JOURNAL OF MULTIDISCIPLINARY SCIENCES AND INNOVATIONS
VOLUME 5, ISSUE 05
MONTHLY JOURNALS
ISSN NUMBER: 2751-4390
IMPACT FACTOR: 9,08

ANALYTICAL REVIEW OF LARGE LANGUAGE MODEL ARCHITECTURES

Raxmanov Mirkomil
Laboratory of Astronomy, Cosmology, and Space Technologies at Institute for Advanced Studies of New Uzbekistan University.
m.raxmanov@newuu.uz
https://doi.org/10.5281/zenodo.20477615

Abstract: Large Language Models (LLMs) have become the foundation of modern Artificial Intelligence systems, enabling breakthroughs in natural language understanding, reasoning, code generation, multimodal learning, and autonomous agents. Recent advances in Transformer-based architectures have significantly improved model capabilities, scalability, and generalization performance. This paper presents a comprehensive analytical review of modern LLM architectures, tracing their evolution from early neural language models to contemporary frontier systems such as GPT, Claude, Gemini, LLaMA, DeepSeek, and Mistral. The study examines core architectural components including attention mechanisms, positional encoding, Mixture-of-Experts (MoE), retrieval-augmented generation (RAG), multimodal extensions, and reasoning-enhanced designs. Furthermore, the paper discusses the strengths and limitations of current architectures and highlights future research directions toward efficient, trustworthy, and autonomous AI systems.

Keywords: Large Language Models, Transformer Architecture, Generative AI, Mixture-of-Experts, Retrieval-Augmented Generation, Artificial Intelligence, Deep Learning.

Аннотация: Большие языковые модели (LLM) стали основой современных систем искусственного интеллекта, обеспечивая прорывы в понимании естественного языка, логическом выводе, генерации программного кода, мультимодальном обучении и автономных агентах. Недавние достижения в архитектурах на основе Transformer значительно улучшили возможности моделей, их масштабируемость и способность к обобщению. В данной статье представлен всесторонний аналитический обзор современных архитектур LLM, прослеживающий их эволюцию от ранних нейронных языковых моделей до современных передовых систем, таких как GPT, Claude, Gemini, LLaMA, DeepSeek и Mistral. Исследование рассматривает ключевые архитектурные компоненты, включая механизмы внимания (attention), позиционное кодирование, архитектуру Mixture-of-Experts (MoE), генерацию с дополнением поиска (Retrieval-Augmented Generation, RAG), мультимодальные расширения и модели с улучшенными механизмами рассуждения. Кроме того, в статье обсуждаются сильные стороны и ограничения современных архитектур, а также выделяются перспективные направления будущих исследований, направленных на создание эффективных, надежных и автономных систем искусственного интеллекта.

Ключевые слова: Большие языковые модели, архитектура Transformer, генеративный искусственный интеллект, Mixture-of-Experts, Retrieval-Augmented Generation, искусственный интеллект, глубокое обучение.

Annotatsiya: Katta Til Modellari (LLM) zamonaviy sun’iy intellekt tizimlarining asosiga aylandi hamda tabiiy tilni tushunish, mantiqiy fikrlash, dastur kodini yaratish, multimodal o‘rganish va avtonom agentlar sohalarida muhim yutuqlarga erishishga imkon berdi. Transformer arxitekturasiga asoslangan so‘nggi yutuqlar modellarning imkoniyatlari, masshtablanuvchanligi va umumlashtirish qobiliyatini sezilarli darajada yaxshiladi. Ushbu maqolada zamonaviy LLM arxitekturalarining keng qamrovli tahliliy sharhi taqdim etilib,

ularning dastlabki neyron til modellaridan tortib GPT, Claude, Gemini, LLaMA, DeepSeek va Mistral kabi zamonaviy ilg‘or tizimlargacha bo‘lgan evolyutsiyasi yoritiladi. Tadqiqot diqqat mexanizmlari (attention), pozitsion kodlash, Mixture-of-Experts (MoE), qidiruv bilan boyitilgan generatsiya (Retrieval-Augmented Generation, RAG), multimodal kengaytmalar hamda fikrlash qobiliyati kuchaytirilgan dizaynlar kabi asosiy arxitektura komponentlarini tahlil qiladi. Shuningdek, maqolada hozirgi arxitekturalarning kuchli va zaif tomonlari muhokama qilinib, samarali, ishonchli va avtonom sun’iy intellekt tizimlarini yaratishga qaratilgan kelajakdagi tadqiqot yo‘nalishlari ko‘rsatib beriladi.

Kalit so‘zlar: Katta Til Modellari, Transformer arxitekturasi, generativ sun’iy intellekt, Mixture-of-Experts, Retrieval-Augmented Generation, sun’iy intellekt, chuqur o‘rganish.

1. Introduction

The emergence of Large Language Models (LLMs) has fundamentally transformed the field of Artificial Intelligence. Unlike traditional Natural Language Processing (NLP) systems that relied on handcrafted features and task-specific architectures, modern LLMs learn language representations directly from massive datasets through self-supervised learning[1,3].

The release of Transformer-based models has enabled machines to generate coherent text, answer questions, write software code, perform reasoning tasks, and interact with users in natural language[2]. Today, LLMs serve as the foundation for intelligent assistants, search systems, scientific research tools, and autonomous AI agents.

The rapid progress of models such as GPT, Claude, Gemini, LLaMA, DeepSeek, and Mistral has created intense competition in architecture design, model efficiency, and reasoning capabilities. Understanding the architectural foundations behind these systems is essential for both researchers and practitioners[4,5,6].

2. Motivation

Several factors motivate the study and research of LLM architectures:

- Increasing demand for intelligent conversational systems.
- Growing importance of AI-powered decision support.
- Need for efficient deployment of large-scale models.
- Development of autonomous AI agents.
- Integration of language, vision, audio, and video processing[7].

Modern LLMs are no longer limited to text generation. They now function as reasoning engines capable of interacting with external tools, databases, and software environments. Consequently, architectural innovations have become critical for improving performance, scalability, and reliability[8].

3. Historical Evolution of Language Models

The development of language models has undergone several transformative stages over the past decades. Each generation introduced new mechanisms for representing linguistic information, capturing contextual dependencies, and improving generalization capabilities[10,11]. The evolution can be broadly categorized into four major phases: Statistical Language Models, Neural Language Models, Recurrent Neural Architectures, and Transformer-based Architectures. Figure 1 illustrates the chronological evolution of language model architectures.

JOURNAL OF MULTIDISCIPLINARY SCIENCES AND INNOVATIONS
VOLUME 5, ISSUE 05
MONTHLY JOURNALS
ISSN NUMBER: 2751-4390
IMPACT FACTOR: 9,08

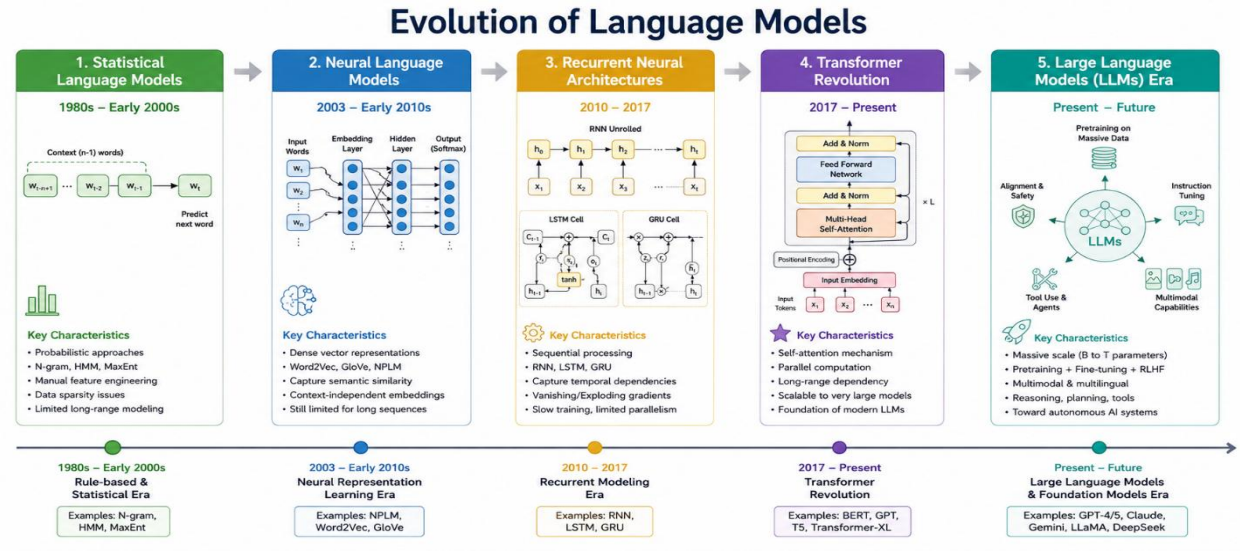

Figure 1. Evolution of Language Models (1980–Present)

3.1 Statistical Language Models

The earliest language models were based on probabilistic approaches designed to estimate the likelihood of a sequence of words[12]. These methods dominated Natural Language Processing (NLP) research from the 1980s until the early 2000s[6,13].

Statistical language modeling relies on the chain rule of probability:

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

However, calculating the complete conditional probability distribution is computationally infeasible due to the enormous vocabulary size. Consequently, simplified assumptions were introduced.

3.1.1 N-gram Models

N-gram models estimate the probability of a word based on a limited number of preceding words[14].

For a trigram model:

$$P(w_i|w_1,\ldots,w_{i-1})\approx P(w_i|w_{i-2},w_{i-1})$$

Architecture

JOURNAL OF MULTIDISCIPLINARY SCIENCES AND INNOVATIONS
VOLUME 5, ISSUE 05
MONTHLY JOURNALS
ISSN NUMBER: 2751-4390
IMPACT FACTOR: 9,08

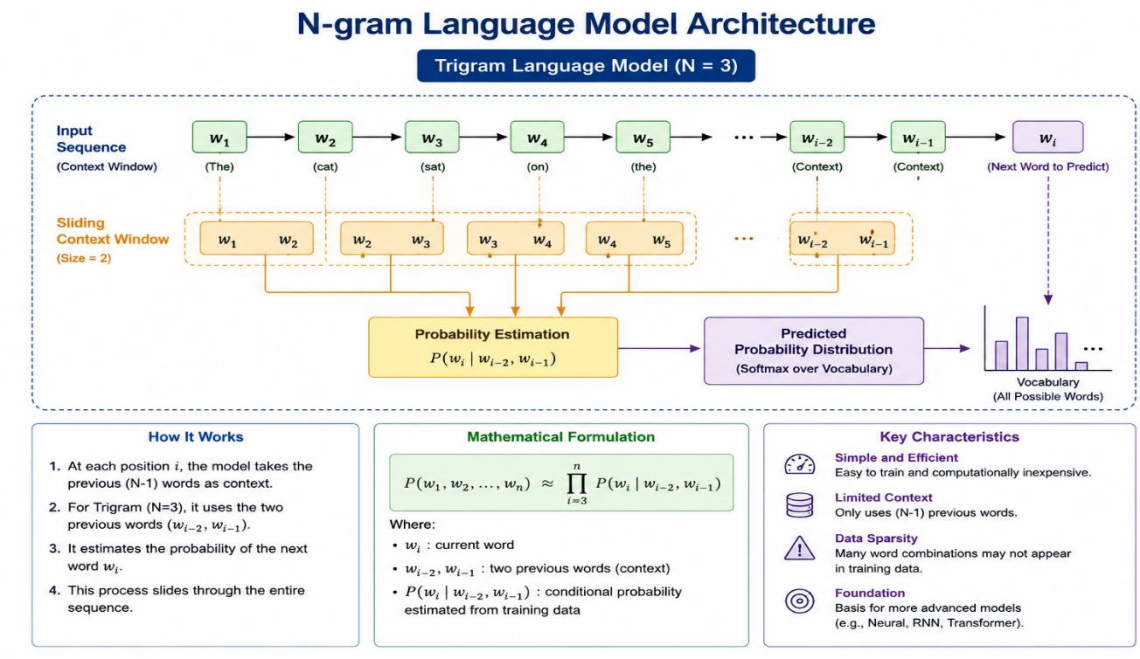

Figure 2. N-gram Language Model Architecture

N-gram models are computationally simple and interpretable. However, they suffer from severe data sparsity because many word combinations never appear in training data[15,16].

3.1.2 Hidden Markov Models (HMMs)

Hidden Markov Models introduced latent states that represent underlying linguistic structures.

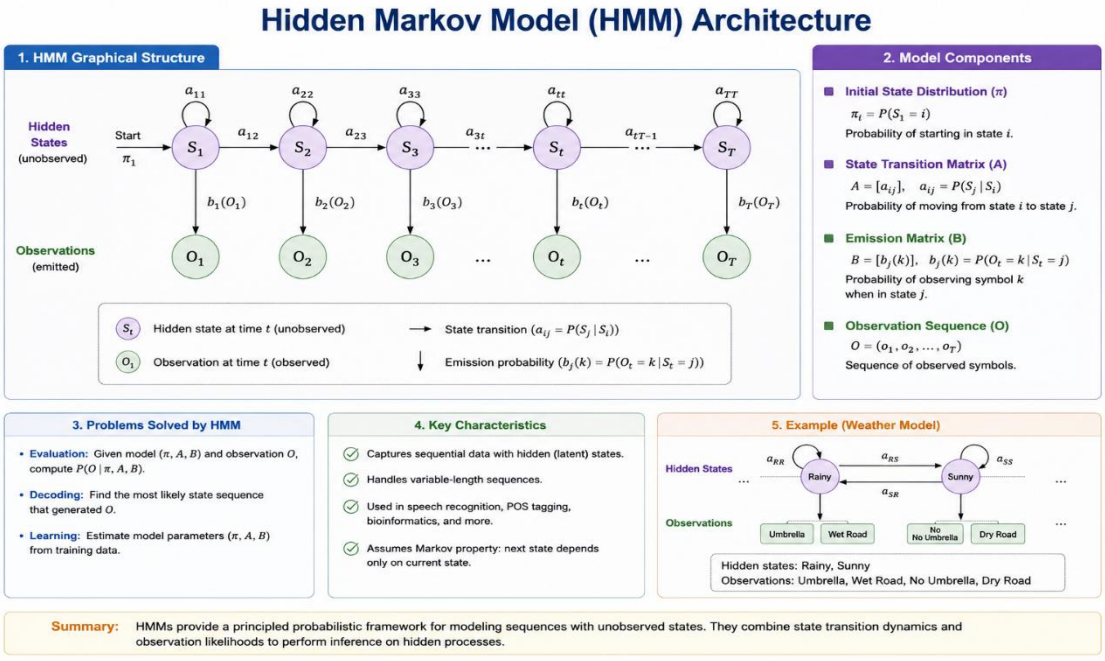

Figure 3. Hidden Markov Model Architecture

The model consists of:

- Hidden states
- Transition probabilities

- Emission probabilities

HMMs became widely used in speech recognition and sequence labeling tasks but struggled to represent long-range dependencies[17].

3.1.3 Maximum Entropy Models

Maximum Entropy (MaxEnt) models improved statistical language modeling by incorporating multiple contextual features.

The probability distribution is defined as:

$$P(w|h)=\frac{1}{Z(h)}\exp\left(\sum_i \lambda_i f_i(h,w)\right)$$

where:

- $f_i(h,w)$ are feature functions,
- λ_i are learned weights,
- $Z(h)$ is the normalization factor.

Although more flexible than N-grams, MaxEnt models still relied heavily on manual feature engineering[18].

3.2 Neural Language Models

The introduction of Artificial Neural Networks marked a significant paradigm shift in language modeling. Instead of treating words as discrete symbols, neural models represented words as dense continuous vectors known as embeddings.

The pioneering work by Bengio et al. (2003) introduced the Neural Probabilistic Language Model (NPLM), which learned distributed word representations automatically[19].

3.2.1 Neural Probabilistic Language Model

The Neural Probabilistic Language Model (NPLM), introduced by Bengio et al. in 2003, marked a significant milestone in the evolution of language modeling. Prior statistical approaches such as N-gram models relied heavily on frequency counts and suffered from severe data sparsity issues due to the exponential growth of possible word combinations. NPLM addressed these limitations by introducing distributed word representations and neural network-based probability estimation, laying the foundation for modern neural language processing systems[20].

The fundamental idea behind NPLM is to represent words as continuous dense vectors, known as word embeddings, instead of discrete symbolic identifiers[21]. By mapping semantically similar words to nearby locations in a low-dimensional vector space, the model can generalize better to unseen word sequences and capture linguistic regularities that traditional statistical models cannot effectively represent. The architecture consists of:

1. Input Layer
2. Embedding Layer
3. Hidden Layer
4. Output Softmax Layer

JOURNAL OF MULTIDISCIPLINARY SCIENCES AND INNOVATIONS
VOLUME 5, ISSUE 05
MONTHLY JOURNALS
ISSN NUMBER: 2751-4390
IMPACT FACTOR: 9,08

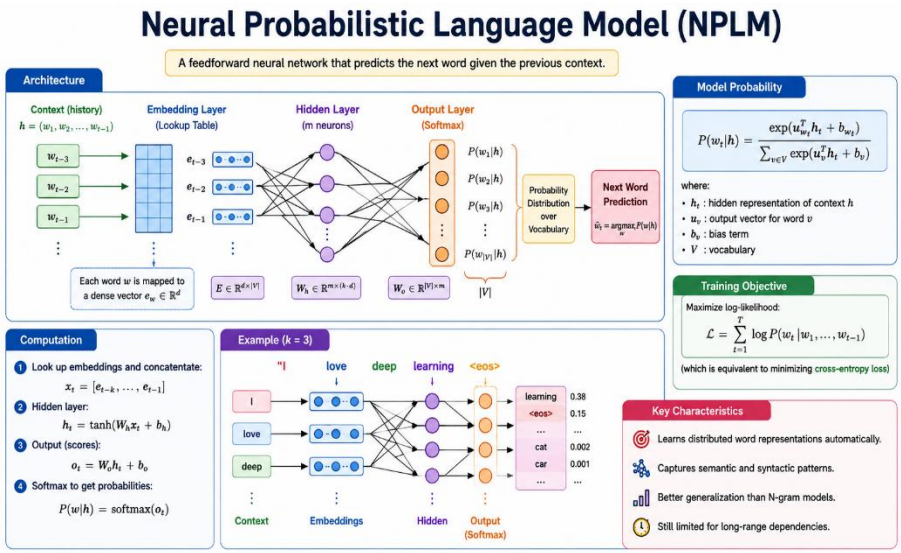

Figure 4. Neural Probabilistic Language Model

The embedding layer transforms words into low-dimensional continuous vectors:

$$w_i \rightarrow e_i \in \mathbb{R}^d$$

This innovation enabled semantic similarity learning and improved generalization[23].

3.2.2 Word2Vec

Word2Vec is one of the most influential word representation learning techniques in Natural Language Processing (NLP). Introduced by Mikolov et al. at Google in 2013, Word2Vec significantly advanced neural language modeling by providing an efficient method for learning distributed word representations from large text corpora. Unlike earlier Neural Probabilistic Language Models (NPLMs), which required computationally expensive hidden layers, Word2Vec employs a lightweight neural architecture that can learn high-quality word embeddings at a much larger scale[24,25,26].

The primary objective of Word2Vec is to map words into a continuous vector space where semantically and syntactically similar words are located close to one another. These learned vector representations capture meaningful linguistic relationships and can be utilized in various downstream NLP tasks, including text classification, machine translation, sentiment analysis, information retrieval, and question-answering systems.

The fundamental principle behind Word2Vec is the Distributional Hypothesis, which states that words appearing in similar contexts tend to have similar meanings. By analyzing neighboring words within a context window, Word2Vec learns vector representations that encode semantic information directly from raw text data[27].

Word2Vec successfully captured semantic relationships such as:

$$\text{Vector}(King) - \text{Vector}(Man) + \text{Vector}(Woman) \approx \text{Vector}(Queen)$$

3.2.3 GloVe

Global Vectors (GloVe) combined matrix factorization and local context learning.

The objective function utilizes global word co-occurrence statistics:

$$J = \sum_{i,j} f(X_{ij}) \left(w_i^T w_j + b_i + b_j - \log X_{ij} \right)^2$$

where X_{ij} denotes word co-occurrence frequency.

Despite substantial improvements, these models exhibited several limitations:

- Static word representations.

- Context-independent embeddings.
- Poor handling of long sequences.
- Limited memory capacity.

These shortcomings led to the development of recurrent architectures.

3.3 Recurrent Neural Architectures

Recurrent Neural Architectures were developed to overcome the limitations of fixed-context neural language models by introducing memory mechanisms capable of processing sequential data. Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Gated Recurrent Units (GRUs) maintain hidden states that allow information from previous time steps to influence future predictions, making them highly effective for language modeling and sequence learning tasks. Although these architectures significantly improved the modeling of temporal dependencies, they suffered from challenges such as vanishing gradients, limited parallelization, and computational inefficiency, which eventually led to the emergence of Transformer-based architectures[28].

3.3.1 Recurrent Neural Networks (RNNs)

The hidden state update is defined as:

$$h_t = f(W_h h_{t-1} + W_x x_t + b)$$

where:

- x_t are current input,
- h_t are hidden state.

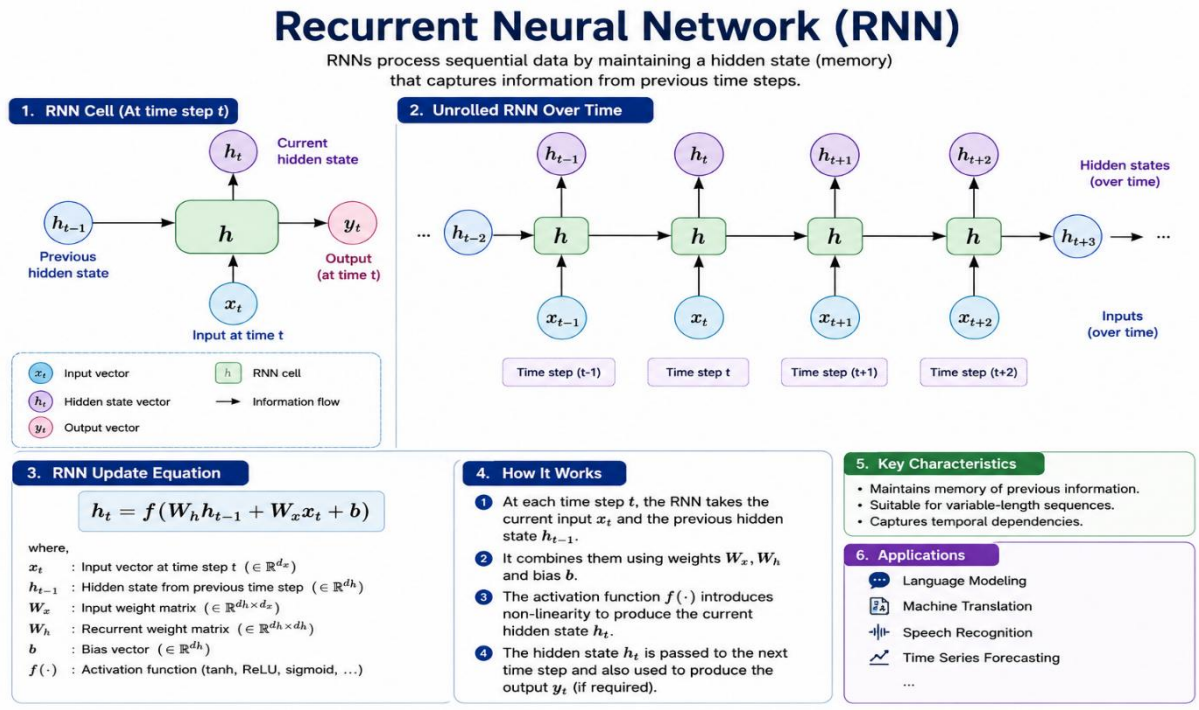

Figure 7. Recurrent Neural Network

The recurrent mechanism allows the network to model temporal dependencies by propagating information through successive hidden states[29]. As shown in the unrolled architecture, the same RNN cell is repeatedly applied across multiple time steps, sharing the same set of parameters throughout the sequence. This parameter sharing significantly reduces the number of trainable parameters and enables the network to process sequences of variable length.

3.3.2 Long Short-Term Memory (LSTM)

LSTM networks were proposed to overcome the vanishing gradient problem. They introduce memory cells and gating mechanisms.

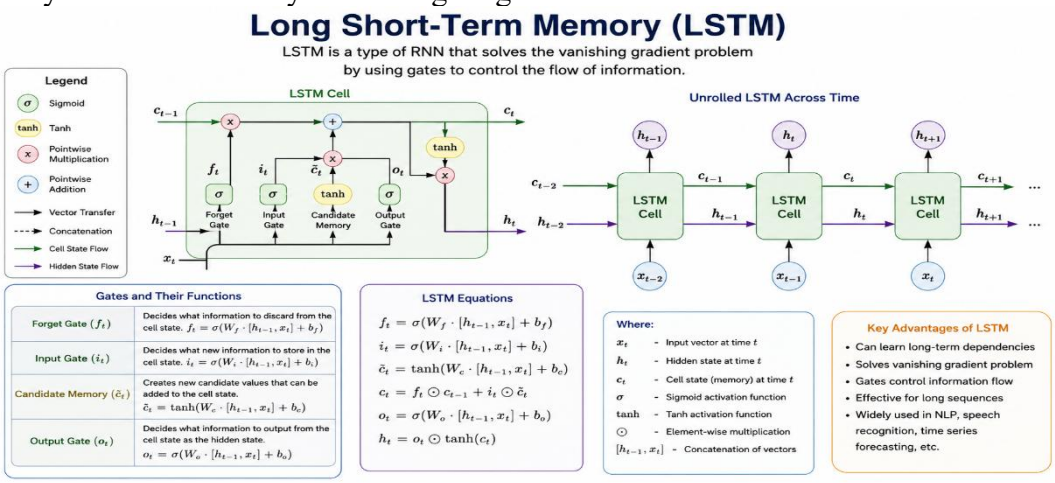

Figure 8. LSTM Cell Architecture

Long Short-Term Memory (LSTM) is an advanced type of Recurrent Neural Network (RNN) introduced by Hochreiter and Schmidhuber in 1997 to address the vanishing gradient problem encountered in traditional RNNs. Unlike standard recurrent networks, LSTMs incorporate a specialized memory structure known as a cell state, which enables the network to preserve and selectively update information over long sequences. This capability allows LSTMs to effectively capture long-range dependencies in sequential data, making them particularly suitable for language modeling, machine translation, speech recognition, and time-series forecasting[30,31].

3.3.3 Gated Recurrent Units (GRUs)

GRUs simplified LSTM architectures while maintaining competitive performance. Gated Recurrent Units (GRUs) are a simplified variant of Long Short-Term Memory (LSTM) networks introduced by Cho et al. in 2014 to improve the ability of Recurrent Neural Networks (RNNs) to capture long-term dependencies while reducing computational complexity.

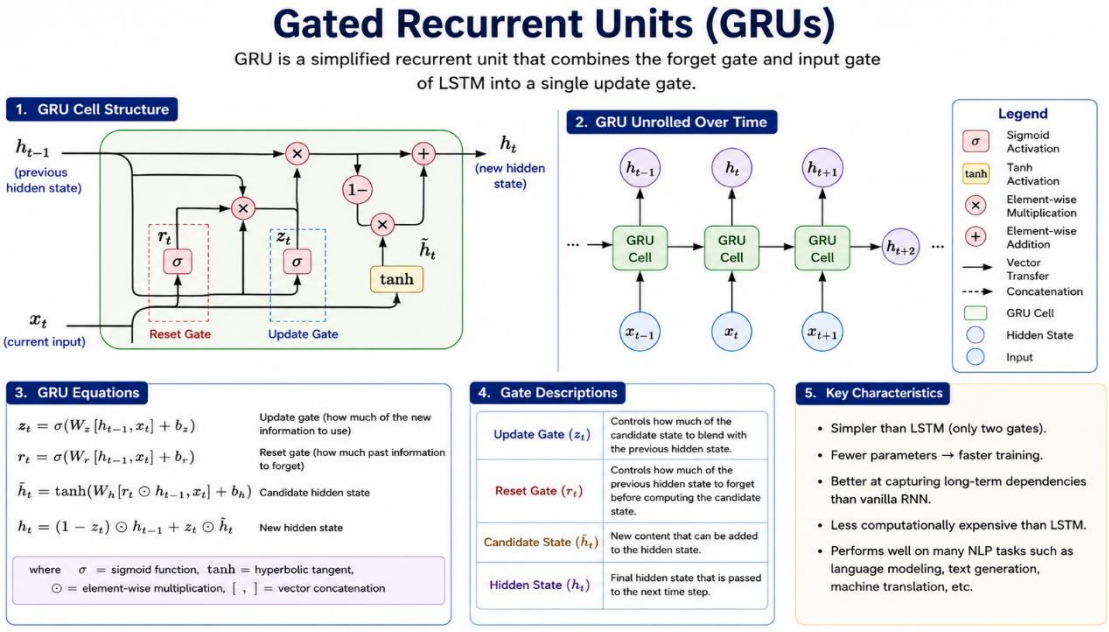

Figure 9. GRU Architecture

Similar to LSTMs, GRUs employ gating mechanisms to regulate the flow of information through the network. However, unlike LSTMs, which utilize separate cell states and three gating

units, GRUs combine memory and hidden states into a single representation and use only two gates: the Update Gate and the Reset Gate. This streamlined design reduces the number of trainable parameters and improves training efficiency without significantly sacrificing performance.

3.4 Transformer Revolution

A breakthrough occurred in 2017 with the publication of *Attention Is All You Need* by Vaswani et al. The Transformer architecture eliminated recurrence entirely and replaced it with self-attention mechanisms[32].

3.4.1 Self-Attention Mechanism

The Scaled Dot-Product Attention mechanism computes contextual representations by measuring the similarity between query and key vectors and using the resulting attention weights to aggregate information from value vectors. By allowing each token to directly attend to all other tokens in a sequence, attention overcomes the limitations of recurrent architectures in modeling long-range dependencies.The attention operation is defined as:

$$Attention(Q,K,V)=Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

where:

- Q = Query matrix
- K = Key matrix
- V = Value matrix

This mechanism enables every token to attend directly to all other tokens.

3.4.2 Transformer Architecture

The Transformer architecture, introduced by Vaswani et al. in the landmark paper *"Attention Is All You Need"* (2017), represents one of the most significant breakthroughs in the history of Artificial Intelligence and Natural Language Processing. Unlike Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks, which process sequences sequentially, the Transformer eliminates recurrence entirely and relies on self-attention mechanisms to model dependencies between tokens. This innovation enables efficient parallel processing, improved scalability, and superior performance on long-range contextual learning tasks.

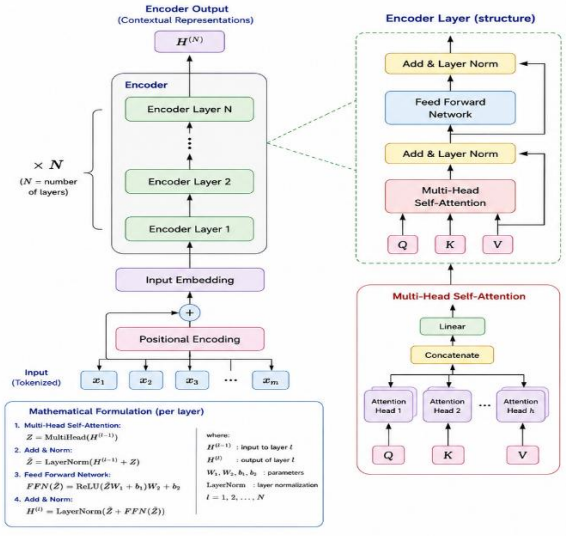

Figure 10. Transformer Encoder Architecture

Transformers scale efficiently to billions and even trillions of parameters. Modern models including GPT, Claude, Gemini, LLaMA, DeepSeek, and Mistral are all based on Transformer variants[33].

The Transformer architecture initiated the era of Foundation Models and Large Language Models. Through large-scale pretraining and fine-tuning, Transformers achieved unprecedented performance across NLP, computer vision, multimodal learning, reasoning, and autonomous agent systems.

Consequently, the Transformer has become the dominant architecture underlying contemporary Artificial Intelligence research and industrial applications.

4. Core Architecture of Modern LLMs

Modern Large Language Models (LLMs) are primarily built upon the Transformer architecture, which serves as the foundational framework for contemporary language understanding and generation systems. Since the introduction of the Transformer in 2017, numerous architectural enhancements have been proposed to improve scalability, contextual understanding, computational efficiency, reasoning capabilities, and multimodal integration. Despite differences among models such as GPT, Claude, Gemini, LLaMA, DeepSeek, and Mistral, most modern LLMs share a common set of core architectural components that enable them to process vast amounts of textual information and learn complex linguistic patterns[34,35].

4.1 Transformer Architecture

The Transformer architecture fundamentally transformed the field of machine learning by replacing recurrence with attention-based computation. It established the foundation for modern foundation models and Large Language Models, including GPT, BERT, T5, Claude, Gemini, LLaMA, DeepSeek, and Mistral. Through large-scale pretraining and self-supervised learning, Transformer-based models have achieved state-of-the-art performance in language understanding, text generation, reasoning, coding, multimodal learning, and autonomous AI systems.

Consequently, the Transformer architecture is widely regarded as the most influential neural network architecture of the modern AI era and serves as the backbone of virtually all contemporary Large Language Models.

4.2 Self-Attention Mechanism

Self-attention computes relationships among tokens by generating:

Query vectors (Q)

Key vectors (K)

Value vectors (V)

The attention score determines how much information should be exchanged between tokens.

This mechanism computes attention weights that determine the relative importance of each token in the sequence.

To improve representation learning, Transformers employ Multi-Head Attention, which enables the model to simultaneously capture different types of relationships, including syntactic structures, semantic dependencies, and long-range contextual interactions.

A key advantage of the Transformer architecture is its ability to process all tokens in parallel, significantly reducing training time compared to recurrent architectures. Furthermore, direct token-to-token attention eliminates the information bottleneck associated with long sequences, enabling the model to effectively capture long-range dependencies[36].

4.3 Positional Encoding

One of the fundamental challenges of the Transformer architecture is the absence of an inherent mechanism for processing sequential order. Unlike Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks, which process tokens sequentially and

naturally preserve positional information, Transformers process all tokens simultaneously through self-attention. Consequently, the model requires an explicit method to represent the position of each token within a sequence. This challenge is addressed through Positional Encoding, a technique that injects positional information into token embeddings before they are processed by the attention layers.

Positional Encoding enables the model to distinguish between tokens appearing in different positions and to capture sequential relationships among words. Without positional information, a Transformer would treat a sentence as an unordered collection of tokens, making it impossible to understand syntax, grammar, or word order.

Positional Encoding is a critical component of Transformer-based architectures. By incorporating sequence order into token representations, it enables self-attention mechanisms to model syntactic and semantic relationships effectively, making it an essential building block of modern Large Language Models[37].

5. Conclusion

Large Language Models represent one of the most significant advancements in Artificial Intelligence. Their success is largely driven by architectural innovations centered around the Transformer paradigm. Recent developments such as Mixture-of-Experts, Retrieval-Augmented Generation, multimodal processing, and agent-oriented frameworks have expanded the capabilities of LLMs far beyond traditional language generation.

As the field continues to evolve, future architectures will likely emphasize efficiency, reasoning, autonomy, and trustworthiness. Understanding these architectural foundations is therefore essential for researchers, developers, and organizations seeking to leverage the full potential of next-generation AI systems.

References

[1] C. E. Shannon, “A Mathematical Theory of Communication,” *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.
[2] F. Jelinek, *Statistical Methods for Speech Recognition*. Cambridge, MA, USA: MIT Press, 1997.
[3] L. R. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition,” *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
[4] A. Berger, S. Della Pietra, and V. Della Pietra, “A Maximum Entropy Approach to Natural Language Processing,” *Computational Linguistics*, vol. 22, no. 1, pp. 39–71, 1996.
[5] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, “A Neural Probabilistic Language Model,” *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.
[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” arXiv:1301.3781, 2013.
[7] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed Representations of Words and Phrases and Their Compositionality,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2013.
[8] J. Pennington, R. Socher, and C. D. Manning, “GloVe: Global Vectors for Word Representation,” in *Proceedings of EMNLP*, 2014, pp. 1532–1543.
[9] J. L. Elman, “Finding Structure in Time,” *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.
[10] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to Forget: Continual Prediction with LSTM," *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, 2000.
[12] K. Cho et al., "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," arXiv:1406.1078, 2014.
[13] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," arXiv:1412.3555, 2014.
[14] A. Vaswani et al., "Attention Is All You Need," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
[15] A. Radford et al., "Improving Language Understanding by Generative Pre-Training," OpenAI, 2018.
[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of NAACL-HLT*, 2019.
[17] A. Radford et al., "Language Models are Unsupervised Multitask Learners," OpenAI Technical Report, 2019.
[18] T. Brown et al., "Language Models are Few-Shot Learners," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
[19] C. Raffel et al., "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.
[20] J. Dai et al., "Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context," in *Proceedings of ACL*, 2019.
[21] Z. Yang et al., "XLNet: Generalized Autoregressive Pretraining for Language Understanding," in *Advances in Neural Information Processing Systems*, 2019.
[22] T. B. Brown et al., "GPT-3: Language Models are Few-Shot Learners," arXiv:2005.14165, 2020.
[23] H. Touvron et al., "LLaMA: Open and Efficient Foundation Language Models," arXiv:2302.13971, 2023.
[24] H. Touvron et al., "LLaMA 2: Open Foundation and Fine-Tuned Chat Models," arXiv:2307.09288, 2023.
[25] A. Chowdhery et al., "PaLM: Scaling Language Modeling with Pathways," arXiv:2204.02311, 2022.
[26] R. Anil et al., "PaLM 2 Technical Report," arXiv:2305.10403, 2023.
[27] Anthropic, "Claude: Constitutional AI and Large Language Models," 2023.
[28] Google DeepMind, "Gemini: A Family of Highly Capable Multimodal Models," arXiv:2312.11805, 2023.
[29] A. Jiang et al., "Mistral 7B," arXiv:2310.06825, 2023.
[30] DeepSeek-AI, "DeepSeek LLM: Scaling Open-Source Language Models with Long Context," 2024.
[31] S. Shazeer et al., "Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer," arXiv:1701.06538, 2017.
[32] N. Shazeer, "Fast Transformer Decoding: One Write-Head is All You Need," arXiv:1911.02150, 2019.
[33] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in *Advances in Neural Information Processing Systems*, 2020.
[34] J. Su et al., "RoFormer: Enhanced Transformer with Rotary Position Embedding," arXiv:2104.09864, 2021.
[35] O. Press, N. A. Smith, and M. Lewis, "Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation," arXiv:2108.12409, 2021.

[36] S. Bubeck et al., "Sparks of Artificial General Intelligence: Early Experiments with GPT-4," arXiv:2303.12712, 2023.
[37] OpenAI, "GPT-4 Technical Report," arXiv:2303.08774, 2023.