A State Of the Art of Multimodal Large Language Models

Nouh CHELGHAM, Fahd FADHA, Noura FAIZ, Hajar FORSI DROBI, and Adil GHALEM

Paul Sabatier University

Abstract. In recent years, there has been a significant surge in research and development surrounding large language models (LLMs) fuelled by advancements in deep learning techniques and the availability of vast amounts of text data. These LLMs, with their capacity to generate coherent and contextually relevant text, have shown remarkable capabilities across various natural language processing (NLP) tasks. However, to further extend the capabilities of these models and bridge the gap between language and other modalities, researchers have started integrating multimodal inputs, such as images, audio, and video, into the architecture of LLMs. This paper presents a review of the current state of multimodal large language models, highlighting their architecture, training methodologies, and applications. We discuss the evolution from unimodal LLMs to MLLMs, outlining key innovations and challenges in this domain. Finally, we discuss the future directions and research challenges in the field of MLLMs, emphasizing the need for better strategies, enhanced interpretability, and scalability to larger datasets and more complex modalities.

Keywords: Language Models · Large Language Models · Multimodal Large Language Models

1 Introduction

In recent years, we've observed a surge of interest in large language models (LLMs) within AI research. These models have demonstrated remarkable capabilities in understanding and generating human-like text. However, as the demand for more versatile and contextually aware AI systems grows, researchers are exploring novel approaches to imbue these models with multimodal capabilities, enabling them to process and generate not only text, but also other modalities such as images, audio, and video. Multimodal large language models (MLLMs) represent the next topic of interest in AI research. By integrating information from multiple modalities, MLLMs have the potential to exhibit a deeper understanding of context, nuances, and semantics than their unimodal counterparts. This paper aims to provide a comprehensive overview of the state of MLLMs, examining their architecture, training methodologies, applications, and challenges. We'll go over the evolution of Language Models (LMs) transitioning into Large Language Models (LLMs), and further, the emergence of Multimodal Large Language Models (MLLMs).

2 An Overview of Language Models

Language modelling aims to compute the probability distribution of a sequence of words, to predict future or absent tokens. It represents a crucial strategy in advancing algorithms tailored for understanding natural language. The evolution of language modelling during the last three decades has been significant, transitioning from early language models primarily focused on text generation to the more recent emphasis on solving complex problems.

2.1 Statistical Language Models (SLM)

Statistical Language Models (SLMs) are rooted in statistical learning methods that gained prominence in the 1990s. These models predict the next word based on recent context. SLMs find wide use in information retrieval (IR) and natural language processing (NLP). However, they face challenges when it comes to modelling long-range dependencies and capturing semantic nuances. This limitation arises from their reliance on fixed-length context windows and the inability to discern complex linguistic patterns beyond the immediate context. Additionally, SLMs struggle with out-of-vocabulary words and are less effective in handling noisy or ambiguous input. Despite these challenges, SLMs remain foundational in NLP, serving as benchmarks for more advanced language models and providing valuable insights into language structure and usage.

2.2 Neural Language Models (NLM)

Neural Language Models (NLMs) employ neural network architectures such as multilayer perceptrons (MLPs) and recurrent neural networks (RNNs) to ascertain the probability of word sequences. These models introduced the notion of distributed word representations, wherein words are embedded into continuous vector spaces. This representation allows NLMs to capture semantic similarities between words and generalize across different contexts more effectively than traditional approaches. By leveraging neural networks, NLMs can model complex linguistic patterns and dependencies, enabling them to generate more coherent and contextually relevant text. Moreover, the flexibility of neural architectures facilitates the incorporation of additional features and data sources, enhancing the model's performance across various Natural Language Processing (NLP) tasks.

2.3 Pre-trained Language Models (PLM)

Pre-trained Language Models (PLMs) involve training bidirectional Language Models to acquire context-aware word representations, a breakthrough that has notably enhanced the efficacy of NLP models. PLMs have established the pre-training and fine-tuning paradigms, revolutionizing the approach to leveraging Language Models for diverse NLP tasks.

2.4 Large Language Models (LLM)

Researchers have observed that scaling up Language Models not only enhances performance but also unveils exceptional abilities like in-context learning, previously latent in smaller counterparts. This realization has given rise to the concept of Large Language Models (LLMs), typically built upon transformer architecture and characterized by billions of parameters. Trained on extensive textual data, these models possess a remarkable ability to comprehend natural language and tackle complex tasks through text generation, striving to become versatile and adept at solving a myriad of tasks.

3 An Overview of Large Language Models

3.1 Background

In contemporary NLP research, Large Language Models stand out, employing Transformer-based architectures with vast parameter spaces, often comprising billions of parameters. Trained on extensive textual data, they exhibit profound natural language understanding and adeptness in text generation. Key areas of LLM research include scaling laws and emergent abilities. [2]

Scaling Laws Scaling laws analyze the relationship between model size, dataset size, and computational resources, impacting model performance. Noteworthy examples include:

1. **The KM scaling law** explores the impact of increasing model and dataset size alongside computational resources on performance, highlighting diminishing returns.
2. **The Chinchilla scaling law** focuses on the relationship between model size and performance, independently of dataset size, particularly emphasizing the benefits for large models.

Understanding these scaling laws provides insights into optimizing model performance while managing computational constraints. [2]

Emergent Abilities Emergent abilities in LLMs manifest as models increase in size, introducing new capabilities like in-context learning, instruction following, and step-by-step reasoning, enhancing performance across linguistic tasks. Studying these emergent abilities is crucial for improving LLM effectiveness in real-world applications.

The interaction between scaling laws and emergent abilities presents a complex area of study. While scaling laws offer predictability in performance trends, emergent abilities introduce transformative potential, requiring nuanced considerations in model evaluation and optimization.[2]

3.2 Pre-training

In large language models (LLMs), pre-training is crucial, shaping the model's abilities through training on extensive corpora. The scale and quality of the pre-training corpus significantly impact LLM capabilities. Additionally, well-designed model architectures and optimization techniques enhance effective pre-training.[2]

Data Sourcing Advanced LLMs rely on acquiring extensive natural language datasets from various sources for development. They primarily use a mix of diverse public textual datasets for pre-training, categorized into general and specialized data. General data, such as webpages, books, and conversational text, enhances language modelling and generalization capabilities due to its accessibility and diversity. Specialized datasets like multilingual, scientific, and code data provide task-specific competences.

Webpages are abundant sources of diverse linguistic knowledge, while conversational text improves LLMs' question-answering skills. Books offer formal textual content suitable for learning linguistic nuances and creating coherent narratives.

Specialized datasets, including multilingual text and scientific data, augment LLM proficiency in specific domains. Code corpus enhances reasoning abilities and precision in LLM-generated results. [2]

Data Pre-processing Data preprocessing is crucial for constructing robust language models. Techniques like quality filtering, de-duplication, privacy preservation, and tokenization refine raw textual data into high-quality corpora. Quality filtering removes noisy or redundant data, while deduplication mitigates the negative effects of duplicate data. Privacy preservation techniques focus on reducing personally identifiable information. Tokenization segments raw text into discrete tokens for model input. [2]

Data Scheduling In LM pretraining, data scheduling involves two main aspects: data mixture and data curriculum. Data mixture ensures a balanced training corpus, promoting LM generalization. Data curriculum structures data presentation for pedagogical effectiveness, sequencing from basic to complex data and adjusting based on LM performance and predefined benchmarks. Works best for specialized LLMs. [2]

Common Pre-Training Tasks

1. **Language Modeling (LM):** Autoregressive Prediction involves predicting the next token in a sequence based on preceding tokens. It's essential for understanding sequential dependencies in text data. Models like GPT-3 rely on LM as their primary pre-training objective. They generate coherent and contextually relevant text by predicting subsequent tokens in a sequence. [2]
2. **Denoising Autoencoding (DAE):** Recovering Corrupted Text tasks involve reconstructing original text from corrupted input by replacing spans with random tokens. This encourages LLMs to learn robust language representations by understanding and correcting input errors. While effective, DAE tasks are more complex to implement compared to LM, limiting their widespread usage in pre-training large language models. [2]

3.3 Architecture

In Large Language Models (LLMs), the Transformer architecture is key for its remarkable scalability and easy parallelization, allowing models to reach unprecedented sizes. LLM architectures are typically categorized into three main types: encoder-decoder, causal decoder, and prefix decoder.[2]

Encoder-Decoder Architecture This architecture, based on the classic Transformer model, consists of stacked encoder and decoder blocks. Encoders use multi-head self-attention layers to encode input sequences, while decoders perform cross-attention on these encoded representations to generate target sequences. It's commonly used in tasks like machine translation and text summarization. Currently, very few LLMs use this architecture, like Flan-T5. [2]

Causal Decoder Architecture In this architecture, the decoder is designed to generate output tokens one at a time in a causal manner, meaning each token is conditioned only on previously generated tokens. This architecture is often used in autoregressive models like GPT series for tasks like text generation. Specifically, GPT-3 has proven the efficacy of this architecture, showcasing impressive in-context learning capabilities in Large Language Models (LLMs). Yet, GPT-1 and GPT-2 lack the advanced abilities observed in GPT-3, indicating the significance of scaling in enhancing this model architecture's capacity. So far, this architecture is the most widely used.[2]

Prefix Decoder Architecture This architecture is similar to the causal decoder but with the addition of a prefix input. The prefix provides contextual information that guides the decoder's generation process. It's particularly useful for tasks where prior context is crucial for generating accurate outputs, such as question answering or dialogue generation. [2]

3.4 Training of the Model

Optimization Settings Training large language models (LLMs) involves careful optimization settings to ensure stability and efficiency.[2]

Batch Training:

- LLMs typically use large batch sizes to improve stability and throughput. For example, batch sizes of 2,048 examples or 4 million tokens are common.
- Dynamic batch size adjustment, as seen in GPT-3, gradually increases batch size during training to stabilize the process. [2]

Learning Rate:

- A warm-up and decay strategy is employed for learning rate adjustment. Initially, a linear warm-up increases the learning rate gradually, followed by cosine decay until convergence. [2]

Optimizer:

- Commonly used optimizers include Adam and AdamW, which are based on adaptive estimates of lower-order moments for first-order gradient-based optimization.
- Adafactor optimizer is designed for conserving GPU memory during training and is used in models like PaLM and T5. [2]

Stabilizing Training:

- Weight decay and gradient clipping are widely used techniques to mitigate training instability issues.
- Some models, like PaLM and OPT, employ strategies such as restarting training from an earlier checkpoint to address training loss spikes.[2]

Training Suggestions

- Combining techniques such as 3D parallelism, ZeRO, and mixed precision training can significantly improve training throughput and memory efficiency.
- Leveraging open-source libraries like DeepSpeed, Colossal-AI, and Alpa that support these techniques can streamline the training process.
- Utilizing early issue detection mechanisms, such as predictable scaling introduced in GPT-4, aids in forecasting model performance and detecting problems early in training. [2]

3.5 Adaptation

After pre-training, Large Language Models (LLMs) can be further adapted through instruction tuning, refining their abilities for specific tasks, and alignment tuning, aligning their behaviors with human values. Efficient tuning and quantization techniques are necessary for model adaptation in resource-limited settings, aiming to optimize performance while minimizing computational resources.

Instruction tuning Instruction tuning involves fine-tuning pre-trained Large Language Models (LLMs) using formatted natural language instances. It enhances LLMs' generalization to unseen tasks, even in multilingual settings. Recent research focuses on its impact on LLMs, guiding instance collection and tuning. [2]

Compared to pre-training, instruction tuning optimizes training objectives and configurations more efficiently. Balancing data distribution and combining tuning with pre-training enhance effectiveness, while multi-stage tuning prevents capacity forgetting. Despite utilizing fewer instances, instruction tuning consistently enhances LLMs' performance across various scales, architectures, and objectives, thereby improving task generalization and domain specialization. [2]

In constructing LLM instances, prioritizing diversity and quality over quantity is crucial for improved generalization. Clear task descriptions and examples play key roles, while integrating chain-of-thought (CoT) examples can enhance reasoning abilities across tasks. However, there's a lack of clear guidelines for annotating human-need instances, making the process somewhat heuristic. [2]

Alignment Tuning Aligning Large Language Models (LLMs) with human values is essential for their effective and safe application across various Natural Language Processing (NLP) tasks. Reinforcement Learning from Human Feedback (RLHF) stands as a prevalent method for achieving this alignment. By fine-tuning LLMs with human feedback data, RLHF enhances their behavior according to criteria like usefulness, honesty, and harmlessness. Nonetheless, RLHF isn't the sole approach available. Other methods for aligning LLMs encompass supervised fine-tuning, rule-based techniques, adversarial testing, and hybrid approaches.[2]

3.6 Evaluation

AI model evaluation, utilizing standard protocols (k-fold cross-validation, holdout validation, LOOCV, bootstrap, and reduced set) is crucial for assessing performance, with each method offering advantages and limitations depending on specific problem requirements and data characteristics. However, as LLMs become more prominent and complex, existing evaluation protocols may not fully capture their capabilities. [2]

Amidst the rapid evolution and widespread adoption of LLMs, comprehensive evaluation frameworks are needed to consider both task-level performance and societal implications. Evaluation methodologies are evolving to include greater human involvement, such as AdaVision and AdaTest, facilitating interactive testing and feedback. Crowd-sourcing test sets, like DynaBench and DynamicTempLAMA, are increasingly used to generate and assess challenging samples. Additionally, tools like DeepTest and CheckList focus on creating diverse and challenging test sets tailored to specific tasks, with fairness considerations. Platforms like HELM and Big-Bench provide multifaceted evaluations, while PromptBench assesses adversarial robustness, revealing vulnerabilities in LLMs to adversarial prompts. [2]

The exploration and advancement of LLMs involve various strategies, from examining open-source frameworks to scrutinizing proprietary technologies. Customization for academic purposes includes rigorous empirical assessments and analyses of specialized, fine-tuned, and base models, essential for understanding their proficiency in intricate reasoning activities such as symbolic, knowledge-driven, and mathematical problem-solving. In the research community, interdisciplinary applications of LLMs are showcased through detailed case studies in non-traditional domains, emphasizing adaptability beyond conventional contexts and addressing complex issues like model bias,

ethical AI development, and data privacy. Empirical evaluation employs advanced statistical methods like metrics analysis, incorporating perplexity for language modeling and F1 scores for task-specific benchmarks, while domain-specific benchmarking and evaluation of base LLMs assess performance accurately within specific domains and quantify generalization ability across diverse NLP tasks.[2]

4 Multimodal Large Language Models

4.1 General Overview of MLLMs

Previous surveys have categorized Multimodal Large Language Models (MLLMs) into four main genres [3]: Multimodal Instruction Tuning (MIT), Multimodal In-Context Learning (M-ICL), Multimodal Chain-of-Thought (M-CoT), and LLM-Aided Visual Reasoning (LAVR).

Multimodal Instruction Tuning (MIT): Instruction tuning refines pre-trained language models (LLMs) using task-specific datasets, integrating multimodal data (text, images, audio) for precise guidance. It aims to generalize across tasks rather than overfitting, enhancing zero-shot learning. Multimodal Instruction Tuning (M-IT) datasets are acquired by adapting existing benchmarks or using self-instruction methods. Incorporating multimodal information involves embedding foreign modalities directly into LLMs, making them versatile multimodal chatbots and task solvers.

Multimodal In-Context Learning (M-ICL): Instead of relying solely on vast datasets, ICL learns through analogy, enabling LLMs to solve complex tasks with few examples, often accompanied by optional instructions. This allows for adaptable generalization to unseen tasks. Employed without additional training, ICL seamlessly integrates into various frameworks during inference, with instruction-tuning further enhancing its efficacy.

Within Multimodal Large Language Models (MLLMs), ICL evolves into Multimodal ICL (M-ICL), encompassing multiple modalities. M-ICL applications range from tackling visual reasoning tasks to instructing LLMs in the use of external tools.

Multimodal Chain-of-Thought (M-CoT): M-CoT models complex chains of reasoning using multimodal inputs to capture intricate relationships and dependencies between concepts across modalities. It facilitates sophisticated reasoning and decision-making by enabling the model to infer logical connections between textual, visual, and audio inputs.

Derived from CoT (Chain-of-Thought), M-CoT embodies human-like cognitive processes by sequencing reasoning steps toward a final answer. Transitioning to Multimodal CoT aims to enhance reasoning abilities across diverse inputs by bridging modalities. This adaptation focuses on acquiring M-CoT skills through various approaches and configuring reasoning chains to empower LLMs to express coherent chains of thought across modalities.

LLM-Aided Visual Reasoning (LAVR): LAVR integrates large language models with visual data for tasks like image captioning and visual question answering, leveraging language understanding and visual perception synergy. Recent research explores combining LLMs with external tools or vision models for visual reasoning tasks, offering advantages such as

Robust generalization: They demonstrate strong zero/few-shot performance on unseen objects or concepts due to extensive pretraining. Emergent capabilities: Stand out at intricate tasks like deciphering nuanced meanings in images, thanks to LLMs' reasoning skills and knowledge. Enhanced interactivity: Facilitating user interactions via intuitive interfaces, such as clicks and natural language queries.

Overall, these four genres represent distinct paradigms for leveraging multimodal data within large language models, each addressing different aspects of multimodal understanding, reasoning, and interaction. By categorizing MLLMs into these genres, researchers can explore and develop novel approaches to harness the power of multimodal information for advancing the capabilities of language models across a wide range of applications and domains.

4.2 Applications

1. **Image Captioning:** Multimodal models can generate descriptive captions for images by analyzing both visual content and textual context. The model learns to associate relevant text with different aspects of the image, producing coherent and informative captions. [8]
2. **Visual Question Answering (VQA):** In VQA tasks, models answer questions about images using both visual and textual information. Multimodal models can effectively reason about the content of images and understand the context of questions to provide accurate answers.
3. **Image Generation from Text:** Similar to DALL-E, multimodal models can generate realistic images based on textual descriptions. By learning the relationships between textual and visual features, these models produce images that align with the provided descriptions.

4.3 Techniques

In the realm of multimodal models, a core architecture has emerged, consisting of an encoder, a joint embedding space, and a language model for generating text responses. This structure remains essential for effectively integrating diverse modalities, despite the myriad of approaches explored in this field. To illustrate these principles, we'll focus on two prominent models: CLIP from OpenAI and the Flamingo model.

Encoder and Joint Embedding Space : The encoder is pivotal in processing input data from various modalities such as text, images, or audio, extracting relevant features crucial for downstream tasks. CLIP (Contrastive Language-Image Pretraining), developed by OpenAI, serves as an excellent example. It excels in translating different data modalities, including text and images, into a unified embedding space. Its innovative use of **natural language supervision** and **contrastive learning** significantly enhances its capabilities.

Natural langage supervision

Traditionally, image models relied on datasets where images were manually annotated with corresponding text descriptions (e.g., ImageNet, MS COCO). However, this manual annotation approach poses scalability challenges due to its labor-intensive nature and high costs. To address this, scalable pre-training methods have emerged, leveraging large-scale language models trained on vast amounts of unstructured text data sourced from the internet. These methods tap into the abundance of textual information available online, eliminating the need for manually labeled or curated datasets. For instance, in the "Learning Transferable Visual Models From Natural Language Supervision"

paper, researchers utilized publicly accessible internet data to construct a novel dataset comprising 400 million pairs of images and corresponding text.

Contrastive learning

Before CLIP, vision-language models primarily relied on classifier or language model objectives. However, CLIP introduced a contrastive objective, which revolutionized the field by enabling scalability and generalization across multiple tasks.

Contrastive learning operates on the basis of distinguishing between right and wrong image-text pairings. Positive pairs consist of data samples deemed similar or related, sharing commonalities or belonging to the same class. On the other hand, negative pairs comprise dissimilar or unrelated instances, serving as contrasting examples to the positive pairs.

In CLIP's architecture, the objective of training is to amplify the similarity between correct pairs while diminishing the similarity between incorrect ones. This approach facilitates the creation of a robust embedding space where different modalities can be effectively compared and interpreted.

Joint Embedding Space

In multimodal models, a joint embedding space plays a crucial role as it serves as the meeting point for vectors encoded from different sources. This space is where computations are conducted, enabling effective fusion and interpretation of multimodal inputs.

Various techniques exist for merging embedding vectors from different encoders into the same space in multimodal models. Common approaches include concatenation, summation or averaging, attention mechanisms, cross-modal fusion layers (such as late fusion and early fusion), among others.

Through contrastive learning, CLIP learns to map both images and text onto a shared embedding space using projection matrices that are jointly trained with encoders from scratch. This process enhances the alignment of representations across modalities, facilitating robust multimodal understanding.

In contrast, Flamingo adopts a distinct approach to project text and image vectors into the same space. It utilizes a component called the Perceiver Resampler, which takes variable-size images as input and, through a combination of Feed Forward and Attention mechanisms, outputs fixed-size vectors. This mechanism ensures consistency in the dimensionality of vectors from different modalities.

Regardless of the specific technique employed, the key consideration is ensuring that vectors from each modality are transformed into vectors of the same dimensionality. This harmonization enables effective multimodal processing and interaction within the joint embedding space.

LLM : If a multimodal model aims to generate textual outputs based on the joint representation of text and images, it may incorporate a Large Language Model (LLM) as part of its architecture. This LLM can take the joint embeddings as input and generate text conditioned on the multimodal context. In the case of Flamingo, the model utilized is Chinchilla, developed by the research team at DeepMind and presented in March 2022.

The principle behind fine-tuning Chinchilla LM for specific tasks, such as text generation conditioned on visual inputs in the case of Flamingo, involves freezing certain layers. "Frozen layers" remain fixed or unaltered during the fine-tuning process. This practice ensures that the knowledge learned from the pre-training phase is retained, as freezing prevents the parameters of these layers from being updated.

Typically, lower layers closer to the input are frozen because they capture more generic features and patterns, while higher layers closer to the output are fine-tuned to adapt to the specific task at hand.

Another crucial component introduced in this process is the GATED XATTN-DENSE layer, composed of feed-forward and cross-attention components. This layer acts as a bridge between existing and frozen LM layers, enabling the language model to optimize its attention towards visual tokens while generating text tokens.

Beyond the GATED XATTN-DENSE layers, several techniques can potentially enhance the performance of Large Language Models (LLMs) in multimodal models. Some of these techniques include hierarchical fusion, dynamic fusion methods, adversarial training, and more.

For example, in another paper [], researchers propose the Structure-Content Neural Language Model (SC-NLM), which introduces a novel approach to manage the dichotomy of sentence structure and content. By separating semantics and syntax, this study demonstrates the potential for achieving improved model performances.

Loss Function

CLIP

L contrastive txt2img =

$$\frac{-1}{N}\sum_{i}^{N}\log(\frac{\exp(L_i^T V_i \beta)}{\sum_j^N \exp(L_i^T V_j \beta)})$$

L contrastive txt2img =

$$\frac{-1}{N}\sum_{i}^{N}\log(\frac{\exp(V_i^T L_i B)}{\sum_j^N \exp(V_i^T L_j \beta)})$$

The objective is to minimize the combined loss of these two components, where β represents a trainable inverse temperature parameter, V_i represents a given image embedding, L_i represents a given text embedding, and N is the number of image-text pairs.

Flamingo :

Calculates the probability of text y given the interleaved images and videos x

$$p(y|x) = \prod_{l=1}^{N} p(y_l|y_{<l}, x_{\leq l})$$

The training loss function was a weighted sum of expected negative log-likelihoods of generated text across all 4 datasets, with λ_m being the training weight of dataset m.

$$\sum_{m=1}^{M} \lambda_m E_{(x,y)\sim D_m}\left[-\sum_{l=1}^{L}\log p(y|x)\right]$$

5 Discussion

LLM and MLLM demonstrate remarkable language understanding capabilities and exhibit versatility in performing various tasks. However, their objectives and final users are frequently ambiguous or extensive. It becomes challenging to conduct thorough testing without clearly delineating the model's intended functions.

The current landscape of AI is predominantly steered by research aimed at optimizing model accuracy, with progress often equated to enhanced prediction quality and accuracy. However, this focus has ushered in a decade where AI advancement has heavily relied on extensive computational resources, often at the expense of environmental considerations and resource efficiency. Data scaling, the prevalent method for boosting model quality, prioritizes enlarging the training dataset over algorithmic optimization. However, this approach carries significant environmental consequences. To manage training times, system resources must scale with dataset size, increasing embodied and operational carbon footprints. Alternatively, keeping system resources fixed extends training time, amplifying the operational energy footprint. [10]

Navigating multimodal tasks requires understanding data interactions and complexities in human communication. Effective handling involves respecting specific properties: Firstly, assessing system performance poses challenges, particularly in identifying systematic failures due to interconnectivity, leading to complex error detection and correction. Secondly, the effectiveness of multimodal linguistic models hinges on design and training decisions, necessitating optimal data mix and evaluation metrics selection for accurate measurement of effectiveness. Lastly, achieving reliable performance and comprehension demands sophisticated modelling of multimodal data. Acknowledging and addressing these properties enhances the robustness and effectiveness of multimodal systems across applications.[6]

Existing evaluation of LLMs spans diverse tasks, revealing their strengths and weaknesses. While excelling in sentiment analysis and text classification, LLMs encounter hurdles in semantic understanding and common-sense reasoning. They show promise in mathematical and logical reasoning but struggle with multilingual contexts and non-Latin languages. Assessing factuality is pivotal, with various methods proposed for alignment with real-world truths. Evaluations also scrutinize robustness, ethics, biases, and trustworthiness, highlighting LLMs' susceptibility to biases and cognitive errors. Addressing these challenges is imperative for enhancing LLMs' reliability and performance.

6 Bibliography

[1] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., ... and Sutskever, I. (2021). Learning Transferable Visual Models From Natural Language Supervision. arXiv preprint arXiv:2103.00020.
ACM Computing Surveys (CSUR), 56(2):Article 30, 2023.
[2]Zhao, Wayne Xin, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, et al. "A Survey of Large Language Models." arXiv, November 24, 2023.
[3]Kiros, Ryan, Ruslan Salakhutdinov, and Rich Zemel. "Multimodal Neural Language Models." In Proceedings of the 31st International Conference on Machine Learning, 595–603. PMLR, 2014.
[4] Huang, Shaohan, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, and Barun Patra. "Language Is Not All You Need: Aligning Perception with Language Models." Advances in Neural Information Processing Systems 36 (2024).
[5]Shengbang Tong and Erik Jones and Jacob Steinhardt , 2023 , "Mass-producing Failures of Multimodal Systems with Language Models" , 2306.12105.
[6]Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. "Recent Advances in Natural Language Processing via Large Pre-trained Language Models: A Survey." [7] Kiros, Ryan, Ruslan Salakhutdinov, and Rich Zemel. "Multimodal Neural Language Models." In Proceedings of the 31st International Conference on Machine Learning, 595–603. PMLR, 2014. [8] Xu, Kelvin, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention." In Proceedings of the 32nd International Conference on Machine Learning, 2048–57. PMLR, 2015. [9] Yan Zeng , Hanbo Zhang , Jiani Zheng ,Jiangnan Xia, Guoqiang Wei, Yang Wei, Yuchen Zhang, Tao Kong . "What Matters in Training a GPT4-Style Language Model with Multimodal Inputs?". [10] Wu, Carole-Jean, Ramya Raghavendra, Udit Gupta, Bilge Acun, Newsha Ardalani, Kiwan Maeng, Gloria Chang, et al. "Sustainable AI: Environmental Implications, Challenges and Opportunities." Proceedings of Machine Learning and Systems 4 (April 22, 2022): 795–813.
[11]Lu, J., Batra, D., Parikh, D., and Lee, S. (2019). Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 2137-2146).