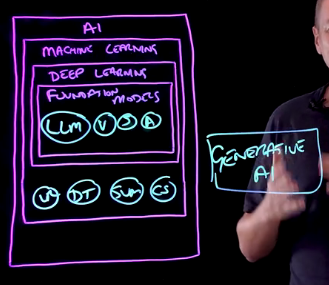
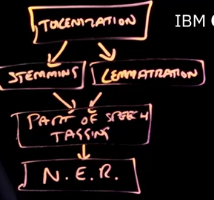
# Course 1 : Intro to AI

* Artificial Intelligence: Simulates human intelligence via algorithms/training
  + Week AI (narrow AI): specific domain (language translators, virtual assistants, AI powered web searches, recommendation engines etc.)
  + Strong AI (generalized AI): AI with diverse capabilities across unrelated tasks + can acquire new skills to face new challenges
  + Super AI (conscious AI): AI with human level consciousness, requiring self-awareness
* Learning
  + Supervised Learning
  + Unsupervised Learning
  + Reinforcement Learning
* Intelligence
  + Human
  + Artificial (machines to perform tasks that requires human intelligence – replaces human involvement)
  + Augmented (machines & humans working side by side – machines giving super abilities to humans / compliments humans abilities)
* Human’s vs machines
  + Humans: generalization, creativity, emotional intelligence
  + Machines: ingesting data, repetitive tasks, accurate
  + Augmented intelligence thrives at combining the strengths of both humans and machines
* Generative AI
  + Creates Content (text, images, video, music etc.) from scratch
  + Does not rely on predefined rules; instead uses deep learning techniques
  + Can use LLM (large language models) for processing & generating human like text
* The evolution from traditional AI to generative AI
  + Repository -> Analytics Platform -> Application [Predictive science]
  + Above + Feedback loop (to repository) [AI]
  + Data (not specific or internal to business) -> LLMs (general) -> Prompting & Tuning (specific) / finetuning -> Application layer + feedback loop (to prompting & Tuning layer)
* AGI Artificial General Intelligence
  + Turing test (chatting with something / someone on the other side and not being able to tell if that’s human or AI)
* Chatbots and smart assistants
* GEN AI Solutions
  + Single modality – text
  + Multiple modal solutions
  + OpenAI
    - GPT 3 – text only llm
  + Google
    - PaLM models – text only
    - Gemini models – multi model
  + Amazon
    - Titan model
  + Meta
    - Llama model
  + Anthropic
    - Claude models
  + Microsoft copilot
  + Dall-E , StyleGAN, SuperResolution, adobe Sensei
  + Voice & music generation (Murf, OpenAIs whisper)
  + Jukedeck, amper music, aiva
  + Video generation (google imagen video, openai sora)
  + IBM watsonx
* Use cases
  + Customer service
  + Voice assistance
  + Recommendations
  + Financial transactions
  + Cyber security
  + Transportation
  + Communication
  + Healthcare
  + Marketing & Sales
* Cognitive Computing
  + Cognitive: Intellectual activities: Thinking, Reasoning, Problem-solving
    - Observe phenomena and evidence
    - Interpret observations to generate a hypothesis
    - Evaluate which hypotheses are correct
    - Decide and act on best option
  + Cognitive Computing: Tech to mimic human cognitive processes
    - Makes machines active partners
    - Perception
      * Interprets the environment, by gathering data
    - Learning
      * ML algo’s
    - Reasoning
      * Analyze patterns and trends etc.
* Generative AI
  + Common types includes
    - VAEs Variational autoencoders (VAEs)
      * Encoder
      * Latent space representation
      * Decoder
    - GANs Generative adversarial networks
      * Generators
      * Discriminators
    - Autoregressive models
      * Creates data sequentially
    - Transformers
      * NLP tasks
      * Encoders / decoders layers
      * E.g GPT generative pre-trained transformers
    - Diffusion models
  + Unimodal vs multimodel models
* LLMs Large Language Models
  + Foundation models (generalized models i.e., capable of handling multiple tasks) – unsupervised training
    - Instead of training multiple models for a bunch of tasks, take a foundation model, trained on a very large dataset, and finetune it for your tasks
    - Disadvantages: Compute + Trust
  + Generative AI
  + Generative AI -> Foundation Models -> LLMs
  + Foundation Models (Unsupervised models) + labeled data (tuning)
  + Prompting -> using a foundation model for your tasks
* Putting things in place



* NLP
  + NLU (natural lang understanding): unstructured text to structured representation
  + NLG (natural lang generation): structured to unstructured text generation
  + Bag of NLP tools
    - Input: unstructured text
    - Tokenization: take a string – break into pieces
      * Stemming (normalizing to get to root word)
      * Lemmatization (arriving at root word by using dictionary like def)
      * Parts of speech tagging
      * Named entity recognition (identifying associations)



* AI Agents
  + Engage with surroundings + collect data (perception using sensors) + execute tasks independently (process data with algos) + decision making (based on knowledge base and logic) + meet goals set by human (drive car) + learning (improve machine learning via feedback loop on past experiences)
  + Social ability + autonomy + reactiveness + proactiveness
* Focus on designing systems not on designing models
  + Systems are modular
  + Easier to adopt
  + Compound AI systems
    - Uses models plus databases and etc., to create a system to solve problems
    - E.g., RAGs
  + Path to answer a query i.e., where to go to get answer to a query is basically Control logic
    - Programmatic control logic (defined by human e.g, via db) – Think Fast – less autonomous
    - LLM (putting model in charge) – tasked with breaking complex big problems into smaller easier ones, then to tackle them – Think Slow – Agentic Approach – more autonomous
      * Ability to reason
      * Ability to act (via tools e.g., search, calculator, custom code, another LLM etc. )
      * Ability to access memory
      * ReAct agent (reason – act)
* Robots
  + Sensors
  + Actuators (motors, hydrollic etc.)
  + Controllers (runs software that controls robot)
  + Cobots (robots that works together with humans)
  + RPA (robotic process automation)
    - Helps create, control virtual robots
* Hugging face – kinda like github for opensource foundation models
* Democratization of AI
* RAG (Retrieval Augmented Generation)
  + Retrieval & generation based models
  + Effective for contextually relevant content generation
  + Query (Prompt)
  + Retrieves relevant docs/pieces from predefined database (retriever models like BM25 or dense retrievers based on neural networks)
  + These docs are then used to augment the input to generative model; thus providing more context
  + Generative model finally gives response based on input query and retrieved docs.
  + Generative models (gpt-3 or gpt-4 or BERT) are more likely to produce incorrect responses – Hallucination
    - They also have a knowledge cutoff date i.e., point of their training completion
    - Also have limited context window; which makes them struggle with tasks requiring long-term context
  + RAG on google cloud
    - Vertex AI: build & deploy ML models
    - BigQuery: backend for retrieval component of RAG
* Using AWS
  + Data storage & Preparation (S3, Glue, Redshift)
  + Model Development (Amazon Sagemaker, AWS deep learning AMIs, AWS lambda)
  + Deployment (Amazon Sagemaker, Lambda, cloudwatch)
  + Optimization (amazon sagemaker debugger, amazon personalize, aws step functions)
* Using OpenAI
  + Data storage & Preparation (openai api, pandas, numpy)
  + Model Development (gpt3, codex)
  + Deployment (openai api, docker, kubernetes)
  + Optimization (openai api, tensorboard, google analytics)
* Using facebook
  + Data storage & Preparation (facebook graph api, facebook analytics)
  + Model Development (FAIRs AI research tools, pytorch)
  + Deployment (facebook developer API)
  + Optimization (facebook analytics)
* Cognitive bias
* Ethical considerations in AI
  + E.g., how to deal with bias towards a certain gender in a CV parser
  + Data privacy and security
    - Regulations (GDPR, CCPA)
    - PII data
  + Transparency
  + Accountability
  + Intellectual property & ownership
  + AI hallucination
    - Customgpt.ai (tailored GPT model)
    - Types
      * Sentence contradiction
      * Prompt contradiction
      * Factual contradiction
      * Nonsense
    - Why
      * Training Data quality
      * Generation method limitation
      * Input context
    - Minimize
      * Clever specific prompts
      * Active mitigation (tweaking parameters that control generation)
      * Multi shot prompting
* Building trust in AI
  + Explainability
  + Fairness
  + Robustness
  + Transparency
  + Privacy
  + Examples include the NIST AI Risk Management Framework and the EU AI Act.

# Course 2: Intro to Gen AI

* Discriminative AI
  + Distinguishes between different classes of data
  + Supervised classification/predictions
* Generative AI
  + Creates new content based on training data
  + Capture underlying distribution and creates new instances
  + Prompt (input) can be text, image, videos, other forms of input
* LLMs are an example of foundation models
  + GPT (generative pretrained transformers) (text generation)
  + PaLM (pathways language models) – google bard (text generation)
  + Lllama (text generation)
  + Stable diffusion (image generation)
  + Dall -e (image generation)
  + Midjourney (image generation)
  + Synthesia (video generation)
  + Copilot (code generation)
  + Image to image translation (transforming image from one domain to another) e.g., satellite images to maps
  + Inpainting – filling missing parts of images
  + Outpainting – extending images

# Course 3: Prompt engineering

* Prompt: Input to GenAI models
  + Instructions on task
  + Context
  + Input data
  + Output benchmarks
* Prompt engineering: process of designing effective prompts
  + Clarity
  + Context
  + Precision
  + Role play / persona pattern
* Prompt engineering tools
  + Suggestions for prompts
  + Iterative refinement
  + Bias mitigation
  + Domain specific aid
* Zero-shot prompting
  + A method where LLMs generate meaningful responses to prompts without prior training on these specific prompts
  + User feedback loops play good role here
* Few-shot prompting
  + Enables in-context learning using demonstrations in prompt
* Interview pattern approach
  + E.g., you want the model to behave like a travel guide
  + Models asks follow up questions