Evaluation Metrics

1. DocVQA Accuracy

Definition:

DocVQA Accuracy measures the model's ability to correctly answer questions based on document images (e.g., scanned forms, invoices, contracts). It tests both visual layout understanding and natural language comprehension.

Formula:

$$\text{Accuracy} = \frac{\text{Number of Correct Answers}}{\text{Total Questions}}$$

Where:

- *Correct Answers*: Model predictions exactly matching ground truth answers
- *Total Questions*: Total visual questions asked on documents

Examples:

1. Given an invoice image, Q: "What is the total due?" → Model answers "₹1,500.00" → Correct
2. Document: Certificate. Q: "What is the date of issue?" → Model replies "15 August 2022" (exact match) → Correct

Applications:

- Visual question answering over documents
- OCR-based form understanding
- Intelligent document processing (IDP)

---

2. Information Extraction F1

Definition:

Measures how accurately a model extracts structured information (entities, slots) from unstructured text. Combines precision (correctly predicted items) and recall (all actual items retrieved).

Formula:

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Where:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad \text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

Examples:

1. Text: "Elon Musk founded SpaceX in 2002 in California."
   Extracted: {ORG: SpaceX, PER: Elon Musk, DATE: 2002} → High F1
2. Text: "Google was established by Larry Page and Sergey Brin."
   Misses one founder or adds wrong info → F1 drops

Evaluation Metrics

Applications:

- Named Entity Recognition (NER)
- Resume parsing, clinical report extraction
- Knowledge base population

---

3. Natural Questions F1

Definition:
Used in open-domain QA, it evaluates token-level overlap between the predicted answer and the ground truth. Useful when answers are not exact matches but partially correct.
Formula:

Same as standard F1, but applied on tokens:

$$\text{Precision} = \frac{\text{Overlapping Tokens}}{\text{Tokens in Prediction}} \quad \text{Recall} = \frac{\text{Overlapping Tokens}}{\text{Tokens in Ground Truth}}$$

Examples:

1. GT: “Barack Hussein Obama”, Prediction: “Barack Obama” → Partial token overlap → F1 ≈ 0.66
2. GT: “United States of America”, Prediction: “USA” → No token overlap → F1 = 0

Applications:

- Open-domain QA systems
- Knowledge-grounded assistants
- Chatbot evaluation

---

4. Exact Match (EM)

Definition:
Binary metric: determines whether the model’s output exactly matches the reference answer. Highly strict, no room for partial correctness.

$$\text{EM} = \begin{cases} 1, & \text{if prediction} = \text{ground truth} \\ 0, & \text{otherwise} \end{cases}$$

Formula:

Examples:

1. GT: “Marie Curie”, Prediction: “Marie Curie” → EM = 1
2. GT: “Saturn”, Prediction: “planet Saturn” → EM = 0 (extra word)

Evaluation Metrics

Applications:

- Reading comprehension
- Short answer evaluation
- QA datasets like SQuAD

---

5. Relevance

Definition:
Assesses whether the retrieved/given content is topically relevant to a user's query. Usually evaluated using scoring models or human judgments.
Formula:
No fixed formula — may use:

- Cosine similarity
- BERT-based scoring
- Human annotations (Likert scale)

Examples:

1. Query: "Photosynthesis steps" → Retrieved: "Plants use chlorophyll to convert sunlight…" → Relevant
2. Query: "India's GDP" → Retrieved: "Population growth in Africa" → Irrelevant

Applications:

- RAG pipelines
- Document retrieval
- Context selection in QA

---

6. Faithfulness

Definition:
Evaluates factual correctness of generated outputs relative to the source context. Penalizes hallucinations or contradictions.
Formula:
No standard formula; typically evaluated using:

- Fact-checking models
- QA over references
- Human judgment

Examples:

1. Input: "Obama was president in 2009." Output: "Barack Obama served as president starting in 2009." → Faithful
2. Input: Same. Output: "Donald Trump began his term in 2009." → Unfaithful

Applications:

- Factual summarization
- RAG-based generation
- Scientific/medical text generation

---

7. Context Recall

Definition:
Measures how many of the relevant context documents (ground truth) were successfully retrieved by the model.
Formula:

$$\text{Context Recall} = \frac{\text{Number of Relevant Contexts Retrieved}}{\text{Total Number of Relevant Contexts (Ground Truth)}}$$

Examples:

1. Ground truth: 5 documents. Retrieved: 4 correct → Recall = 0.8
2. Ground truth: 3. Retrieved: only 1 → Recall = 0.33

Applications:

- Retrieval-Augmented Generation
- Multi-hop QA
- Knowledge retrievers (e.g., DPR, BM25)

---

8. Dialogue Coherence

Definition:
Checks whether a chatbot's responses follow logical, topical, and contextual consistency with the conversation history.
Formula:
No fixed formula; evaluated using:

- Coherence models
- Human rating scales (1–5)
- Discourse modeling

Examples:

1. User: "What's the weather like?"

Bot: "It's sunny and warm today." → Coherent

2. User: "Tell me a joke."

Bot: "India's independence was in 1947." → Incoherent

Applications:

- Conversational agents
- Customer service bots
- Virtual assistants

---

9. Intent Match

Definition:

Checks whether the model correctly identifies and acts on the user's intended goal. Important for task-oriented systems.

Formula:

$$\text{Intent Match} = \begin{cases} 1, & \text{if Detected Intent} = \text{True Intent} \\ 0, & \text{otherwise} \end{cases}$$

Examples:

1. User: "Remind me to drink water." → Detected: Reminder → Match = 1
2. User: "Book me a table." → Detected: Weather Inquiry → Match = 0

Applications:

- Virtual assistants
- Command and control systems
- Voice bots (Alexa, Siri)

---

10. GPTScore

Definition:

Uses a GPT model to evaluate generated text by assigning scalar scores or pairwise rankings based on fluency, coherence, and informativeness.

Formula:

No universal formula. Can involve:

- Prompted scoring
- Log-likelihoods
- Pairwise preferences

Evaluation Metrics

Examples:

1. Prompt GPT-4: “Rate this response on coherence (1–5)” → Output: 4 → Score = 4
2. Given two summaries, ask GPT: “Which is better?” → Uses preference to score

Applications:

- Model evaluation without humans
- Preference-based RL training (RLHF)
- Summary and generation grading

---

11. Rubric Evaluation Accuracy

Definition:
Measures how well the model output satisfies pre-defined criteria (e.g., grammar, content, structure), often used in structured assessments.
Formula:

$$\text{Rubric Accuracy} = \frac{\text{Number of Rubric Conditions Satisfied}}{\text{Total Number of Rubric Conditions}}$$

Examples:

1. Rubric: Grammar, Structure, Relevance, Detail
   Output satisfies 3/4 → Accuracy = 0.75
2. Essay meets only 1 criterion → Accuracy = 0.25

Applications:

- Essay grading
- Formal answer evaluation
- Generative model benchmarking

---

12. ROUGE-L

Definition:
Evaluates summary quality using the longest common subsequence (LCS) between reference and generated text, capturing fluency and phrase-level similarity.
Formula:

- LCS = Longest Common Subsequence
- Precision = LCS / Gen Length
- Recall = LCS / Ref Length
- F1 = Harmonic Mean of Precision and Recall

Evaluation Metrics

Examples:
1. Reference: “The dog barked at night.”
Generated: “The dog barked loudly.” → Partial LCS → Medium ROUGE-L
2. Reference: “India won the match.”
Generated: “India won the match.” → ROUGE-L = 1

Applications:
- Text summarization
- Headline generation
- Story simplification

---

13. BERTScore

Definition:
Calculates semantic similarity between reference and prediction using contextualized BERT embeddings, capturing meaning beyond exact words.

Formula:

$$\text{BERTScore} = \text{Avg. Cosine Similarity between aligned token embeddings}$$

Examples:
1. Ref: “Dogs are friendly.”
Gen: “Canines are kind.” → High semantic similarity → High BERTScore
2. Ref: “Paris is the capital of France.”
Gen: “Eiffel Tower is in Europe.” → Low semantic overlap → Low score

Applications:
- Paraphrase detection
- Machine translation
- Semantic summarization

14. BLEU (Bilingual Evaluation Understudy)

Definition:
BLEU measures the overlap of n-grams between a machine-generated sentence and one or more reference sentences. It focuses on precision — how many of the generated words are also in the reference — and is used widely in machine translation.

Formula:

$$\text{BLEU} = BP \times \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

Where:
- BPBPBP: Brevity Penalty to penalize short outputs
- pnp_npn: Modified n-gram precision for n=1 to N (usually N=4)
- wnw_nwn: Weight for each n-gram level (typically uniform)

Examples:
1. Ref: "The cat is on the mat"
Gen: "The cat is on mat" → 4-gram match partially missed → Lower BLEU
2. Ref: "He is playing football."
Gen: "He is playing football." → Perfect match → BLEU = 1.0

Applications:
- Machine Translation (MT)
- Text generation tasks
- Summarization (less common due to precision bias)

---

15. COMET (Crosslingual Optimized Metric for Evaluation of Translation)

Definition:
COMET is a neural metric that evaluates translation quality using a pretrained multilingual encoder. It considers both source and reference sentences to estimate adequacy and fluency.

Formula:
Learned function:

$$\text{COMET} = f(\text{source}, \text{hypothesis}, \text{reference})$$

Where f is a neural regression model predicting human judgment scores.

Examples:
1. Source (en): “I love my dog.”
Hypothesis: “J’adore mon chien.”
Reference: “J’aime mon chien.” → Semantic match → High COMET score
2. Source: “It is raining.”
Hypothesis: “Le soleil brille.” (Sun is shining) → Incorrect → Low COMET

Applications:
- Machine Translation evaluation
- Cross-lingual summarization
- Reference-free MT scoring (with COMET-QE)

---

16. chrF++ (Character n-gram F-score)

Definition:
chrF++ evaluates text generation quality based on character-level n-gram overlap (plus some word-level matching), making it robust to morphology and minor word order variations.

Formula:

$$\text{chrF} = (1 + \beta^2) \cdot \frac{\text{Precision} \cdot \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}}$$

- Precision & Recall are computed over character n-grams
- Typically β=2 to favor recall

Examples:

1. Ref: "unbelievable"

Gen: "unbeleivable" → Small typo → High chrF

2. Ref: "The weather is good."

Gen: "Climate is nice." → Different wording → Low chrF++

Applications:

- Machine translation for morphologically rich languages
- Text simplification
- Spelling-robust scoring

---

17. Factuality Score

Definition:

Factuality Score evaluates how factually accurate the generated text is with respect to known or verifiable information. Often derived from QA-based methods or fact-checking classifiers.

Formula:

No fixed formula; based on:

- Fact-checking model outputs
- QA over source documents
- Human annotations (True/False labels)

Examples:

1. Source: "The capital of France is Paris."

Output: "France's capital is Paris." → Factual

Output: "France's capital is Lyon." → Not factual → Low score

Applications:

- Scientific or medical summarization
- News generation
- RAG system outputs

---

18. Faithfulness Score (Reused)

Definition:

Sometimes reused or calculated differently across tasks, this version focuses on checking whether model generations remain grounded in the input context — especially in summarization or generation from evidence.

Evaluation Metrics

Formula:

$$\text{Faithfulness} = \frac{\text{Number of Factually Consistent Sentences}}{\text{Total Generated Sentences}}$$

Examples:

1. Input: Wikipedia article on Einstein.
 Summary: “Einstein developed relativity.” → Faithful
 Summary: “Einstein won a Grammy.” → Hallucinated → Not faithful

Applications:

- Abstractive summarization
- Dialogue generation
- LLM hallucination detection

---

19. Creativity Score

Definition:
Measures the novelty or inventiveness of the model’s output. Often scored manually or by prompting GPT models to rate novelty, uniqueness, and surprise value.

Formula:
No standard formula; usually:

- Human rating (Likert scale)
- Model-based scoring (e.g., GPT: “Rate the creativity from 1–5”)

Examples:

1. Prompt: “Write a story about a clock that eats time.” → Creative response = High score
2. Prompt: “Tell a joke.”
 Model says: “Why did the chicken cross the road?” → Overused → Low score

Applications:

- Story generation
- Ad copywriting
- Creative writing tools

---

20. Story Coherence

Definition:
Evaluates the logical flow and consistency of narrative elements in a generated story. Focuses on character consistency, event ordering, and cause-effect chains.

Formula:
No formal formula; judged by:

- Coherence classifiers
- Human judgment (e.g., consistency score out of 5)

Examples:

1. Story: "She woke up, then ate breakfast, and left for work." → Logically coherent
2. Story: "He died in chapter 2 but fought dragons in chapter 4." → Incoherent timeline

Applications:

- Story generation (e.g., novel writing AI)
- Game narrative generation
- Script and scene modeling

---

21. Recall@K

Definition:
Measures whether at least one of the correct answers appears in the top-K results retrieved by a model. Common in retrieval and ranking tasks.
Formula:

$$\text{Recall@K} = \frac{\text{Number of Relevant Items in Top-K}}{\text{Total Number of Relevant Items (Ground Truth)}}$$

Examples:

1. Query: "Who discovered penicillin?"
 Top-5 results include "Alexander Fleming" → Recall@5 = 1
2. Top-5 results: "Newton, Darwin, Pasteur…" → Missed correct answer → Recall@5 = 0

Applications:

- Document ranking
- RAG retrievers
- Multi-hop QA

---

22. Precision@K

Definition:
Measures how many of the top-K retrieved items are relevant. Unlike Recall@K, it penalizes irrelevant results.
Formula:

$$\text{Precision@K} = \frac{\text{Number of Relevant Items in Top-K}}{K}$$

Examples:

1. Top-5 docs: 3 are relevant → Precision@5 = 3/5 = 0.6
2. Top-10 docs: only 2 are relevant → Precision@10 = 0.2

Applications:

- Search engines
- QA retrieval systems
- Evaluation of document retrievers

Evaluation Metrics

23. Mean Reciprocal Rank (MRR)

Definition:
Evaluates how early in the ranked list the first relevant result appears. The reciprocal rank of the first correct answer is averaged over multiple queries.
Formula:

$$\mathrm{MRR} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\mathrm{rank}_i}$$

Where rank_i is the position of the first relevant document for query i.
Examples:

1. Correct doc at rank 1 → Reciprocal = 1
2. Correct doc at rank 5 → Reciprocal = 1/5 = 0.2

Applications:

- QA and search
- RAG retriever evaluation
- Legal or academic document search