

# Evaluation of Model Generation

CS XXX: Introduction to Large Language Models

# Contents

- Evaluating Generation
- BLEU
- ROUGE
- BERT
- LLM Judges

# Evaluating Generation

- We've seen perplexity as an automatic measure to evaluate language models

$$PP(x_{1:L}) = \exp \left( \frac{1}{L} \sum_{i=1}^L \log \left( \frac{1}{p(x_i | x_{1:i-1})} \right) \right)$$

- However, perplexity alone is insufficient to tell us about how well a model is solving some downstream task (e.g. translation or summarization)

# Evaluating Generation

- How good is a model's generated text?
- Imagine you are evaluating an LLM for English-to-French translation.
  - **Input (Source Sentence):**  
"The cat is on the mat."
  - **Reference (Ground Truths):** These are correct translations written by human experts.
    - Ref 1:** "Le chat est sur le tapis."
    - Ref 2:** "Le chat est posé sur le tapis."
  - **Candidate Output (LLM's Response):**  
"Le chat est sur la moquette."

# Evaluating Generation

- How good is a model's generated text?
- Imagine you are evaluating an LLM for English-to-French translation.
  - **Input (Source Sentence):**  
"The cat is on the mat."
  - **Reference (Ground Truths):** These are correct translations written by human experts.
    - Ref 1:** "Le chat est sur le tapis."
    - Ref 2:** "Le chat est posé sur le tapis."
  - **Candidate Output (LLM's Response):**  
"Le chat est sur la moquette."

The candidate output is not identical to the references.  
**How do we measure the goodness of the translation?**

# Evaluating Generation

- **Confusion Matrix:** A confusion matrix is a performance evaluation tool for classification models. It provides a summary of the model's predictions compared to the actual values.

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	<b>True Positives</b>	<b>False Positives</b>
Predicted Negative (0)	<b>False Negatives</b>	<b>True Negatives</b>

# Evaluating Generation

- Confusion Matrix for text-generation

	Word in the reference	Word not in the reference
Word predicted by model	<b>True Positives</b> Number of words in the candidate that are also in the reference	<b>False Positives</b> Number of words in the candidate that are not present in the reference
Word not predicted by model	<b>False Negatives</b> Number of words in the reference that are not present in the candidate	<b>True Negatives</b> Number of words not predicted by the model that are also not in the reference.

# Evaluating Generation

- Candidate translation (output of the model):

Israeli officials responsibility of airport safety

- Reference Translation (Ground Truth):

Israeli officials are responsible for airport security

- True Positives (TP) = 3
- False Positives (FP) = 3
- False Negatives (TN) = 4



# Evaluating Generation

- F-Measure – Precision and Recall of Words

$$\text{Precision (P)} = \frac{\text{True Positives (TP)}}{\text{True Positive (TP)} + \text{False Positives (FP)}}$$

$$\text{Recall (R)} = \frac{\text{True Positives (TP)}}{\text{True Positive (TP)} + \text{False Negatives (FN)}}$$

$$\text{F1 (F-Measure)} = \frac{2 \times P \times R}{P + R}$$

# Evaluating Generation

- Candidate translation (output of the model):  
Israeli officials responsibility of airport safety
- Reference Translation (Ground Truth):  
Israeli officials are responsible for airport security

$$P = \frac{3}{6} = \frac{1}{2}$$

$$R = \frac{3}{7}$$

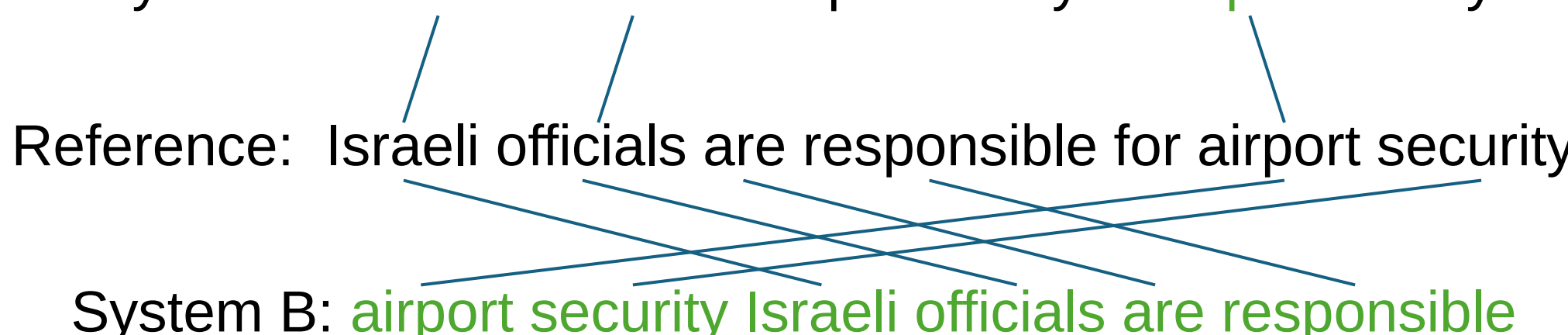
$$F1 = \frac{2 \times \frac{1}{2} \times \frac{3}{7}}{\frac{1}{2} + \frac{3}{7}} = 0.46$$

# Evaluating Generation

System A: Israeli officials responsibility of airport safety

Reference: Israeli officials are responsible for airport security

System B: airport security Israeli officials are responsible



Metric	System A	System B
Precision	50%	100%
Recall	43%	100%
F-Measure	46%	100%

Flaw: no penalty for reordering

# BLEU

The BLEU (Bilingual Evaluation Understudy) (Papineni et al. 2002) score is a metric used to evaluate the quality of machine-generated translations compared to reference translations. The BLEU score for a corpus of candidate translation sentences is a function of the n-gram word precision over all the sentences combined with a brevity penalty computed over the corpus as a whole.

$$\text{BLEU} = BP \times e^{\frac{1}{N} \sum_{n=1}^N \log p_n}$$

- Brevity Penalty (BP): is a factor used to penalize overly short translations. Since shorter translations tend to have higher precision (fewer words, less chance of incorrect ones), the brevity penalty prevents the BLEU score from being unfairly high for translations that are much shorter than the reference.

$$BP = \begin{cases} 1, & \text{if } c > r \\ e^{1-\frac{r}{c}}, & \text{if } c \leq r \end{cases} \quad c = \text{length of hypothesis translation}, r = \text{length of closest reference translation}$$

# BLEU

$$\text{BLEU} = BP \times e^{\frac{1}{N} \sum_{n=1}^N \log p_n}$$

- $BP = \begin{cases} 1, & \text{if } c > r \\ e^{1-\frac{r}{c}}, & \text{if } c \leq r \end{cases}$   $c$  = length of hypothesis translation,  $r$  = length of closest reference translation
- Precision  $p_n = \frac{\text{Number of n-grams in system and reference translations}}{\text{Number of n-grams in system translation}}$
- $N$  is usually 4

# BLEU

$$\text{BLEU} = BP \times e^{\frac{1}{N} \sum_{n=1}^N \log p_n}$$

**Candidate:** He He He eats tasty fruit (6 words)

**Reference1:** He eats a sweet apple (5 words)

**Reference2:** He is eating a tasty apple (6 words)

- Closest reference translation to the candidate (6 words) is Reference 2 (6 words).
- If two references are equally close, the shorter one is chosen based on character count.

$$c = r = 6$$

# BLEU

$$\text{BLEU} = BP \times e^{\frac{1}{N} \sum_{n=1}^N \log p_n}$$

$$N = 2$$

**Candidate:** He He He eats tasty fruit

**Reference1:** He eats a sweet apple

**Reference2:** He is eating a tasty apple

- $BP = \begin{cases} 1, & \text{if } c > r \\ e^{1-\frac{r}{c}}, & \text{if } c \leq r \end{cases}$   $c = \text{length of hypothesis translation}, r = \text{length of closest reference translation}$

$$BP = e^{1-\frac{6}{6}} = 1$$

Brevity Penalty (BP) is 1, which means no penalty is applied because the candidate's length matches the reference.

# BLEU

- Precision Calculation
  - Count: The number of occurrences of the n-gram in the candidate
  - Clipped Count: Limit the count to the maximum count in any reference



# BLEU

## ○ Precision Calculation

- **Candidate:** He He He eats tasty fruit
- **Reference1:** He eats a sweet apple
- **Reference2:** He is eating a tasty apple

Unigram	Correct	Count	Clipped Count
he	Yes	3	1
eats	Yes	1	1
tasty	Yes	1	1
fruit	no	1	0
Total		6	3

$$p_1 = \frac{\text{Clipped number of correct predicted 1-grams}}{\text{Number of total predicted 1-grams}} = \frac{3}{6}$$

# BLEU

## ○ Precision Calculation

- **Candidate:** He He He eats tasty fruit
- **Reference1:** He eats a sweet apple
- **Reference2:** He is eating a tasty apple

Bigram	Correct	Count	Clipped Count
he he	No	2	0
he eats	Yes	1	1
eats tasty	No	1	0
tasty apple	Yes	1	1
Total		5	2

$$p_2 = \frac{\text{Clipped number of correct predicted 2-grams}}{\text{Number of total predicted 2-grams}} = \frac{2}{5}$$

# BLEU

$$\text{BLEU} = BP \times e^{\frac{1}{N} \sum_{n=1}^N \log p_n}$$

$$\text{BLEU} = 1 \times e^{\frac{1}{2}(\log \frac{1}{2} + \log \frac{2}{5})}$$

$$\text{BLEU} = 0.447$$

- The BLEU score is between 0 and 1. A higher BLEU score indicates better translation quality, but values closer to 1 are difficult to achieve, especially for longer texts where some variation is expected.

# ROUGE

- ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a widely used evaluation metric for assessing the quality of text generation tasks, such as summarization and machine translation. Unlike BLEU, which primarily focuses on precision, ROUGE considers both recall and precision, making it well-suited for evaluating how much of the human-written reference text is captured by the model-generated output. It compares candidate against reference using n-grams, skip-bigrams, and longest common subsequences to measure overlap and relevance.

# ROUGE

- ROUGE is a set of metrics, rather than just one. The main ones that will be discussed are:
  - ROUGE-N
  - ROUGE-L
  - ROUGE-S

# ROUGE

- ROUGE-N

- ROUGE-N evaluates the quality of a model's output by measuring the overlap of n-grams (sequences of n words) between the model-generated text (candidate) and the human-written reference. The N in ROUGE-N corresponds to the size of the n-gram being compared:
  - ROUGE-1 measures the overlap of unigrams (single words) between the candidate and reference.
  - ROUGE-2 focuses on bigrams (pairs of consecutive words).
  - ROUGE-3 assesses the overlap of trigrams (triplets of consecutive words), and so on.
- This metric helps to determine how much of the model's output matches the structure and content of the reference text at different levels of granularity.

# ROUGE

## ROUGE-N

- $\text{Precision}(P_{\text{ROUGE-N}}) = \frac{\text{number of overlapping n-grams}}{\text{total n-grams in the candidate}}$
- $\text{Recall}(R_{\text{ROUGE-N}}) = \frac{\text{number of overlapping n-grams}}{\text{total n-grams in the reference}}$
- $F1(F1_{\text{ROUGE-N}}) = 2 \cdot \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$

# ROUGE

- ROUGE-N

- Reference: "The cat sat on the mat."
- Candidate: "The cat sat on mat."

N = 2

- Reference bigrams = [('The', 'cat'), ('cat', 'sat'), ('sat', 'on'), ('on', 'the'), ('the', 'mat')]
- Candidate bigrams = [('The', 'cat'), ('cat', 'sat'), ('sat', 'on'), ('on', 'mat')]
- Overlapping bigram count = 3
- Total Candidate Bigrams = 4
- Total Reference Bigrams = 5



# ROUGE

## ROUGE-N

- $P_{ROUGE-2} = \frac{3}{4} = 0.75$
- $R_{ROUGE-2} = \frac{3}{5} = 0.6$
- $F1_{ROUGE-2} = \frac{2 \times 0.75 \times 0.6}{0.75 + 0.6} = 0.667$

# ROUGE

- ROUGE-L

- ROUGE-L is based on the Longest Common Subsequence (LCS) rather than n-grams. The LCS captures sentence-level structure similarity, making it useful when word order matters but minor modifications are acceptable. LCS is the longest sequence of words appearing in order in both summaries (but not necessarily consecutively).

# ROUGE

## ROUGE-L

- $\text{Precision}(P_{\text{ROUGE-L}}) = \frac{\text{number of tokens in LCS}}{\text{total tokens in the candidate}}$
- $\text{Recall}(R_{\text{ROUGE-L}}) = \frac{\text{number of tokens in LCS}}{\text{total tokens in the reference}}$
- $F1(F1_{\text{ROUGE-L}}) = 2 \cdot \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$

# ROUGE

- ROUGE-L
  - Reference: "The cat sat on the mat."
  - Candidate: "The cat sat on mat."
  - LCS = The cat sat on mat
  - LCS tokens = 4

# ROUGE

## ROUGE-L

- $P_{ROUGE-L} = \frac{4}{5} = 0.8$
- $R_{ROUGE-L} = \frac{4}{6} = 0.667$
- $F1_{ROUGE-L} = \frac{2 \times 0.8 \times 0.667}{0.8 + 0.667} = 0.727$

# ROUGE

- ROUGE-S

- ROUGE-S stands for ROUGE-Skip-Bigram. Unlike ROUGE-N (which considers only consecutive n-grams), ROUGE-S measures how many skip-bigrams from the reference summary appear in the candidate (generated) summary.
- A skip-bigram is any ordered pair of words appearing in the same order in the sentence, even if words are skipped in between.

# ROUGE

## ROUGE-S

- $\text{Precision}(P_{\text{ROUGE-S}}) = \frac{\text{number of overlapping skip-bigrams}}{\text{total skip-bigrams in candidate}}$
- $\text{Recall}(R_{\text{ROUGE-S}}) = \frac{\text{number of overlapping skip-bigrams}}{\text{total skip-bigrams in reference}}$
- $F1(F1_{\text{ROUGE-S}}) = 2 \cdot \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$

# ROUGE

- ROUGE-S

Reference: "The cat sat on the mat."

Candidate: "The cat sat on mat."

- Reference skip-bigrams = [  
('The', 'cat'), ('The', 'sat'), ('The', 'on'), ('The', 'the'), ('The', 'mat'),  
('cat', 'sat'), ('cat', 'on'), ('cat', 'the'), ('cat', 'mat'),  
('sat', 'on'), ('sat', 'the'), ('sat', 'mat'),  
('on', 'the'), ('on', 'mat'),  
('the', 'mat')  
]

- Candidate skip-bigrams = [  
('The', 'cat'), ('The', 'sat'), ('The', 'on'), ('The', 'mat'),  
('cat', 'sat'), ('cat', 'on'), ('cat', 'mat'),  
('sat', 'on'), ('sat', 'mat'),  
('on', 'mat')  
]

Overlapping skip-bigrams = 10, Reference skip-bigrams = 15, Candidate skip-bigrams = 10



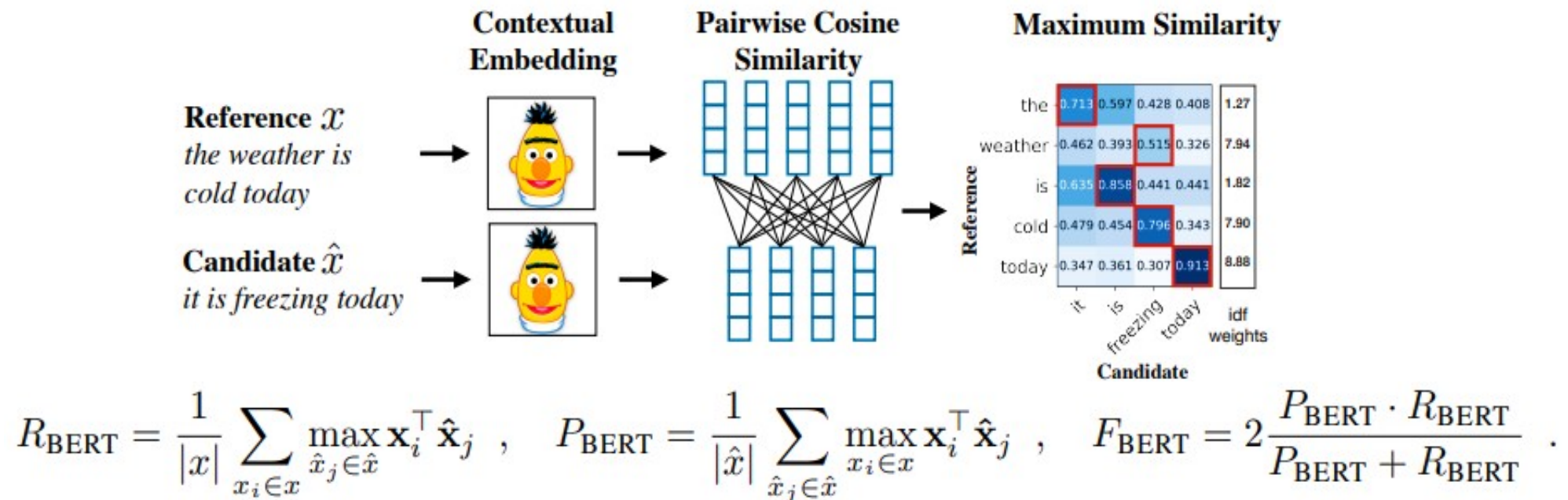
# ROUGE

## ROUGE-S

- $P_{ROUGE-S} = \frac{10}{10} = 1$
- $R_{ROUGE-S} = \frac{10}{15} = 0.667$
- $F1_{ROUGE-S} = 2 \cdot \frac{1 \times 0.667}{1 + 0.667} = 0.800$

# BERT Score

- Measures the similarity between model predictions and a set of ground truths by computing cosine similarities between word embeddings of the reference and candidate sentences generated using BERT.
- Assigns a score between 0 and 1 for each reference (ground truth) and candidate (prediction) pair



# BERT Score

- Reference Sentence

“dog runs”

- Candidate Sentence

“puppy moves”

We create word embeddings using BERT for the both the reference sentence and the candidate sentence. For simplicity, we use 3-dimensional embeddings (as we have seen before, actual BERT embeddings are much larger)

# BERT Score

- Reference Sentence

“dog runs”

$$\text{dog} = \begin{bmatrix} 0.2 \\ 0.5 \\ 0.7 \end{bmatrix}, \quad \text{runs} = \begin{bmatrix} 0.9 \\ 0.8 \\ 0.6 \end{bmatrix}$$

- Candidate Sentence

“puppy moves”

$$\text{puppy} = \begin{bmatrix} 0.1 \\ 0.3 \\ 0.5 \end{bmatrix}, \quad \text{moves} = \begin{bmatrix} 0.85 \\ 0.75 \\ 0.65 \end{bmatrix}$$

# BERT Score

- Cosine Similarity between two vectors **a** and **b**

$$\cos(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \times \|\mathbf{b}\|}$$

- Computer pair-wise cosine similarity for e.g. “dog” vs. “puppy”

$$\| \text{"dog"} \| = \sqrt{0.2^2 + 0.5^2 + 0.7^2} = 0.883$$

$$\| \text{"puppy"} \| = \sqrt{0.1^2 + 0.3^2 + 0.5^2} = 0.592$$

$$\cos(\text{dog}, \text{puppy}) = \frac{(0.2 \times 0.1) + (0.5 \times 0.3) + (0.7 \times 0.5)}{0.883 \times 0.592} = 0.995$$

# BERT Score

- Pair-wise cosine similarities
  - Cosine similarity between “dog” and “puppy” = **0.995**
  - Cosine similarity between “dog” and “moves” = 0.867
  - Cosine similarity between “runs” and “puppy” = 0.792
  - Cosine similarity between “runs” and “moves” = **0.998**

# BERT Score

- Precision

$$P_{\text{BERT}} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} \mathbf{x}_i^\top \hat{\mathbf{x}}_j$$

Candidate vs. Reference

puppy:  $\max(0.995, 0.792) = 0.995$

moves:  $\max(0.867, 0.998) = 0.998$

$$P_{\text{BERT}} = \frac{1}{2} (\max(0.995, 0.792) + \max(0.867, 0.998))$$

$$P_{\text{BERT}} = \frac{1}{2} (0.995 + 0.998) = 0.9965$$

# BERT Score

~ Recall

$$R_{\text{BERT}} = \frac{1}{|x|} \sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} \mathbf{x}_i^\top \hat{\mathbf{x}}_j$$

Reference vs. Candidate

dog:  $\max(0.995, 0.867) = 0.995$

runs:  $\max(0.792, 0.998) = 0.998$

$$R_{\text{BERT}} = \frac{1}{2} (\max(0.995, 0.867) + \max(0.792, 0.998))$$

$$R_{\text{BERT}} = \frac{1}{2} (0.995 + 0.998) = 0.9965$$



# BERT Score

- F1

$$F_{\text{BERT}} = 2 \frac{P_{\text{BERT}} \cdot R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}}$$

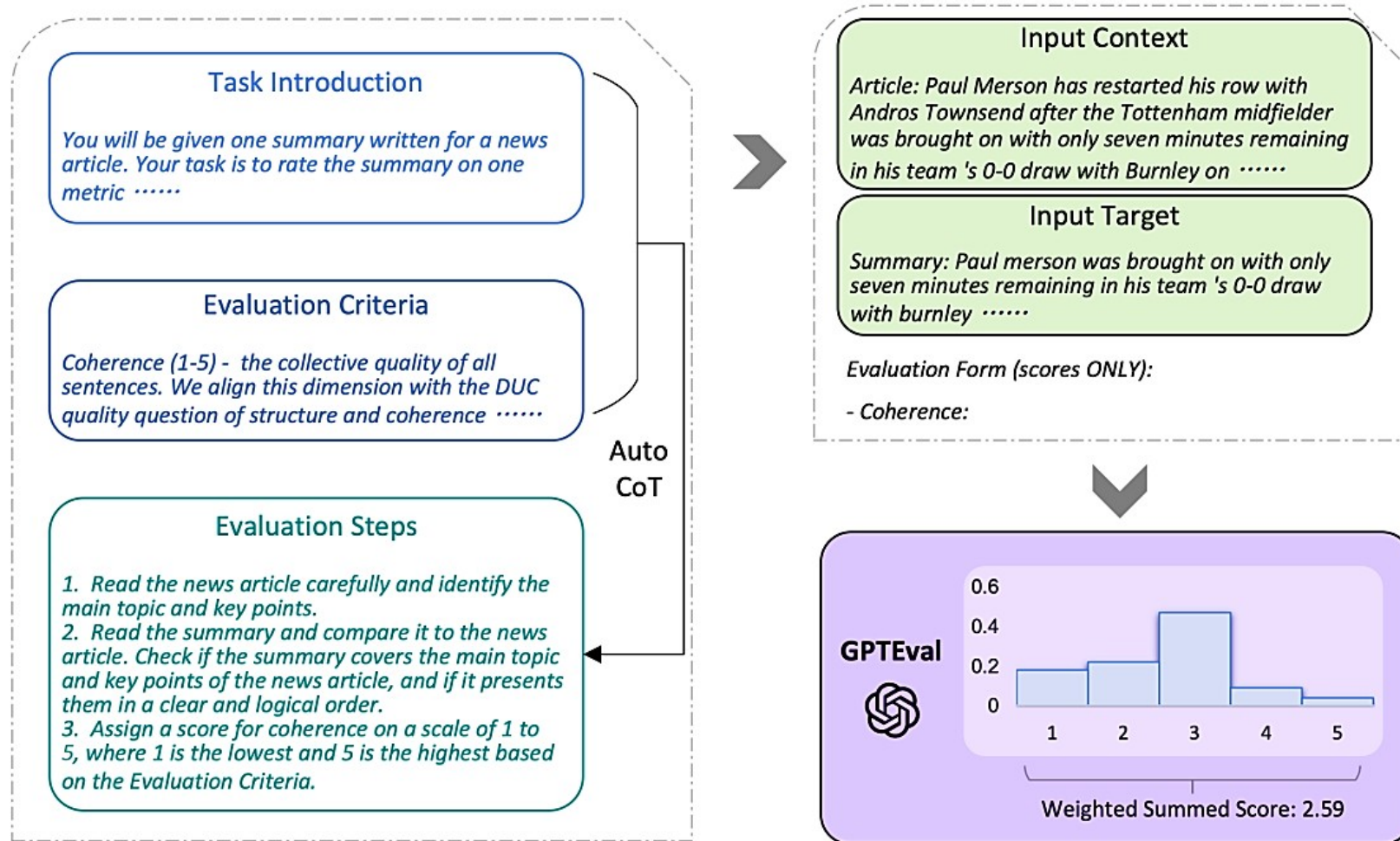
$$F_{\text{BERT}} = \frac{2 \cdot 0.9965 \cdot 0.9965}{0.9965 + 0.9965}$$

$$F_{\text{BERT}} = 0.9965$$

- Since the candidate words closely match the reference words based on cosine similarities, the BERT-Score ( $F_{\text{BERT}}$ ) is very high, indicating strong semantic similarity between the sentences.

# Evaluating Generation – Machine Translation

- LLMs as Judges: Use a powerful LLM like GPT4 to assign a score to generated text based on some evaluation criteria. (GPTEval, Liu et al., 2023)



# References

- Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). BLEU: A method for automatic evaluation of machine translation.
- Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. Text Summarization Branches
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding.
- Liu, Y., Zhang, Y., Wei, J., et al. (2023). GPT Eval: Leveraging GPT models for automatic evaluation of free-form text.