Meta AI LlaMa perplexity score - 13 February 2025

From Srinivasan Kannan <ka.shrinivaasan@gmail.com>
To <ksrinivasan@krishna-iresearch.org>
Date 2025-02-13 16:24

[14:50, 2/13/2025] Srinivasan Kannan: compute perplexity of "related quantity mathematics sciences one dealing. related quantity mathematics sciences one dealing. related quantity integers sciences one dealing . related parity mathematics sciences one dealing . related group mathematics sciences one dealing . related parity mathematics sciences one dealing"
[14:50, 2/13/2025] Meta AI: A fascinating task!

To compute the perplexity of the given text, I'll use a simple language model. Perplexity is a measure of how well a model predicts a sample of text. Lower perplexity indicates better prediction.

First, I'll tokenize the text:

1. related
2. quantity
3. mathematics
4. sciences
5. one
6. dealing
7. related
8. quantity
9. mathematics
10. sciences
11. one
12. dealing
13. related
14. quantity
15. integers
16. sciences
17. one
18. dealing
19. related
20. parity
21. mathematics
22. sciences
23. one
24. dealing
25. related
26. parity
27. mathematics
28. sciences
29. one
30. dealing

Next, I'll calculate the probability of each word given the context. Since the text is quite repetitive, I'll use a simple frequency-based approach.

Here's a rough estimate of the word frequencies:

- related: 5
- quantity: 3
- mathematics: 5
- sciences: 5
- one: 5
- dealing: 5
- integers: 1
- parity: 2

Using these frequencies, I can estimate the probability of each word. For example, the probability of the word "related" is 5/30, since it appears 5 times out of 30 total words.

Now, let's calculate the perplexity using the formula:

Perplexity = 2^(-1/N * ∑(log2(p(word))))

where N is the total number of words, and p(word) is the probability of each word.

Plugging in the numbers, I get:

Perplexity ≈ 2^(-1/30 * (∑(log2(p(word)))))

≈ 2^(-1/30 * (5*log2(5/30) + 3*log2(3/30) + 5*log2(5/30) + 5*log2(5/30) + 5*log2(5/30) + 5*log2(5/30) + 1*log2(1/30) + 2*log2(2/30)))

≈ 2^(-1/30 * (-15.51))

≈ 2^0.517

≈ 1.51

So, the perplexity of the given text is approximately 1.51. This is extremely low, indicating that the text is highly repetitive and predictable.

In fact, the perplexity is so low that it's almost like the model is "memorizing" the text rather than truly understanding its meaning. This is not surprising, given the repetitive nature of the text.
[14:51, 2/13/2025] Srinivasan Kannan: compute perplexity of "related quantity mathematics sciences one dealing."
[14:52, 2/13/2025] Meta AI: A much simpler task!

To compute the perplexity of the given text, I'll use a simple language model. Since the text is short, I'll use a frequency-based approach.

Here's the tokenized text:

1. related
2. quantity
3. mathematics
4. sciences
5. one
6. dealing

Since there's only one instance of each word, the probability of each word is 1/6.

Using the perplexity formula:

Perplexity = 2^(-1/N * ∑(log2(p(word))))

where N is the total number of words, and p(word) is the probability of each word.

Plugging in the numbers, I get:

Perplexity = 2^(-1/6 * (∑(log2(1/6))))
= 2^(-1/6 * (6_log2(1/6)))
= 2^(-1/6 * (6_(-2.585)))
= 2^(2.585)
≈ 6.00

So, the perplexity of the given text is approximately 6.00. This is a relatively moderate perplexity, indicating that the text is somewhat predictable, but not extremely so.

For comparison, a perplexity of:

- 1-2 indicates a highly predictable text (e.g., a repetitive sequence)
- 2-10 indicates a moderately predictable text (e.g., a simple sentence)
- 10-100 indicates a less predictable text (e.g., a paragraph of text)
- 100+ indicates a highly unpredictable text (e.g., a large corpus of text)

Keep in mind that this is a very simple estimate, and perplexity can be influenced by many factors, including the language model used and the context in which the text is evaluated.