Back

Software Applications
Graded Assignment • 10 min

Due Oct 20, 8:59 AM CEST

Your grade: 80%

Your latest: **80%** • Your highest: **80%** • To pass you need at least 80%. We keep your highest score.

Next item →

1. In the videos, we described using either supervised learning or a prompt-based development process to build a restaurant review sentiment classifier. Which of the following statements about prompt-based development is correct? 1 / 1 point

- (•) Prompt-based development is generally much faster than supervised learning.
- ( ) Prompt-based development requires that you collect hundreds or thousands of labeled examples.
- ( ) Prompt-based development requires that you collect hundreds or thousands of unlabeled examples (meaning reviews without a label B to say if it is positive or negative sentiment).
- ( ) If you want to classify reviews as positive, neutral, or negative (3 possible outputs) there is no way to write a prompt to do so: An LLM can generate only 2 outputs.

Correct
Prompt-based development allows you to take advantage of an LLM's ability to carry out sentiment classification, so you can get up and running very quickly because you don't need to train a model from scratch.

2. What is a token in the context of a large language model (LLM)? 1 / 1 point

- (•) A word or part of a word in either the input prompt or LLM output.
- ( ) A unit of cryptocurrency (like bitcoin or other "crypto tokens") that you can use to pay for LLM services.
- ( ) The part of the LLM output that has primarily symbolic rather than substantive value (as in, "the court issued a token fine", or "the LLM generated a token output").
- ( ) A physical device or digital code to authenticate a user's identity.

Correct
Tokens in the context of LLMs refer to a unit of text. Common words are typically represented by a single token, while uncommon words may be broken into two or more tokens.

3. What are the major steps of the lifecycle of a Generative AI project? 1 / 1 point

- (•) Scope project → Build/improve system → Internal evaluation → Deploy and monitor
- ( ) Scope project → Internal evaluation → Build/improve system → Deploy and monitor
- ( ) Scope project → Internal evaluation → Deploy and monitor → Build/improve system

Correct
This sequence accurately represents the recommended steps in the lifecycle of a Generative AI project. You first scope the project, then build or improve the system, followed by internal evaluation, and finally, deployment and monitoring.

4. You are building a customer service chatbot. Why is it important to monitor the performance of the system after it is deployed? 1 point

- ( ) In case customers say something that causes the chatbot to respond in an unexpected way, monitoring lets you discover problems and fix them.
- (•) Because of the LLM's knowledge cutoff, we must continuously monitor the knowledge cutoff and update its knowledge frequently.
- ( ) Every product should be monitored to track customer satisfaction -- this is good practice for all software.
- ( ) This is false. So long as internal evaluation is done well, further monitoring is not necessary.

Incorrect
While it is true that an LLM's knowledge of the world is frozen at the moment of its training, updating the model constantly by carrying out additional training would be an expensive and inefficient task. The primary concern with a chatbot is that it may behave in unexpected ways, like hallucinating or outputting harmful content. Monitoring the chatbot for issues like this will help you improve the reliability and safety of the system.

5. You are working on using an LLM to summarize research reports. Suppose an average report contains roughly 6,000 words. Approximately how many tokens would it take an LLM to process 6,000 input words? (Assume 1 token = 3/4 words, or equivalently, 1 word \approx 1.333 tokens). 1 / 1 point

- ( ) 4,500 tokens (6000 * 3/4)
- ( ) 6,000 tokens
- (•) 8,000 tokens (about 6000 * 1.333)
- ( ) 14,000 tokens (about 6000 * 1.333 + the original 6000 words)

Correct
A token typically represents a single common word or a part of the word. This means that a word can represent anywhere between 1 or more tokens. Therefore, an LLM typically requires more tokens to process the input number of words.