* Transformer model vs. MoE (Mixture-of-Experts Layer) model?
  + Transformer:
* GPT model behind? Is it Transformer? Predicting the next word related?
* Decoder mode vs. others in Transformer model?
* Decoder-only architecture like GPT models?
* Gemini uses Transformer & MoE model research
* ~~Multi-headed attention vs. self-attention in Transformer?~~
* ~~Does GPT use RNN (recurrent neural network)?~~
* GPT: “we fine-tune the model’s behavior using reinforcement learning with human feedback ([RLHF⁠](https://openai.com/index/learning-from-human-preferences/)).”
* Does Gemini have one?
* ~~Convolutional vs. recurrent neural networks~~
* If all AI are developed based on Transformer model, then how some is better than others?
* <https://openai.com/index/gpt-4-research/>
* [~~MoE Research~~](https://arxiv.org/abs/1701.06538) ~~Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer (BEGINNING OF MOE)~~
* [Transformer Research](https://research.google/blog/transformer-a-novel-neural-network-architecture-for-language-understanding/)
* [~~https://blog.google/technology/ai/google-gemini-next-generation-model-february-2024/#architecture~~](https://blog.google/technology/ai/google-gemini-next-generation-model-february-2024/#architecture)
* [~~https://www.larksuite.com/en\_us/topics/ai-glossary/problem-that-ai-is-trying-to-solve~~](https://www.larksuite.com/en_us/topics/ai-glossary/problem-that-ai-is-trying-to-solve)
* [Llama 3 Herd of Models](https://www.youtube.com/watch?v=uXt6rYXnV8U) - VIDEO
  + <https://arxiv.org/pdf/2407.21783>
* [~~https://www.cloudflare.com/learning/ai/what-is-large-language-model/~~](https://www.cloudflare.com/learning/ai/what-is-large-language-model/)
* [~~https://aws.amazon.com/what-is/large-language-model/~~](https://aws.amazon.com/what-is/large-language-model/) ~~“ The underlying transformer is a set of~~[~~neural networks~~](https://aws.amazon.com/what-is/neural-network/)~~that consist of~~ **~~an encoder and a decoder~~** ~~with self-attention capabilities.”~~
* <https://medium.com/data-science-at-microsoft/how-large-language-models-work-91c362f5b78f>
* [~~https://www.architectureandgovernance.com/applications-technology/mixture-of-experts-moe-architecture-a-deep-dive-and-comparison-of-top-open-source-offerings/~~](https://www.architectureandgovernance.com/applications-technology/mixture-of-experts-moe-architecture-a-deep-dive-and-comparison-of-top-open-source-offerings/)
* <https://medium.com/towards-data-science/understanding-googles-switch-transformer-904b8bf29f66>
* [~~Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity~~](https://arxiv.org/abs/2101.03961)
* [FastMoE (Meta)](https://arxiv.org/abs/2103.13262)
  + Open-source MoE model instead of Switch Transformer from GEMINI
* [LLaMA: Open and Efficient Foundation Language Models](https://arxiv.org/abs/2302.13971)
* [Introducing Meta Llama 3: The most capable openly available LLM to date](https://ai.meta.com/blog/meta-llama-3/)
* [Mixtures of Experts Unlock Parameter Scaling for Deep RL](https://arxiv.org/abs/2402.08609?ref=maginative.com)
  + [Deep RL](https://www.geeksforgeeks.org/what-is-reinforcement-learning/)
* [DeepSeekMoE: Towards Ultimate Expert Specialization in Mixture-of-Experts Language Models](https://arxiv.org/abs/2401.06066)
* [~~https://www.architectureandgovernance.com/applications-technology/mixture-of-experts-moe-architecture-a-deep-dive-and-comparison-of-top-open-source-offerings/~~](https://www.architectureandgovernance.com/applications-technology/mixture-of-experts-moe-architecture-a-deep-dive-and-comparison-of-top-open-source-offerings/)
* General:
  + Artificial intelligence is the field of computer science that researches methods of giving machines the ability to perform tasks that require human intelligence. Machine learning is an artificial intelligence technique that gives computers access to very large datasets and teaches them to learn from this data. Machine learning software finds patterns in existing data and applies those patterns to new data to make intelligent decisions. Deep learning is a subset of machine learning that uses deep learning networks to process data.
  + Traditional machine learning methods require human input for the machine learning software to work sufficiently well. A data scientist manually determines the set of relevant features that the software must analyze. This limits the software’s ability, which makes it tedious to create and manage.
  + On the other hand, in deep learning, the data scientist gives only raw data to the software. The deep learning network derives the features by itself and learns more independently. It can analyze unstructured datasets like text documents, identify which data attributes to prioritize, and solve more complex problems.
* Neural Network:
  + 3 layers: input, hidden, output
  + Deep neural network architecture (deep learning networks): several hidden layers with millions of artificial neurons linked together. Hae weights representing the connections between a node and others.
* Transformer model:
  + Type of neural network architecture that transforms or changes an input sequence into an output sequence. They do this by learning context and tracking relationships between sequence components.
  + Self-attention mechanism: for example, given a sentence, model looks at different parts of the sentence ALL AT ONCE and determine which parts are the most important
  + Input embeddings: the input sequence is encoded into mathematical vectors. Vectors carry semantic and syntax information. Attributes are learned during the training process.
* Tokens embedding are formed parallel
  + Positional encoding: tokens embedding doesn’t carry their sequence. A set of functions generate unique positional signal that is added to the embedding of each token.
  + Transformer block: multiple transformer blocks stacked togehter
    - Multi-head self-attention mechanism: weigh the importance of different tokens within the sentence, enables the grouping of relevant tokens for context.
    - Position-wide feed-forward neural network: has additional components that help the transformer model train and function more efficiently
  + Linear and softmax blocks
  + This new model uses multi-dimensional vectors (word embeddings) to represent words, which overcome the limitation of recognizing similar word meanings.
* MoE:
  + Top k-gating network: choose *k* experts in the model with weights to process the input.
* Recurrent neural network:
  + RNNs process data sequences one element at a time in cyclic iterations. The process starts with the input layer receiving the first element of the sequence. The information is then passed to a hidden layer, which processes the input and passes the output to the next time step. This output, combined with the next element of the sequence, is fed back into the hidden layer. This cycle repeats for each element in the sequence, with the RNN maintaining a hidden state vector that gets updated at each time step. This process effectively enables the RNN to remember information from past inputs.
* Convolutional neural network:
  + CNNs are designed for grid-like data, such as images, where spatial hierarchies and locality are key. They use convolutional layers to apply filters across an input, capturing local patterns through these filtered views. For example, in image processing, initial layers might detect edges or textures, and deeper layers recognize more complex structures like shapes or objects.
  + Early ML used numerical tables to represent words.
* LLM:
  + Based on transformer model.
  + Use word embeddings => understand the context and phrases with similar meanings, relationships
  + The size of the model is generally determined by an empirical relationship between the model size, the number of parameters, and the size of the training data.
  + Training is performed using a large corpus of high-quality data. During training, the model iteratively adjusts parameter values until the model correctly predicts the next token from an the previous squence of input tokens. It does this through self-learning techniques which teach the model to adjust parameters to maximize the likelihood of the next tokens in the training examples.
* GEMINI:
  + “This includes making Gemini 1.5 more efficient to train and serve, with a new [Mixture-of-Experts](https://arxiv.org/abs/1701.06538) (MoE) architecture.”
  + “Depending on the type of input given, MoE models learn to selectively activate only the most relevant expert pathways in its neural network. This specialization massively enhances the model’s efficiency. Google has been an early adopter and pioneer of the MoE technique for deep learning through research such as [~~Sparsely-Gated Mo~~E](https://arxiv.org/abs/1701.06538), [GShard-Transformer](https://arxiv.org/abs/2006.16668), [Switch-Transformer,](https://arxiv.org/abs/2101.03961) [M4](https://blog.research.google/2019/10/exploring-massively-multilingual.html) and more.”
* Done Readings:
  + <https://aws.amazon.com/what-is/neural-network/>
  + <https://aws.amazon.com/what-is/transformers-in-artificial-intelligence/>

[SENTIMENT ANALYSIS](https://www.altexsoft.com/blog/sentiment-analysis-types-tools-and-use-cases/)

**Benchmarking NLP Systems**

📌 Purpose: Compare WSD performance against human-labeled data in benchmark datasets (e.g., SemEval, OntoNotes, Senseval).

📌 Insight: Helps determine if an NLP model achieves human-level accuracy in WSD.

Example

If a model correctly predicts 85% of word senses in a benchmark dataset but humans achieve 95%, then there's room for improvement.

🔹 Use Case: Train AI models to reach human-level disambiguation accuracy.

* The fifth step to analyze sentence structure is to use software tools that can help you to automate or simplify some of the tasks involved in sentence analysis. Software tools are applications or programs that can perform various functions, such as parsing, tagging, annotating, generating, translating, or checking sentences. Some examples of software tools for sentence analysis are AntConc, TreeForm, Praat, ELAN, Sketch Engine, and Grammarly.
* Dependency parsing (Standford paper)
* Grammatical analysis using spaCy:
  + *import spacy*
  + *# Load the English language model*
  + *nlp = spacy.load("en\_core\_web\_sm")*
  + *text = "The quick brown rabbit jumps over the lazy frogs."*
  + *doc = nlp(text)*
  + *# Tokenization and POS tagging*
  + *print("Tokens and POS tags:")*
  + *for token in doc:*
  + *print(token.text, token.pos\_)*
  + *# Dependency parsing*
  + *print("\nDependency parsing:")*
  + *for token in doc:*
  + *print(token.text, "-->", token.dep\_, "-->", token.head.text)*
* <https://www.youtube.com/watch?v=zd4vajdUoV0> – Python
* <https://www.youtube.com/watch?v=DkY_RZrOoqY&t=3s> – Extracting Info from Text
* <https://www.youtube.com/watch?v=bqHJ6Glae1w> - spaCy

**Identifying Dataset Bias**

📌 Purpose: Check if a benchmark dataset contains skewed word sense distributions.

📌 Insight: If a dataset over-represents one meaning of a word, AI models trained on it may learn biased or incorrect sense distributions.

Example

A benchmark dataset might have "bank" referring to finance 90% of the time but rarely as a riverbank.

AI trained on this dataset will incorrectly favor the financial meaning, even when "bank" is in a nature-related context.

🔹 Use Case: Detect imbalances and improve dataset quality for training AI.

**Enhancing Word Embeddings & Pretraining**

📌 **Purpose**: Detect if **word embeddings (e.g., Word2Vec, FastText, BERT embeddings)** capture multiple word senses.  
📌 **Insight**: Some embeddings mix **multiple meanings** into a single vector, reducing accuracy.

**Example**

* If word2vec("bank") gives **one vector** for both meanings, it might cause errors in NLP tasks.
* Benchmark WSD data can help **train sense-specific embeddings** (e.g., using **Sense2Vec** or **contextual BERT embeddings**).

🔹 **Use Case**: Improve embeddings for **disambiguation-aware NLP models**.