School of CIT –
Social Computing
Research Group

Technical
University
of Munich

Advanced Natural Language Processing
CIT4230002

Prof. Dr. Georg Groh
Edoardo Mosca, M.Sc.

Lecture 1
Large Language Models

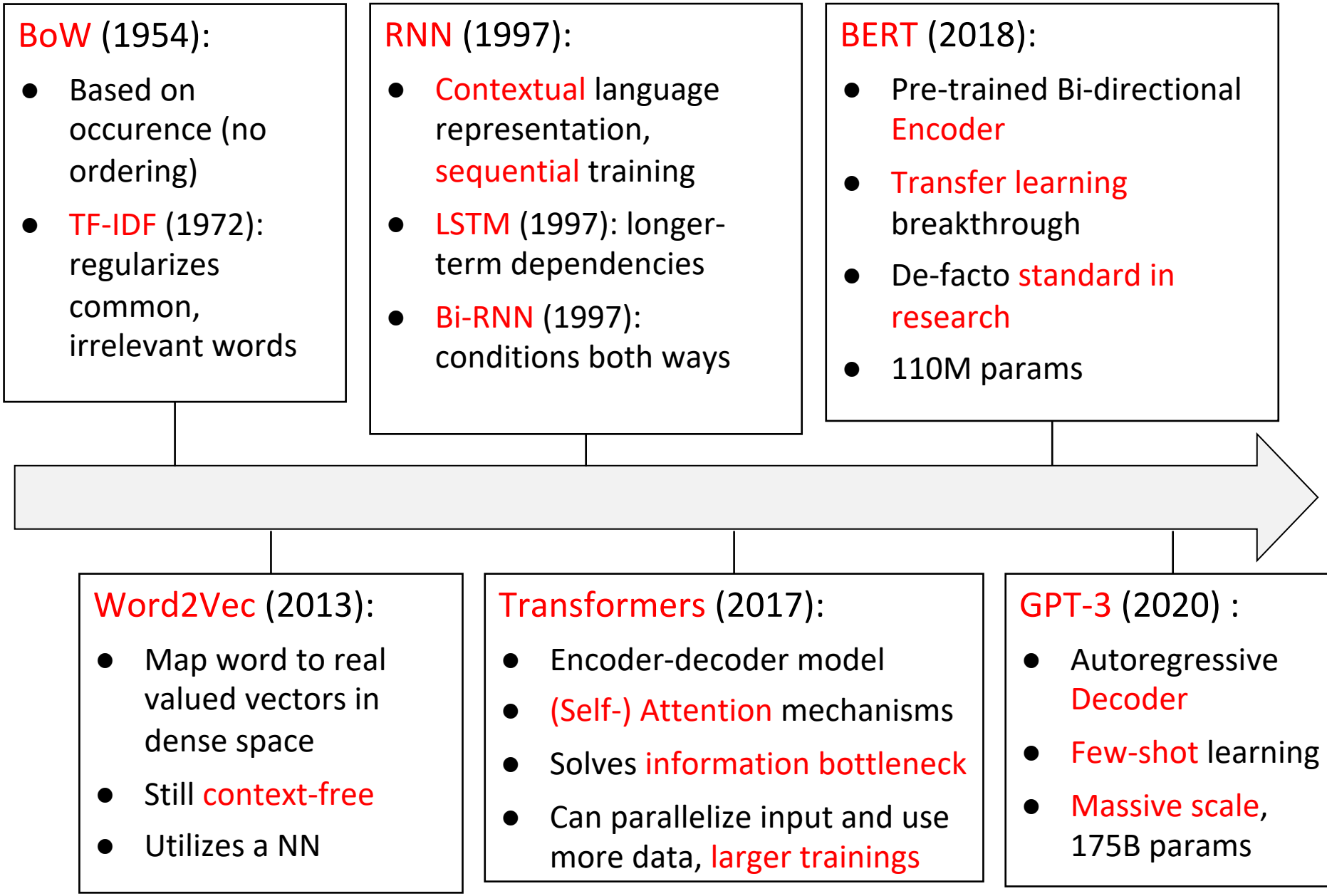
Language Models

- Language Modeling is the task of predicting what word comes next

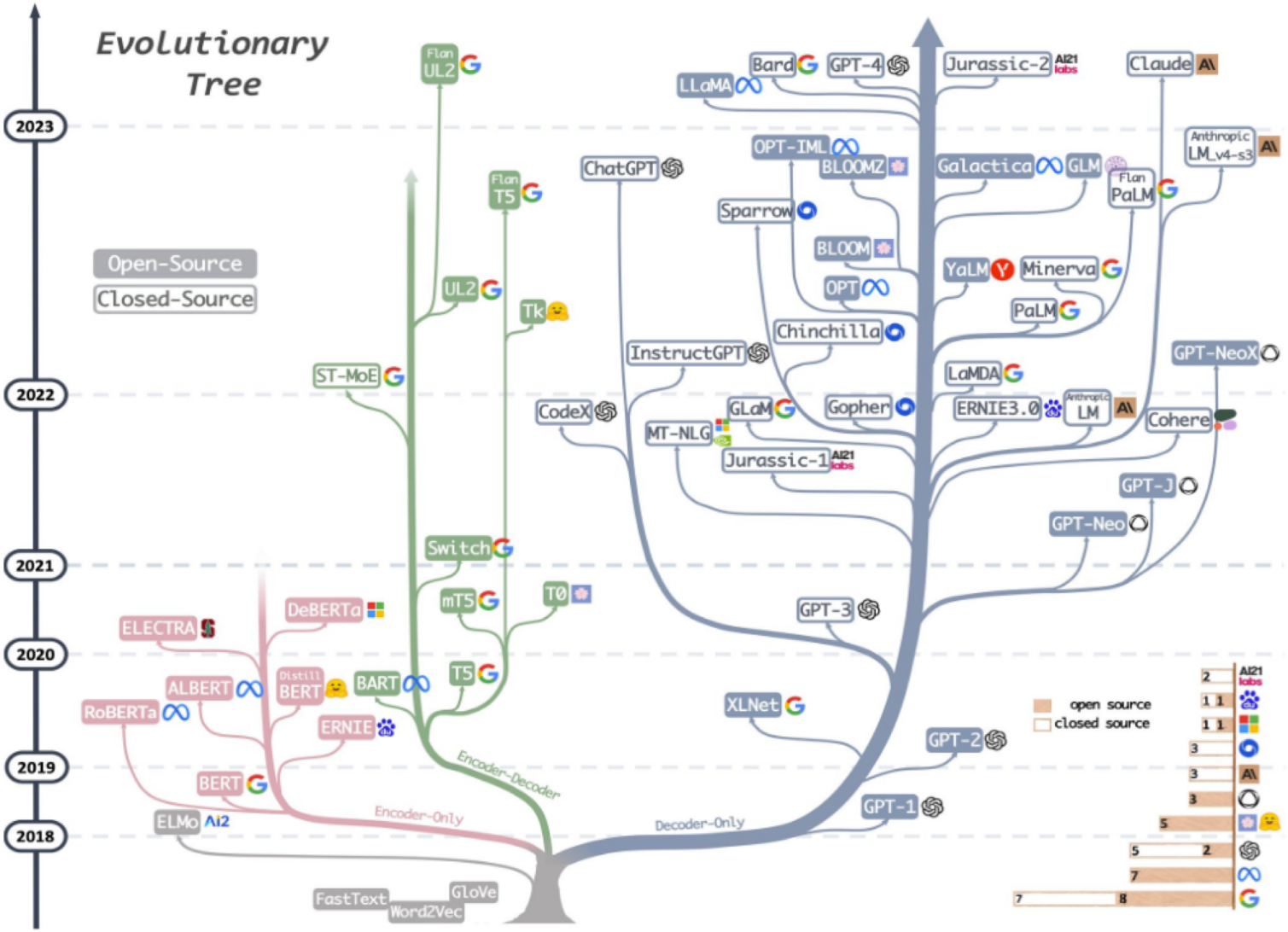
$$P(\boldsymbol{x}^{(t+1)} | \boldsymbol{x}^{(t)}, \ldots, \boldsymbol{x}^{(1)})$$

- Language models are a great indicator of overall progress in the field.
- If you improve at language modeling, you improve at (almost) everything else.

Language Models | Timeline



Language Models | Evolution of Transformers



- Transformers families have evolved, growing in size and capabilities
- Today: Encoder- and Decoder-Only, Encoder-Decoder in the next block

Recap | Attention

Variants

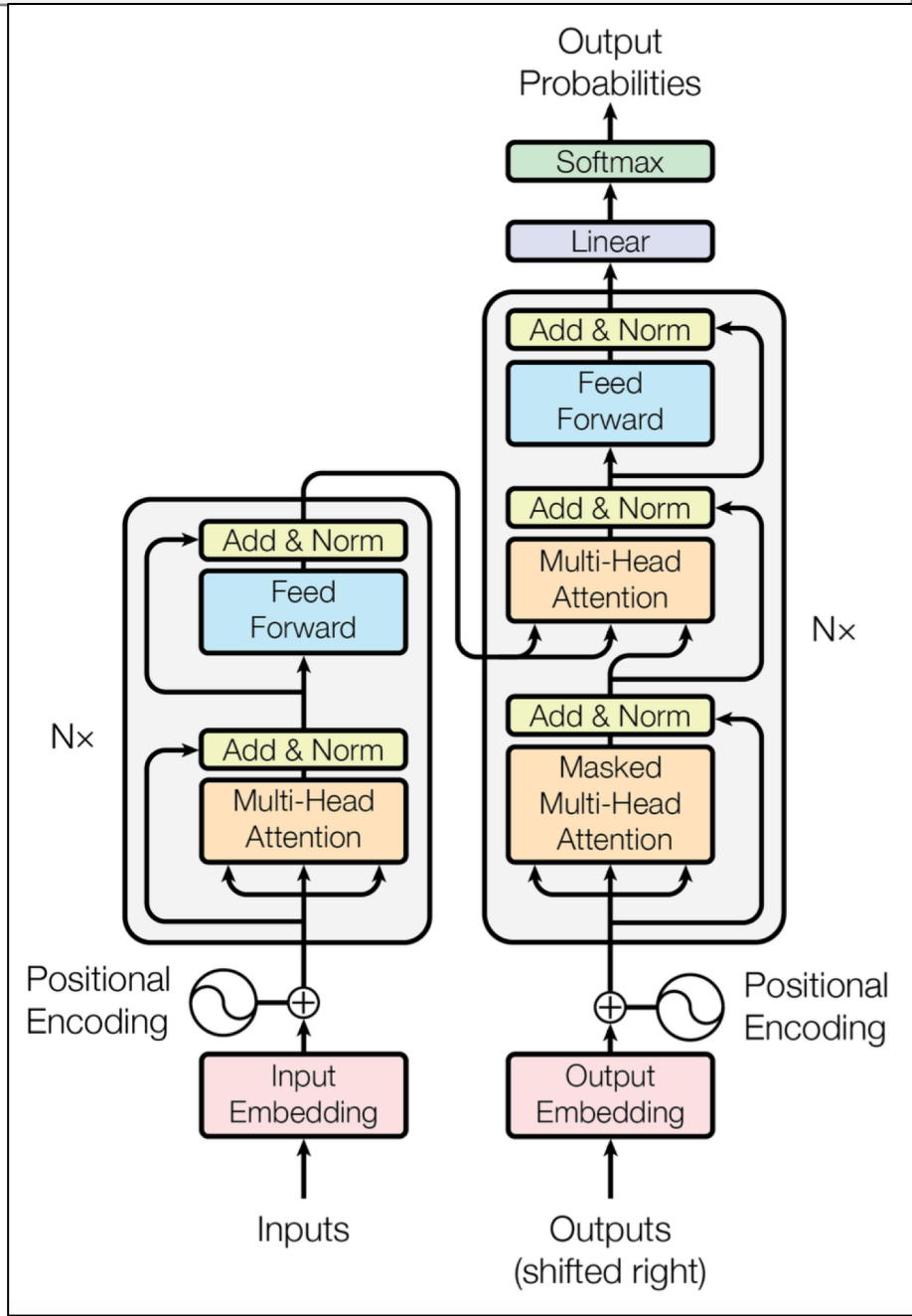
- keys: $\boldsymbol{k}_1, \boldsymbol{k}_2, \ldots, \boldsymbol{k}_N \in \mathbb{R}^{d1}$, values: $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_N \in \mathbb{R}^{d1}$ and query $\boldsymbol{q} \in \mathbb{R}^{d2}$
  - global vs local: are all states used or just a subset
- attention scores: $\boldsymbol{e}_i = \boldsymbol{q}^T \boldsymbol{k}_i \in \mathbb{R}^N$
  - choice of transformation: do you use a simple dot-product with query or use more complex function
- attention distribution: $\alpha_i = \text{softmax}(\boldsymbol{e}_i) \in \mathbb{R}^N$
  - scaling by dimension d_k useful for dot-products with high magnitudes
- attention output: $\boldsymbol{a}_t = \sum_{i=1}^{N} \alpha_i \boldsymbol{v}_i \in \mathbb{R}^{d1}$ (also often denoted as $\boldsymbol{c}_t$ (context vector))
  - soft vs hard: do you use the weighted average or just choose the state with the highest probability

Benefits

- alignment allows for interpretability
- capture different representation subspaces via multi-head attention
- self-attention: allows the inputs to interact with each other ("self")

Recap | Transformer

- Encoder-decoder architecture
- Input words embedded statically, positional encoding adds order
- Input sentence fed all at once. (Self-) multi-head-attention learns intra-input relationships.
- Decoder same as encoder. But self attention is masked for previous tokens.
- Encoder-decoder attention learns input-output alignment.
- Linear layer maps to whole vocabulary, then softmax.
- Generally, good performance on variety of tasks, however is costly to train



BERT

- "BERT": Bidirectional Encoder Representations from Transformers.
  - Basically, a transformer encoder
- LMs are normally uni-directional (autoregressive) => BERT conditions both ways thanks to masking
- Trained on BooksCorpus (800 million words) and English Wikipedia (2,500 million words)
- Pre-training tasks
  - Masked LM prediction: intra sentence level
  - next sentence prediction (NSP): inter sentence level
- BERT versions
  - BERT-Base: 12-blocks, 768-dim-hidden state vectors, 12 attention heads. (110M parameters)
  - BERT-Large: 24-blocks, 1024-dim-hidden state vectors, 16 attention heads. (340M parameters)
- BERT in-short: massive data + MLM + NSP* + transformer

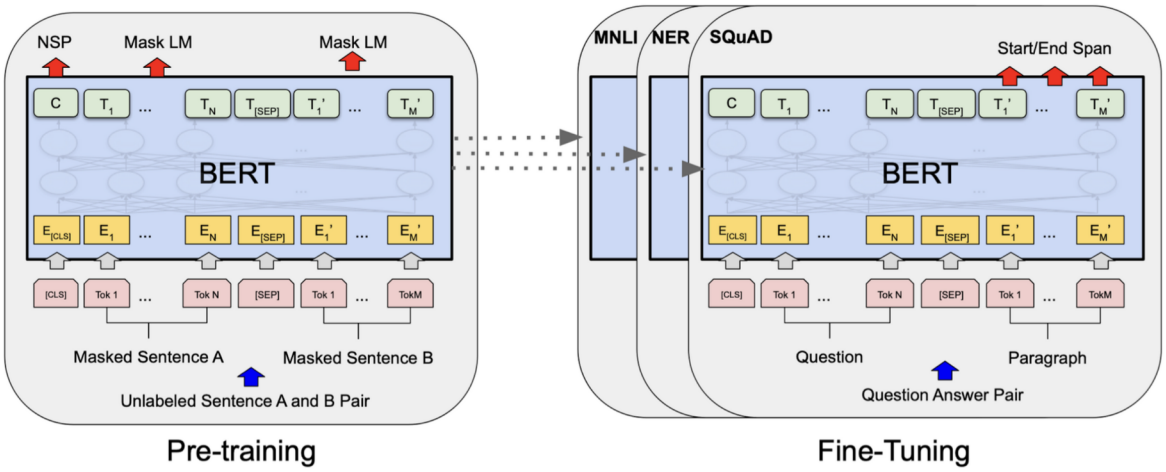
*some authors claim NSP is not too relevant for BERT

Encoder vs Decoder

BERT | Use Cases and Extensions

Use Cases

- "Pre-train once, finetune many times": much faster than full training!
- Language Modeling -> Sentiment Analysis, Question Answering, etc.



Extensions

- RoBERTa: mainly just train BERT on larger data batches and remove next sentence prediction.
- SentenceBERT: BERT + siamese architecture and triplet loss generates better document embeddings, great for similarity-based retrieval.
- DistilBERT: a smaller (40%), faster (60%) BERT. Same architecture distilled with a teacher-student setting. It retains 97% of the performance.

BERT | Use Cases and Extensions

Use Cases

-
-

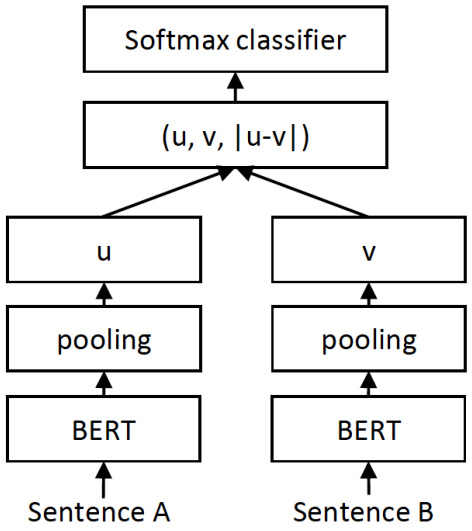


Figure 1: SBERT architecture with classification objective function, e.g., for fine-tuning on SNLI dataset. The two BERT networks have tied weights (siamese network structure).

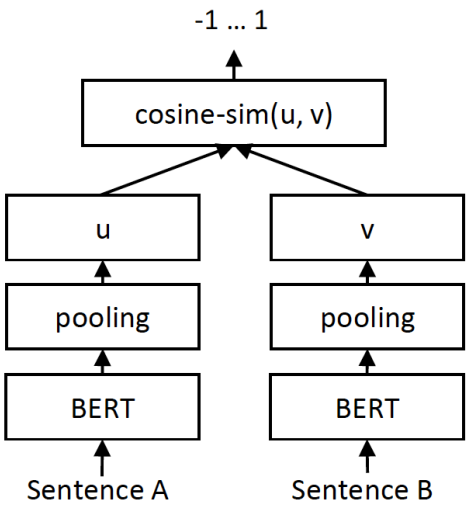


Figure 2: SBERT architecture at inference, for example, to compute similarity scores. This architecture is also used with the regression objective function.

[9]

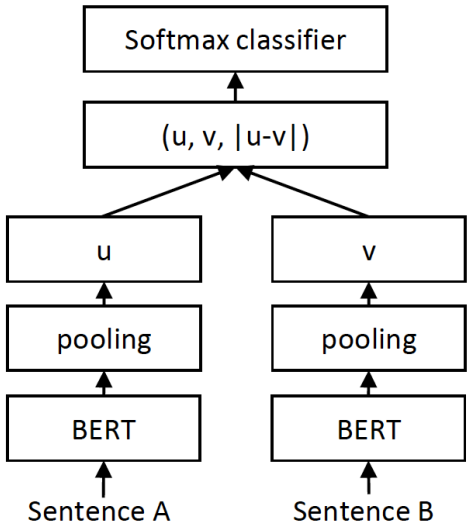
Ext

-
- SentenceBERT: BERT + siamese architecture and triplet loss generates better document embeddings, great for similarity-based retrieval.
- DistilBERT: a smaller (40%), faster (60%) BERT. Same architecture distilled with a teacher-student setting. It retains 97% of the performance.

BERT | Use Cases and Extensions

Use Cases

-
-



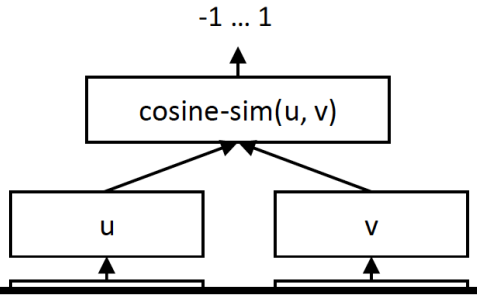


Figure 1: SBERT architecture with classificati
jective function, e.g., for fine-tuning on SNLI d
The two BERT networks have tied weights (si
network structure).

Triplet Objective Function. Given an anchor sentence a, a positive sentence p, and a negative sentence n, triplet loss tunes the network such that the distance between a and p is smaller than the distance between a and n. Mathematically, we minimize the following loss function:

$$max(||s_a - s_p|| - ||s_a - s_n|| + \epsilon, 0)$$

with s_x the sentence embedding for $a/n/p$, $||\cdot||$ a distance metric and margin ϵ. Margin ϵ ensures that s_p is at least ϵ closer to s_a than s_n. As metric we use Euclidean distance and we set $\epsilon = 1$ in our experiments. [9]

Ext

- sentence prediction.
- SentenceBERT: BERT + siamese ar
  better document embeddings, gr
- DistilBERT: a smaller (40%), faster
  distilled with a teacher-student se
  performance.

GPT

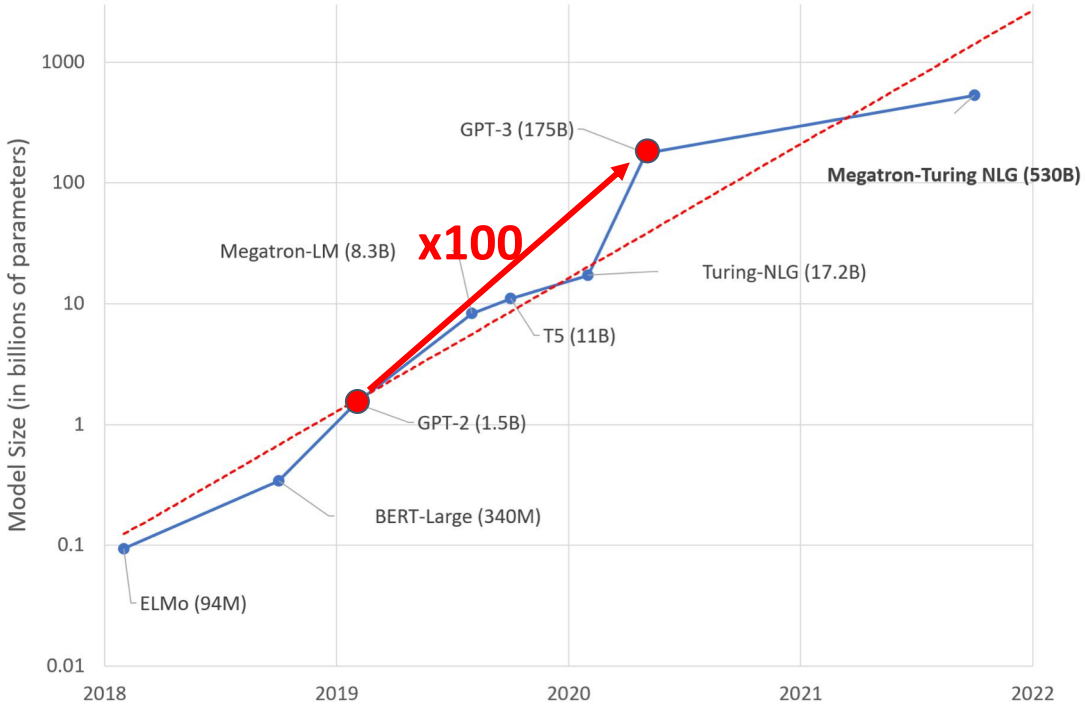
- GPT: Generative Pretrained Transformer.
  - Parallel line of research w.r.t. BERT (from OpenAI)
  - Transformer-based decoder (thus autoregressive), simply trained on language modeling.
  - Still requires a lot of training data to become good at a task.
  - More suited for text generation than BERT.

- GPT-2 is larger and introduces meta-learning
  - Similar architecture, no big changes
  - Provide task description together with input sample, this allows to perform a new task even if we did not train for it.
  - Didn’t work too well, but scaling architecture seems promising

GPT-3

- What if take GPT-2, make it substantially bigger, use more data and really leverage huge computational power.

- Result: Generative Pretrained Transformer 3 (GPT-3)



- Basically the same architecture as GPT-2, but X100 parameters

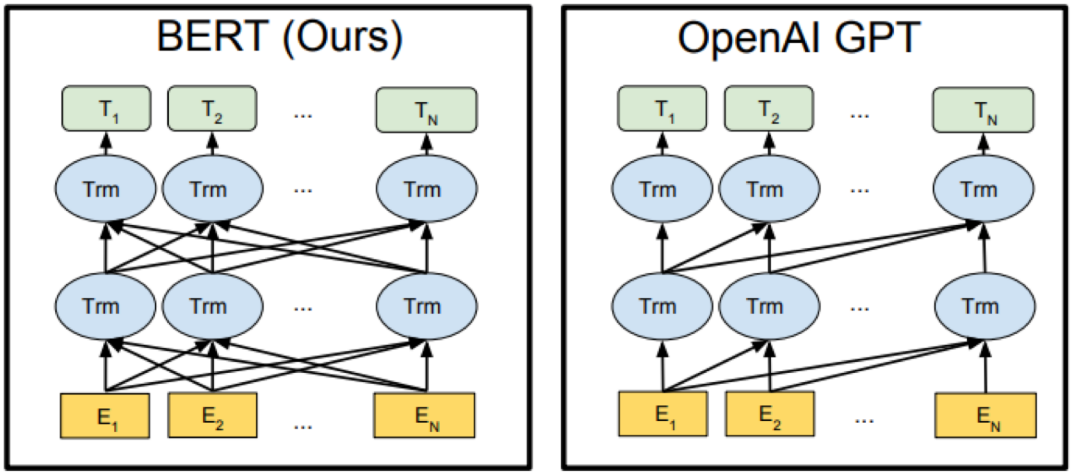
GPT-3 | Motivation

- The previous SOTA needs either substantial fine-tuning (BERT) or even additional task-specific architectures on top (ELMo).
- This implies the need for task-specific datasets and task-specific fine-tuning.
- So far small task-specific datasets lead to very narrow distributions
- Humans are few shot learners

- GPT-3 paper title: Language Models are Few-Shot Learners
- Lesson learned: scaling up language models drastically improves task-agnostic zero/few-shot performance
- No need to retrain the model's parameter. Prompting a few examples via text is enough for the model to "learn" a new task.
- What is prompting? What is few-shot? Next lecture!

GPT-3 | Architecture and Size

- Like GPT-2: autoregressive and simply trained on language modeling.



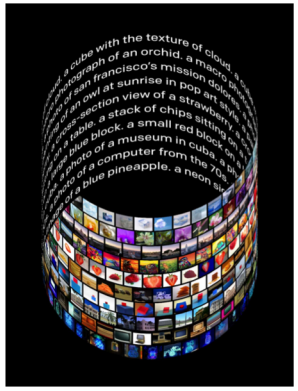
Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or "GPT-3"	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

Table 2.1: Sizes, architectures, and learning hyper-parameters (batch size in tokens and learning rate) of the models which we trained. All models were trained for a total of 300 billion tokens.

GPT-3 | Applications

- GitHub Copilot: can accurately generate (often working) code from a text prompt. Highly focused on web development and SQL.

- AI Dungeon: generates text-based adventure games.



- Dall-E (2): generates images and digital art based on text prompts. Results look much more realistic than previous methods.

GPT-3 | Limitations

NLP-specific Limitations

- Not good at text synthesis (starts repeating itself when generating long documents or passages)
- Easily hallucinates, i.e. it confidently produces false facts.
- Having difficulty within "common sense physics" and with understanding cause-effect relationships.

General Limitations

- Still exhibits undesirable behavior, including known racial, gender, and religious biases included in dataset
- Only trained on text, not knowledge-grounded
- It is LARGE (like, seriously!) and model weights are not available

Can bi-directionality (BERT) be better sometimes?

- Definitely not for text generation! But it could for sentence/token classification, filling gaps tasks, NER, QA, and relationship extraction.

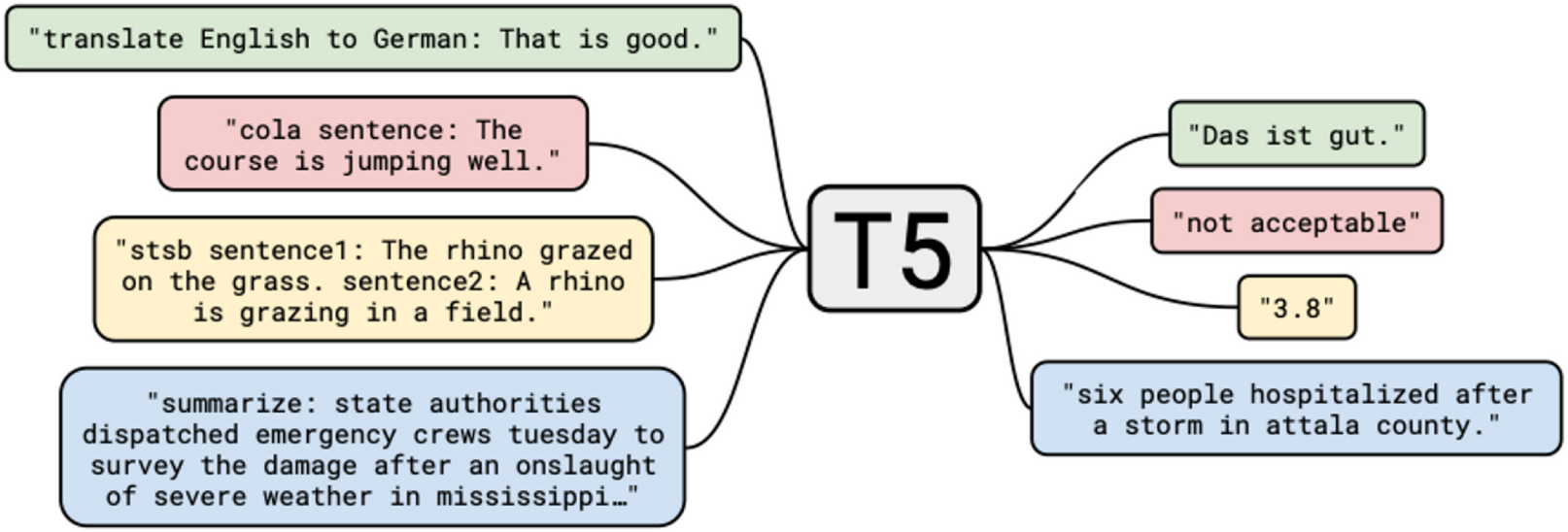
GPT-3| vs BERT Recap

	BERT	GPT-3
Size	340M parameters, trained on ~3.3 Billion tokens	175B parameters, trained on ~500 Billion tokens
Architecture	Bidirectional, made of transformer encoder blocks	Autoregressive, made of transformer decoder blocks
Training	Masked LM + next sentence prediction	Simple Language Modeling
Usage	Use as encoder + fine-tune extra layers on downstream task	Use as-is for any task with few-shot learning techniques

Evolution of LLMs

Following LLMs | T5 2019

- Text-to-Text-Transfer-Transformer (Encoder-Decoder)
  - The T5 paper explores which combinations work best for transfer learning (bidirectional vs autoregressive vs prefix-LM, masks length, masks frequency, training objectives etc.)
- Same model (slight modification of original Transformer Encoder+Decoder as in "Attention is All You Need" paper), loss function, and hyperparameters across multiple tasks
  - Task specific prefix which leads to soft cluster of weights

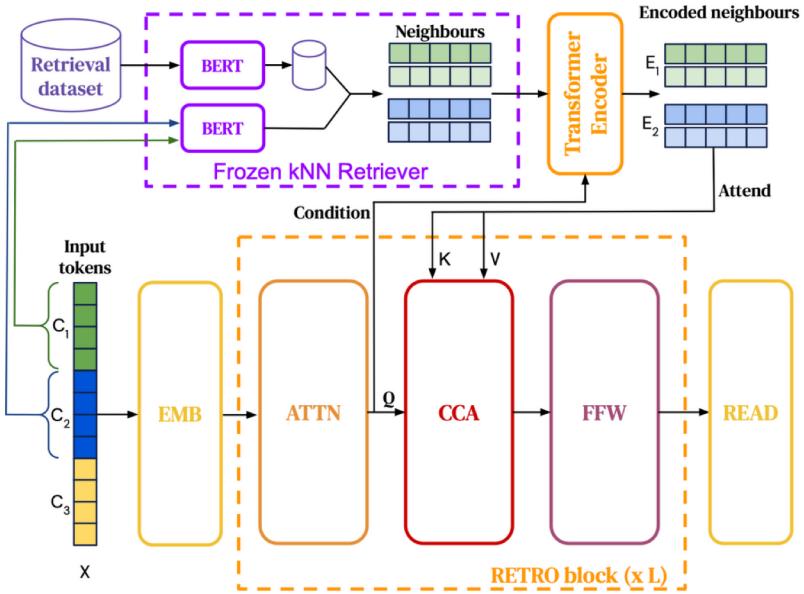


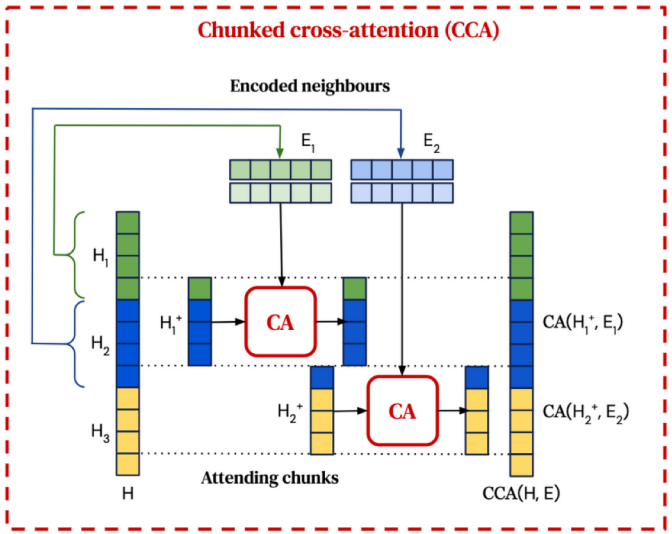
Following LLMs | RETRO 2022

- Retrieval-Enhanced TRansfOrmer
- Enc.-Dec.-Transf. Language model that predicts the next words by conditioning on document chunks retrieved from a large corpus.
- Goal: needs fewer parameters (4% of GPT-3 model size) by leveraging training data also at inference time.
- Intuitively, it makes sense to split language information ("I need to study, thus I go to _") from factual information ("The football world cup in 2016 was won by _")
- Architecture: interleave RETRO blocks and traditional Transf. Blocks

$$\textsc{Retro}(H, E) \triangleq \textsc{Ffw}(\textsc{Cca}(\textsc{Attn}(H), E)), \quad \text{and} \quad \textsc{Lm}(H) \triangleq \textsc{Ffw}(\textsc{Attn}(H))$$

E: Retrieved Nighbours, H: output of prev. layer





[5]

Following LLMs | Mixtral 2023

- Open-source LLMs are also catching up
- #Parameters between 1B and 200B (popular choices 7B, 13B, 33B, 70B)
- Mixtral is a Sparse Mixture of Experts (SMoE). Second only to GPT-4.

[10]

- At every layer, for each token, a router chooses 2 of 8 models (~7B "experts") to process the input further.
- It would be a ~47B model, but is fast as a ~12B one.
- Outperforms 70B models while being more efficient.

Following LLMs | Mixtral 2023

- Open-so
- #Parame
- Mixtral is

MoE: see [11]

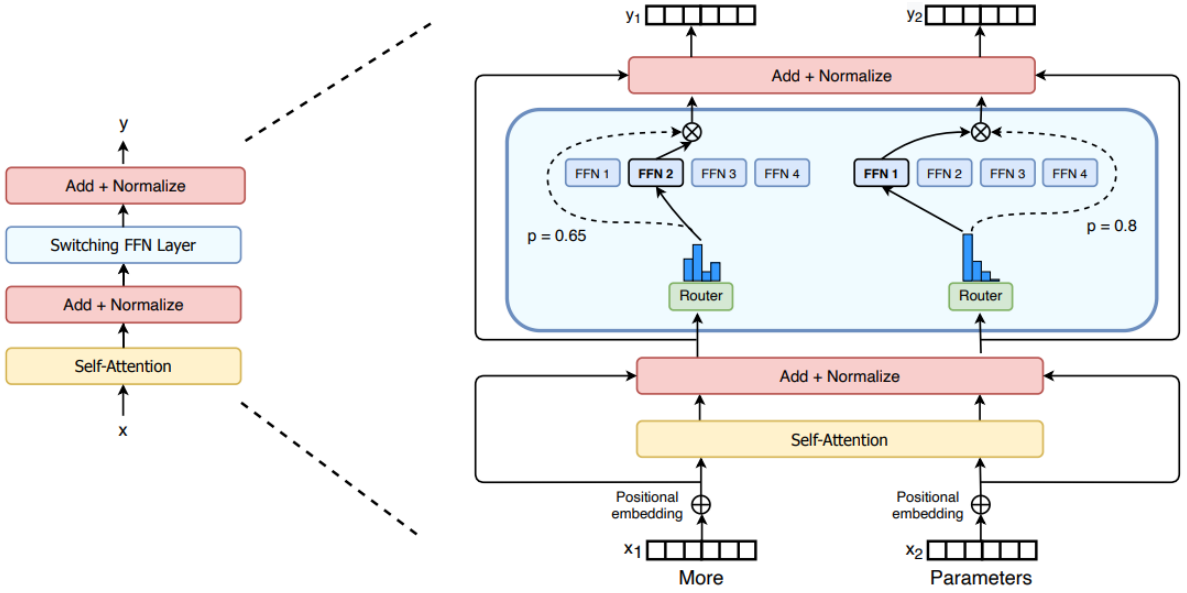
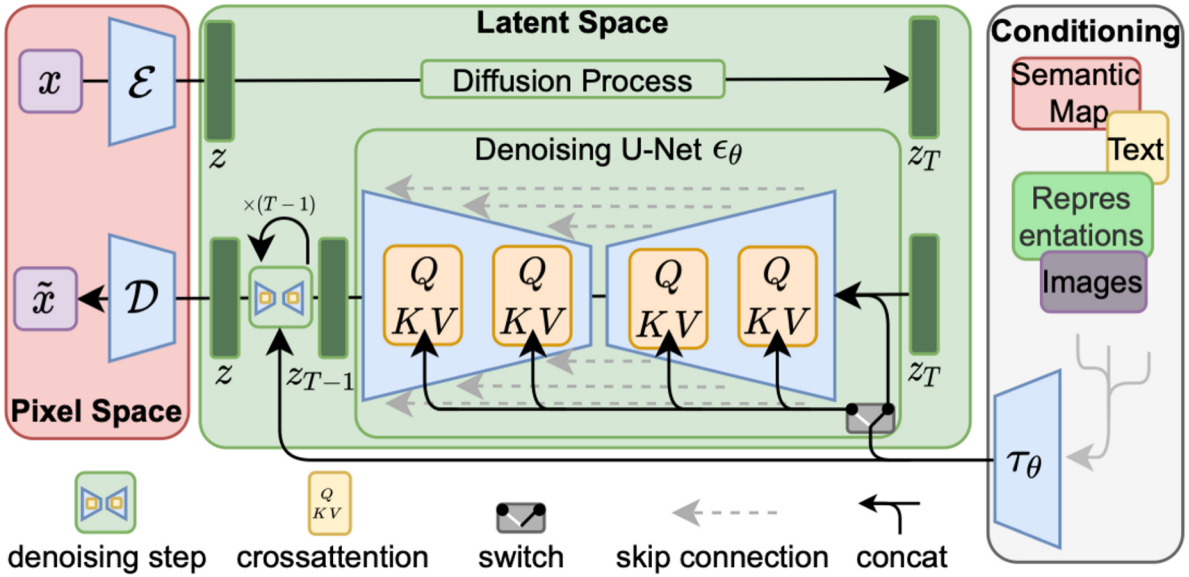


Figure 2: Illustration of a Switch Transformer encoder block. We replace the dense feed forward network (FFN) layer present in the Transformer with a sparse Switch FFN layer (light blue). The layer operates independently on the tokens in the sequence. We diagram two tokens (x_1 = "More" and x_2 = "Parameters" below) being routed (solid lines) across four FFN experts, where the router independently routes each token. The switch FFN layer returns the output of the selected FFN multiplied by the router gate value (dotted-line).

Following LLMs | Stable Diffusion 2022f

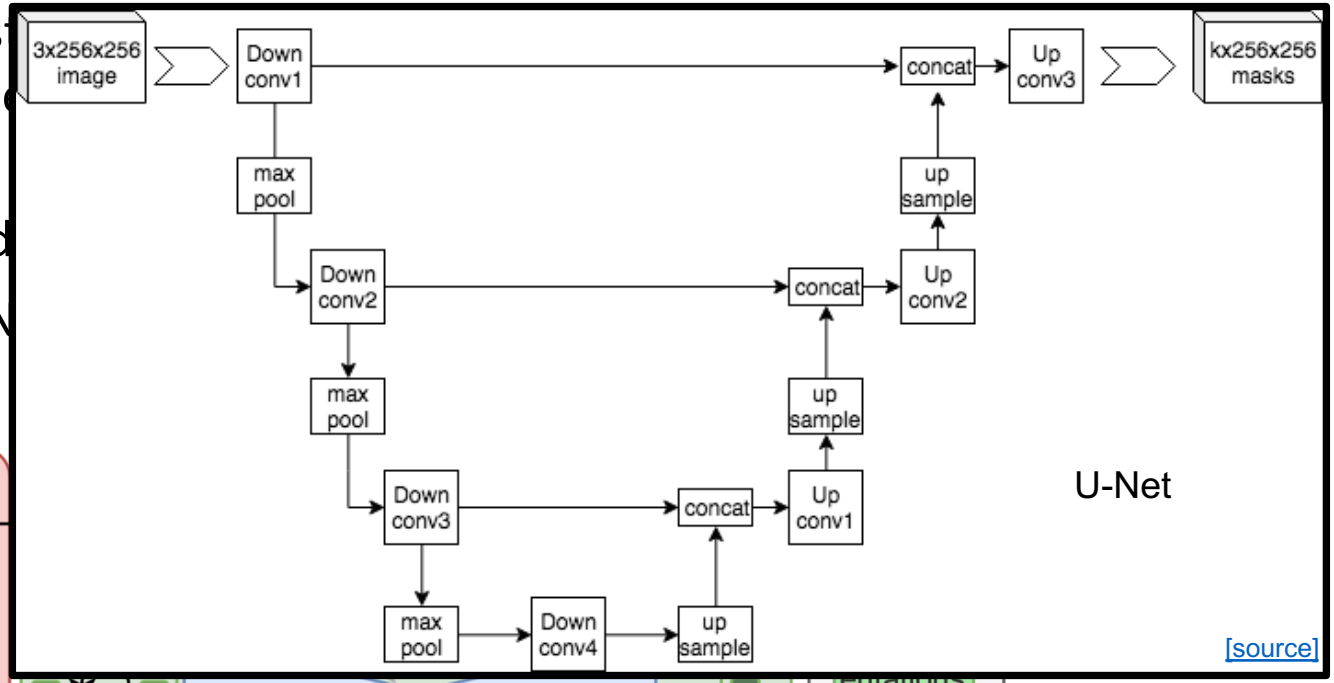
- Diffusion model: apply iteratively Gaussian noise on training images in the latent space. Then train neural nets (U-nets) to denoise and recover the previous state. Voilà, you have an image generator ←→ DALLE2.
- Now encode text with an LLM and use it to condition the denoising process.
- BERT as encoder in the paper, CLIP (GPT-3) in production
- Potential for NLP besides text-to-image? Yes!

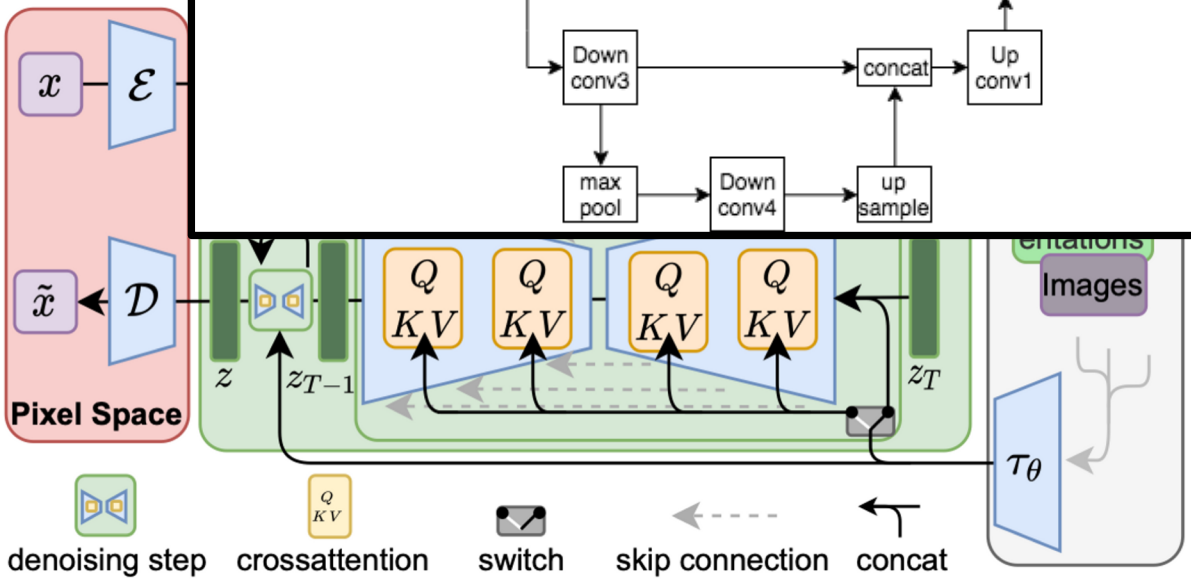


[6]

Following LLMs | Stable Diffusion 2022f

- Diffusion model: apply iteratively Gaussian noise on training images in the latent space. Then train neural nets (U-nets) to denoise and recover the previous st
- Now encode t
process.
- BERT as encod
- Potential for N





[6]

Following LLMs | Sta

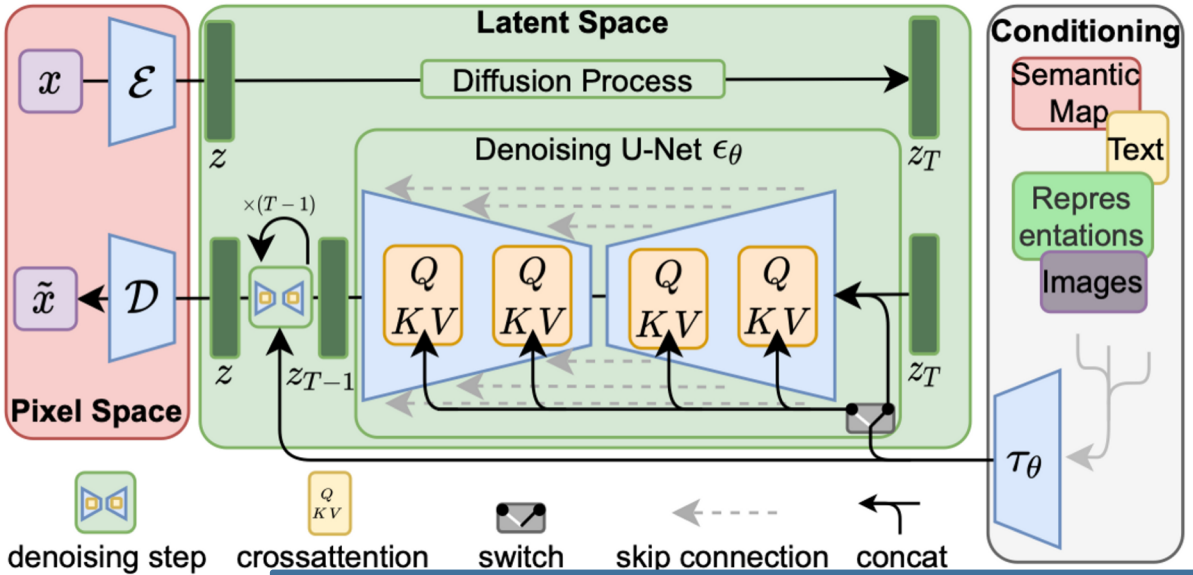
- Diffusion model: app
  the latent space. The
  the previous state. V
- Now encode text wit
  process.
- BERT as encoder in th
- Potential for NLP bes

To pre-process y from various modalities (such as language prompts) we introduce a domain specific encoder τ_θ that projects y to an intermediate representation $\tau_\theta(y) \in \mathbb{R}^{M \times d_\tau}$, which is then mapped to the intermediate layers of the UNet via a cross-attention layer implementing $\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right) \cdot V$, with

$$Q = W_Q^{(i)} \cdot \varphi_i(z_t),\ K = W_K^{(i)} \cdot \tau_\theta(y),\ V = W_V^{(i)} \cdot \tau_\theta(y).$$

Here, $\varphi_i(z_t) \in \mathbb{R}^{N \times d_\epsilon^i}$ denotes a (flattened) intermediate representation of the UNet

[6]

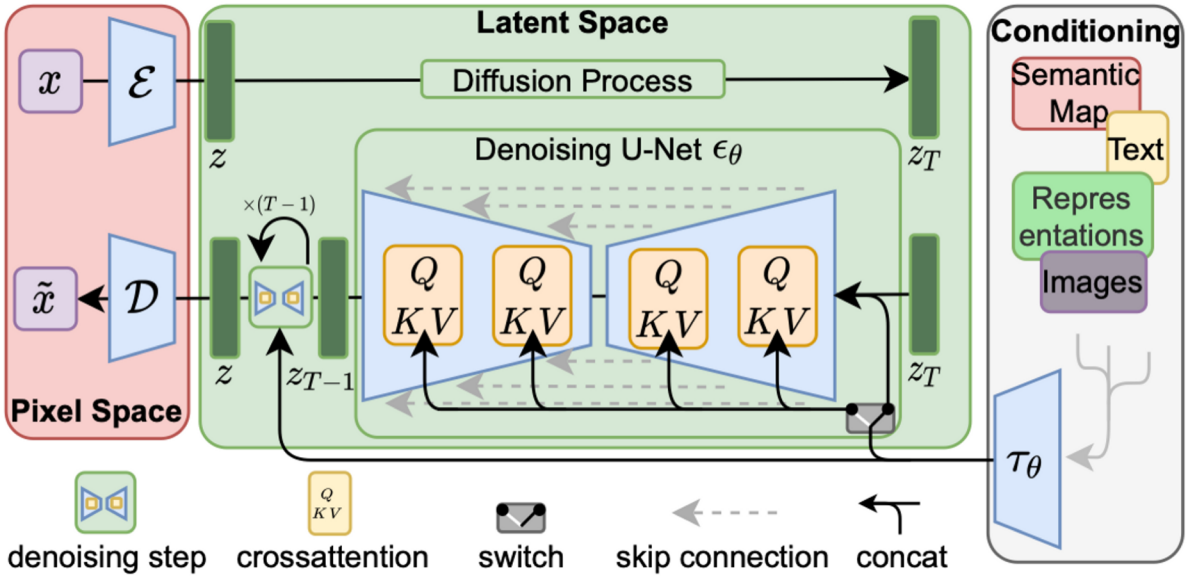


$$L_{LDM} := \mathbb{E}_{\mathcal{E}(x),y,\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(z_t, t, \tau_\theta(y))\|_2^2\right]$$

Following LLMs | Stable Diffusion 2022f

- Diffusion model: apply iteratively Gaussian noise on training images in the latent space. Then train neural nets (U-nets) to denoise and recover the previous state. Vo
- Now encode text with process.
- BERT as encoder in th
- Potential for NLP besides text-to-image? Yes!

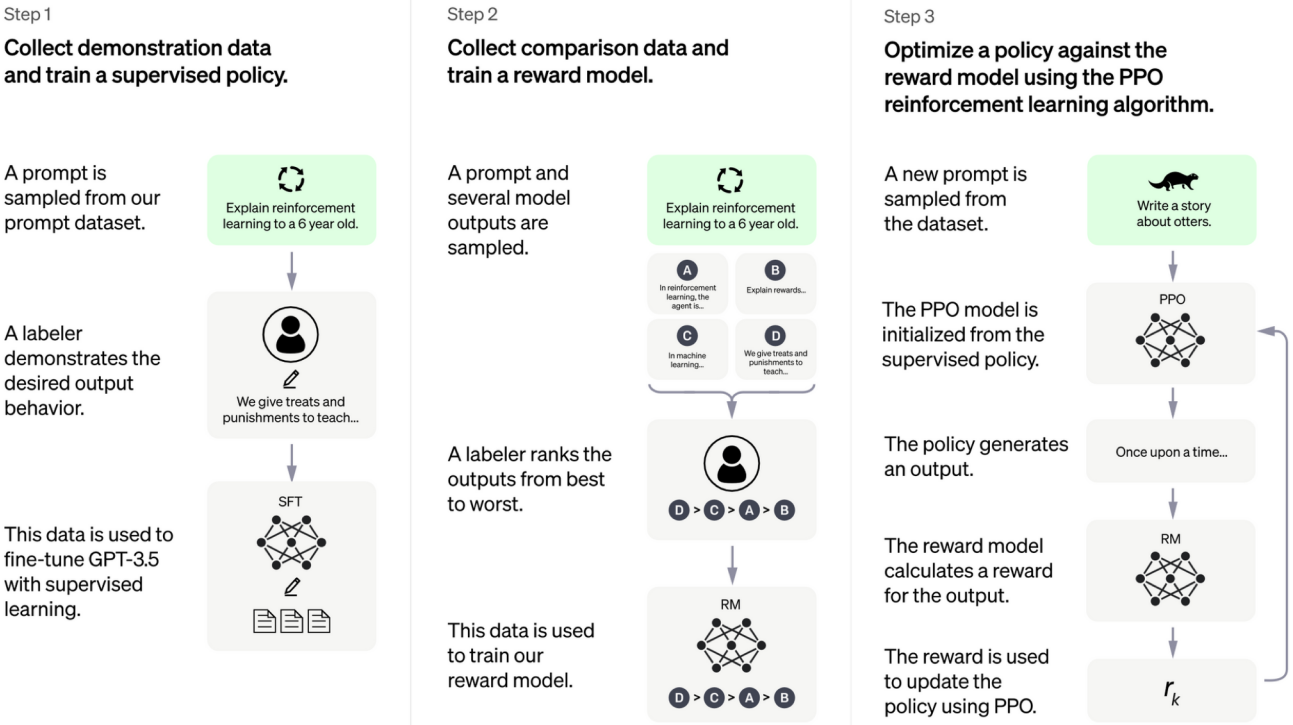
CLIP: trained on (image, caption) pairs: encode each into same dim. latent space,
contrastive objective: max cosine similarity btw related image / caption vector pairs, minimize btw. unrelated ones
→ model is able to tell in how far does the caption text relate to the image



[6]

Following LLMs | ChatGPT

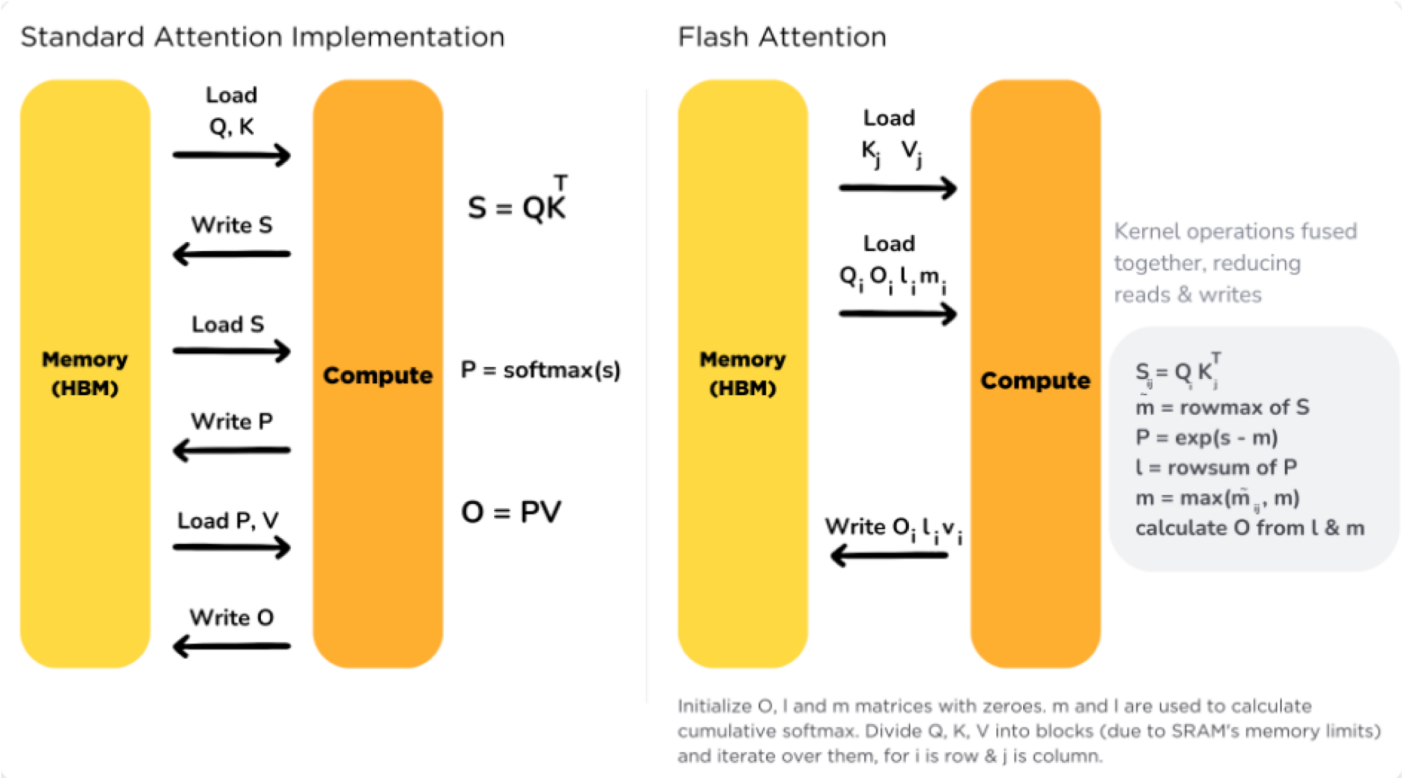
- ChatGPT: GPT-3.5 + supervised training on dialogue + reinforcement learning with human feedback (RLHF, next lecture!).



- ChatGPT probably doesn't remember previous prompts, its webapp does
- OpenAI acknowledged that ChatGPT can produce false facts
- Jailbreaks: despite filters, can instruct about how to perform crimes

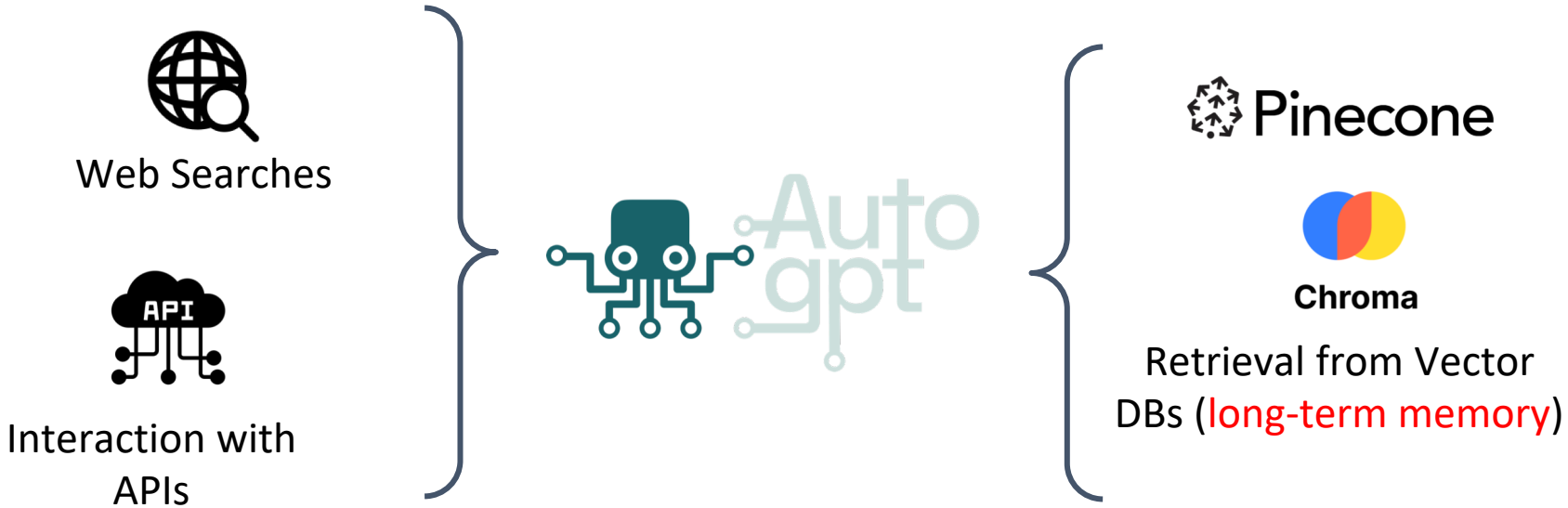
Following LLMs | Flash Attention

- Self-attention is the bottleneck (quadratic time) for scaling transformers architectures. It moves information back and forth from HBM to SRAM while performing one operation at the time.
- Flash Attention loads keys, queries, and values all at once, fusing the operations of of attention, and writes them back.
- Flash attention is becoming the default for Transformers 🤗



Following LLMs | From Models To Agents

- LLM Agents (AutoGPT, AgentGPT, ChatGPT Plus) extend the capabilities of LLMs by enabling them to interact with the real world.



- These capabilities also enable LLMs to solve more complex tasks by (1) breaking them into smaller steps, (2) acquiring the necessary knowledge, (3) execute the steps.

Following LLMs | Notable Mentions

GPT-4 (OpenAI)

- Best foundation model to date, likely not for long, details unknown
- For a while only Claude (Anthropic) came close

BLOOM (BigScience)

- Open source: largest collaboration of AI researchers
- 46 natural languages and 13 programming languages

Galactica (Meta)

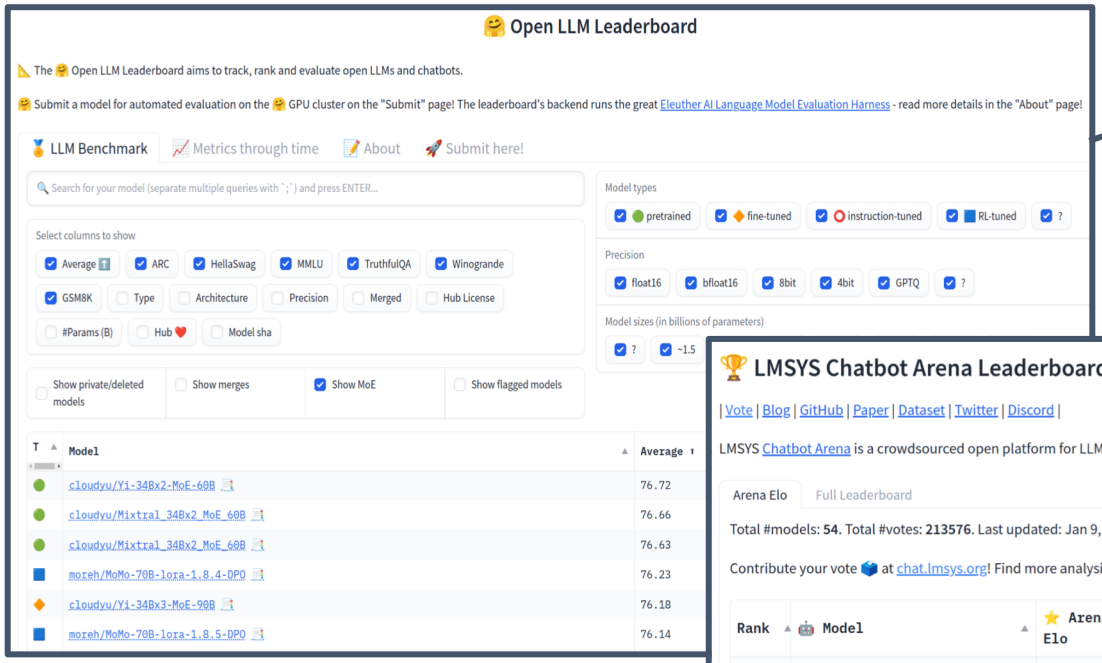
- Scientific LLM: train on massive corpus of science publications
- Demo was withdrawn two days after launch because off concerns regarding misleading authoritative publications

LLama (Meta)

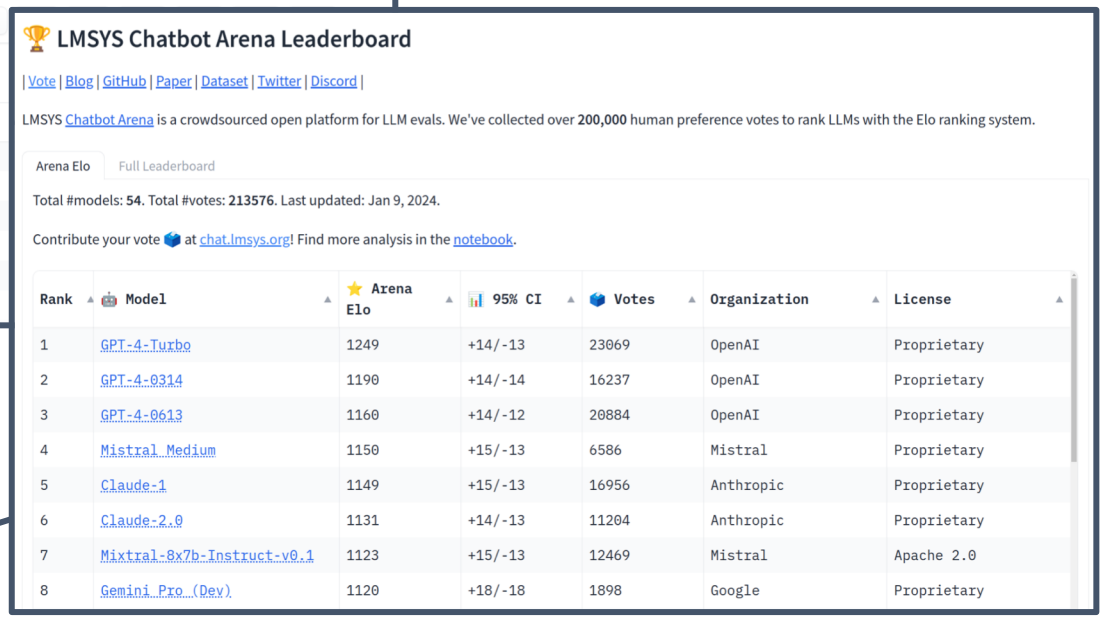
- Open-source standard, gave rise to a series of strong LLMs: Alpaca, Vicuna, Zephyr, Falcon, etc..

Following LLMs | Who's the Best?

- Of course, it depends what you are trying to do. But you can keep track of..



OpenLLM Leaderboard: customizable benchmark of open-source models on Hugging Face



Chatbot Arena: ELO ranking of chat models (both close and open-source)

Trends

- Attention is still the main workhorse and might stay for a while, but not as certain as last year (Attention-free e.g. Mamba?).
- LLMs stopped scaling in size and open-source options are catching up fast. Also higher data quality reduces needs for large architectures (TinyLLaMa, Phi-2).
- Going beyond the autoregressive way has lots of potential (stable diffusion).
- Architecture details seems to become more and more irrelevant. The important factors seem to be:
  - Broad architecture type (e.g. BERT vs GPT)
  - Training objective
  - Fine-tuning techniques
  - Data quality and quantity
- Multi-modality, multi-linguality, efficiency, and safety are now core targets of future LLMs.
- (Almost) everyone uses pre-trained LLMs instead of building and training from scratch.

Research Directions

- Prompting, auto-prompting
- Fine-tuning techniques (next lecture!)
- Post-hoc explainability (in a few blocks)
- Robustness and adversarial attacks/defenses
- Low-resource languages (next block)
- Retrieval Augmented Generation (RAG) and grounding (in a few blocks)
- Evaluation (next block)
- Multi-modality and text to Image/Video (in a few blocks)

References

[1] Vaswani et al. 2017. Transformer

[2] Devlin et al. 2018. BERT

[3] Brown et al. 2020. GPT-3

[4] Raffel et al. 2019. T5

[5] Borgeaud et al. 2021. Retro

[6] Rombach et al. 2021. Stable Diffusion (applied to text: [6a] paper1 , [6b] paper2

[7] Transformers Catalogue

[8] Kaplan et al. 2020. Scaling Laws

[9] Reimers, Gurevych: 2019 S-BERT

[10] Mixtral: at HuggingFace (URL 2024)

[11] Mixtral: Link to MoE Blog Post (URL 2024)

[12] Explain DALLE2 Blog post (URL 2024)

[13] FlashAttention at HuggingFace (URL2024)

[14] Link to Blog Post about ChatGPT RLHF (URL 2024)

Study Approach

Minimal

- work with the slides + skim references 5, 6, 6b, 8, 12, 14

Standard

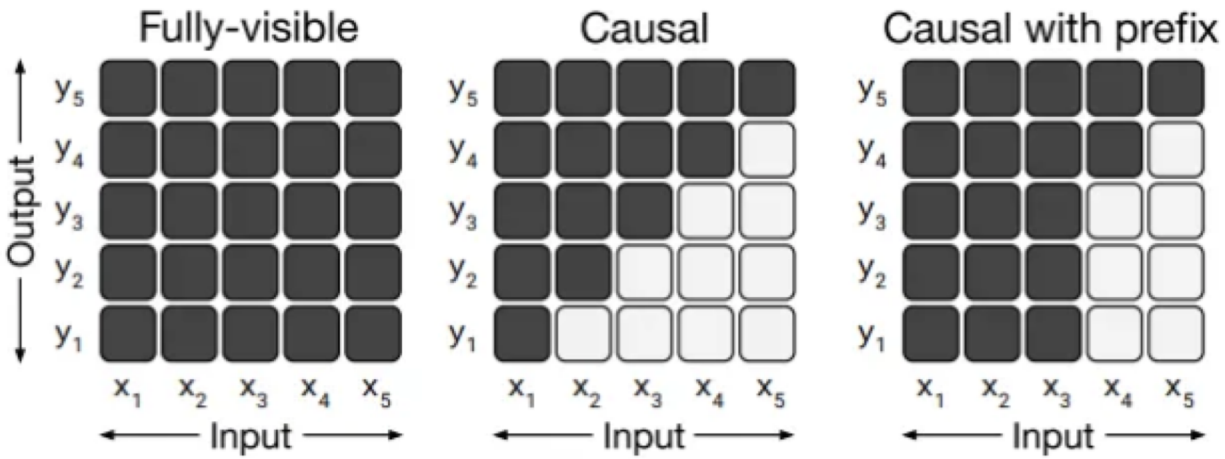
- minimal approach + skim rest of references + read 2 out of 5, 6, 6b, 8, 12, 14 in depth

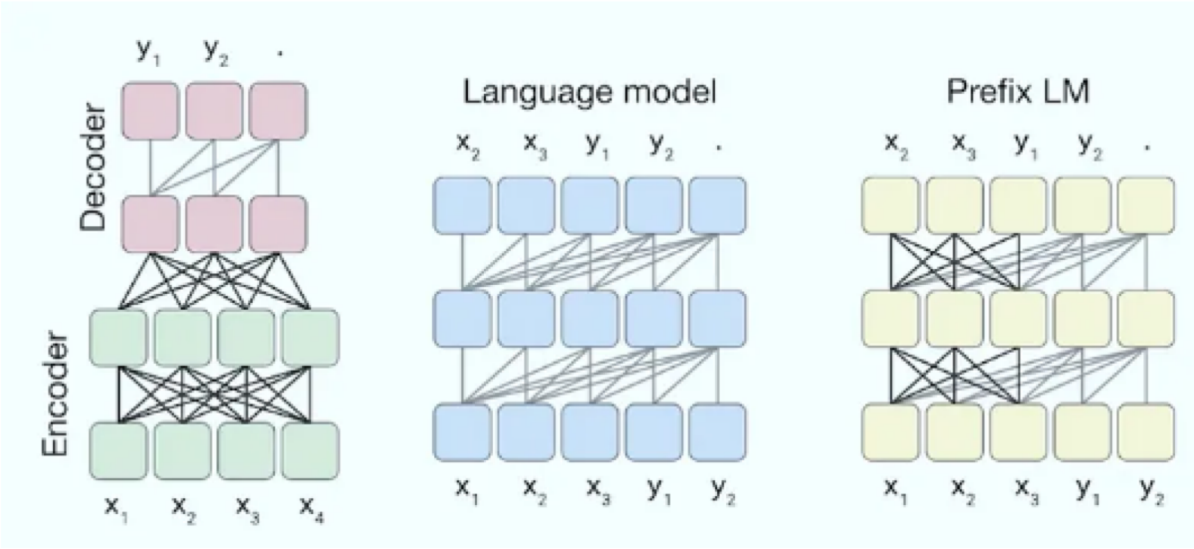
In-Depth

- standard approach + read rest of references in depth

See you next time!

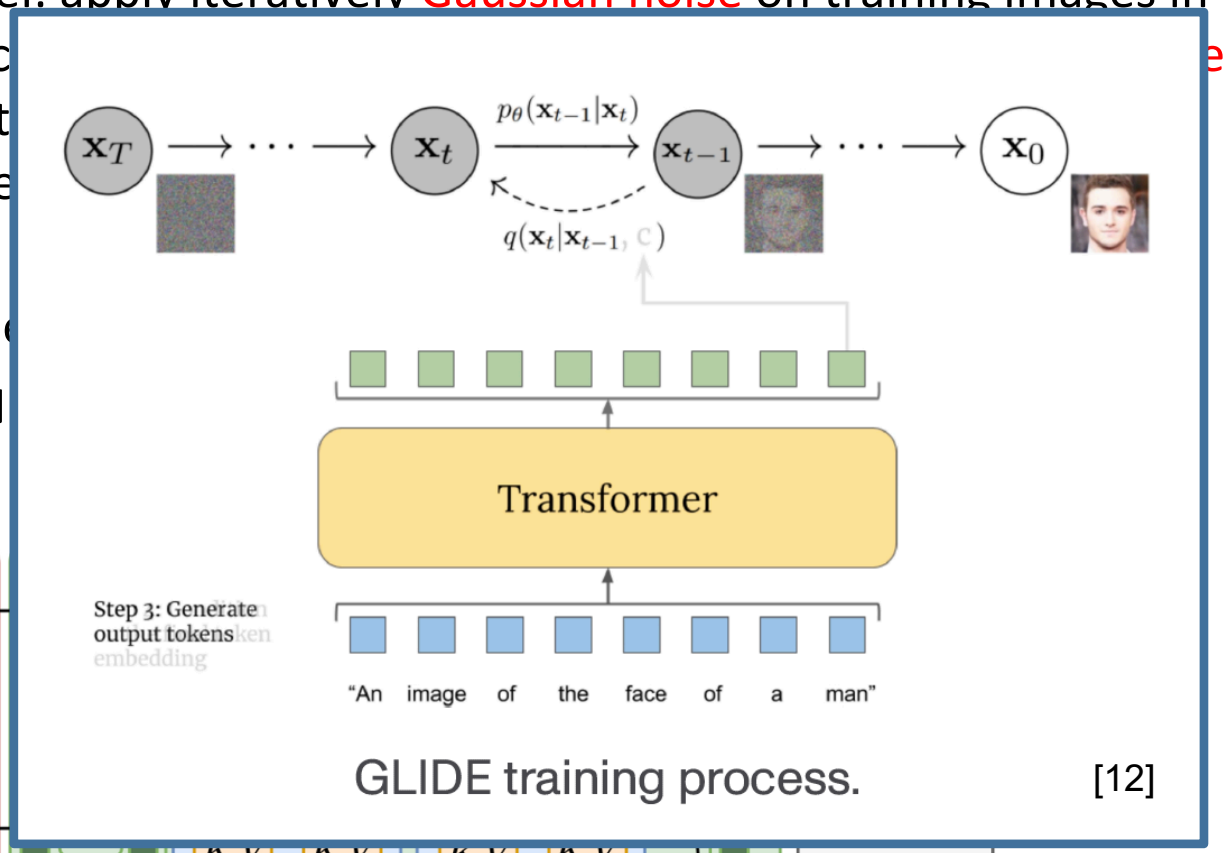
Backup | Attention Masking and Model Architectures



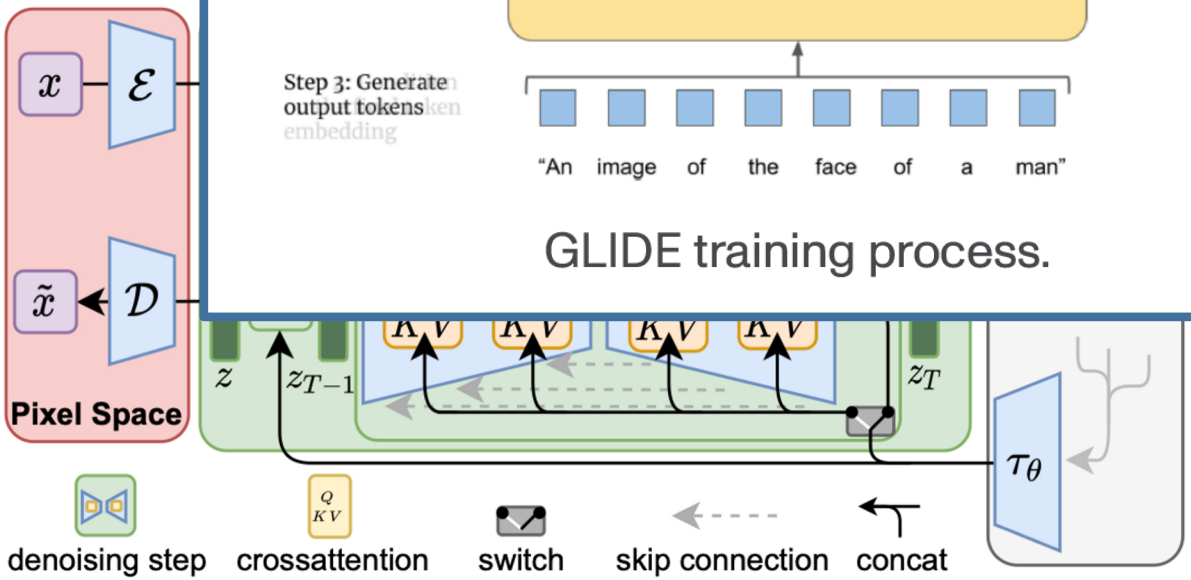


Following LLMs | Stable Diffusion 2022f

- Diffusion model: apply iteratively Gaussian noise on training images in the latent spac … er the previous st…
- Now encode te… process.
- BERT as encod…
- Potential for N…



GLIDE training process. [12]



[6]

Backup| Considerations about Scaling

- Refers to compute budget, not model size
  - However, more compute only useful on larger model
- Small models plateau early large model reaches lower loss
- Train until optimality, not convergence