COSC478

Special Topic: Generative Artificial Intelligence Using Large Language Models

Lecture 4 – Metrics for Evaluating Large Language Models

Wednesday 24 July 2024

Semester 2, 2024

Trevor Nesbit

The need for critical analysis of output from tools like ChatGPT

Should we just accept what ChatGPT tells us without thinking about it?

True story about a student answering an open ended question in a Learn quiz who didn't 'sanity check' what ChatGPT responded with…

"…as an AI language model I cannot… "

Today

Why do we need metrics to evaluate large language models?

What metrics can we use and how do they work?

Discussion

Why do we need to evaluate large language models?

What are metrics used for in general?

Why would we want to use metrics to evaluate large language models?

Why do we need to evaluate large language models?

Understanding Performance:

- Evaluation helps us understand the model's performance across various tasks and domains, which is essential for identifying its strengths and weaknesses.

Safety and Fairness:

- Through evaluation, we can assess the model's safety and fairness, ensuring it doesn't generate harmful or biased content.

Model Improvement:

- Evaluation results provide valuable feedback for model improvement, guiding future research and development efforts.

User Trust:

- Regular evaluation and transparency about a model's capabilities and limitations can help build user trust in the technology.

What are metrics used for in general?

Performance Evaluation:

- Metrics provide a quantifiable measure to assess the performance or effectiveness of processes, systems, or strategies.

Goal Setting and Tracking:

- They help in setting clear, measurable goals and tracking progress towards achieving them, facilitating informed decision-making.

Continuous Improvement:

- By identifying areas of strength and weakness, metrics enable continuous improvement through targeted actions and interventions.

Why would we want to use metrics to evaluate large language models?

Performance Assessment:

- Metrics provide a standardized way to measure how well a model performs on various tasks, ensuring it meets desired benchmarks.

Comparative Analysis:

- They allow for the comparison of different models, helping to identify which models are more effective or efficient.

Improvement Tracking:

- Metrics help track improvements over time, showing how updates and changes impact the model's performance.

Identifying Weaknesses:

- They highlight areas where the model may be lacking, guiding further development and refinement efforts

What metrics can we use to evaluate large language models?

- There are a number of them – seems to be on the increase
- Five quite common/important ones are:
  - Perplexity:
  - BLEU (Bilingual Evaluation Understudy)
  - ROUGE (Recall-Oriented Understudy for Gissing Evaluation)
  - Human Evaluation
  - Zero-shot Evaluation:

- We will look at Perplexity and BLEU today; and Human Evaluation next week

What is Perplexity use for?

- Measures the uncertainty of a language model's predictions

- What does Perplexity need?
- To calculate perplexity, you need a language model (which provides the probability distribution), a test dataset (to provide the sequences of words), and the actual outcomes (to compare with the model's predictions). The lower the perplexity, the better the language model is at predicting the test data.

Using Perplexity

Our subset is “the cat sat on the mat”

Our probability distribution is:

P(“the”) = 0.1

P(“cat” | “the”) = 0.2

P(“sat” | “the cat”) = 0.3

P(“on” | “the cat sat”) = 0.1

P(“the” | “the cat sat on”) = 0.1

P(“mat” | “the cat sat on the”) = 0.1

Using Perplexity

- PP(W) is the perplexity of the sequence of words W
- $P(w_1, w_2, w_3, \ldots w_N)$ is the probability of the sequence of words W
- N is the number of words
- $PP(W) = P(w_1, w_2, w_3, \ldots w_N)^{-1/N}$

Using Perplexity

In our example

- $P(w_1, w_2, w_3, \ldots w_6) = 0.1 * 0.2 * 0.3 * 0.1 * 0.1 * 0.1$
- N is the number of words = 6
- $PP(W) = (0.1 * 0.2 * 0.3 * 0.1 * 0.1 * 0.1)^{-1/6}$
- PP(W) = 7.418

Using Perplexity

- A perplexity of 7.418 means that, on average, the model was as uncertain about its next word prediction as if it were choosing uniformly and independently among 7.418 possibilities.

- In other words, when the model is predicting the next word in a sequence, it's as if it's picking randomly from 7.418 words.

- A lower perplexity score is generally better because it means the model's predictions are more certain.

- A perplexity score of 100 means the model is as confused as guessing randomly, while a perplexity of 1 means the model has perfect predictive power.

Using BLEU

- Used for evaluating the quality of text which has been machine-translated from one natural language to another

- An example:
  - Suppose we have the sentence “I went to University to drink coffee”.
  - We use a large language model to translate the sentence into Te Reo Māori and the model produces: “I whakawhiti ahau ki te whare wānanga ki te inu kawhe”
  - We also consult a fluent speaker of Te Reo Māori and their translation is ““Haere ahau ki te whare wānanga ki te inu kawhe”

Using BLEU

- BLEU is based on the precision of n-grams (continuous sequences of n items) in the text generated by the model that are present in the text generated by the person fluent in Te Reo Māori.

- In this example we will look at unigrams (n=1) and bigrams (n=2)
- Unigrams in the text generated by the model are: (11 in total)
- “I”, “whakawhiti”, “ahau”, “ki”, “te”, “whare”, “wānanga”, “ki”, “te”, “inu”, “kawhe”

Using BLEU

- Unigrams in text generated for the fluent Te Reo Māori speaker are:
- “Haere”, “ahau”, “ki”, “te”, “whare”, “wānanga”, “ki”, “te”, “inu”, “kawhe”
- The unigrams in common are (9 in total):
- “ahau”, “ki”, “te”, “whare”, “wānanga”, “ki”, “te”, “inu”, “kawhe”
- Precision for unigrams = number of common unigrams / total unigrams in text generated by model = 9/11 = 0.818

Using BLEU

- Bigrams in text generated by model (10 in total)
- “I whakawhiti”, “whakawhiti ahau”, “ahau ki”, “ki te”, “te whare”, “whare wānanga”, “wānanga ki”, “ki te”, “te inu”, “inu kawhe”
- Bigrams in text from fluent speaker
- “Haere ahau”, “ahau ki”, “ki te”, “te whare”, “whare wānanga”, “wānanga ki”, “ki te”, “te inu”, “inu kawhe”
- Common bigrams: (8 in total)
- “ahau ki”, “ki te”, “te whare”, “whare wānanga”, “wānanga ki”, “ki te”, “te inu”, “inu kawhe”]
- Precision for bigrams = Number of common bigrams / Total bigrams in text from model translation = 8 / 10 = 0.8

Using BLEU

- The BLEU score is usually the geometric mean of the n-gram precisions, multiplied by a brevity penalty if the machine translation is shorter than the reference.
- In this case, the lengths are the same, so no brevity penalty applies.
- BLEU = (Precision for unigrams * Precision for bigrams)^(1/2) = sqrt(0.818 * 0.8) = 0.809
- The BLEU score for this example is 0.809.
- A BLEU score of 1 represents a perfect match with the reference translation, while a score of 0 represents no match. Therefore, a score of 0.713 indicates a relatively good match with the reference.

Using BLEU

In practice, BLEU is calculated using up to 4-grams (n=4), and multiple reference translations can be used to account for the fact that there can be multiple correct translations.