LLMs for Low Resource Languages in Multilingual, Multimodal and Dialectal Settings

https://llm-low-resource-lang.github.io

EACL 2024, 21th March, 2024

QCRI
معهد قطر لبحوث الحوسبة
Qatar Computing Research Institute
جامعة حمد بن خليفة
HAMAD BIN KHALIFA UNIVERSITY

Speakers

Firoj Alam
Scientist

Shammur Chowdhury
Scientist

Sabri Boughorbel
Scientist

Maram Hasanain
Post Doctoral
Researcher

Qatar Computing Research Institute

Content

- Introduction **[20 mins]**
- Models and their capabilities for low-resource languages **[70 mins]**
  - NLP models [40 mins]
  - Multimodality [25 mins]
    - Overview
      - Multimodality
      - Speech
  - QA [5 mins]
- Coffee break **[30 mins]**
- Prompting + Benchmarking Tool **[60 mins]**
  - Prompt Engineering [40 mins]
    - Prompting techniques
    - Cross-/multi-lingual prompting
  - Prompt and Benchmarking tools [15 mins]
  - QA: [5 mins]
- Other Related Aspects [**20 mins**]

Introduction

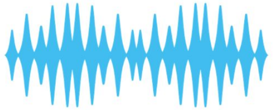
Introduction

Low Resources Languages

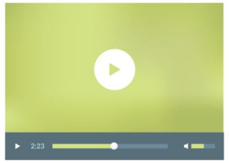
- Approximately ~7,000 languages
- Majority of the internet content are in English
- Mostly categorized as lack of
  - labeled/annotated datasets
  - unlabelled datasets

Text

Image



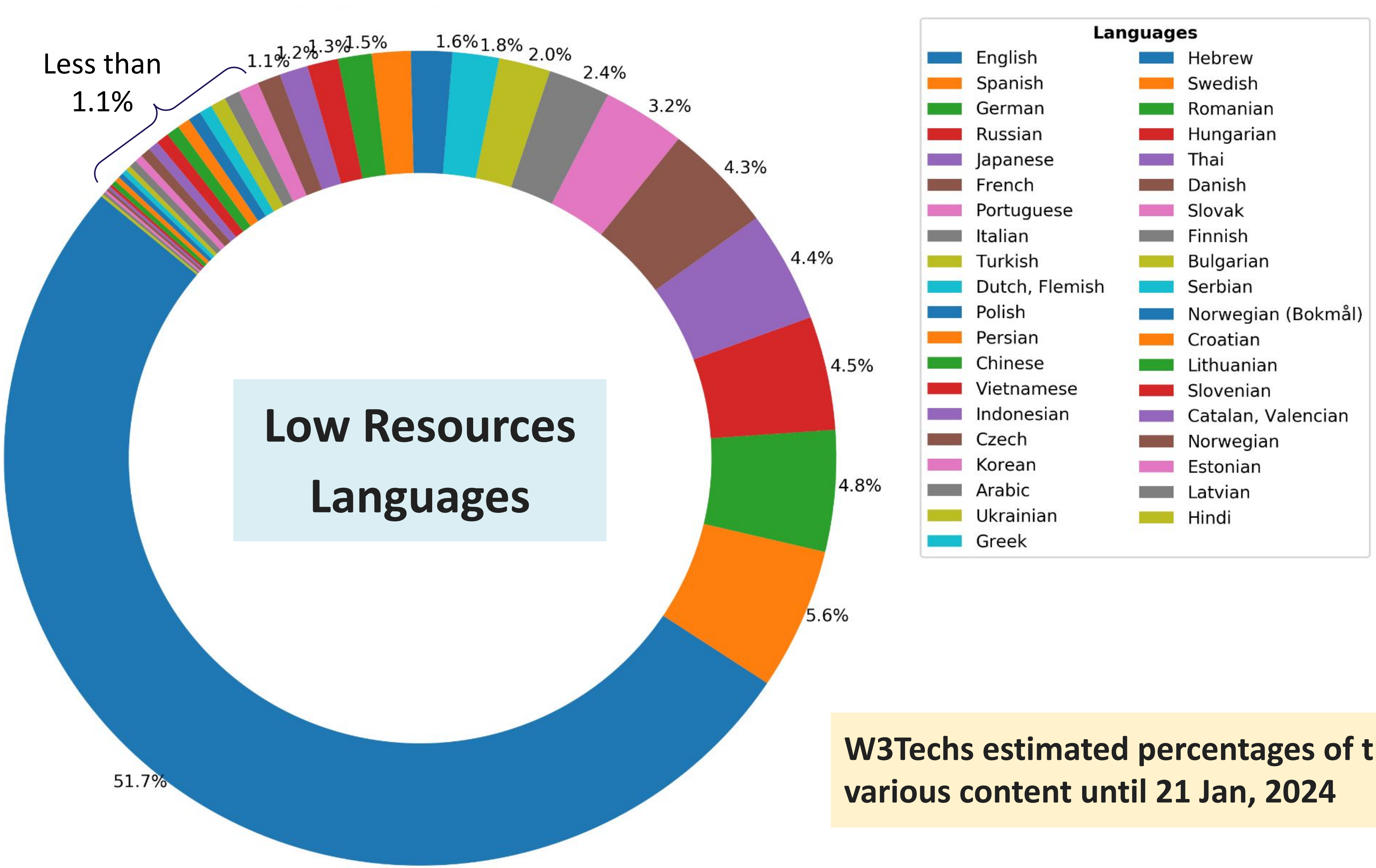
Speech



Video

Image: DALL-E

Introduction



Low Resources Languages

W3Techs estimated percentages of the various content until 21 Jan, 2024

Introduction

Low Resources Languages: Categorization

0- The Left-Behinds (exceptionally limited resources: impossible effort to lift them up in the digital space)

1- The Scraping-Bys (some amount of unlabeled data)

2- The Hopefuls (small set of labeled datasets)

3- The Rising Stars (strong web presence, a thriving cultural community online)

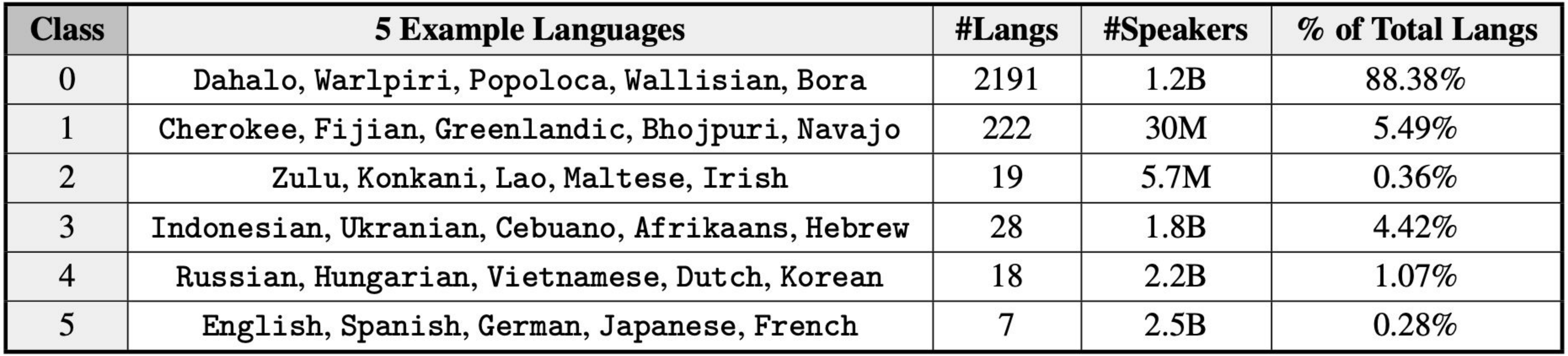
4- The Underdogs (serious amounts of resource, a large amount of unlabeled data, dedicated NLP communities)

5- The Winners (dominant online presence, massive effort to develop resources and technologies)

The State and Fate of Linguistic Diversity and Inclusion in the NLP World (Joshi et al. ACL 2020)

Introduction

Low Resources Languages: Categorization



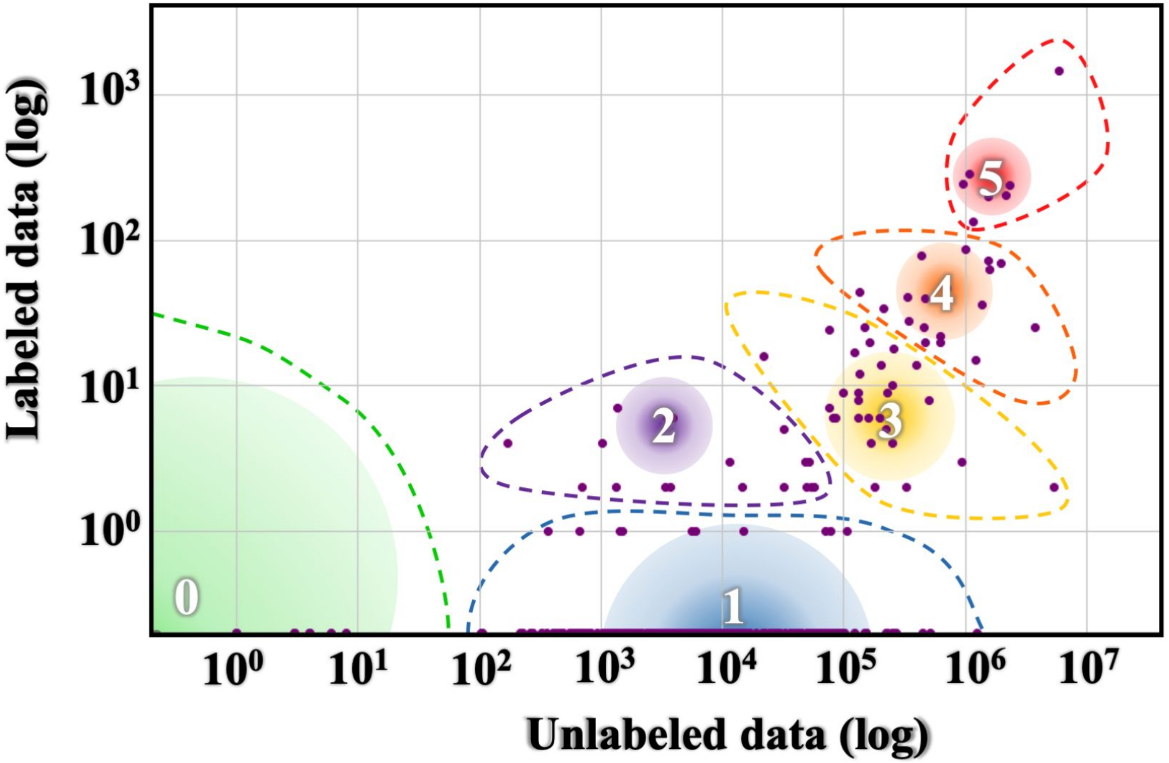
Class	5 Example Languages	#Langs	#Speakers	% of Total Langs
0	Dahalo, Warlpiri, Popoloca, Wallisian, Bora	2191	1.2B	88.38%
1	Cherokee, Fijian, Greenlandic, Bhojpuri, Navajo	222	30M	5.49%
2	Zulu, Konkani, Lao, Maltese, Irish	19	5.7M	0.36%
3	Indonesian, Ukranian, Cebuano, Afrikaans, Hebrew	28	1.8B	4.42%
4	Russian, Hungarian, Vietnamese, Dutch, Korean	18	2.2B	1.07%
5	English, Spanish, German, Japanese, French	7	2.5B	0.28%

Number of languages, number of speakers, and percentage of total languages for each language class.

The State and Fate of Linguistic Diversity and Inclusion in the NLP World (Joshi et al. ACL 2020)

Introduction

Low Resources Languages: Categorization



Language Resource Distribution: The size of the gradient circle represents the number of languages in the class. The color spectrum VIBGYOR, represents the total speaker population size from low to high.

The State and Fate of Linguistic Diversity and Inclusion in the NLP World (Joshi et al. ACL 2020)

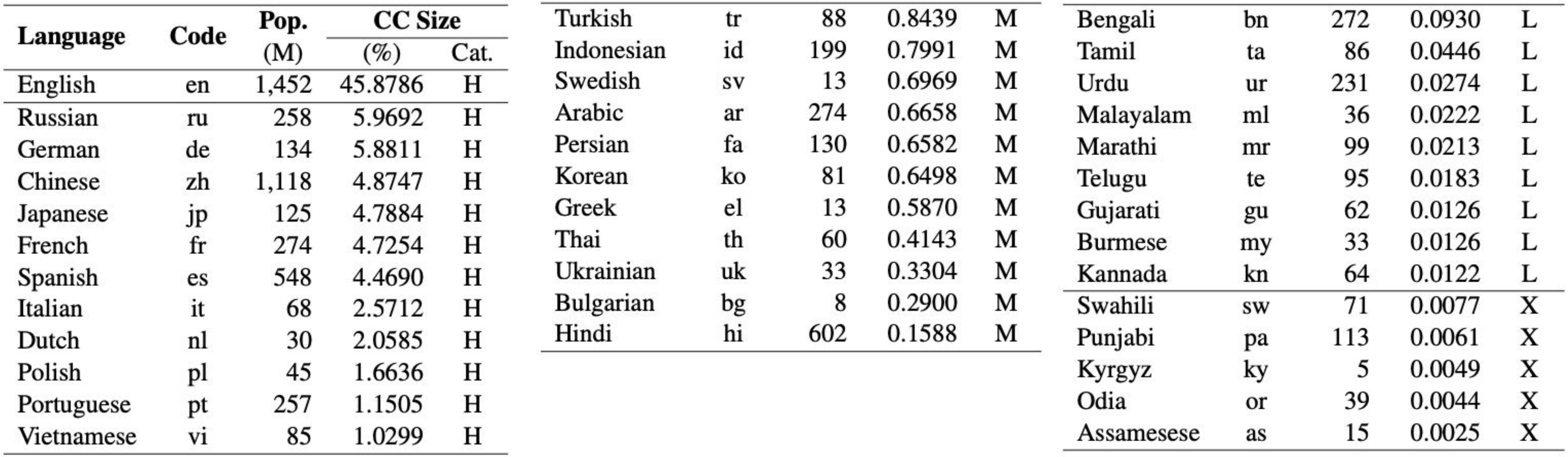
Introduction

Low Resources Languages: Categorization

High (H, > 1%)

Medium (M, > 0.1%)

Low (L, > 0.01%),
Extremely-Low (X, < 0.01%)



Language	Code	Pop. (M)	CC Size (%)	Cat.
English	en	1,452	45.8786	H
Russian	ru	258	5.9692	H
German	de	134	5.8811	H
Chinese	zh	1,118	4.8747	H
Japanese	jp	125	4.7884	H
French	fr	274	4.7254	H
Spanish	es	548	4.4690	H
Italian	it	68	2.5712	H
Dutch	nl	30	2.0585	H
Polish	pl	45	1.6636	H
Portuguese	pt	257	1.1505	H
Vietnamese	vi	85	1.0299	H
Turkish	tr	88	0.8439	M
Indonesian	id	199	0.7991	M
Swedish	sv	13	0.6969	M
Arabic	ar	274	0.6658	M
Persian	fa	130	0.6582	M
Korean	ko	81	0.6498	M
Greek	el	13	0.5870	M
Thai	th	60	0.4143	M
Ukrainian	uk	33	0.3304	M
Bulgarian	bg	8	0.2900	M
Hindi	hi	602	0.1588	M
Bengali	bn	272	0.0930	L
Tamil	ta	86	0.0446	L
Urdu	ur	231	0.0274	L
Malayalam	ml	36	0.0222	L
Marathi	mr	99	0.0213	L
Telugu	te	95	0.0183	L
Gujarati	gu	62	0.0126	L
Burmese	my	33	0.0126	L
Kannada	kn	64	0.0122	L
Swahili	sw	71	0.0077	X
Punjabi	pa	113	0.0061	X
Kyrgyz	ky	5	0.0049	X
Odia	or	39	0.0044	X
Assamesese	as	15	0.0025	X

Languages, language codes, numbers of speakers (first and second), data ratios in the CommonCrawl corpus and language categories.

ChatGPT Beyond English: Towards a Comprehensive Evaluation of Large Language Models in Multilingual Learning (Lai et al. EMNLP 2023)

Different Era of NLP

Task-specific architectures

No more task-specific architectures

Lots of task specific data

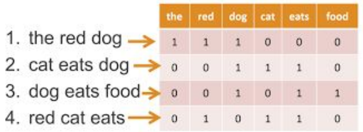
Small amount of task specific data

Few examples

Neural Language Model (2003)

Multitask learning for NLP tasks (2008)

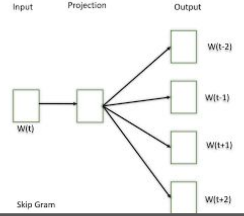
2010



	the	red	dog	cat	eats	food
1. the red dog →	1	1	1	0	0	0
2. cat eats dog →	0	0	1	1	1	0
3. dog eats food →	0	0	1	0	1	1
4. red cat eats →	0	1	0	1	1	0

Bag of words
SVM, RF, Logistic Regression, CRF etc..

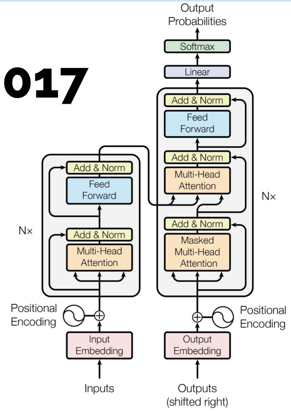
Word Embedding
Deep Learning:
LSTM, CNN



2013

2014
Sequence to sequence learning

2017



Transformer
Attention methods

Pretraining,
Fine-tuning
BERT, RoBERTa, GPT,
…

GPT-3
Few-shot learners

2020

2022

Large Language Model + Prompting

ChatGPT
GPT-3.5

GPT-4
Llama 2,
Falcon,
Palm2

Bloom, Palm, Llama,

Feature Engineering

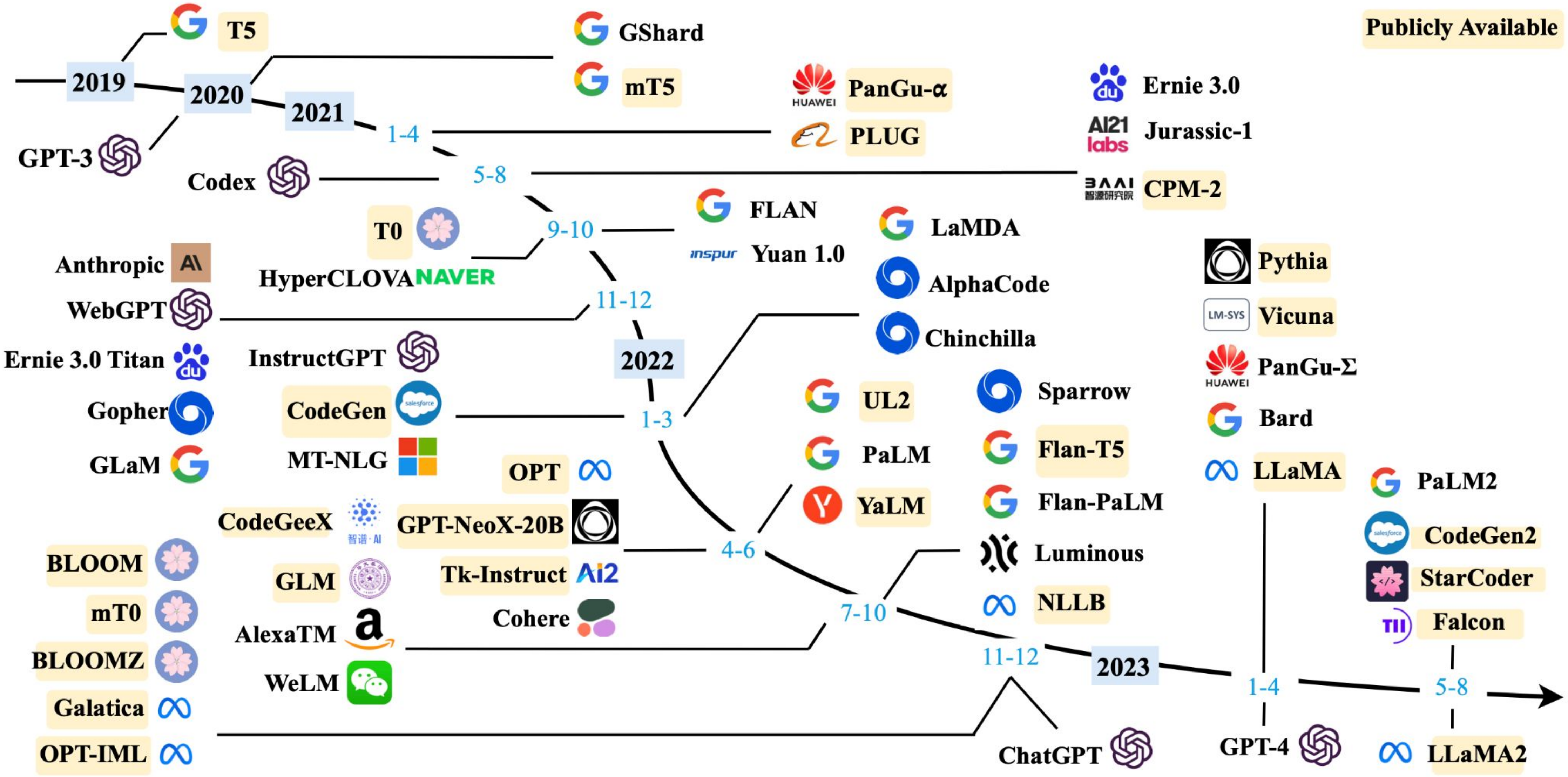
Network Architectures Engineering

Prompt Engineering

Objective Engineering

Machine learning

Recent Models



A Survey of Large Language Models (Zhao et al. 2023)

LLMs Capabilities

LLMs can seemingly "solve" most problems

World Knowledge
- MMLU (57 subcategories/ subjects)
- Natural Questions

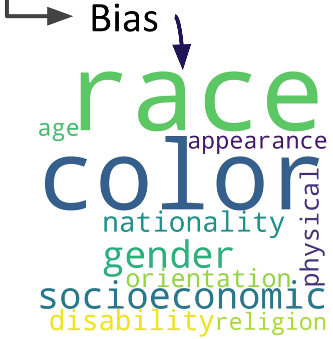
…….

Common Sense Reasoning, Morality
- Grounded Situations (HellaSwag)
- Ethics (Morality): Justice, well being,
- Physical common sense (PIQA)
- Information Seeking (SituatedQA)
- Elementary school science facts (OpenBookQA)
- Grad-school science questions
- Natural science questions

Reading Comprehension

Belebele dataset
Multilingual

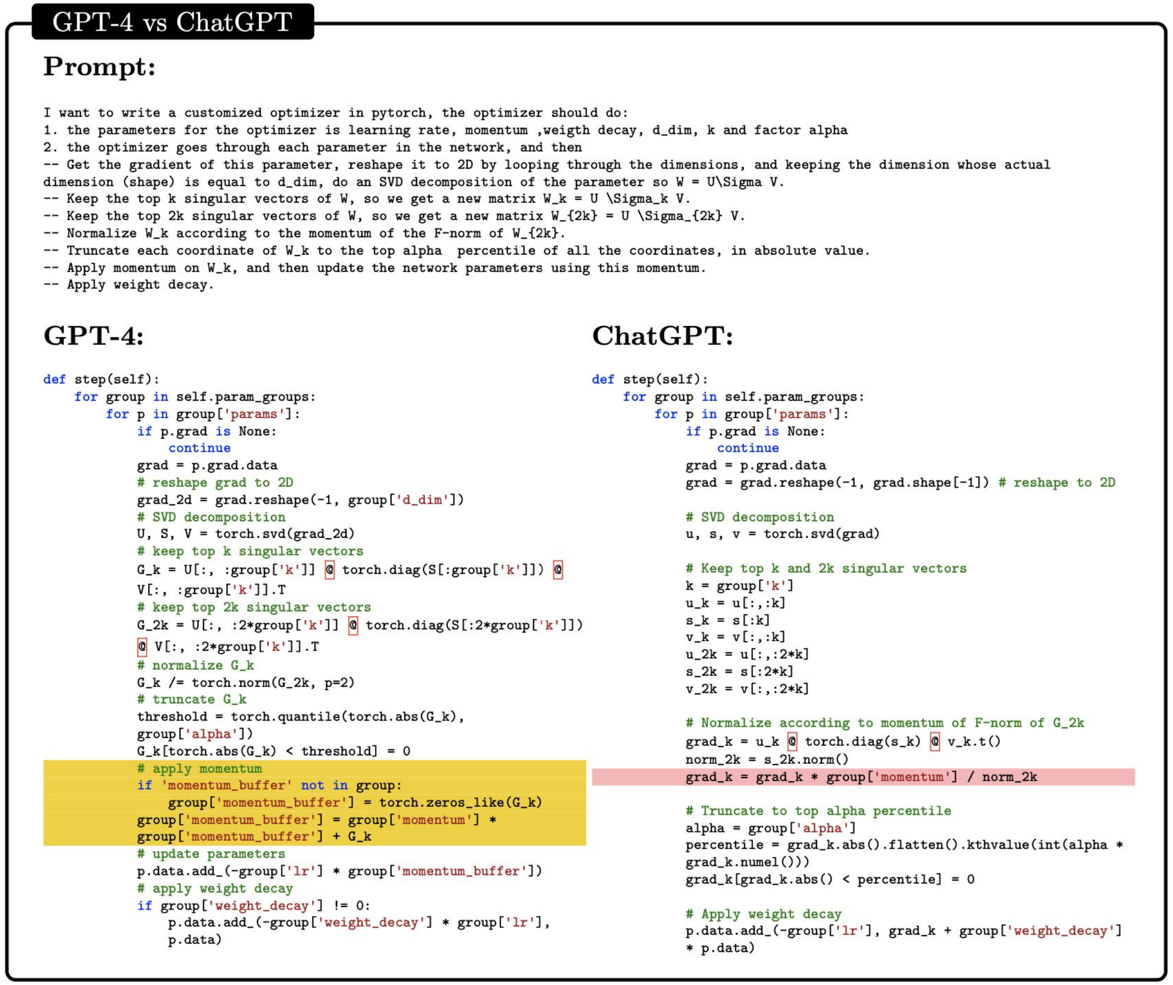
Misinformation, Factuality and Bias
- Factuality (TruthfulQA)
- Stereotype
- Bias



Standard NLP Tasks
- Summarization
- Sarcasm
- Offensive
- Dialect Identification
- Natural Language Inference
- Question Answering

LLMs Capabilities: Example

Programming: Deep Learning



GPT-4 vs ChatGPT

Prompt:

```
I want to write a customized optimizer in pytorch, the optimizer should do:
1. the parameters for the optimizer is learning rate, momentum ,weigth decay, d_dim, k and factor alpha
2. the optimizer goes through each parameter in the network, and then
-- Get the gradient of this parameter, reshape it to 2D by looping through the dimensions, and keeping the dimension whose actual
dimension (shape) is equal to d_dim, do an SVD decomposition of the parameter so W = U\Sigma V.
-- Keep the top k singular vectors of W, so we get a new matrix W_k = U \Sigma_k V.
-- Keep the top 2k singular vectors of W, so we get a new matrix W_{2k} = U \Sigma_{2k} V.
-- Normalize W_k according to the momentum of the F-norm of W_{2k}.
-- Truncate each coordinate of W_k to the top alpha  percentile of all the coordinates, in absolute value.
-- Apply momentum on W_k, and then update the network parameters using this momentum.
-- Apply weight decay.
```

GPT-4:

```
def step(self):
    for group in self.param_groups:
        for p in group['params']:
            if p.grad is None:
                continue
            grad = p.grad.data
            # reshape grad to 2D
            grad_2d = grad.reshape(-1, group['d_dim'])
            # SVD decomposition
            U, S, V = torch.svd(grad_2d)
            # keep top k singular vectors
            G_k = U[:, :group['k']] @ torch.diag(S[:group['k']]) @
            V[:, :group['k']].T
            # keep top 2k singular vectors
            G_2k = U[:, :2*group['k']] @ torch.diag(S[:2*group['k']])
            @ V[:, :2*group['k']].T
            # normalize G_k
            G_k /= torch.norm(G_2k, p=2)
            # truncate G_k
            threshold = torch.quantile(torch.abs(G_k),
            group['alpha'])
            G_k[torch.abs(G_k) < threshold] = 0
            # apply momentum
            if 'momentum_buffer' not in group:
                group['momentum_buffer'] = torch.zeros_like(G_k)
            group['momentum_buffer'] = group['momentum'] *
            group['momentum_buffer'] + G_k
            # update parameters
            p.data.add_(-group['lr'] * group['momentum_buffer'])
            # apply weight decay
            if group['weight_decay'] != 0:
                p.data.add_(-group['weight_decay'] * group['lr'],
                p.data)
```

ChatGPT:

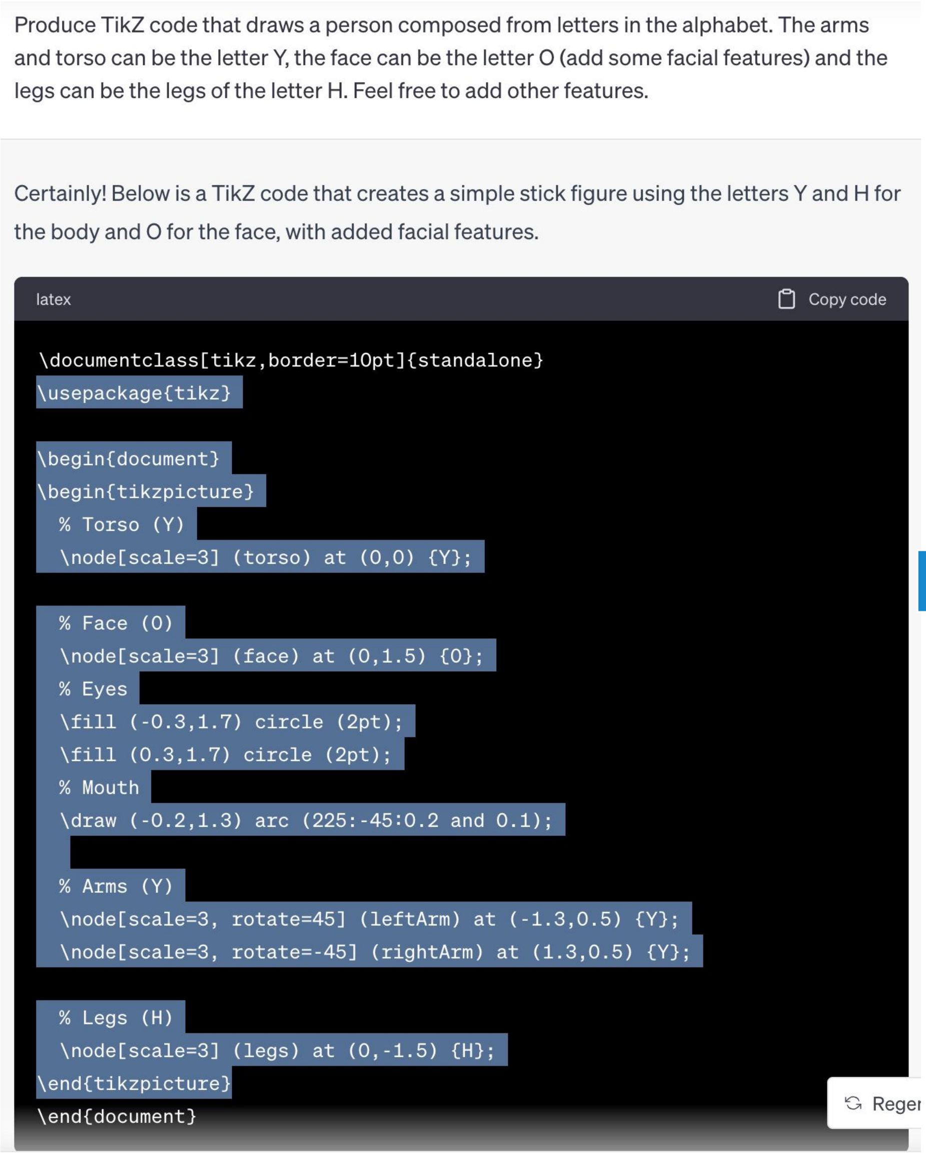
```
def step(self):
    for group in self.param_groups:
        for p in group['params']:
            if p.grad is None:
                continue
            grad = p.grad.data
            grad = grad.reshape(-1, grad.shape[-1]) # reshape to 2D

            # SVD decomposition
            u, s, v = torch.svd(grad)

            # Keep top k and 2k singular vectors
            k = group['k']
            u_k = u[:,:k]
            s_k = s[:k]
            v_k = v[:,:k]
            u_2k = u[:,:2*k]
            s_2k = s[:2*k]
            v_2k = v[:,:2*k]

            # Normalize according to momentum of F-norm of G_2k
            grad_k = u_k @ torch.diag(s_k) @ v_k.t()
            norm_2k = s_2k.norm()
            grad_k = grad_k * group['momentum'] / norm_2k

            # Truncate to top alpha percentile
            alpha = group['alpha']
            percentile = grad_k.abs().flatten().kthvalue(int(alpha *
            grad_k.numel()))
            grad_k[grad_k.abs() < percentile] = 0

            # Apply weight decay
            p.data.add_(-group['lr'], grad_k + group['weight_decay']
            * p.data)
```

GPT-4 vs ChatGPT at implementing a **customized Pytorch optimizer module**. It shows the astuteness of GPT-4 vs the mistake by ChatGPT at 'applying momentum'.

Sparks of Artificial General Intelligence: Early experiments with GPT-4 (Bubeck et al., Arxiv 2023)

LLMs Capabilities: Example

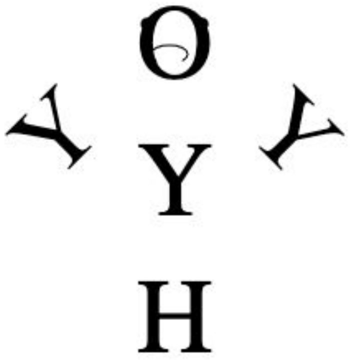
Asking to draw an image using tikz in Latex



Produce TikZ code that draws a person composed from letters in the alphabet. The arms and torso can be the letter Y, the face can be the letter O (add some facial features) and the legs can be the legs of the letter H. Feel free to add other features.

Certainly! Below is a TikZ code that creates a simple stick figure using the letters Y and H for the body and O for the face, with added facial features.

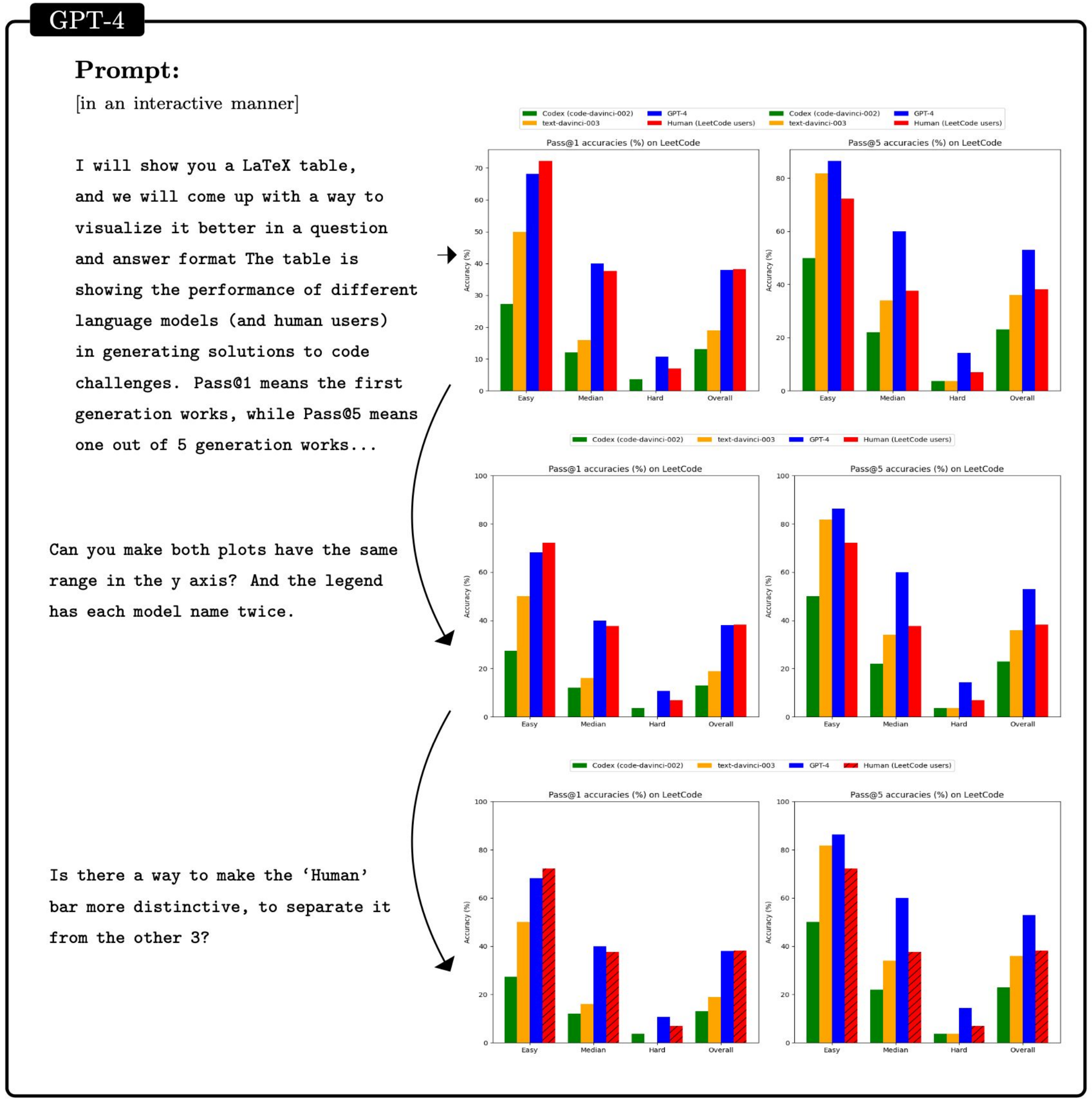
```latex
\documentclass[tikz,border=10pt]{standalone}
\usepackage{tikz}

\begin{document}
\begin{tikzpicture}
  % Torso (Y)
  \node[scale=3] (torso) at (0,0) {Y};

  % Face (O)
  \node[scale=3] (face) at (0,1.5) {O};
  % Eyes
  \fill (-0.3,1.7) circle (2pt);
  \fill (0.3,1.7) circle (2pt);
  % Mouth
  \draw (-0.2,1.3) arc (225:-45:0.2 and 0.1);

  % Arms (Y)
  \node[scale=3, rotate=45] (leftArm) at (-1.3,0.5) {Y};
  \node[scale=3, rotate=-45] (rightArm) at (1.3,0.5) {Y};

  % Legs (H)
  \node[scale=3] (legs) at (0,-1.5) {H};
\end{tikzpicture}
\end{document}
```



Sparks of Artificial General Intelligence: Early experiments with GPT-4 (Bubeck et al., Arxiv 2023)

LLMs Capabilities: Example

Data Visualization using Latex



Prompt:

[in an interactive manner]

I will show you a LaTeX table, and we will come up with a way to visualize it better in a question and answer format The table is showing the performance of different language models (and human users) in generating solutions to code challenges. Pass@1 means the first generation works, while Pass@5 means one out of 5 generation works...

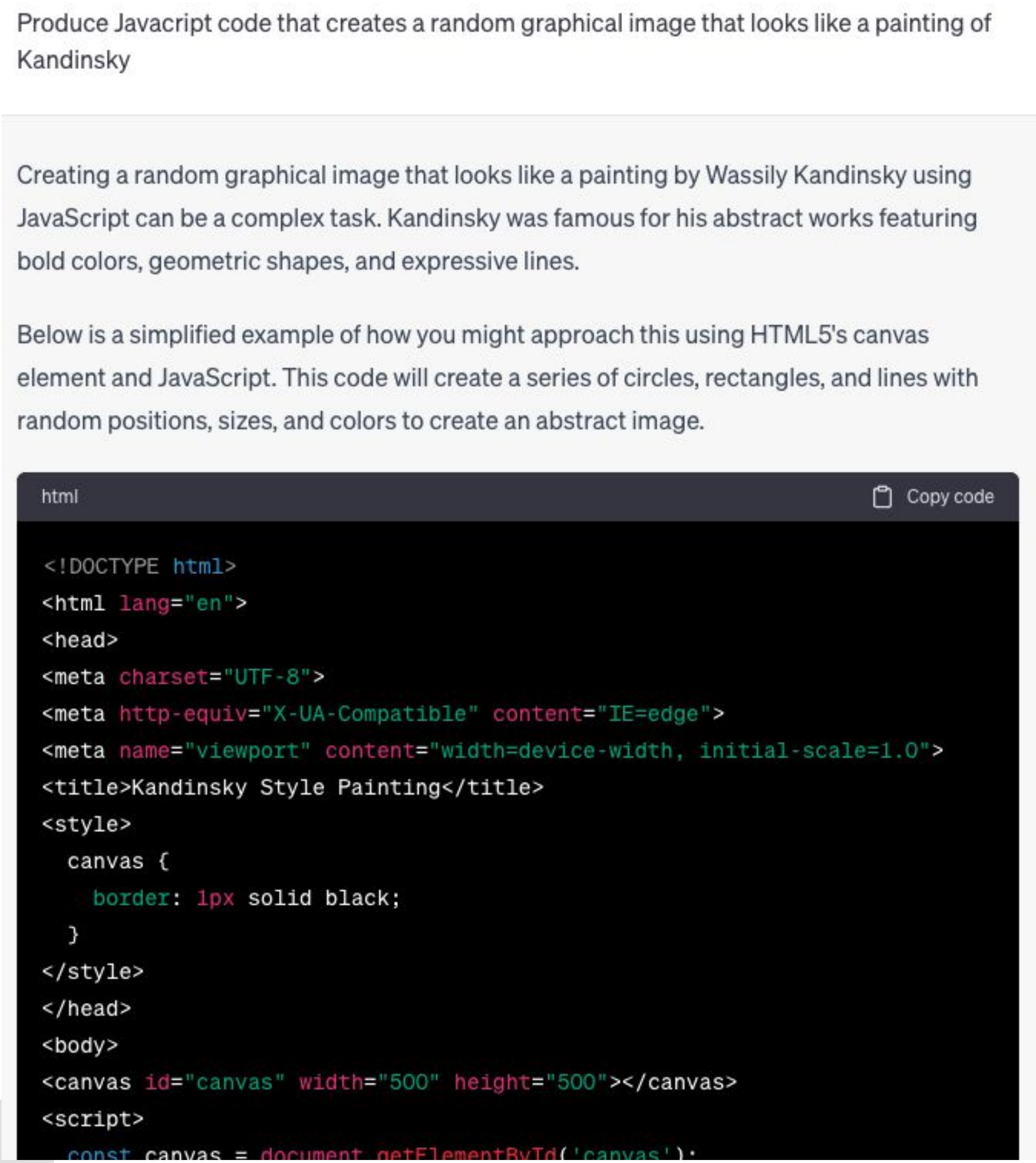
Can you make both plots have the same range in the y axis? And the legend has each model name twice.

Is there a way to make the 'Human' bar more distinctive, to separate it from the other 3?

Sparks of Artificial General Intelligence:
Early experiments with GPT-4 (Arxiv: 2023)

LLMs Capabilities: Example

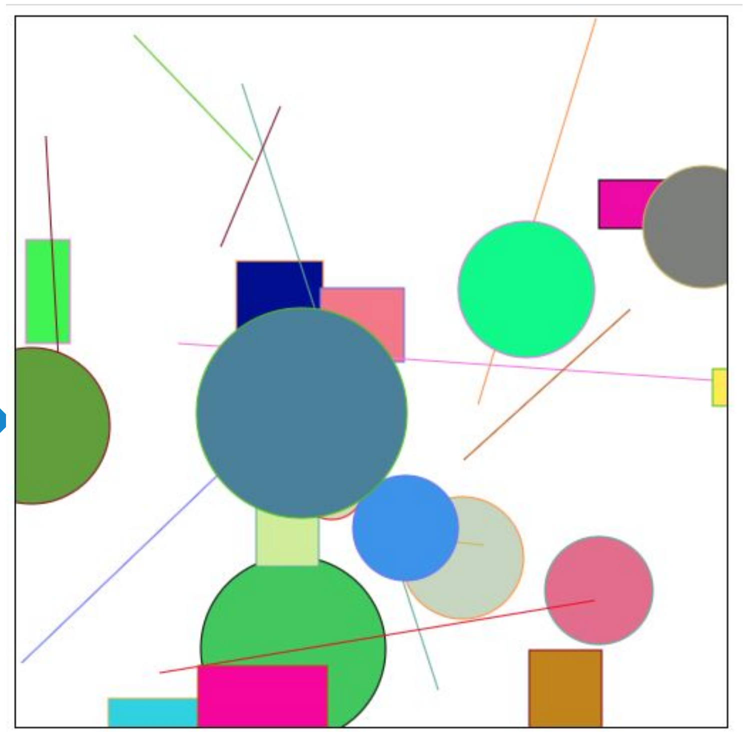
Kandinsky's Abstract painting


Produce Javacript code that creates a random graphical image that looks like a painting of Kandinsky

Creating a random graphical image that looks like a painting by Wassily Kandinsky using JavaScript can be a complex task. Kandinsky was famous for his abstract works featuring bold colors, geometric shapes, and expressive lines.

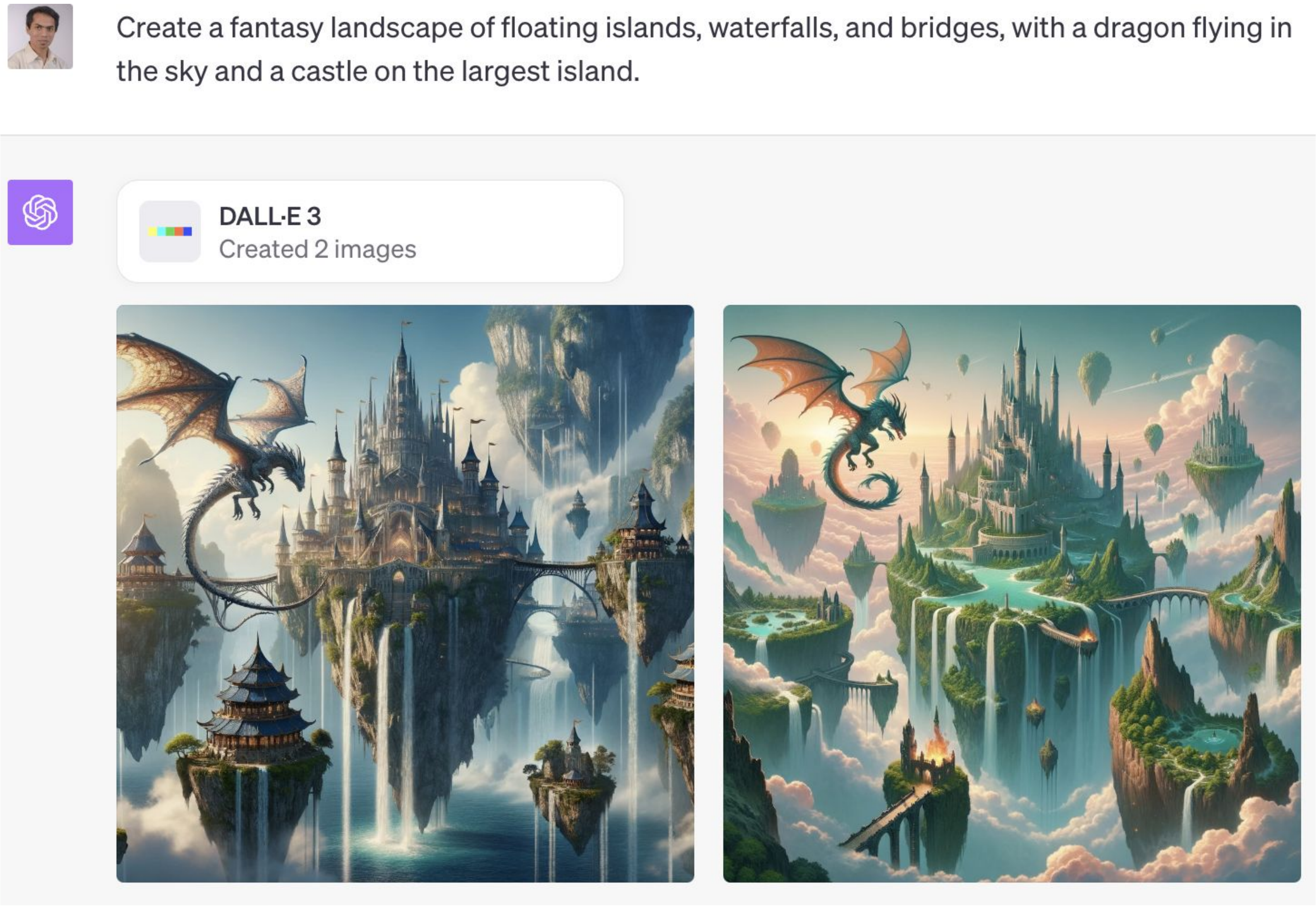
Below is a simplified example of how you might approach this using HTML5's canvas element and JavaScript. This code will create a series of circles, rectangles, and lines with random positions, sizes, and colors to create an abstract image.

```html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Kandinsky Style Painting</title>
<style>
  canvas {
    border: 1px solid black;
  }
</style>
</head>
<body>
<canvas id="canvas" width="500" height="500"></canvas>
<script>
  const canvas = document.getElementById('canvas');
```



Sparks of Artificial General Intelligence: Early experiments with GPT-4 (Arxiv: 2023)

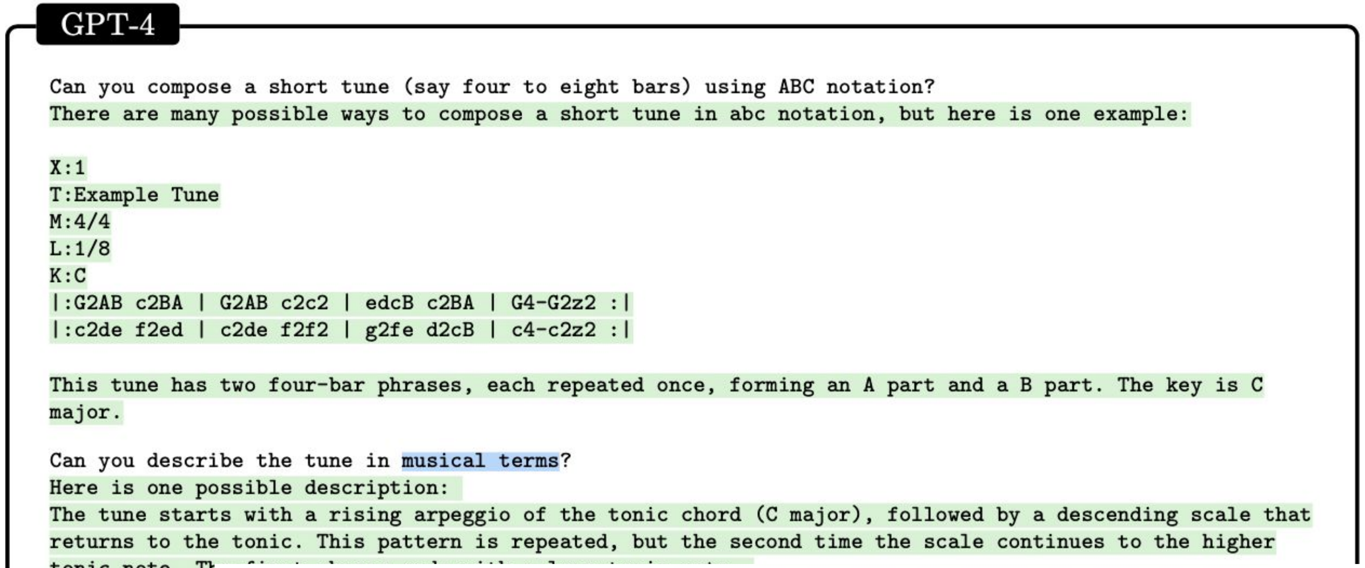
LLMs Capabilities: Example



Create a fantasy landscape of floating islands, waterfalls, and bridges, with a dragon flying in the sky and a castle on the largest island.

LLMs Capabilities: Example

Music tone generator



GPT-4

Can you compose a short tune (say four to eight bars) using ABC notation?

There are many possible ways to compose a short tune in abc notation, but here is one example:

```
X:1
T:Example Tune
M:4/4
L:1/8
K:C
|:G2AB c2BA | G2AB c2c2 | edcB c2BA | G4-G2z2 :|
|:c2de f2ed | c2de f2f2 | g2fe d2cB | c4-c2z2 :|
```

This tune has two four-bar phrases, each repeated once, forming an A part and a B part. The key is C major.

Can you describe the tune in musical terms?

Here is one possible description:

The tune starts with a rising arpeggio of the tonic chord (C major), followed by a descending scale that returns to the tonic. This pattern is repeated, but the second time the scale continues to the higher tonic note. Th

The second phr

the tonic. The

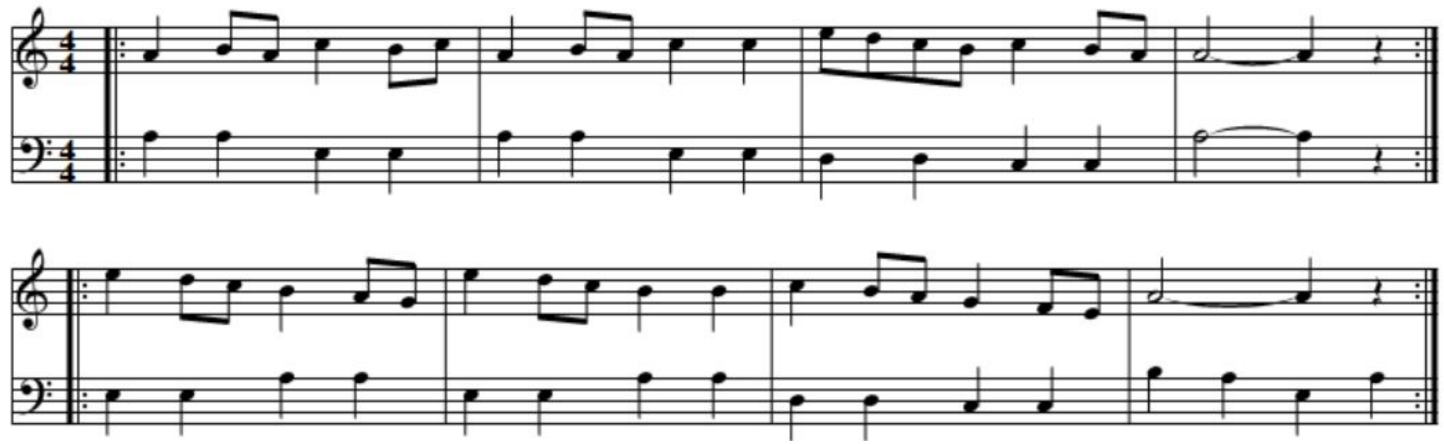
stepwise desce

The A part and

simple and cat

I like part A,

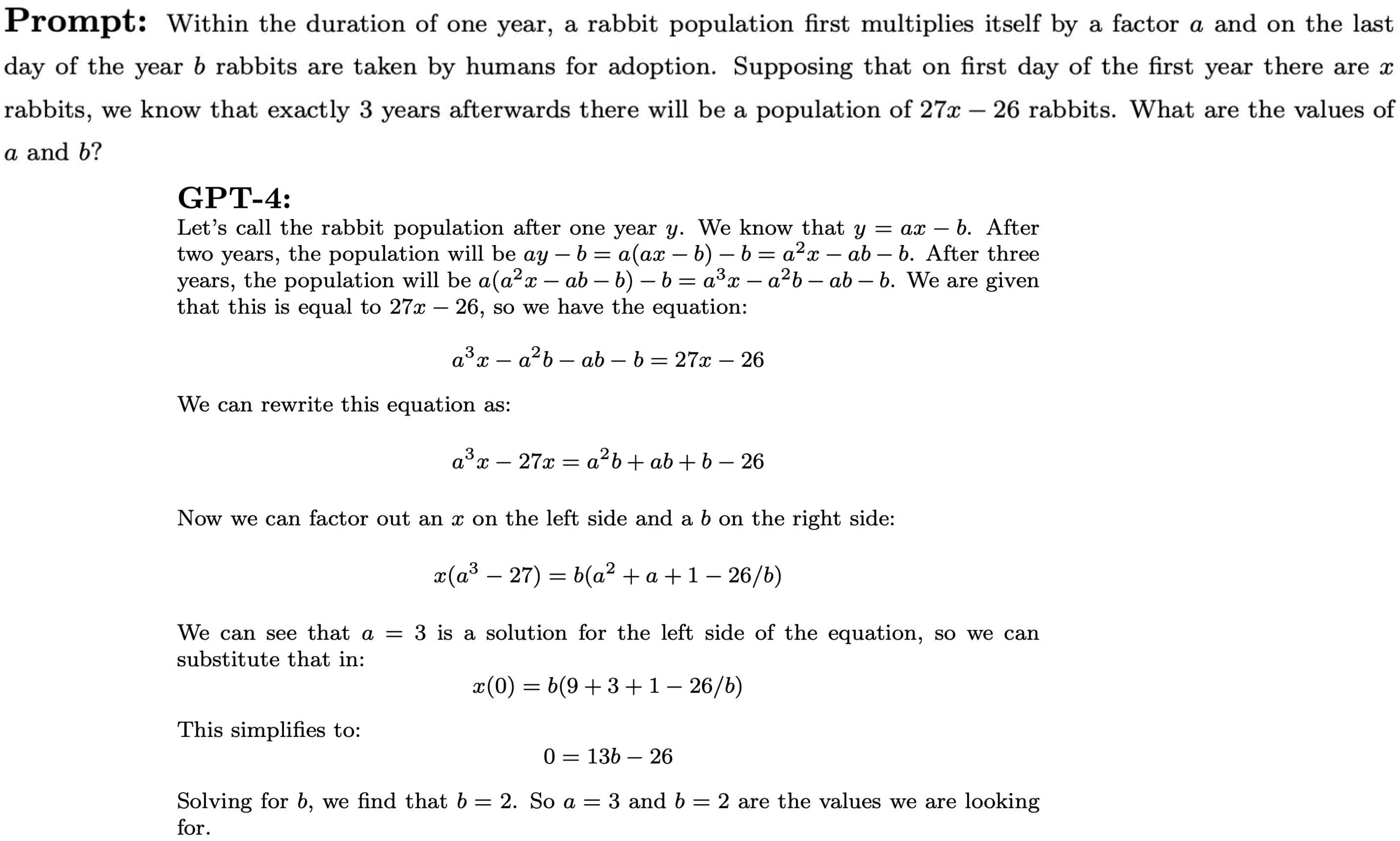
The sheet music of the final tune:



Sparks of Artificial General Intelligence: Early experiments with GPT-4 (Arxiv: 2023)

LLMs Capabilities: Example

Mathematical abilities



Prompt: Within the duration of one year, a rabbit population first multiplies itself by a factor a and on the last day of the year b rabbits are taken by humans for adoption. Supposing that on first day of the first year there are x rabbits, we know that exactly 3 years afterwards there will be a population of $27x - 26$ rabbits. What are the values of a and b?

GPT-4:

Let's call the rabbit population after one year y. We know that $y = ax - b$. After two years, the population will be $ay - b = a(ax - b) - b = a^2x - ab - b$. After three years, the population will be $a(a^2x - ab - b) - b = a^3x - a^2b - ab - b$. We are given that this is equal to $27x - 26$, so we have the equation:

$$a^3x - a^2b - ab - b = 27x - 26$$

We can rewrite this equation as:

$$a^3x - 27x = a^2b + ab + b - 26$$

Now we can factor out an x on the left side and a b on the right side:

$$x(a^3 - 27) = b(a^2 + a + 1 - 26/b)$$

We can see that $a = 3$ is a solution for the left side of the equation, so we can substitute that in:

$$x(0) = b(9 + 3 + 1 - 26/b)$$

This simplifies to:

$$0 = 13b - 26$$

Solving for b, we find that $b = 2$. So $a = 3$ and $b = 2$ are the values we are looking for.

Sparks of Artificial General Intelligence: Early experiments with GPT-4 (Arxiv: 2023)

LLMs Capabilities

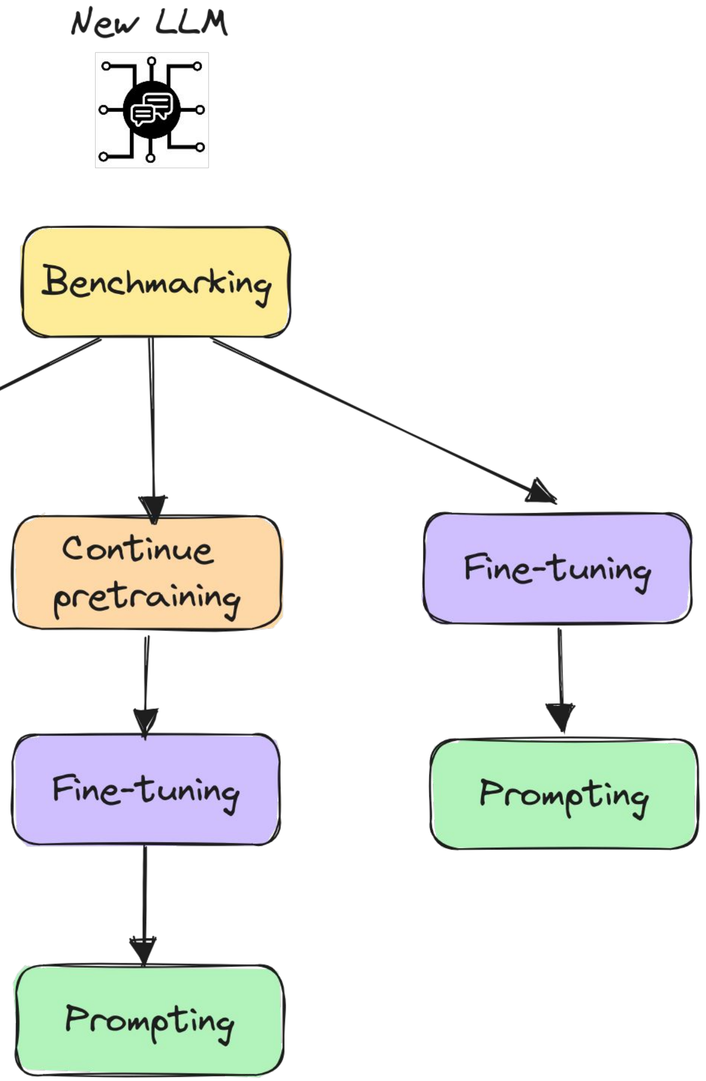
Can LLMs seemingly "solve" most problems?

What about the performance?

Can LLMs achieve same performance as SOTA for all languages, dialects, modality, and tasks?

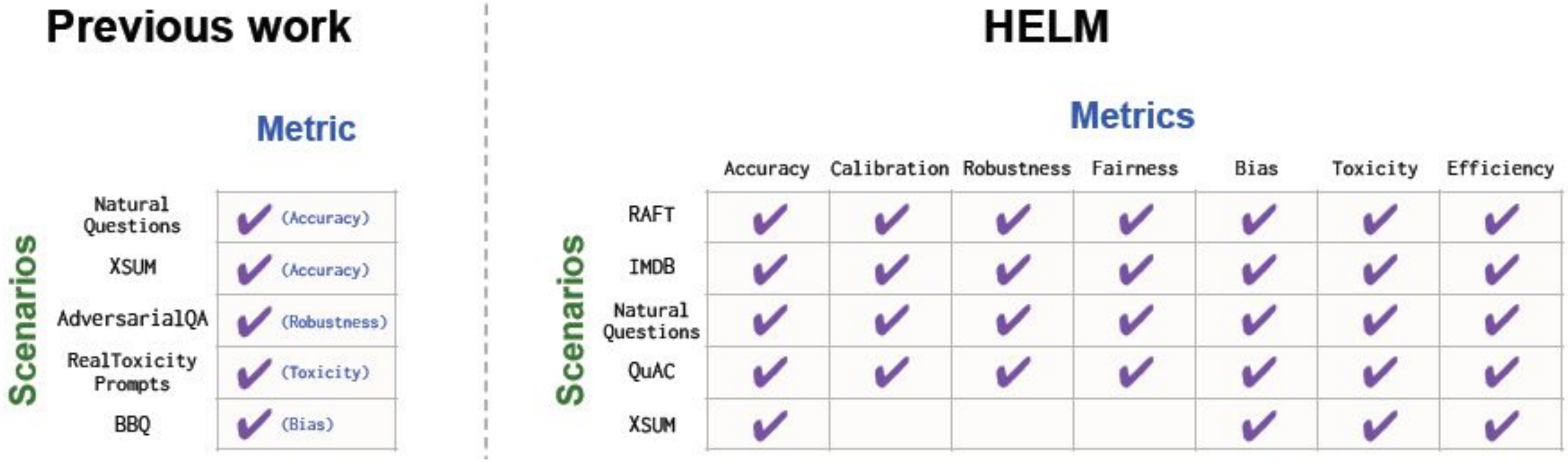
Do we need to build language specific LLMs?

What Could an Workflow Towards Building an LLM?



Benchmarking Studies: English

- **HELM:** Holistic Evaluation of Language Models
- Defines a taxonomy of tasks
- Systematically evaluate tasks using 7 categories of metrics
  - Metrics reflect a range of societal considerations
  - Accuracy, calibration, robustness, fairness, bias, toxicity, efficiency.



Previous work

Scenarios	Metric
Natural Questions	✓ (Accuracy)
XSUM	✓ (Accuracy)
AdversarialQA	✓ (Robustness)
RealToxicity Prompts	✓ (Toxicity)
BBQ	✓ (Bias)

HELM

Scenarios	Accuracy	Calibration	Robustness	Fairness	Bias	Toxicity	Efficiency
RAFT	✓	✓	✓	✓	✓	✓	✓
IMDB	✓	✓	✓	✓	✓	✓	✓
Natural Questions	✓	✓	✓	✓	✓	✓	✓
QuAC	✓	✓	✓	✓	✓	✓	✓
XSUM	✓				✓	✓	✓

Holistic evaluation of language models (Arxiv, 2023)

Benchmarking Studies: Arabic

Tasks, Datasets

- 33 tasks
- 61 datasets
- 46 hours of speech
- 30 sentences for TTS

Models:

- **NLP:** GPT-3.5, GPT-4, BloomZ
- **ASR:** Whisper, USM
- **TTS:** Amazon Polly, QCRI TTS

330+ sets of experiments
296K data points

TASKS

- **Word Segmentation, Syntax & Information Extraction** **(e.g., POS tagging)**
- **Factuality, Disinformation & Harmful Content Detection** **(e.g., Hate Speech & Propaganda Detection)**
- **Semantics** **(e.g., Semantic Textual Similarity and Natural Language Inference)**
- **Demographic & Protected Attributes** **(e.g., Gender and User Country Detection)**
- **Sentiment, Stylistic & Emotion Analysis** **(e.g., Stance Detection, Sarcasm Detection)**
- **Machine Translation** **(e.g., English-Arabic and Arabic dialects)**
- **News Categorization**
- **Question Answering**

DATASETS

- XNLI
- XGLUE
- XQuAD
- ASAD
- Aqmar
- SANAD
- MADAR
- QASR
- WikiNews
- Conll2006
- ANERcorp

EVALUATION

- Accuracy
- F1
- Macro-F1
- Micro-F1
- Weighted-F1
- BLEU
- WER
- Pearson Correlation
- Jaccard Similarity

MODELS

- GPT-3.5
- GPT-4
- BLOOMZ

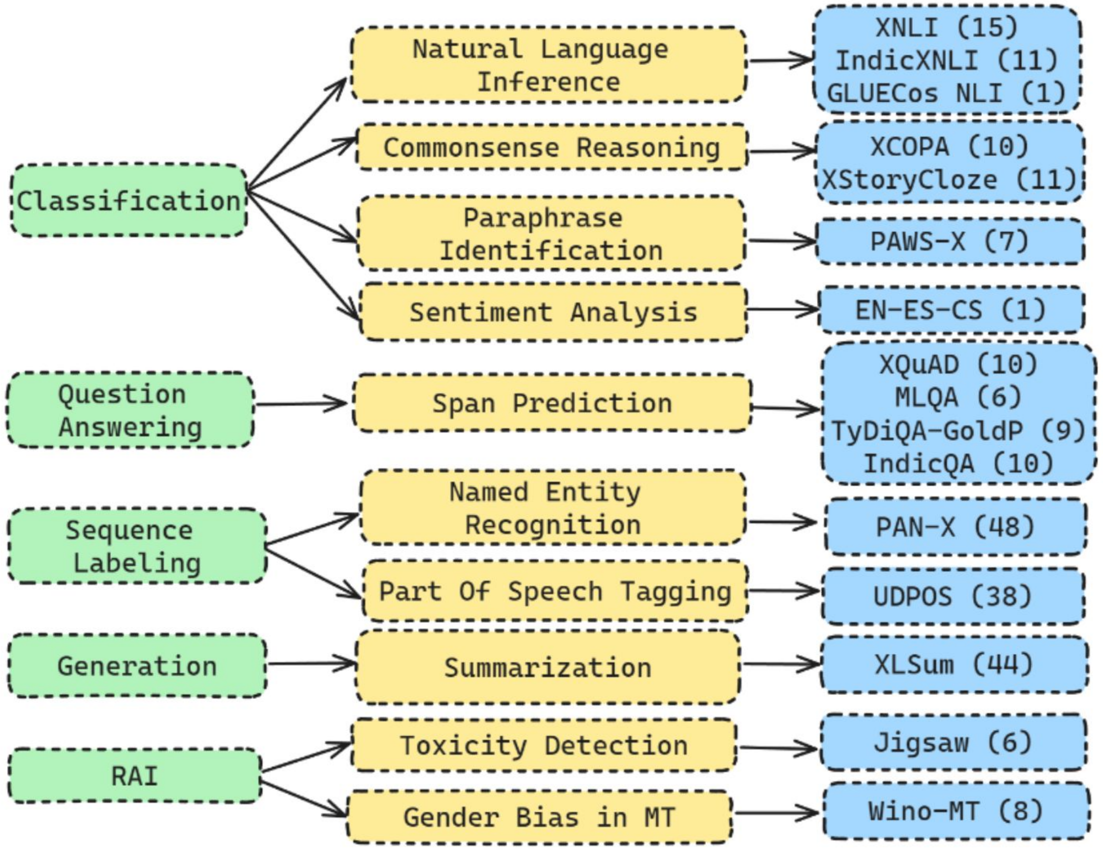
LEARNING

- Zero-shot
- Few-shot

Benchmarking Arabic AI with Large Language Models (EACL, 2024)

Benchmarking Studies: Multilingual

MEGA evaluates models on standard NLP benchmarks, covering 16 NLP datasets across 70 typologically diverse languages



Classification → Natural Language Inference → XNLI (15), IndicXNLI (11), GLUECos NLI (1)

Classification → Commonsense Reasoning → XCOPA (10), XStoryCloze (11)

Classification → Paraphrase Identification → PAWS-X (7)

Classification → Sentiment Analysis → EN-ES-CS (1)

Question Answering → Span Prediction → XQuAD (10), MLQA (6), TyDiQA-GoldP (9), IndicQA (10)

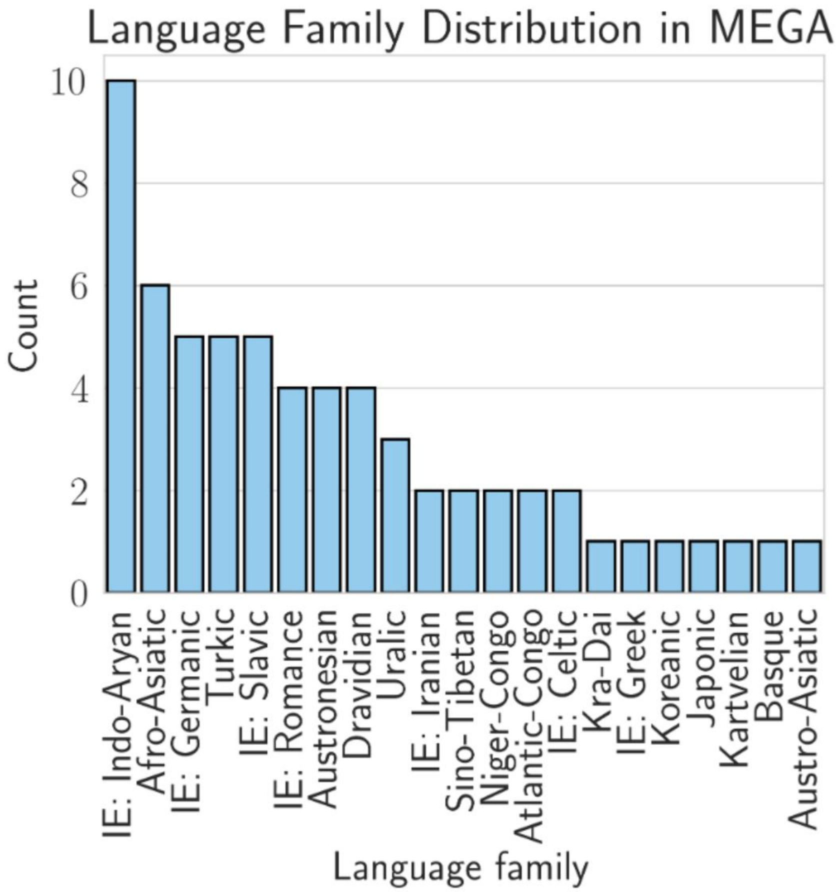
Sequence Labeling → Named Entity Recognition → PAN-X (48)

Sequence Labeling → Part Of Speech Tagging → UDPOS (38)

Generation → Summarization → XLSum (44)

RAI → Toxicity Detection → Jigsaw (6)

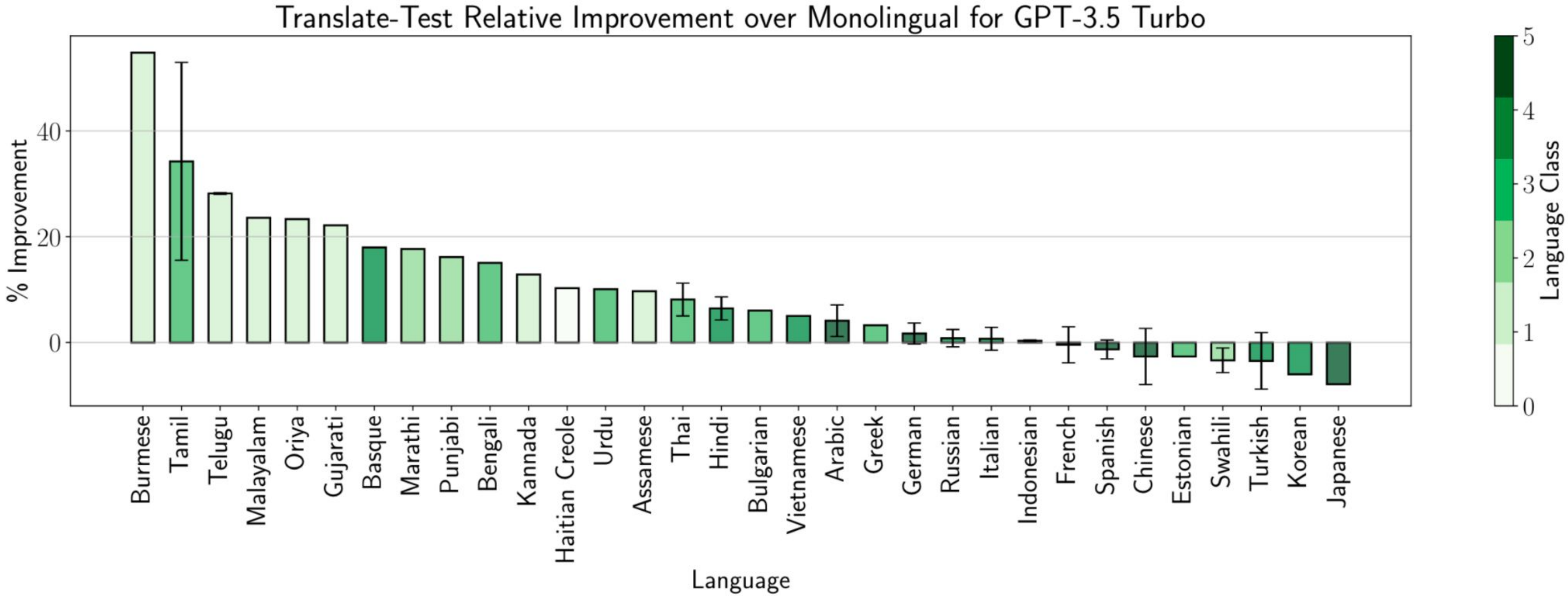
RAI → Gender Bias in MT → Wino-MT (8)



MEGA: Multilingual Evaluation of Generative AI (Ahuja et al, 2024)

Benchmarking Studies: Multilingual

- LLMs still vastly underperform on (especially low-resource) non-English languages



MEGA: Multilingual Evaluation of Generative AI (Ahuja et al, 2024)

Benchmarking Studies: Multilingual

21 datasets covering 8 different common NLP application tasks

- ChatGPT fails to generalize to low-resource and extremely low-resource languages (e.g., Marathi, Sundanese, and Buginese).
- ChatGPT shows more weakness in inductive reasoning than in deductive or abductive reasoning
- ChatGPT suffers from the hallucination problem

A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity (Bang et al, IJCNLP-AACL 2024)

Benchmarking Studies: Multilingual

37 diverse languages, characterizing high-, medium-, low-, and extremely low-resource languages

- ChatGPT’s zero-shot learning performance is generally worse than the SOTA
- The importance of task-specific models is higher
- ChatGPT’s performance is generally better for English than for other languages, especially for higher-level tasks that require more complex reasoning abilities

ChatGPT Beyond English: Towards a Comprehensive Evaluation of Large Language Models in Multilingual Learning (Bang et al, 2024)

Thank You

https://llm-low-resource-lang.github.io/