Introduction to Large Language Models

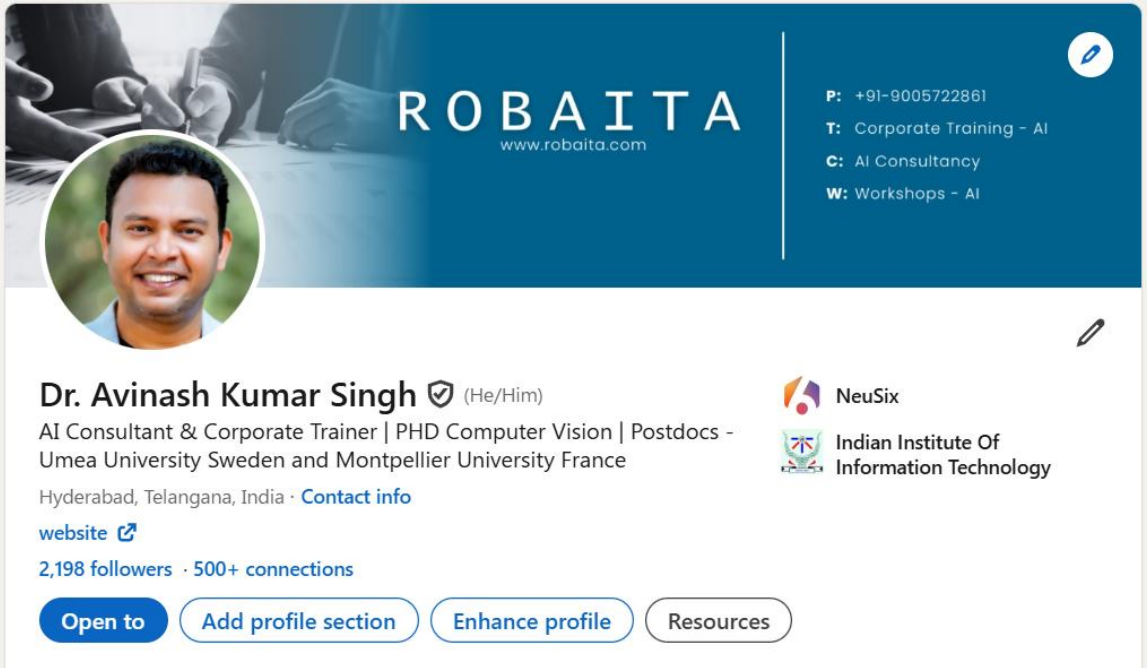
Journey and Evaluation Parameters

Dr. Avinash Kumar Singh

AI Consultant and Coach, Robaita

Dr. Avinash Kumar Singh

- ❑ **Possess** 15+ years of **hands-on expertise** in Machine Learning, Computer Vision, NLP, IoT, Robotics, and Generative AI.
- ❑ **Founded** Robaita—an initiative **empowering** individuals and organizations to **build, educate, and implement** AI solutions.
- ❑ **Earned** a Ph.D. in Human-Robot Interaction from IIIT Allahabad in 2016.
- ❑ **Received** postdoctoral fellowships at Umeå University, Sweden (2020) and Montpellier University, France (2021).
- ❑ **Authored** 30+ research papers in **high-impact** SCI journals and international conferences.
- ❑ Unlearning, learning, making mistakes …

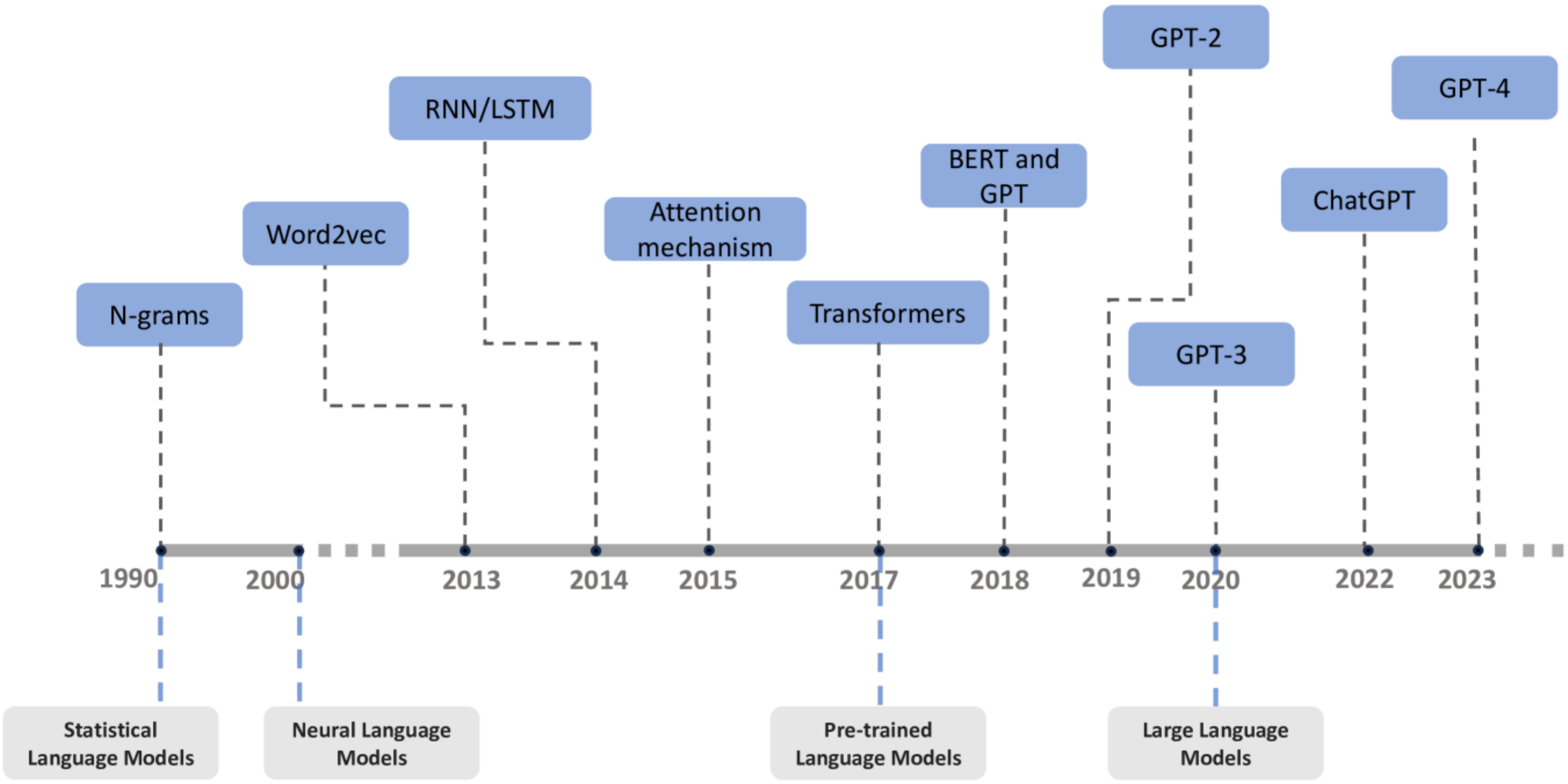


https://www.linkedin.com/in/dr-avinash-kumar-singh-2a570a31/

Things to be discussed

- What Are Large Language Models and How Do They Work?
- Popular LLMs and Benchmarks
- Evaluation Metrices:
    - Accuracy
    - Precision
    - Recall
    - F1-score
    - Confusion matrix interpretation
    - BLEU Score
    - ROUGE for language models, Perplexity (LLM-specific)
- Prompt Engineering

Language Models Journey



History, Development, and Principles of Large Language Models—An Introductory Survey, https://arxiv.org/html/2402.06853v1

Large Language Models

What is a Language Model

A language model is a probabilistic model that assigns a probability to a sequence of words and predicts the likelihood of the next word in a sentence, given the previous words.

Statistical Language Model (SLM):

A language model estimates the probability distribution over sequences of words. Given a sequence of words $w_1, w_2, w_3, \ldots, w_n$, a language model computes:

$$P(w_1, w_2, \ldots, w_n) = \prod_{i=1}^{n} P(w_i \mid w_1, \ldots, w_{i-1})$$

To calculate the probability of a sentence (or sequence of words),we multiply the conditional probability of each word given the previous words.

Chapter 3: N-gram Language Models , Jurafsky, D., & Martin, J. H. (2023). *Speech and Language Processing* (3rd ed. draft). Stanford University.
https://web.stanford.edu/~jurafsky/slp3/

Large Language Models

Predict the next word

I want to drink a hot cup of ____

Training Corpus

1. I want to drink a hot cup of coffee
2. Every morning, I drink a hot cup of coffee before work
3. He prefers a hot cup of tea in the evening
4. She needs a hot cup of coffee to wake up
5. After dinner, they drank a hot cup of tea
6. I always start my day with a hot cup of black coffee
7. On cold days, people enjoy a hot cup of cocoa
8. I want to drink a hot cup of coffee quickly
9. They like to have a hot cup of herbal tea after yoga
10. I usually order a hot cup of coffee at Starbucks

Large Language Models

Predict the next word

I want to drink a hot cup of ____

Training Corpus

1. I want to drink a hot cup of coffee
2. Every morning, I drink a hot cup of coffee before work
3. He prefers a hot cup of tea in the evening
4. She needs a hot cup of coffee to wake up
5. After dinner, they drank a hot cup of tea
6. I always start my day with a hot cup of black coffee
7. On cold days, people enjoy a hot cup of cocoa
8. I want to drink a hot cup of coffee quickly
9. They like to have a hot cup of milk after yoga
10. I usually order a hot cup of coffee at Starbucks

Derivation

1. Expression to find the next word

$$P(w_1, \ldots, w_9) \approx \prod_{i=1}^{9} P(w_i \mid w_{i-2}, w_{i-1})$$

2. Chain rule of probability

$$P(w_1) \cdot P(w_2 \mid w_1) \cdot P(w_3 \mid w_1, w_2) \cdot P(w_4 \mid w_2, w_3) \cdot \cdots \cdot P(w_9 \mid w_7, w_8)$$

3. If we assume the next word is coffee

$$P(\text{"I want to drink a hot cup of coffee"}) = P(w_1, w_2, \ldots, w_9)$$

$$= P(w_1) \cdot P(w_2 \mid w_1) \cdot P(w_3 \mid w_1, w_2) \cdot \cdots \cdot P(w_9 \mid w_1, \ldots, w_8)$$

Phrase	Count	Probability
cup of coffee	5	= 5/10 (0.5)
cup of tea	2	= 2/10 (0.2)
cup of milk,	1	= 1/10 (0.1)
Total ("cup of X")	10	

4. The predicted word would be coffee

The Issues with Statistical Model

History, Development, and Principles of Large Language Models—An Introductory Survey

Zhibo Chu

Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China

University of Science and Technology of China, Hefei, China

Shiwen Ni

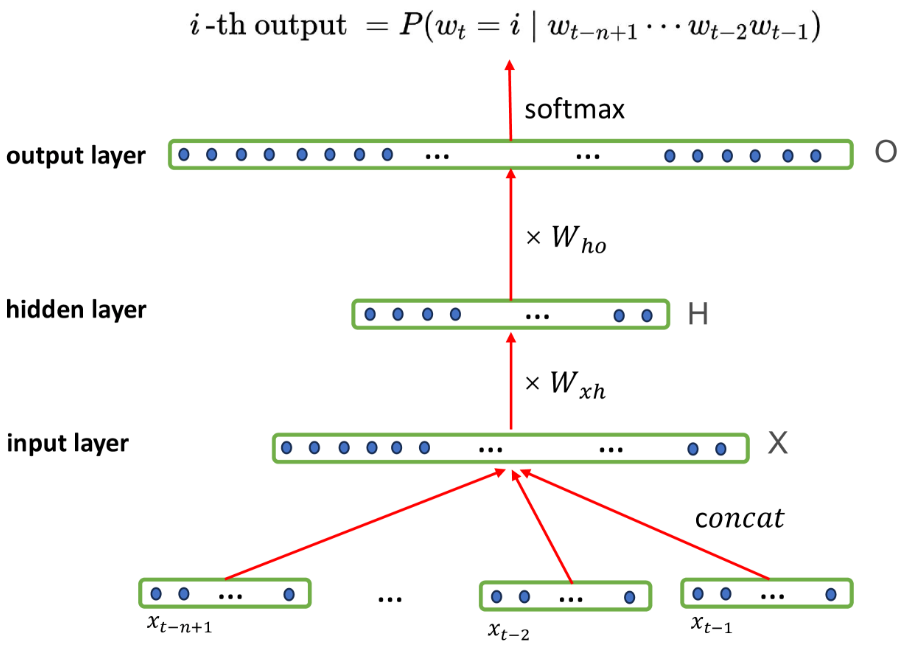
corresponding author Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China

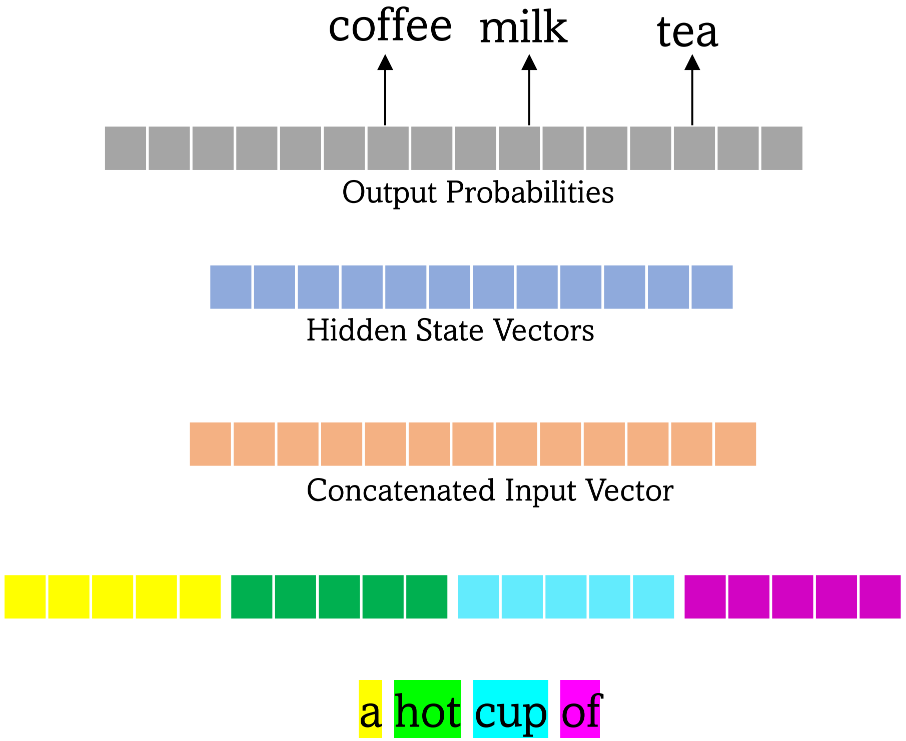
conditional probabilities, it is necessary to pre-compute and save $C(X)$required for the conditional probability computation, where X is a sentence of length n. The number of possible sentences X grows exponentially with the size of the vocabulary. For instance, with 1000 different words, there exist 1000^n potential sequences of length n. However, excessively large values of n pose storage limitations. Typically, n is confined to 2 or 3, causing each word to relate to only its first 1 or 2 preceding words, ultimately leading to a reduction in the model's accuracy.

Large Language Models

Neural Language Models

Neural Language Models: NLMs (Bengio et al., 2000; Mikolov et al., 2010; Kombrink et al., 2011) leverage neural networks to predict the probabilities of subsequent words within sequences. They effectively handle longer sequences and mitigate the limitations associated with small n in SLMs. Before delving into neural networks, let's grasp the concept of

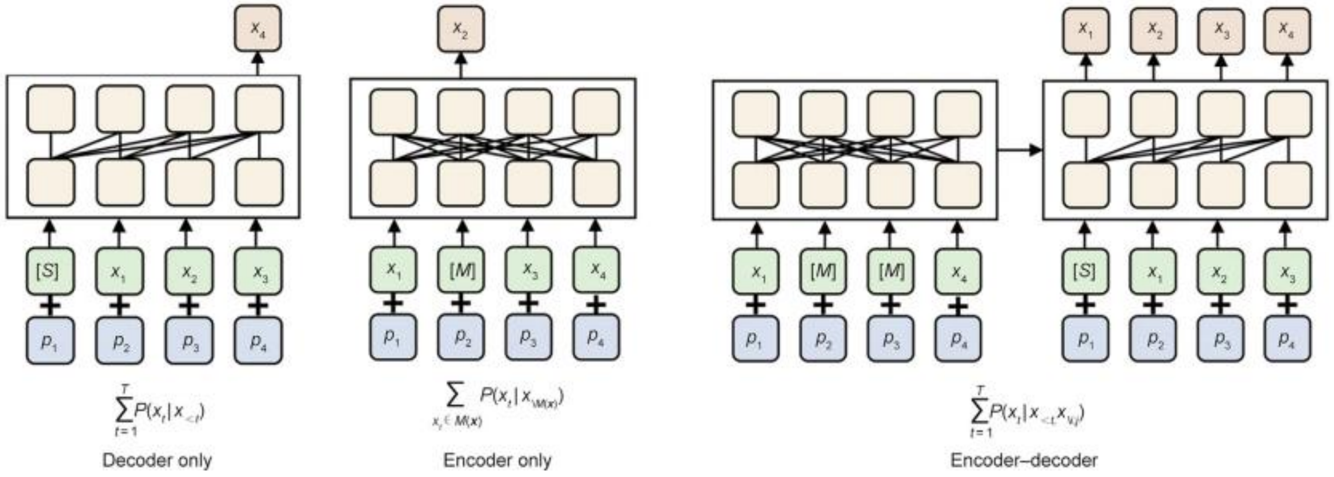




Large Language Models

Large Language Model

Pre-trained Language Model: PLMs undergo initial training using an extensive volume of unlabeled text, enabling them to grasp fundamental language structures such as vocabulary, syntax, semantics, and logic — a phase termed pre-training. Subsequently, this comprehensive language model can be applied to various NLP tasks like machine translation, text summarization, and question-answering systems. To optimize its performance, models need to be trained a second time on a smaller dataset customized for a specific downstream task — a phase known as fine-tuning. This is the "pre-training and fine-tuning" learning paradigm. We can use a visual example to understand the "pre-training and fine-tuning", as follows: in



GPT

BERT

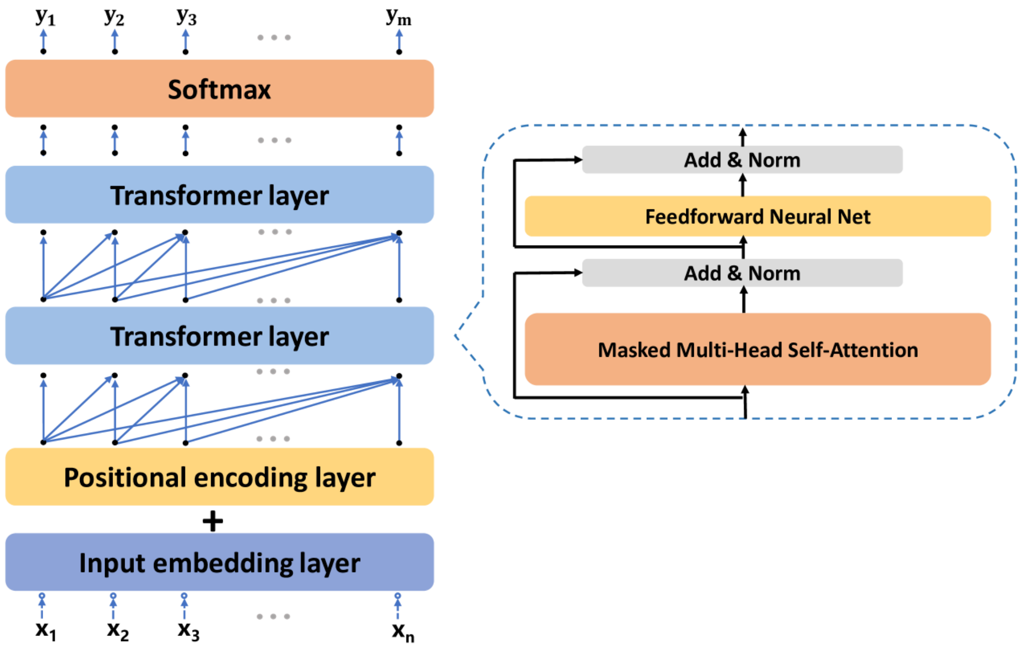
Bidirectional and Auto-Regressive Transformers (BART)

Wang, H., Li, J., Wu, H., Hovy, E., & Sun, Y. (2023). Pre-trained language models and their applications. *Engineering*, *25*, 51-65.

Large Language Models

"A Large Language Model is a transformer-based neural network trained to model the probability distribution over sequences of words or tokens, enabling tasks such as text generation, summarization, translation, and question answering."

- **Bommasani et al., 2021**, *On the Opportunities and Risks of Foundation Models*



Examples: GPT-4, BERT

LLaMA, Claude, Gemini, Mistral

Large Language Models

LLM Datasets

Dataset Name	Size	Dataset Information	Languages	URL
Common Crawl	Petabyte-scale	Web pages, blogs, news articles, forums	100+	https://commoncrawl.org
The Pile	825 GB	Academic papers, books, GitHub, StackExchange, Wikipedia, PubMed, etc.	Primarily English	https://pile.eleuther.ai
Wikipedia	~20 GB (English)	Encyclopedic articles	300+	https://dumps.wikimedia.org
OpenWebText2	~40 GB	High-quality content from web links in Reddit	Primarily English	https://github.com/EleutherAI/openwebtext2
RedPajama	~1.2 TB	Common Crawl, C4, Books, GitHub, Wikipedia, StackExchange	Primarily English	https://www.together.xyz/blog/redpajama
The Stack	3.1 TB	Source code from GitHub in 30+ programming languages	30+ (programming languages)	https://huggingface.co/datasets/bigcode/the-stack
arXiv + PubMed	10+ GB	Scientific papers in physics, math, medicine, biology	Primarily English	https://pubmed.ncbi.nlm.nih.gov/download https://www.kaggle.com/datasets/Cornell-University/arxiv

Large Language Models Evaluation

Total Cats = 6 +1

Total Dogs = 2 + 12

True Positive & True Negative: When Actual and Predicted values are same.

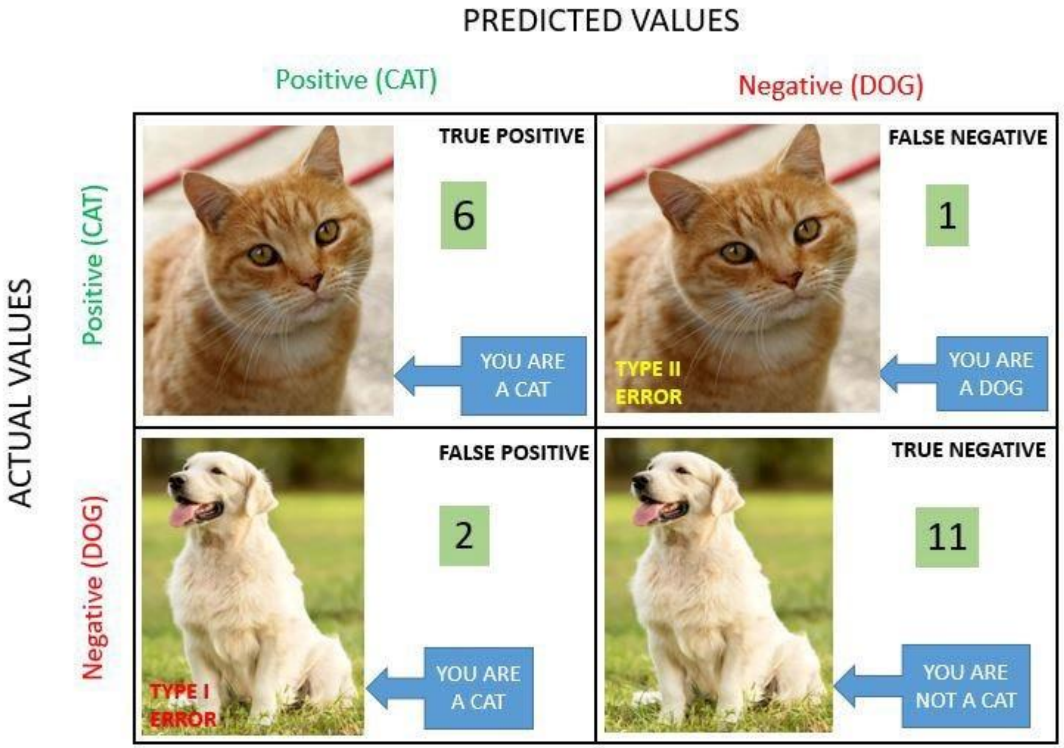
TP = 6, TN = 11

False Positive: When the actual Value was negative “dog” and the system predicted positive “cat “.

FP = 2

False Negative: When the actual value was “cat” and the system is predicted it “dog”

FN = 1



Confusion Matrix

Large Language Models Evaluation

TP = 6, TN = 11

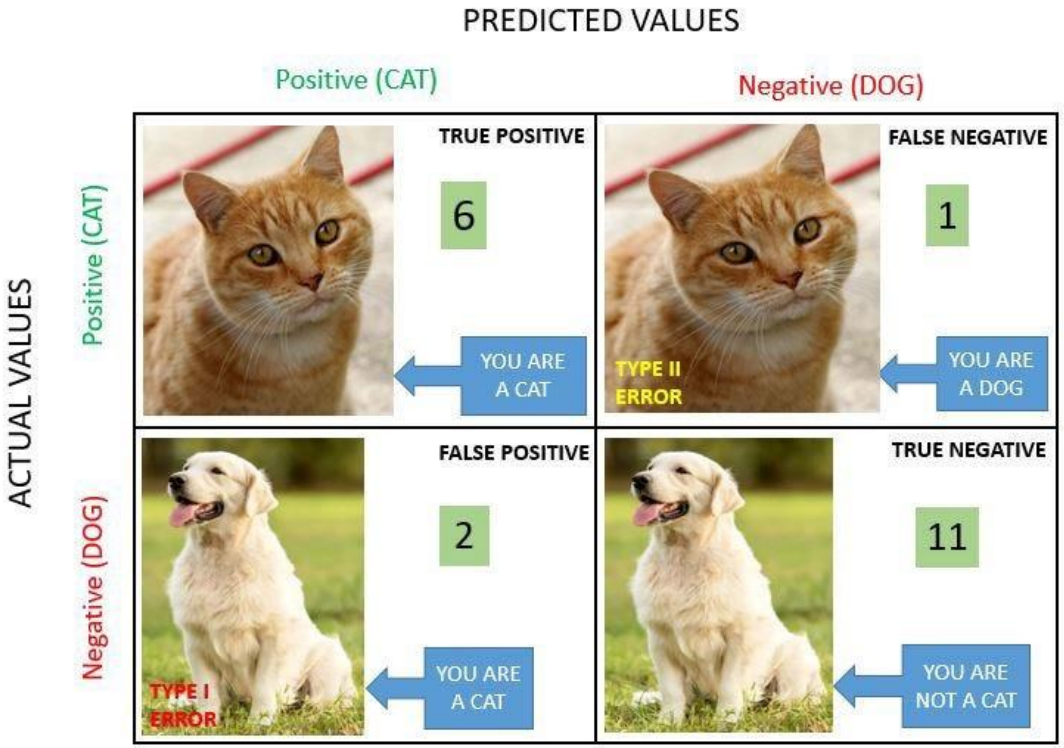
FP = 2, FN = 1

$$Accuracy = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + FP + FN}$$

$$Precision = \frac{\text{TP}}{\text{TP} + FP}$$

$$Recall = \frac{\text{TP}}{\text{TP} + FN}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$



Confusion Matrix

Large Language Models Evaluation

BLEU (Bilingual Evaluation Understudy) – *compares n-gram overlaps between prediction and reference*

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

Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002).

Let's calculate it for bigram

$$\text{BLEU-2} = \text{BP} \cdot \exp\left(\frac{1}{2}(\log p_1 + \log p_2)\right)$$

Actual: The cat is on the mat

Unigram: the, cat, is, on, the, mat

Bigram: the cat, cat is, is on, on the, the mat

Predicted: The cat sat on the mat

$$p1 = \frac{5}{6}$$

Unigram: the, cat, sat, on, the, mat

Bigram: the cat, cat sat, sat on, on the, the mat

$$p2 = \frac{3}{6}$$

$$BLEU - 2 = 1 * e^{\left(\frac{1}{2}\left(log\frac{5}{6} + log\frac{3}{6}\right)\right)}$$

$$\boxed{0.645}$$

BP stands for **Brevity Penalty**. It is used to penalize machine-generated text that is **too short** compared to the reference

BLEU score is precision-oriented (counts how many n-grams match), but without a length penalty, a model could **cheat** by just outputting short sequences.

BP solves this by lowering the BLEU score when the generated output is shorter than the reference.

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-\frac{r}{c})} & \text{if } c \leq r \end{cases}$$

Large Language Models Evaluation

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) Score

Lin, C.-Y. (2004).
ROUGE: A Package for Automatic Evaluation of Summaries

ROUGE (Recall-Oriented) is used in summarization. The most common are:

- **ROUGE-1**: Overlap of unigrams

$$\textbf{ROUGE-1(Precision)} = 5/6,\ \textbf{ROUGE-1(Recall)} = 5/6,\ \textbf{ROUGE-1(F1)} = \frac{2 * \frac{5}{6} * \frac{5}{6}}{\frac{5}{6} + \frac{5}{6}}$$

- **ROUGE-2**: Overlap of bigrams

$$\textbf{ROUGE-2(Precision)} = 3/6,\ \textbf{ROUGE-1(Recall)} = 3/6,\ \textbf{ROUGE-1(F1)} = \frac{2 * \frac{3}{6} * \frac{3}{6}}{\frac{3}{6} + \frac{3}{6}}$$

- **ROUGE-L**: Longest Common Subsequence (LCS)

  Longest Sequence (5) = The cat [mismatch/gap] on the mat

$$\textbf{ROUGE-L(Precision)} = 5/6,\ \textbf{ROUGE-1(Recall)} = 5/6,\ \textbf{ROUGE-1(F1)} = \frac{2 * \frac{5}{6} * \frac{5}{6}}{\frac{5}{6} + \frac{5}{6}}$$

Some Benchmarks

Model	MMLU	HumanEval	GSM8K	TruthfulQA
GPT-4	86.40%	88.00%	94.00%	59.00%
Claude 2	81.60%	71.00%	88.00%	58.00%
Gemini 1.5 Pro	84.00%	83.00%	92.00%	62.00%
Claude 3 Opus	88.70%	90.00%	95.00%	68.00%
GPT-3.5	70.00%	48.10%	57.10%	47.00%
LLaMA 2 70B	79.00%	67.00%	83.00%	52.00%
Mixtral 8x7B	84.10%	74.00%	87.00%	58.50%
Mistral 7B	70.00%	55.00%	65.00%	47.00%
Command R+	75.20%	60.50%	78.00%	53.10%
Gemma 7B	65.00%	45.00%	58.00%	41.00%

MMLU (Massive Multitask Language Understanding)
- **What it tests:** Knowledge and reasoning across 57 academic subjects like history, law, math, medicine, etc.
- **Use case:** Checks how well a model performs on real-world, high school to graduate-level exams.

HumanEval
- **What it tests:** Code generation and reasoning.
- **Use case:** Given a prompt (like a function definition), the model needs to generate correct Python code that passes test cases.

GSM8K (Grade School Math 8K)
- **What it tests:** Basic arithmetic and word problem-solving.
- **Use case:** Models solve grade-school level math problems using step-by-step reasoning.

TruthfulQA
- **What it tests:** The ability to give **truthful** answers, especially in tricky or misleading questions.
- **Use case:** The model is asked questions where giving a common but false answer is easy (e.g., urban myths).

Thanks for your time