ELOQUENT sensemaking

Conference and Labs of the Evaluation Forum - Information Access Evaluation meets Multilinguality, Multimodality, and Visualization - CLEF 2025

OutstandingOutsiders

Task

Given:

→ **Context: English text** (large, up to 35k tokens)

1. Teacher (other teams):

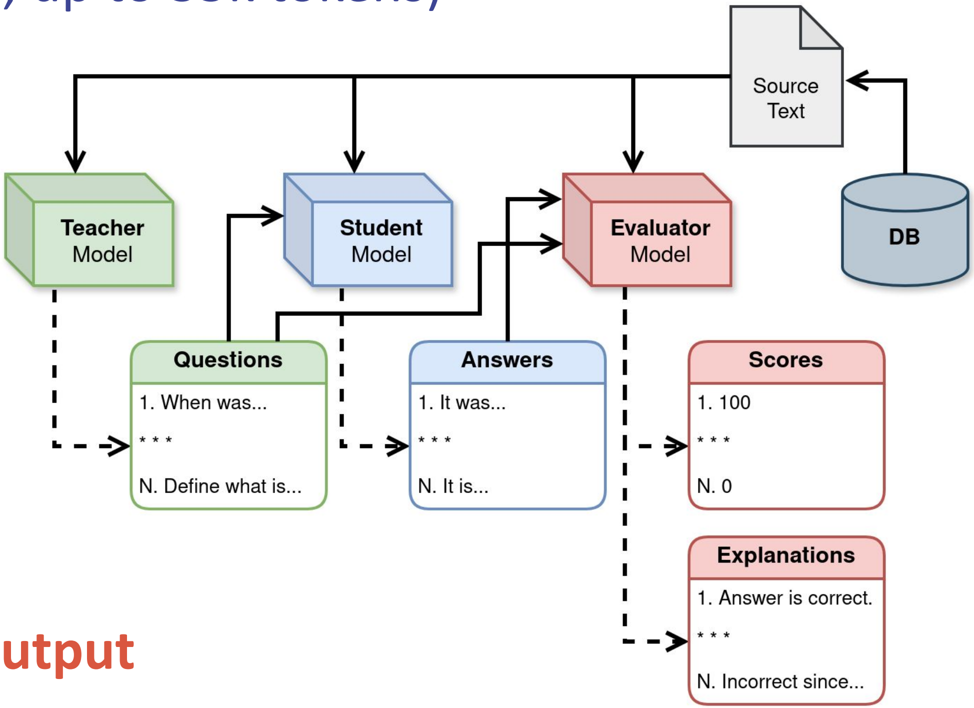
   → **Question output**

2. Student (other teams):

   → **Answer output**

3. **Evaluator:**

   ↦ **Score** (+ Explanation) **output**



Why bother?

*“LLMs store immense amounts of information and easily answer questions about it. **Can they limit their knowledge to information provided in given materials?**”* - task provider.

- Potential help for real teachers and evaluators
- Testing of models knowledge constraints
- Exploring different aspects of source-based QA task

Problems to deal

Too large context

Hard to fine-tune a specific model locally because:

- No available data with a comparable context-question size
- Models with such large context window take a lot of time to finetune
- Manual evaluation of predictions is exhausting

Solution

Use ollama models

- **Various open source models to run inference** (e.g. deepseek, llama and llava, qwen, gemma)
- **Different parameter sizes** available for some models
- Framework **optimized for multiple GPUs** using free video RAM as a buffer for input tokenization besides storing the model weights

Choosing models

Tech limitations:

- **GPU memory available** (eventually used **2** x **NVIDIA A30** (24Gb) or **3** x **NVIDIA RTX A4000** (16Gb) from UFAL cluster)
- **Context window size** (should be extended from the **ollama**'s cropped default)

General limitations:

- **JSON output format** (optional - can be post corrected)

Preliminary experiments

Goal: generate a training data for the Evaluator e.g. **Q**,**A**,**C**-to-**S** which is, in fact, **text-to-int** task

Data: QA dataset with answers including source quotes was parsed to **Question, Answer, Context** variables - then expanded with **wrong** (prev/next string in sequence) **Answers** in 1:1 ration to correct ones

Run: inference of **llama3.3** as Evaluator where input's **Context** is composed out of prev, current (actually relevant), and next items in the data sequence, and output prompted to be a JSON containing a **Score**

Preliminary experiments

Results:

- Some predictions with incorrect JSON formatting (e.g. missed punctuation) - can be fixed by regex
- Considering 1:1 ratio of correct-wrong **Answers**, the **Scores** were not so strict in proportions - **many initially right Answers scored zero**

Conclusions:

Decided to forget about creating a training dataset or using old llamas

Applied approach

Models used (**128k context window size**):

Qwen3 - 30b - a3b

MoE type transformer
Thinking/Not-thinking modes

Gemma3 - 27b

Decoder-only type transformer
Sliding Window Attention
Function calling head (structured output)

Prompt engineering

Decisions to make (based on a devset):

- **Define order of the given information** (e.g. task or Inputs first)
- **Describe task definition** ("You are an evaluator…")
- **Give output format limitations** (e.g. JSON with exactly 2 fields)
- **State knowledge boundaries** ("Given the current context…")

System prompt:

```
You are a fair teacher who grades students' answers. Evaluate the quality of
the *Answer* specifically in response to the *Question* considering the
*Context* provided.
Format your entire response as a single JSON object containing 'score' (an
integer between 0 and 100, where 100 is best) and 'explanation' (a string
briefly justifying the score).
```

Changeable prompt:

```
Question: {question}
Answer: {answer}
And given the following context: {text_fragments}
```

Generated output:

```
Potential thinking…
{“score”: {int}, “explanation”: {str}}
```

Applied approach

System prompt:

```
You are a fair teacher who grades students' answers. Evaluate the quality of
the *Answer* specifically in … a 'score' (an integer between 0 and 100,
where 100 is best) and 'explanation' (a string briefly justifying the score).
```

Changeable prompt:

```
Question: What is a result of misplacing punctuation marks in machine
translation?
Answer: What is a result of misplacing punctuation marks in machine
translation is not discussed in the provided text.
And given the following context: Computation Graphs For our example neural
network from Section … book published in year 2020.
```

Generated output:

```
The text talks about … thus the score is 100.
{"score": 100, "explanation": "The given answer is correct, the text does not
mention misplacing punctuation marks in machine translation."}
```

Example

System prompt:

`You are a fair teacher who grades students' answers. Evaluate the quality of the *Answer* specifically in … a 'score' (an integer between 0 and 100, where 100 is best) and 'explanation' (a string briefly justifying the score).`

Changeable prompt:

`Question:` **What was the structure of trade in the Roman Empire?**

`Answer:` **In hospitals, dehydration is commonly treated with infusions.**

`And given the following context:` **CHAPTER OUTLINE 7.1 The Daily Life of a Roman Family 7.2 Slavery in the Roman Empire 7.3 The Roman Economy: Trade, Taxes, and Conquest … Jewish population during the imperial period.**

Generated output:

{“score”: 10, “explanation”: “The answer is entirely unrelated to the question. It is the full text of a chapter on the Roman Empire. There is no attempt to answer the question at all. Therefore the score is the lowest possible.”}

Example

Findings

It’s difficult to pick “the right” model when multiple of the same quality are available - too large and detailed context for a simple eyeball check of the generated outputs

Fine-tuning models is inefficient in comparison to prompt-engineering of the newest models (lack of time and equal in style training datasets)

Thanks for attention