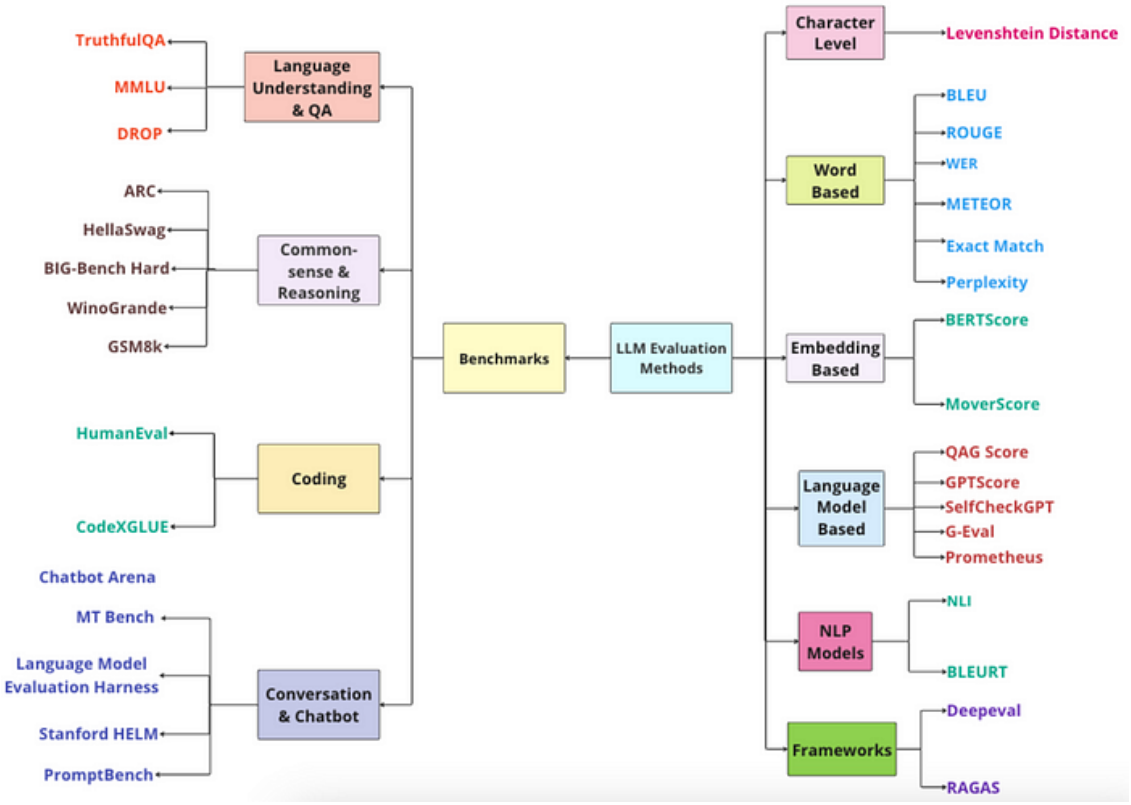
จุดประสงค์ของเอกสารนี้จัดทำขึ้นเพื่อรวบรวม สรุป วิธีการประเมินโมเดลทางภาษาต่างๆ รวมถึงการประเมิน RAG แอพพลิเคชั่น



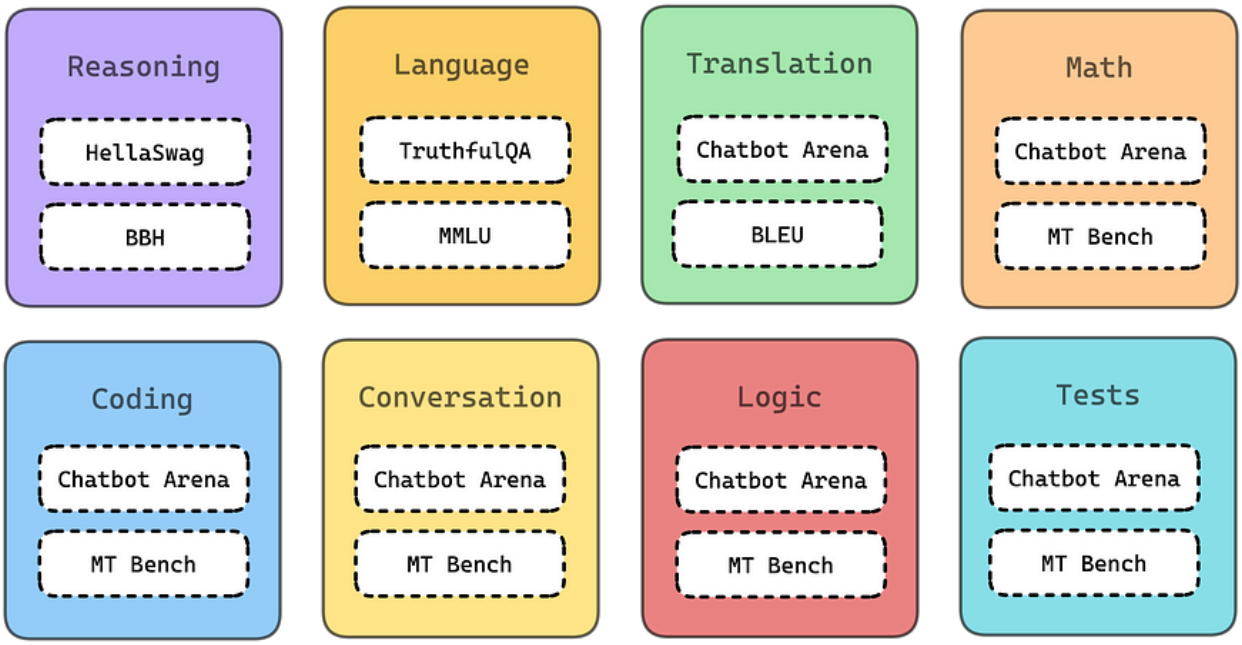
ข้อแตกต่างระหว่าง LLM Benchmarking Vs. Evaluation

Benchmarking จะใช้สำหรับการทดสอบที่เป็นมาตรฐาน ที่มีการกำหนดชุดข้อมูล การวัดเมทริกซ์ที่เป็นมาตรฐานของ benchmark นั้นๆ อยู่แล้ว เพื่อที่จะใช้สำหรับทดสอบโมเดลทางภาษาในแต่ประเภทงาน เช่นการอ่าน การคำนวณ การเขียนโค้ด และช่วยให้นักวิจัยสามารถนำผลลัพธ์ที่ได้ ไปเปรียบเทียบกับโมเดลทางภาษาอื่นๆ กับประเภทต่างๆ เพื่อนำไปวิเคราะห์ในการเลือกใช้งานโมเดลในลำดับต่อไปได้

Evaluation ในขณะที่ evaluation นั้นค่อนข้างยืดหยุ่นมากกว่า ในการประเมินโมเดลทางภาษา ซึ่งเหมาะกับการนำไปใช้ในการประเมินแอพพลิเคชั่นที่เป็นการใช้งานจริง เช่นการประเมินความน่าเชื่อถือของคำตอบ การประเมินความเป็นกลางในการตอบคำถาม

1.LLM Benchmarking

Benchmarking จัดทำขึ้นเพื่อประเมินประเมินโมเดลทางภาษา กับชุดข้อมูลมาตรฐาน ที่แสดงถึงประสิทธิภาพในการทำงานของโมเดลกับงานนั้นๆ โดยในที่นี้จะยกตัวอย่างชุด Benchmarking บางส่วน เนื่องจากการใช้งานจริงเราอาจจะไม่ได้ใช้งานในส่วนนี้ในการประเมินโมเดลเพราะไม่ได้ออกแบบมาเพื่องานเฉพาะ



1.1 Language understanding

1.1.1 TruthfulQA

Benchmark ที่ประกอบไปด้วย 817 คำถาม ใน 38 categories ที่เกี่ยวข้องกับด้านสุขภาพ กฎหมาย การเงิน และ การปกครอง จุดประสงค์เพื่อให้โมเดลหลีกเลี่ยง หรือป้องกันคำถาม ความเข้าใจผิดๆของคน

1.1.2 MMLU (Massive multitask language understanding)

เป็นชุดข้อมูลที่ประกอบไปด้วย task มากกว่า 57 task โดยประกอบไปด้วย การคำนวณเลข ประวัติศาสตร์ กฎหมาย และอื่นๆ และรูปแบบข้อมูลเป็นแบบ multiple choice จุดประสงค์ใช้ในการประเมินโมเดล pre-train โดยการโฟกัสที่การทำ few-shot และ zero-shot

1.2 Common-sense and reasoning benchmarks

1.2.1 ARC (AI2 Reasoning Challenge)

เป็นชุดคำถามและคลังข้อความที่รวบรวมเอาไว้สำหรับประเมินงาน question answering ที่ถามคำถามเกี่ยวกับวิทยาศาสตร์ โดยจะมีตัวเลือกมาให้

1.2.2 HellaSwag

เป็นชุดข้อมูลที่ใช้สำหรับประเมินความสามารถในการให้เหตุผล ที่มีตัวเลือกภายในชุดคำถาม และมีข้อมูลมากกว่า 10000 ข้อมูล

1.3 Coding Benchmarks

1.3.1 HumanEval

เป็นชุดข้อมูลที่ประเมินทางด้าน programming กว่า 164 งาน ออกแบบมาเพื่อประเมินการเขียนโค้ดของโมเดล

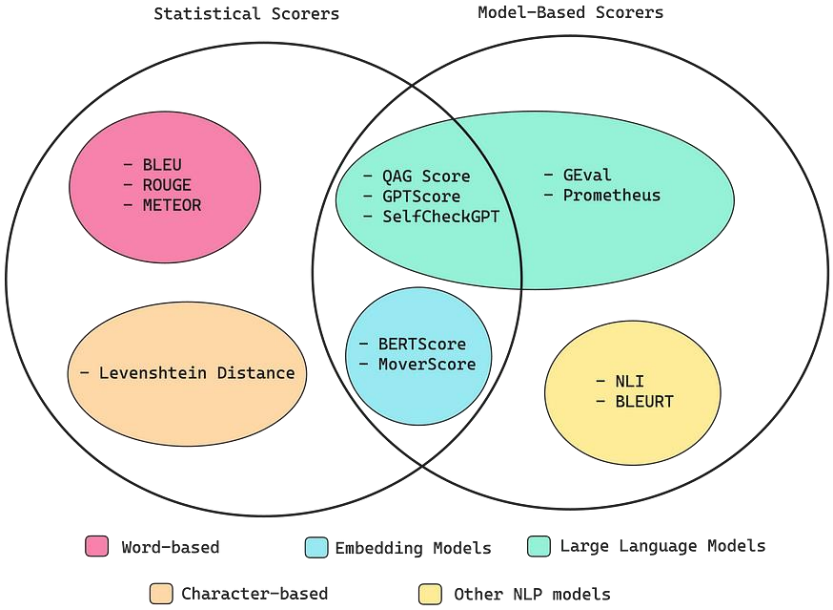
1.4 Limitation of Benchmarks

แม้ว่าการประเมินจะมีความสำคัญในการประเมินศักยภาพของ LLMs (Language Models) แต่มันก็มีข้อจำกัดที่ควรคำนึงถึงได้แก่

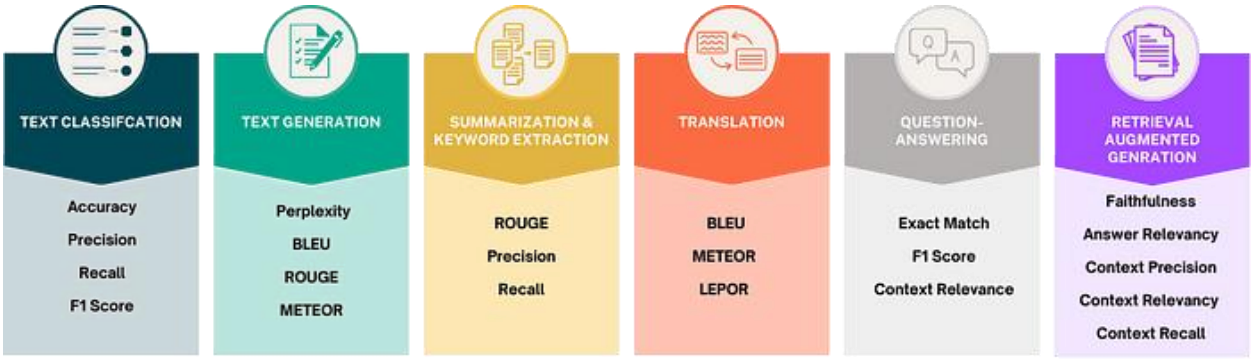
1. ความสอดคล้องในโดเมน การประเมินมักไม่สามารถปรับให้ตรงกับโดเมนหรือบริบทเฉพาะที่ LLMs ถูกนำมาใช้ได้ดีพอ เช่น การวิเคราะห์ทางกฎหมายหรือการตีความทางการแพทย์ ทำให้มีช่องว่างในการประเมินประสิทธิภาพของ LLMs ในการใช้งานเฉพาะทางต่าง ๆ ที่หลากหลาย

2. อายุขัยของมาตรฐาน เมื่อมาตรฐานการประเมินใหม่ ๆ ถูกนำมาใช้ LLMs มักไม่สามารถทำได้ดีเท่ากับการเทียบกับมาตรฐานของมนุษย์ แต่ภายในเวลา 1-3 ปี จำเป็นจะต้องมีการอัพเดทความรู้ใหม่

2. LLM Evaluation Metrics



มีวิธีการมากมายในการคำนวณ เมทริก เช่นการคำนวณจาก neural network, จาก embedding model, การใช้ LLM ในการคำนวณ หรือการประยุกต์ใช้การคำนวณแบบสถิติ ซึ่งในแต่ละเมทริกซ์ก็จะสามารถนำไปใช้กับ task แต่ละ task เช่น การประเมินด้าน text generation ก็จะใช้ perplexity, BLEU score ในการวัดค่าความแม่นยำ ซึ่งแต่ละเมทริกก็จะขึ้นอยู่กับวัตถุประสงค์ของงานดังรูปด้านล่าง



2.1 Statistical Score

2.1.1 Word Error rate (WER)

การคำนวณ WER (Word Error Rate) มาจากระยะห่างของ Levenshtein ซึ่งทำงานในระดับคำแทนที่จะเป็นระดับพยางค์ WER

จากนั้นสามารถคำนวณค่า WER ได้ดังนี้:

Word error rate can then be computed as:

$$WER = \frac{S + D + I}{N} = \frac{S + D + I}{S + D + C}$$

where

- S is the number of substitutions,
- D is the number of deletions,
- I is the number of insertions,
- C is the number of correct words,
- N is the number of words in the reference (N=S+D+C)

WER ถูกใช้อย่างแพร่หลายในการประเมินประสิทธิภาพของระบบรู้จำเสียง (ASR) และการแปลภาษาเพราะมันให้ข้อมูลเกี่ยวกับอัตราความผิดพลาดที่เกิดขึ้นเมื่อระบบประมวลผลคำ

2.1.2 Exact match

เป็นการวัดแบบ match กันระหว่างข้อความที่สร้างจากโมเดลเทียบกับข้อความอ้างอิง เหมาะสำหรับชุดข้อมูลที่มีการถามและคำตอบผลลัพธ์เป็นคำตอบสั้นๆเท่าที่ตรงตัวเท่านั้น เช่นสมมติคำถามคือประเทไทยมีเมืองหลวงชื่อว่า ให้ reference text คือ กรุงเทพ ในขณะที่โมเดลตอบว่า ฉันคิดว่ากรุงเทพหรือเชียงใหม่ นั้นเท่ากับว่าโมเดลจะถูกว่า score เท่ากับ 1 เนื่องจากข้อความที่โมเดลตอบมามีคำว่ากรุงเทพ ทั้งที่โมเดลให้คำตอบไม่แน่ชัดก็ตาม หากคำในโมเดลไม่มีการ match จะเท่ากับ 0

2.1.3 Perplexity

เป็น metric ที่ใช้สำหรับวัดการสร้างข้อความเป็นหลักโดยจะต้องอาศัยช่วงการฝึกโมเดลหรือการ evaluate โมเดลในการคำนวณเมทริกซ์ เนื่องจากต้องการมีการคิดค่า probability distribution ของโมเดลที่ทำนาย โดยเมทริกซ์ประเภทนี้จะไม่ใช้ reference text ในการเทียบคำตอบ โดยยิ่งค่าน้อยยิ่งดี

ตัวอย่างการคำนวณ Link: https://medium.com/nlplanet/two-minutes-nlp-perplexity-explained-with-simple-probabilities-6cdc46884584

2.1.4 BLEU

เป็นเมทริกที่นิยมใช้ในงานด้าน machine translation โดยวัดความแมนยำจากการ overlap n-gram ระหว่างข้อความโมเดลสร้างกับ reference text เป็นเพียงการประเมินหยาบๆ โดยที่ไม่ได้ดูความหมายของบริบท

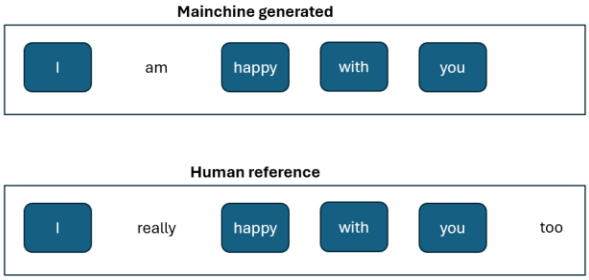
Example

BLEU Score:
Candidate Translation: The quick brown fox jumps over the lazy dog.
Unigrams=["The", "quick", "brown", "fox", "jumps", "over", "the", "lazy", "dog"]= 9
Reference Translation: The quick brown dog jumps over the lazy fox.
Unigrams=["The", "quick", "brown", "dog", "jumps", "over", "the", "lazy", "fox"]=9
Unigram Precision
matching unigrams = 9 (all)
Unigram Precision $= \frac{\text{(Overlapping 1-grams)}}{\text{(Total 1-grams in Candidate Translation)}} = \frac{9}{9} = 1$
matching bigrams: ["The quick", "quick brown", "jumps over", "over the", "the lazy"]=5
Bigram Precision $= \frac{\text{(Overlapping 2-grams)}}{\text{(Total 2-grams in Candidate Translation)}} = \frac{5}{8}$
matching trigrams: ["The quick brown", "jumps over the", "over the lazy"] = 3
Trigram Precision $= \frac{\text{(Overlapping 3-grams)}}{\text{(Total 3-grams in Candidate Translation)}} = \frac{3}{7}$
matching 4-grams: ["jumps over the lazy"]
4-gram Precision $= \frac{\text{(Overlapping 4-grams)}}{\text{(Total 3-grams in Candidate Translation)}} = \frac{1}{6}$
PB = 1 since length is same
BLEU Score=1* $\exp((1/4) * (\log(1) + \log(5/8) + \log(3/7) + \log(1/6)))$ **= 0.46**

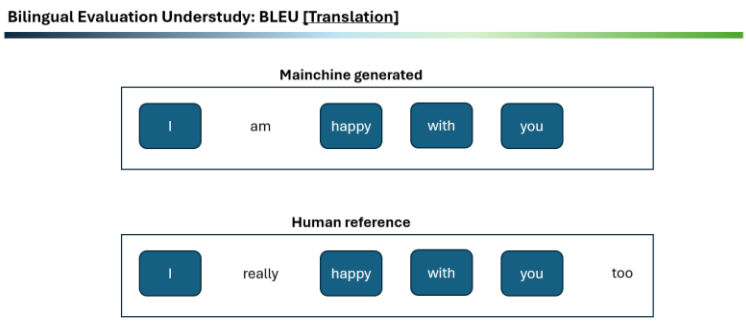
With Python Library:

```
from nltk.translate.bleu_score import sentence_bleu

reference = [['The', 'quick', 'brown', 'fox', 'jumps',
'over', 'the', 'lazy', 'dog']]

candidate = ['The', 'quick', 'brown', 'dog', 'jumps',
'over', 'the', 'lazy', 'fox']


score = sentence_bleu(reference, candidate)


print(score)


0.46
```

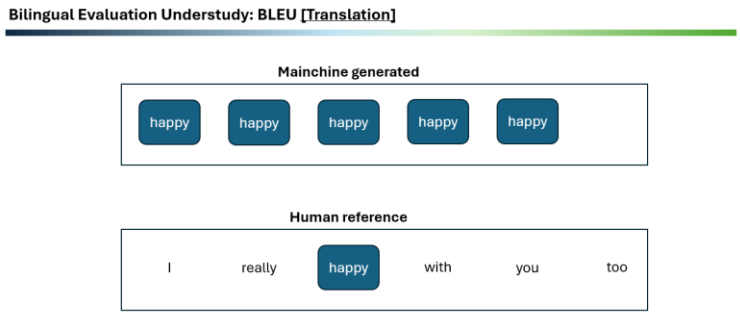
Bilingual Evaluation Understudy: BLEU [Translation]



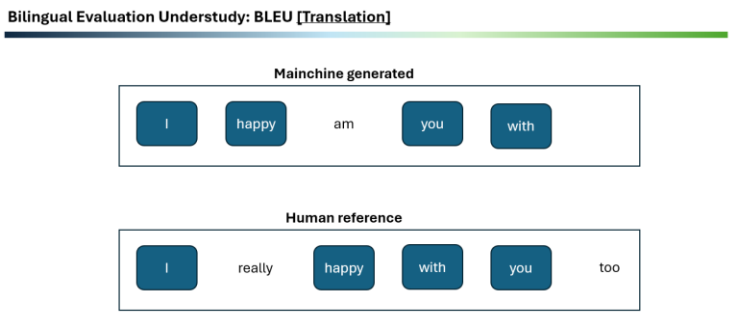
$$\text{Unigram Precision} = \frac{\text{No. of word matches}}{\text{No. of words generated}} = \frac{4}{5}$$



$$\text{Unigram Precision} = \frac{\text{No. of word matches}}{\text{No. of words generated}} = \frac{4}{5}$$



$$\text{Modified Unigram Precision} = \frac{CLIP(\text{No. of word matches})}{\text{No. of words generated}} = \frac{1}{5}$$



$$\text{Modified Unigram Precision} = \frac{CLIP(\text{No. of word matches})}{\text{No. of words generated}} = \frac{4}{5}$$

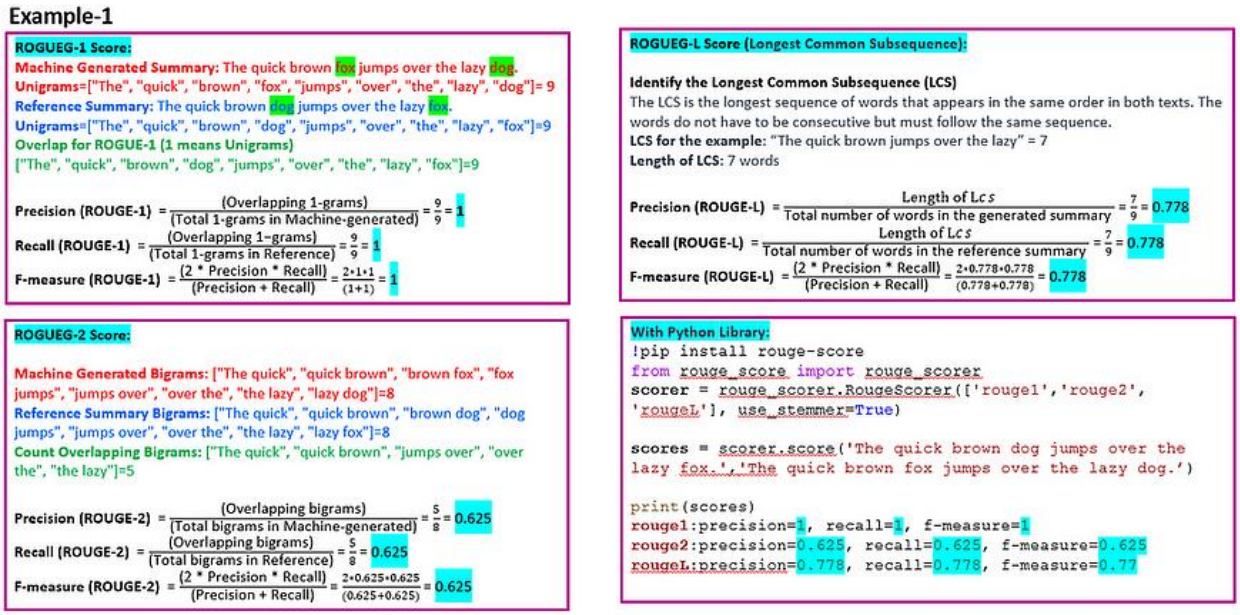
จากตัวอย่างข้างต้น credit: P'ming จุดเด่นของโมเดลนี้คือการใช้งานที่ง่ายและเร็ว ในขณะที่สามารถนำไปใช้งานกับงานประเภทอื่นที่ไม่ใช่ machine translation ได้ แต่ก็จะมีข้อเสียในเรื่องของการตีความหมายที่ลึกซึ้ง ลักษณะโครงสร้างประโยค และการตัดคำของหลากหลายภาษาที่แตกต่างกัน

2.1.5 ROUGE

เป็นเมทริกซ์ที่นิยมใช้ในการประเมินงานด้านการสรุปข้อความ โดยจะเป็นการเปรียบเทียบระหว่างข้อความที่สรุป กับ reference summary โดยอาจจะเป็น ref ที่มาจากคนเขียนขึ้นเอง เพื่อเปรียบเทียบกัน โดยการคำนวณจะมีการวัดกันของ n-

grams overlap ลำดับของข้อความ และ word-pair โดยทั่วไป ROUGE จะมีอยู่สามแบบ ROUGE-1 และ ROUGE-2 ที่ใช้วัด n-gram ในขณะที่ ROUGE-L ใช้วัดประโยคที่ยาวที่สุดที่มีการเหมือนกัน ไม่จำเป็นต้องต่อเนื่องกัน แต่ยังคงเรียบตามลำดับ โดยเมทริกของ ROUGE จะออกมาอยู่ในรูปแบบ 3 รูปแบบได้แก่

1. Precision (ความแม่นยำ) สัดส่วนของ n-gram (ลำดับของ n คำ) ในการสรุปที่โมเดลสร้างขึ้นที่ปรากฏในสรุปข้อความอ้างอิง
2. Recall (ความครอบคลุม) สัดส่วนของ n-gram ในสรุปข้อความอ้างอิงที่ปรากฏในสรุปที่โมเดลสร้างขึ้น
3. F1-Score (F1 Score) ค่าเฉลี่ยฮาร์โมนิกของความแม่นยำและความครอบคลุม เพื่อทำให้เกิดความสมดุลระหว่างสองตัวชี้วัด



https://medium.com/nlplanet/two-minutes-nlp-learn-the-rouge-metric-by-examples-f179cc285499

แม้ว่าเมทริกซ์ rouge จะใช้งานได้ง่ายและเร็ว แต่ก็ยังมีข้อจำกัดในเรื่องความเข้าใจบริบท และต้องการ reference ในการอ้างอิง

2.1.6 METEOR

เป็นเมทริกซ์ที่พัฒนามาจาก BELU score โดยทที่จะไม่ได้แค่พิจารณา exact word แต่ยังรวมถึง word ในรูปแบบของ stemming หรือ synonyms สำหรับการประเมินการแปลภาษา โดยมีการรักษาสมดุลระหว่าง precision และ recall รวมถึงสามารถกำหนด penalty สำหรับความแตกต่างระดับคำ การคำนวณจะต้องคำนวณ Precision และ Recall จากการ exact ในรูปแบบที่ลดรูป stem และ synonym แล้ว จากนั้นนำไปหา harmonic mean ของ recall และ precision และกำหนด penalty เพื่อเพิ่มบทลงโทษสำหรับลำดับคำที่ไม่ถูกต้อง

*F-mean=10*P*R/(R+9*P) ; R = Recall, P = Precision*

Penalty=0.5(# of chunks/# of matches)**3 ; chunk = จำนวนคำที่ตรงกันต่อเนื่อง, match จำนวนคำที่ตรงกัน*

*Score=(1−Penalty)*F-mean*

Example

METEOR Score:
Candidate Translation: The quick brown fox jumps over the lazy dog.
Unigrams=["The", "quick", "brown", "fox", "jumps", "over", "the", "lazy", "dog"]= 9
Reference Translation: The quick brown dog jumps over the lazy fox.
Unigrams=["The", "quick", "brown", "dog", "jumps", "over", "the", "lazy", "fox"]=9
Total matches (overlap) = 9 words matches

Unigram Precision (P) = $\frac{\text{(Overlapping 1-grams)}}{\text{(Total 1-grams in Candidate Translation)}} = \frac{9}{9} = 1$

Unigram Recall (R) = $\frac{\text{(Overlapping 1-grams)}}{\text{(Total 1-grams in Reference Translation)}} = \frac{9}{9} = 1$

Harmonic (F) Mean = $\frac{10PR}{R+9P} = \frac{10*1*1}{1+9*1} = 1$

There are three chunks: (fox, dog) and (dog, fox) are the same chunk

Penalty = $0.5*(\frac{\text{\# of chunks}}{\text{\# of matches}})^{**}3 = 0.5*(\frac{3}{7})^{**}3 =$ **0.03935**

Score = (1-Penalty)*Fmean =(1-0.03935)*1 = 0.961

With Python Library:

```
import nltk
nltk.download('punkt')
nltk.download('wordnet')
from nltk.translate.meteor_score import meteor_score
from nltk import word_tokenize

prediction = " The quick brown fox jumps over the lazy dog"
reference = " The quick brown dog jumps over the lazy fox "

results = meteor_score([word_tokenize(reference)], word_tokenize(prediction),
alpha = 0.9, beta = 3, gamma = 0.5)

Print(results)
0.956
```

แม้ว่าจะเป็นเมทริกที่พัฒนาจาก BELU score แต่ก็ยังมีข้อจำกัดในเรื่องของการประเมินกับภาษาอื่นๆที่นอกจากภาษาอังกฤษ รวมถึงการ tokenize ที่จะต้องเป็นแบบเดียวกัน

2.1.7 CiDER

CIDEr เป็นเมทริกที่ออกแบบมาเพื่อประเมินคุณภาพของข้อความบรรยายภาพ (image captioning) โดยคำนึงถึงความเหมือนในระดับการใช้คำ, ความสัมพันธ์ในเชิงโครงสร้างของคำ, และการให้ความสำคัญกับคำที่สำคัญและมีความหมายเฉพาะในบริบทนั้น ๆ CIDEr คำนวณโดยใช้การเปรียบเทียบระหว่างข้อความบรรยายที่สร้างขึ้น (candidate caption) กับข้อความบรรยายที่เป็นตัวอย่าง (reference captions) หลาย ๆ อัน การคำนวณจะมีการใช้ TF-IDF เพื่อให้ความสำคัญกับคำที่สำคัญและไม่ซ้ำในบริบท จากนั้นนำค่า TF-IDF ที่ได้ไปหา cosine-similarity ในการวัดความคล้ายคลึงกันของข้อความที่สร้างกับ ข้อความ reference จะนั้นนำไปหาค่าเฉลี่ยคะแนน cider ของ n-grams ต่างๆ

ข้อดีคือใช้ สามารถ match word ที่มากขึ้น แต่ในขณะเดียวกันจำเป็นจะต้องมี list reference ที่เพิ่มขึ้น

https://www.youtube.com/watch?v=3nZF99Z4CIc

2.2 Model-Based Scorers

2.2.1 G-Eval

เป็นเมทริกซ์ที่ใช้ llm โมเดล GPT4 ในการประเมินที่ทำให้เหมือนกับการตัดสินใจของมนุษย์ ด้วยการใส่ prompt ขั้นตอนการประเมิน (CoTs) โดยมีการกำหนดเกณฑ์การให้คะแนน ซึ่งวิธีการนี้จะช่วยทำให้สามารถประเมินความหมายของคำตอบได้ ทั้งนี้การประเมินความน่าเชื่อถือก็จะต้องอ้างอิงตามโมเดลที่ใช้ประเมิน (สามารถดูตัวอย่างได้ใน deep eval)

2.2.2 Prometheus

มีหลักการทำงานคล้าย G-eval เพียงแต่โมเดลที่นำมาใช้ในการประเมินเป็น Llama-2-chat ซึ่งผ่านการ finetune กับ 100K feedback และถูกออกแบบมาให้เป็น open-source สำหรับการใช้งานฟรี ข้อแตกต่างอีกข้อของ Prometheus คือจำเป็นจะต้องมี reference ในการประเมิน

2.3 Combining Statistical and Model-Based Scorers

2.3.1 GPTScore

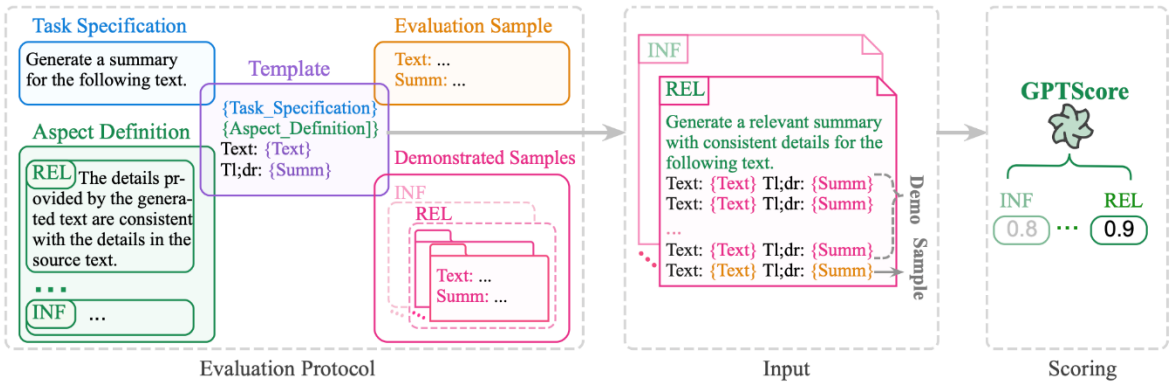


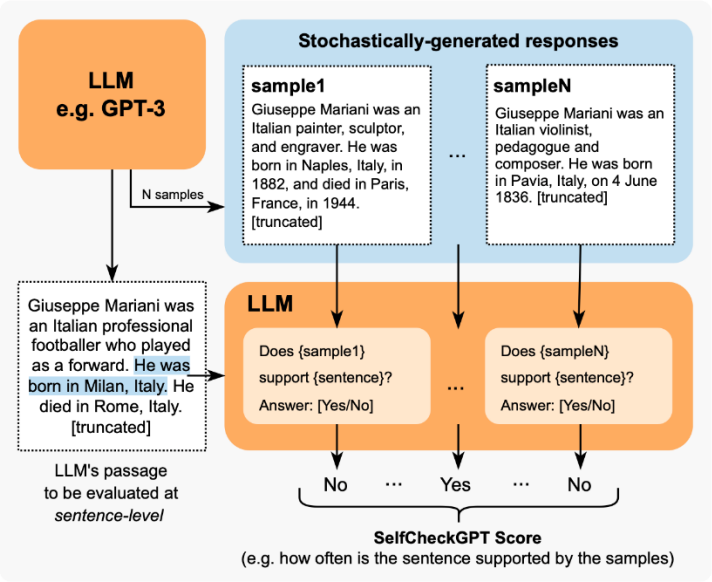
Figure 2. The framework of GPTSCORE. We include two evaluation aspects *relevance (REL)* and *informative (INF)* in this figure and use the evaluation of *relevance (REL)* of the text summarization task to exemplify our framework.

https://wandb.ai/vincenttu/blog_posts/reports/Evaluating-Generative-Models-with-GPTScore--VmlldzozNTI2NjQy

เป็นการประเมินที่อาศัยความสามารถของ zero-shot instruction โดยมีการกำหนด prompt ที่จะประกอบไปด้วย specification และ aspect ตามรูปด้านบน โดยงานวิจัยได้มีการทดสอบกับโมเดลที่หลากหลาย และพบว่าทำงานได้ดีกับ GPT3 วิธีการลักษณะนี้คล้ายกับ G-eval

2.3.2 SelfCheckGPT

เป็นวิธีการตรวจสอบ Hallucinate โดยการสุ่มตัวอย่างเพื่อเช็คความถูกต้อง ของการตอบจากโมเดล โดยมีแนวคิดว่าคำตอบที่ถูกสุ่มขึ้นมาจะต้องมีแนวโน้มและข้อเท็จจริงที่สอดคล้องกันนแต่ละครั้ง หากมีการ hallucinate เกิดขึ้นแสดงว่าข้อเท็จจริงจากการสุ่มมีการขัดแย้งกันเอง



2.3.3 QAG Score

เป็นเมทริกซ์ที่ใช้ประโยชน์จาก LLM ในการประเมิน โดยที่รูปแบบการประเมินจะทำการสกัดข้อมูลจาก llm output โดยแต่ละข้อมูลที่ถูกสกัดจะนำมาตั้งถามกับ ground truth ว่าเป็นจริงหรือเท็จ 0 หรือ 1 ทำให้การคำนวณเมทริกซ์สุดท้ายมีความน่าเชื่อถือมากขึ้น เนื่องจากรูปแบบการถามเพื่อวัดประสิทธิภาพ มีเพียงแค่ใช่กับไม่

จากตัวอย่างเช่น

Martin Luther King Jr., the renowned civil rights leader, was assassinated on April 4, 1968, at the Lorraine Motel in Memphis, Tennessee. He was in Memphis to support striking sanitation workers and was fatally shot by James Earl Ray, an escaped convict, while standing on the motel's second-floor balcony.

ข้อมูลที่สกัดออกมา

Martin Luther King Jr. assassinated on the April 4, 1968

คำถามที่สอดคล้องกัน

Was Martin Luther King Jr. assassinated on the April 4, 1968?

จากนั้นจะทำการถามคำถามว่าเห็นด้วยกับประโยคกล่าวอ้างนี้ไหม yes หรือ no

2.4 RAG Metrics

2.3.1 Faithfulness

ความสอดคล้องเป็นเมตริกของ RAG ที่ประเมินว่า LLM ที่ใช้ RAG กำลังสร้างผลลัพธ์ของ LLM ที่สอดคล้องตามข้อเท็จจริงกับข้อมูลที่นำเสนอในบริบทการสืบค้นหรือไม่ โดยหลักการคือ

- ใช้ LLM เพื่อ extract information
- สำหรับแต่ละ information ตรวจสอบว่ามันสอดคล้องหรือขัดแย้งกับแต่ละโหนด retriever context
- นับจำนวน information ที่เป็นจริงทั้งหมด (‘ใช่’ และ ‘ไม่ทราบ’) และหารด้วยจำนวน information ทั้งหมดที่สร้างขึ้น

2.3.2 Answer Relevancy

ความเกี่ยวข้องของคำตอบเป็นเมตริกของ RAG ที่ประเมินว่าสามารถสร้างคำตอบที่กระชับตรงตามคำถามอินพุตได้หรือไม่ และสามารถคำนวณได้โดยการหาสัดส่วนของประโยคในผลลัพธ์ของ LLM ที่เกี่ยวข้องกับข้อมูลนำเข้า (input)

2.3.3 Contextual Precision

เป็นเมตริกของ RAG ที่ประเมินคุณภาพของตัวสืบค้นในกระบวนการ RAG ของคุณ เมื่อเราพูดถึงเมตริกเชิงบริบท เราสนใจความเกี่ยวข้องของบริบทการสืบค้นเป็นหลัก contextual precision สูงหมายถึงโหนดที่เกี่ยวข้องในการสืบค้นถูกจัดเรียงลำดับได้ถูกต้อง สิ่งนี้สำคัญเพราะ LLM ให้ความสำคัญกับข้อมูลในโหนดที่ปรากฏก่อนในบริบทการสืบค้นมากกว่า ซึ่งส่งผลต่อคุณภาพของผลลัพธ์สุดท้าย

2.3.4 Contextual Recall

เป็นเมตริกเพิ่มเติมสำหรับการประเมิน Retriever-Augmented Generator (RAG) คำนวณโดยการหาสัดส่วนของประโยคใน expect output หรือ ground truth ซึ่งสามารถเชื่อมโยงกับโหนดในบริบทการสืบค้นได้ คะแนนที่สูงขึ้นแสดงถึงความสอดคล้องกันมากขึ้นระหว่างข้อมูลที่สืบค้นได้กับผลลัพธ์ที่คาดหวัง

2.3.5 Contextual Relevancy

Contextual Relevancy วัดสัดส่วนของประโยคในบริบทที่ถูกดึงมาที่เกี่ยวข้องกับอินพุตที่กำหนด การมีค่า Contextual Relevancy สูงแสดงถึงความเกี่ยวข้องที่ดีของบริบทที่ถูกดึงมาใช้

2.5 Model-Based Scorers

2.3.1 BLEURT (Bilingual Evaluation Understudy with Representations from Transformers)

ใช้รูปแบบกาฝึกโมเดล machine learning เพื่อให้สามารถจับความคล้ายคลึงกันของความหมายระหว่างประโยค reference กับผลลัพธ์ ที่ไม่เกี่ยวข้องกันได้ โดยฝึกจากชุดข้อมูล WMT Metrics Shared Task dataset ที่เป็นการให้ rating จากคน

Input: Bud Powell était un pianiste de légende. Reference: Bud Powell was a legendary pianist.	BLEURT
Candidate 1: Bud Powell was a legendary pianist.	*1.01*
Candidate 2: Bud Powell was a historic piano player.	*0.71*
Candidate 3: Bud Powell was a New Yorker.	*-1.49*

ข้อจำกัดของเมทริกซ์คือพึ่งพาความสามารถของโมเดลและการนำไปใช้อาจจะใช้ได้ดีแค่ในโดเมนที่โมเดลนี้ถูกเทรนมา ซึ่งอาจไม่เหมาะกับการนำไปใช้ในปัจจุบัน รวมถึงไม่ค่อยมีการอัพเดตแล้ว

2.6 Fine-tuning metrics

ในเชิงของ finetune-metric คือการวัดประเมินภายในตัว LLM ว่ามีความรู้ที่เพิ่มเติมเกิดขึ้นไหม รวมถึงในแง่ของพฤติกรรมการตอบคำถาม

2.6.1 Hallucination

เป็นการวัดการตอบของโมเดลที่แต่งเติมขึ้นมาเองและไม่ตรงกับข้อเท็จจริง โดยภายใน deepeval ไลบรารี่ จะนำวิธีการเดียวกับ SelfCheckGPT มาใช้ในการประเมิน

2.6.2 Toxicity

ใช้ในการประเมินภาษาที่ไม่เหมาะสม เป็นอันตราย โดยสามารถใช้โมเดล Detoxify ที่ถูกฝึกเพื่อให้คะแนนความ toxic ได้ อย่างไรก็ตามวิธีอาจจะไม่เหมาะกับทุกงาน เนื่องจากบางบริบท คำที่อันตราย หรือหยาบคายอาจจะไม่ใช่ เจตนาในแง่ร้าย จึงสามารถใช้วิธีการ G-eval และกำหนดเงื่อนไขทดแทนได้

2.6.3 Bias

ใช้ในการประเมินอคติทางการเมือง เพศ สังคม ซึ่งจะใช่วิธีการ geval และกำหนดเงื่อนไขเช่น criteria= Bias - determine if the actual output contains any racial, gender, or political bias.

3. Evaluation tool or framework

3.1 DeepEvals

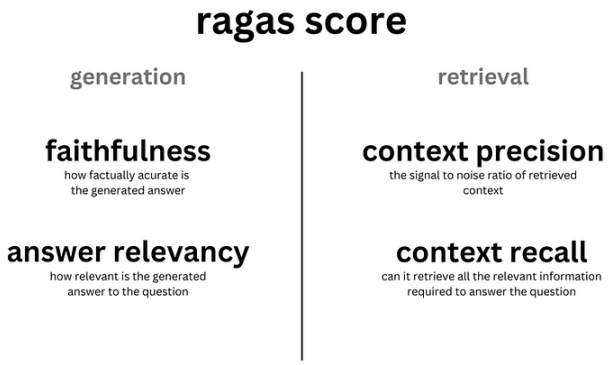
เป็นเฟรมเวิร์คที่สนใจเฉพาะเรื่องของการประเมินที่เกี่ยวข้องกับงานทางด้าน NLP ที่รวบรวมเมทริกซ์ที่สำคัญต่างๆ และเหมาะสมกับการพัฒนา NLP ในปัจจุบัน นอกจากนี้ยังรองรับการสังเคราะห์ข้อมูลเพิ่ม รวมถึงการทำ unit test

ข้อดีของเฟรมเวิร์กนี้คือ มีเมทริกซ์ที่รองรับกับงาน nlp จำพวกงานถามตอบในปัจจุบันที่ดี และเหมาะกับการประเมินที่ต้องการ llm ช่วยในการประเมิน

3.2 Langsmith

เป็นเฟรมเวิร์คที่ถูกสร้างขึ้นมาเพื่อจุดประสงค์ในการติดตาม มอนิเตอร์ผลลัพธ์ที่ได้ทำการ process ในการดำเนินการ llm ซึ่งในส่วนของ langsmith อาจไม่ได้ provide เรื่องของการประเมินมากนัก อาจจะต้องทำการ integrate ร่วมกับไลบรารี่อื่นๆ เช่น langchain หรือ deepeval ในการประเมินผลลัพธ์ แต่ข้อดีของมันคือทำให้เราสามารถ trace แต่ละ transaction ได้ว่าทำอะไรไปบ้าง

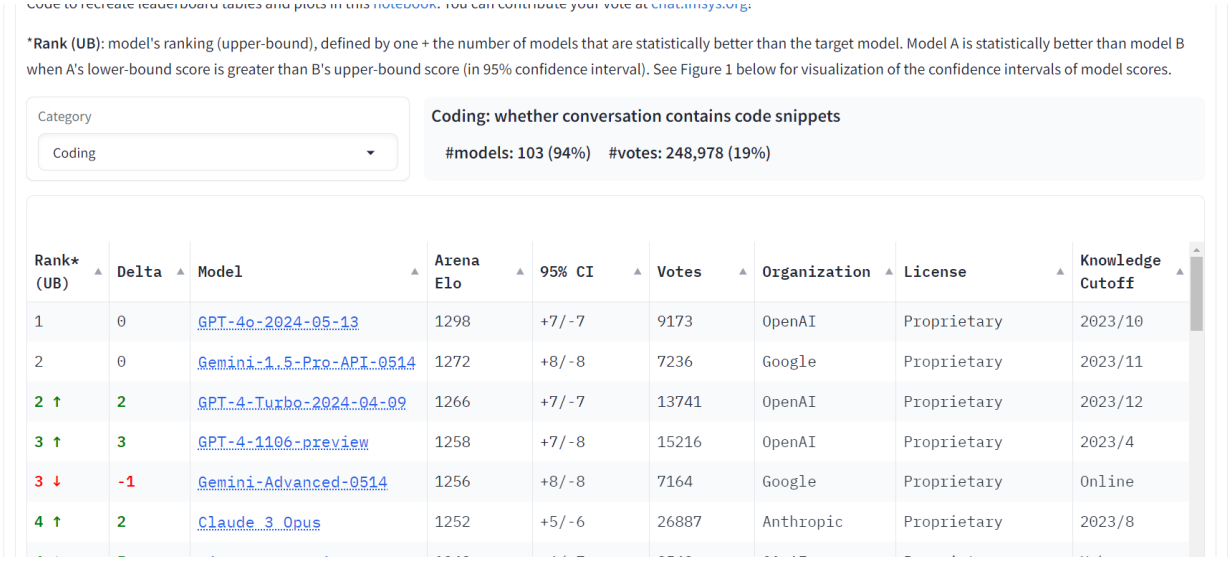
3.3 RAGAS



เป็นเฟรมเวิร์คที่มีการรวบรวมนำเอาเมทริกที่กล่าวไปแล้วก่อนหน้านี้ เช่น faithfulness, context precision, answer relevancy, context recall นำมารวมอยู่ในเฟรมเวิร์คให้อยู่ในรูปแบบ end to end evaluation เพื่อให้ง่ายในการนำไปใช้กับ RAG application

4. How to select LLM evaluation

ปัจจัยที่สำคัญอีกอย่างในกรณีที่เราต้องการใช้โมเดล LLM ในการช่วยเราประเมินผลลัพธ์ที่ได้ โมเดลที่ประเมินจำเป็นจะต้องมีประสิทธิภาพความเข้าใจในภาษาและมีความคิดหรือแนวคิดคล้ายกับมนุษย์ประเมิน เพื่อที่จะทำให้ผลลัพธ์ที่ออกมาใกล้เคียงกับการประเมินด้วยคนมากที่สุด เพราะฉะนั้นเพื่อช่วยให้การตัดสินใจ เลือกโมเดลทางภาษามาใช้ในการประเมินได้ดียิ่งขึ้น จำเป็นจะต้องดูตาราง leaderboard ที่มีการเปรียบเทียบกันในเว็บไซต์ ซึ่งแต่ละโมเดลก็อาจจะมีความเชี่ยวชาญในงานเฉพาะแต่ละงานที่ต่างกันไป เช่น เว็บ https://chat.lmsys.org/?leaderboard



*Rank (UB): model's ranking (upper-bound), defined by one + the number of models that are statistically better than the target model. Model A is statistically better than model B when A's lower-bound score is greater than B's upper-bound score (in 95% confidence interval). See Figure 1 below for visualization of the confidence intervals of model scores.

Category: Coding

Coding: whether conversation contains code snippets

#models: 103 (94%) #votes: 248,978 (19%)

Rank* (UB)	Delta	Model	Arena Elo	95% CI	Votes	Organization	License	Knowledge Cutoff
1	0	GPT-4o-2024-05-13	1298	+7/-7	9173	OpenAI	Proprietary	2023/10
2	0	Gemini-1.5-Pro-API-0514	1272	+8/-8	7236	Google	Proprietary	2023/11
2 ↑	2	GPT-4-Turbo-2024-04-09	1266	+7/-7	13741	OpenAI	Proprietary	2023/12
3 ↑	3	GPT-4-1106-preview	1258	+7/-8	15216	OpenAI	Proprietary	2023/4
3 ↓	-1	Gemini-Advanced-0514	1256	+8/-8	7164	Google	Proprietary	Online
4 ↑	2	Claude 3 Opus	1252	+5/-6	26887	Anthropic	Proprietary	2023/8

Reference :

- https://medium.com/@vipra_singh/building-llm-applications-evaluation-part-8-fcfa2f22bd1c
- https://www.confident-ai.com/blog/llm-evaluation-metrics-everything-you-need-for-llm-evaluation
- https://klu.ai/glossary/llm-evaluation