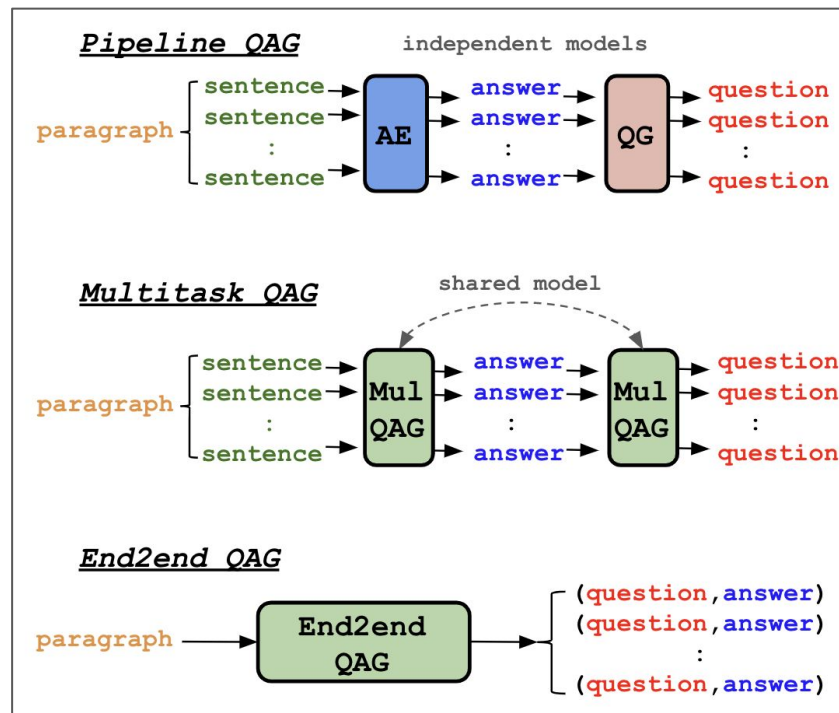 - Prediction per sentence
- ***Multitask QAG***
  - QG + AE (Shared Model)
  - Prediction per sentence
- ***End2end QAG***
  - QAG (Single Model)
  - Prediction per paragraph

- Slow Inference
- More Storage
- More QA Pairs
- Stable across languages

*QG: Question Generation

*AE: Answer Extraction

A Practical Toolkit for Multilingual Question and Answer Generation
Asahi Ushio, Fernando Alva-Manchego, Jose Camacho-Collados

4

Three QAG Approaches

- *Pipeline QAG*
  - QG + AE (Separate Model)
  - Prediction per sentence

  - Slow Inference
  - More Storage
  - More QA Pairs
  - Stable across languages

- *Multitask QAG*
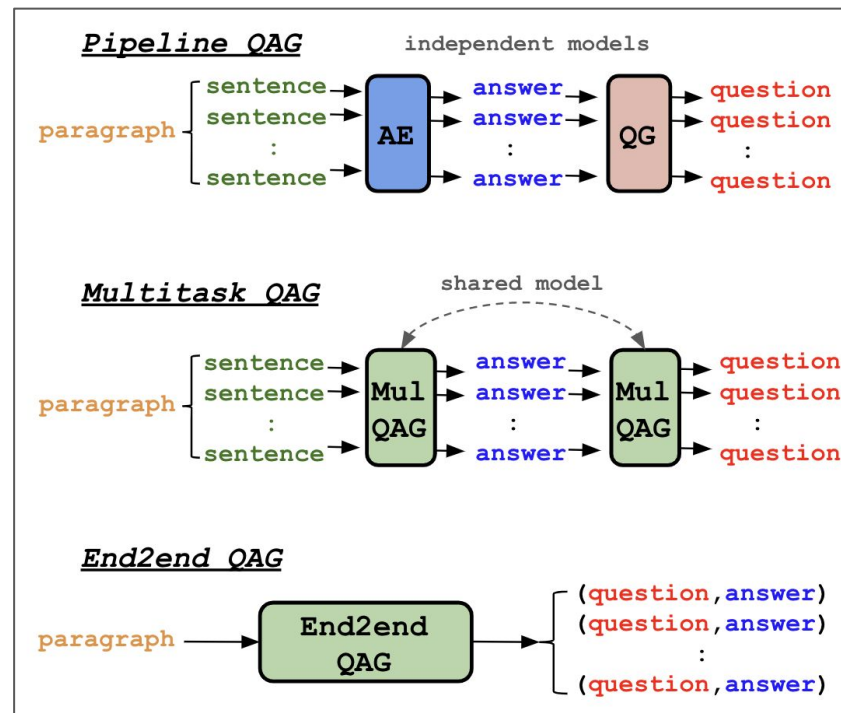 - QG + AE (Shared Model)
  - Prediction per sentence

  - Slow Inference
  - Low Storage
 - More QA Pairs
  - Not the best in some languages

- *End2end QAG*
  - QAG (Single Model)
  - Prediction per paragraph

*QG: Question Generation

*AE: Answer Extraction

A Practical Toolkit for Multilingual Question and Answer Generation
Asahi Ushio, Fernando Alva-Manchego, Jose Camacho-Collados

5

Three QAG Approaches

- ***Pipeline QAG***
  - ○ QG + AE (Separate Model)
  - ○ Prediction per sentence

  - Slow Inference
  - More Storage
  - More QA Pairs
  - Stable across languages

- ***Multitask QAG***
  - ○ QG + AE (Shared Model)
  - ○ Prediction per sentence

  - Slow Inference
  - Low Storage
  - More QA Pairs
  - Not the best in some languages

- ***End2end QAG***
  - ○ QAG (Single Model)
  - ○ Prediction per paragraph

  - Fast Inference
  - Low Storage
  - Less QA Pairs
  - Failure in some languages

*QG: Question Generation

*AE: Answer Extraction
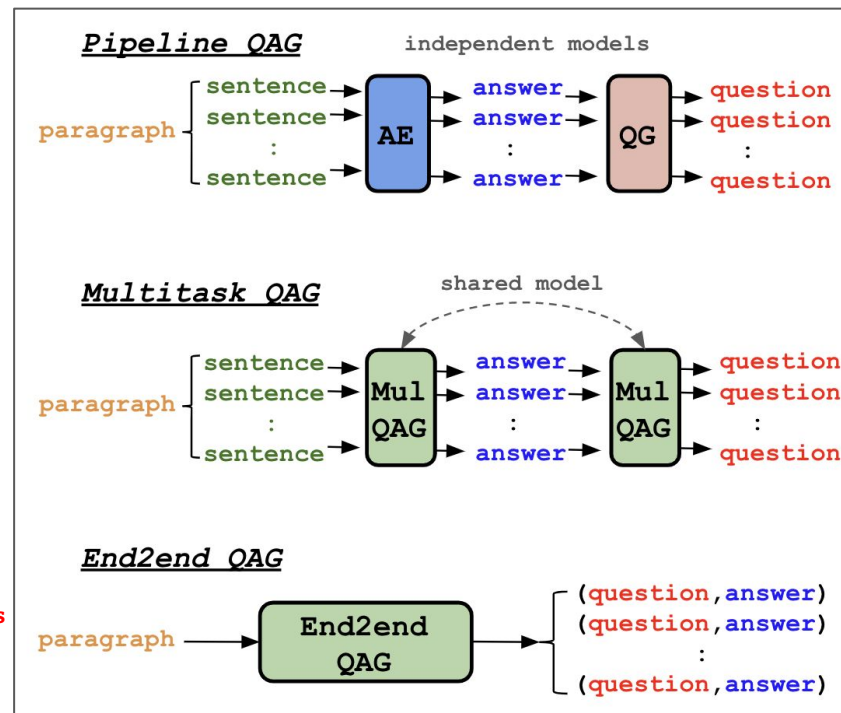
lmqg: A Python Library for QAG

- What I can do with _lmqg_? 🤔
  - QA Generation with fine-tuned models in 8 languages En/Ja/Es/Fr/De/It/Ru/Ko.
    - Check available models at MODEL CARD or https://huggingface.co/lmqg.
  - QAG model training/evaluation.
    - Check available datasets at https://huggingface.co/lmqg.
  - Rest API for QA Generation.

- Link → https://github.com/asahi417/lm-question-generation

Examples of *lmqg*

```python
from lmqg import GridSearcher

# instantiate AE trainer
trainer_ae = GridSearcher(
    dataset_path="lmqg/qg_squad",
    input_types="paragraph_sentence",
    output_types="answer",
    model="t5-large")

# train AE model
trainer_ae.train()

# instantiate QG trainer
trainer_qg = GridSearcher(
    dataset_path="lmqg/qg_squad",
    input_types="paragraph_answer",
    output_types="question",
    model="t5-large")

# train QG model
trainer_qg.train()
```

(i) Model Training

```python
from lmqg import QAAlignedF1Score

# gold reference and generation
ref = [
"question: What makes X?, answer: Y",
"question: Who made X?, answer: Y"]
pred = [
"question: What makes X?, answer: Y",
"question: Who build X?, answer: Y",
"question: When X occurs?, answer: Y"]

# compute QAAligned BS
scorer = QAAlignedF1Score(
    base_metric="bertscore")
scorer.get_score(pred, ref)

# compute QAAligned MS
scorer = QAAlignedF1Score(
    base_metric="moverscore"
)
scorer.get_score(pred, ref)
```

(ii) Model Evaluation

```python
from lmqg import TransformersQG

# instantiate model
model = TransformersQG(
    model="lmqg/t5-base-squad-qg",
    model_ae="lmqg/t5-base-squad-ae"
)

# input paragraph
x = """William Turner was an English
painter who specialised in watercolour
landscapes. One of his best known
pictures is a view of the city of
Oxford from Hinksey Hill."""

# generation
model.generate_qa(x)
[
  (
    "Who was an English painter
    specialised in watercolour
    landscapes?",
    "William Turner"
  ),
  (
    "Where is William Turner's
    view of Oxford?",
    "Hinksey Hill."
  )
]
```

(iii) QA Generation

A Practical Toolkit for Multilingual Question and Answer Generation
Asahi Ushio, Fernando Alva-Manchego, Jose Camacho-Collados

8

Rest API for QA Generation with _lmqg_

Build

- Build/Run Local (command line):

```
export API_TOKEN={Your Huggingface API Token}
uvicorn app:app --reload --port 8088
uvicorn app:app --host 0.0.0.0 --port 8088
```

- Build/Run Local (docker):

```
docker build -t lmqg/app:latest . --build-arg api_token={Your Huggingface API Token}
docker run -p 8080:8080 lmqg/app:latest
```

API Description

You must pass the huggingface api token via the environmental variable `API_TOKEN`. The main endpoint is `question_generation`, which has following parameters,

Parameter	Description
input_text	input text, a paragraph or a sentence to generate question
language	language
qg_model	question generation model
answer_model	answer extraction model

A Practical Toolkit for Multilingual Question and Answer Generation
Asahi Ushio, Fernando Alva-Manchego, Jose Camacho-Collados

***AutoQG* https://autoqg.net/**

- Multilingual
  - En/Ja/Es/Fr/De/It/Ru/Ko
- Switch QAG Type:
  - Pipeline/End2end/Multitask
- Switch LMs
  - T5/BART/Flan-T5
- QG mode: Specify Answer

Thank you!!