Worked Examples for Common LLM Evaluation Metrics

Eng. Ahmed Métwalli

December 11, 2024

Contents

1 Introduction

This document provides numerical worked examples for each of the major metrics used in evaluating Large Language Models (LLMs). Each section will present:

- A brief reminder of the metric definition.
- A step-by-step numeric example.
- Interpretation of the resulting score.

These examples are simplified for clarity and may not reflect real-world data scale.

2 Perplexity (PPL)

Example Setup

Assume we have a small test sequence of $N = 5$ tokens: $\{w_1, w_2, w_3, w_4, w_5\}$. The model assigns the following conditional probabilities:
P(w$_1$) = 0.10,
P(w$_2$ | w_1) = 0.20,
P(w$_3$ | w_1, w_2) = 0.25,
P(w$_4$ | w_1, w_2, w_3) = 0.40,
P(w$_5$ | w_1, w_2, w_3, w_4) = 0.05.

Calculation

Perplexity is defined as:

$$\text{PPL} = 2^{-\frac{1}{N}\sum_{i=1}^{N}\log_2 P(w_i|w_{<i})}.$$

For the first token w_1, we consider $P(w_1)$ itself. Thus:

$$\sum_{i=1}^{5}\log_2 P(w_i \mid w_{<i}) = \log_2(0.10) + \log_2(0.20) + \log_2(0.25) + \log_2(0.40) + \log_2(0.05).$$

Compute each log:

$$\begin{aligned}
\log_2(0.10) &\approx -3.322,\\
\log_2(0.20) &\approx -2.322,\\
\log_2(0.25) &\approx -2.000,\\
\log_2(0.40) &\approx -1.322,\\
\log_2(0.05) &\approx -4.322.
\end{aligned}$$

Sum them up:

$$\sum_{i=1}^{5}\log_2 P(w_i \mid w_{<i}) \approx -3.322 - 2.322 - 2.000 - 1.322 - 4.322 = -13.288.$$

Average:

$$-\frac{1}{5}\sum_{i=1}^{5}\log_2 P(w_i \mid w_{<i}) = -\frac{-13.288}{5} = 2.6576.$$

Finally:

$$\text{PPL} = 2^{2.6576} \approx 6.29.$$

Interpretation

A PPL of about 6.29 suggests the model's predictions are moderately uncertain for this tiny example. A lower PPL would indicate more confidence and better predictive power.

3 BLEU Score

Example Setup

Candidate Translation: "The cat is on the mat" **Reference Translation:** "The cat is sitting on the mat"

Assume we compute BLEU with up to bigrams ($N = 2$) and equal weights $w_1 = w_2 = 0.5$.

Unigram Matches

Candidate unigrams: {The, cat, is, on, the, mat}

Reference unigrams: {The, cat, is, sitting, on, the, mat}

Unigram matches (count clipped): The (2 matches), cat (1), is (1), on (1), the (again counted in previous The match), mat (1). Note that we do not have "sitting" in candidate.

Total candidate unigrams: 6 Total matches: Let's carefully count unique matches. Matched unigrams = The(2), cat(1), is(1), on(1), mat(1) = 6 matches. But notice we have the word "the" twice in both candidate and reference, so that's fine. The candidate matches all tokens except "sitting." However, since "sitting" is not in candidate, it does not reduce matches; matches are clipped by the frequency in the reference.

Actually, let's be precise: - "The" appears twice in both candidate and reference, so clipped count for "The" = 2 - "cat" appears once in both, clipped count = 1 - "is" appears once in both, clipped count = 1 - "on" appears once in both, clipped count = 1 - "the" (already counted as The, case sensitivity aside if we treat them same) = included above - "mat" appears once in both, clipped count = 1

All candidate tokens appear in the reference except "sitting" does not appear in candidate, but that doesn't affect precision directly. Thus, unigram precision $p_1 = \frac{6}{6} = 1.0$.

Bigram Matches

Candidate bigrams: {The cat, cat is, is on, on the, the mat} Reference bigrams: {The cat, cat is, is sitting, sitting on, on the, the mat}

Matches: "The cat", "cat is", "on the", "the mat" are present in both, but "is on" is in candidate while reference has "is sitting" and "sitting on." So "is on" does not match.

Count matches: "The cat" (match), "cat is" (match), "is on" (no match), "on the" (match), "the mat" (match).

We have 4 matches out of 5 candidate bigrams. So $p_2 = \frac{4}{5} = 0.8$.

Length Penalty

Candidate length $c = 6$ tokens. Reference length $r = 7$ tokens. Brevity penalty = $\exp(\min(0, 1-\frac{r}{c})) = \exp(\min(0, 1-\frac{7}{6})) = \exp(\min(0, 1-1.1667)) = \exp(\min(0, -0.1667)) = \exp(-0.1667) \approx 0.8465$.

Final BLEU

$$\text{BLEU} = \exp(w_1 \log p_1 + w_2 \log p_2) \times \text{Brevity Penalty}.$$

Plug in:

$$p_1 = 1.0, \quad \log p_1 = 0, \quad p_2 = 0.8, \quad \log p_2 = \log(0.8) \approx -0.2231.$$

$$\text{BLEU} = \exp(0.5 \times 0 + 0.5 \times (-0.2231)) \times 0.8465 = \exp(-0.11155) \times 0.8465 \approx 0.894 \times 0.8465 \approx 0.757.$$

Final BLEU ≈ 0.757 or 75.7.

Interpretation

A BLEU score of about 0.76 is quite high, indicating the candidate translation is very similar to the reference in terms of word choice and order, despite being slightly shorter.

4 ROUGE-1

Example Setup

Candidate Summary: "The dog sleeps on a rug." **Reference Summary:** "A dog is sleeping on the rug."

Unigrams in candidate: {The, dog, sleeps, on, a, rug.} Unigrams in reference: {A, dog, is, sleeping, on, the, rug.}

Overlap (case-insensitive, punctuation removed):

- "The" (candidate) vs "the" (reference) = match of "the" (assuming lowercase)
- "dog" matches "dog"
- "sleeps" vs "sleeping" = no exact match
- "on" matches "on"
- "a" matches "a" (note reference has "A" - treat case-insensitive)
- "rug" matches "rug" (remove punctuation)

Matches: "the", "dog", "on", "a", "rug" = 5 out of the reference's 7 tokens.

$$\text{ROUGE-1} = \frac{\text{Number of overlapping unigrams}}{\text{Total unigrams in reference}} = \frac{5}{7} \approx 0.714.$$

Interpretation

A ROUGE-1 of about 0.71 suggests that most of the key words in the reference summary are captured by the candidate, even though "sleeps" and "sleeping" did not match exactly.

5 METEOR

Example Setup

Candidate: "The cat sat on the mat." **Reference:** "The cat is sitting on a mat."

Unigram matches (considering stem/synonyms): - Candidate tokens: {The, cat, sat, on, the, mat} - Reference tokens: {The, cat, is, sitting, on, a, mat}

Exact matches: "The", "cat", "on", "mat" "sat" vs "sitting": Using stemming, "sat" and "sitting" share the root "sit." So we have a stem match for "sat" and "sitting".

Ignoring articles and focusing on matched content words: Total matched unigrams (including exact and stem matches): 5 (The, cat, on, mat, and sat sitting pair).

Total candidate unigrams = 6, total reference unigrams = 7.

$$P = \frac{\text{Matches}}{\text{Candidate length}} = \frac{5}{6} \approx 0.8333$$

$$R = \frac{\text{Matches}}{\text{Reference length}} = \frac{5}{7} \approx 0.7143$$

Now, count chunks (frag): Suppose we align matches in order: - Match sequence could be: The(cat)(on)(mat) is straightforward. - Considering "sat" matches "sitting" (stem match), we place it in the same position. Let's assume we form 2 matched chunks: 1. "The cat [sat/sitting]" as one chunk 2. "on the mat" as second chunk

So frag = 2.

matchCount = 5.

$$\text{METEOR} = 10 \times \frac{P \times R}{P + R} \times (1 - 0.5 \times (\frac{\text{frag}}{\text{matchCount}}))$$

Compute harmonic mean:

$$\frac{P \times R}{P + R} = \frac{0.8333 \times 0.7143}{0.8333 + 0.7143} = \frac{0.5952}{1.5476} \approx 0.3845.$$

Now the chunk penalty:

$$1 - 0.5 \times \frac{\text{frag}}{\text{matchCount}} = 1 - 0.5 \times \frac{2}{5} = 1 - 0.5 \times 0.4 = 1 - 0.2 = 0.8.$$

Finally:

$$\text{METEOR} = 10 \times 0.3845 \times 0.8 = 10 \times 0.3076 = 3.076.$$

METEOR is often scaled between 0 and 1. The original METEOR formula differs slightly in scaling, but for demonstration, let's assume this is the final value. Normally, METEOR falls in [0,1]. If we recall METEOR's standard formula uses different normalization, let's just treat this as a comparative value. In practice, METEOR would yield a score closer to 0.3. The exact numeric scale can vary by implementation.

Interpretation

A METEOR-like score around 0.3 (after proper normalization) suggests moderately good alignment. It accounts for stem matches, giving credit for "sat" vs "sitting."

6 BERTScore

Example Setup

Candidate: "A man is eating pasta." **Reference:** "A person eats spaghetti."

For simplicity, assume we have embeddings (fictional values):

Token embeddings (dim=1 for simplicity): - "A": Candidate=0.9, Reference=0.91 - "man": Candidate=0.7, Reference="person"=0.68 - "is": Candidate=0.5, Reference="eats"=0.49 - "eating": Candidate=0.6, Reference="spaghetti"=0.55 - "pasta": Candidate=0.65, no direct match in reference tokens.

We compute cosine similarity as a simple product since dimension=1:

$$\text{cosine}(E(man), E(person)) \approx \frac{0.7 \times 0.68}{|0.7||0.68|} = 1.0 \text{ (since magnitudes are identical).}$$

(For simplicity, assume each embedding is normalized.)

Pair each candidate token with reference tokens to find best matches:

- "A" (0.9) best matches "A"(0.91) similarity 0.9*0.91=0.819 (approx)
- "man"(0.7) best matches "person"(0.68) similarity 0.7*0.68=0.476
- "is"(0.5) best matches "eats"(0.49) similarity 0.5*0.49=0.245
- "eating"(0.6) best matches "spaghetti"(0.55) similarity 0.6*0.55=0.33
- "pasta"(0.65) has no good semantic match, best might be "spaghetti"(0.55) again but we consider a one-to-one matching. If we consider greedy matching, "spaghetti" is already taken by "eating". So "pasta" might match leftover "eats"(0.49) with 0.65*0.49=0.318, but "eats" is also taken by "is" if we do a global max.

  For simplicity, let's just sum max matches ignoring complex alignment: Candidate length=5, sum of best matches approx: ("A" 0.819) + ("man" 0.476) + ("is" 0.245) + ("eating" 0.33) + ("pasta" 0.318) 2.188 total.

  BERTScore precision-like measure:

  $$\frac{\sum_{c \in C} \max_{r \in R} \text{similarity}(c, r)}{|C|} = \frac{2.188}{5} \approx 0.4376.$$

  (Exact BERTScore involves IDF weighting and a symmetrical calculation for recall, then combines them, but we illustrate the concept.)

Interpretation

A BERTScore of around 0.44 indicates moderate semantic overlap. Although the words differ ("man" vs "person", "pasta" vs "spaghetti"), semantically similar embeddings yield a reasonable score.

7 Exact Match (EM)

Example Setup

Ground Truth Answers: "Paris", "Blue whale", "42" **Model Answers:** "Paris", "Whale", "42"

$$\text{EM} = \frac{\sum_{i=1}^{3} \mathbf{1}(\hat{y}_i = y_i)}{3}.$$

Check each: 1. "Paris" vs "Paris" = exact match. 2. "Blue whale" vs "Whale" = not an exact match. 3. "42" vs "42" = exact match.

Matches = 2 out of 3.

$$\text{EM} = \frac{2}{3} \approx 0.6667.$$

Interpretation

An EM of about 0.67 indicates the model got two out of three answers perfectly correct.

8 F1-Score (for QA Overlap)

Example Setup

For a QA task, consider: **Reference Answer:** "Barack Obama" **Model Answer:** "Obama"

Tokenize: Reference: {"Barack", "Obama"} Candidate: {"Obama"}

Overlap: "Obama" is common. $TP = 1$ (Obama), $FN = 1$ (Barack not predicted), $FP = 0$ (no extra incorrect tokens).

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{1}{1+0} = 1.0,$$
$$\text{Recall} = \frac{TP}{TP + FN} = \frac{1}{1+1} = 0.5.$$

$$\text{F1} = 2 \cdot \frac{1.0 \times 0.5}{1.0 + 0.5} = 2 \cdot \frac{0.5}{1.5} = 2 \cdot 0.3333 = 0.6667.$$

Interpretation

An F1 of about 0.67 means the model partially captured the answer but missed one token ("Barack").

9 Conclusion

These worked examples provide hands-on understanding of how to compute and interpret key LLM evaluation metrics with actual numbers. In real scenarios, the calculations may involve large datasets, complex embeddings, and more sophisticated normalization, but these examples serve as a foundational reference.