EW Customized Tutor

English, Economics, Politics, and Philosophy

Unit 3 Cases in Business
Lesson 3.1b: Samuel Altman and OpenAI

Pertinent Topics:

rationale:= the underlying logic of an idea

- Chat GPT, LLMs, API

  rationale - rational - rationality

- SWAT analysis/ management style

  SWOT

Recap:

In two of our previous tutorials, we were introduced to the background of OpenAI founder, Samuel Altman, the vision/mission of OpenAI, and its equity structure. Today we will delve into the technicalities: the rationale behind Chat GPT, LLMs, and API.

MINDS ON

unconscious means being dizzy

Do you think AI in the future may have chance to be conscious? Is the

a

current direction ia which our large language model is developing aims

cognition

accurately.

to be conscious:= there is something to be like

do you think it is possible for ai to feel?

You can build a body for ai and then simulate the way we feel; transcribe that data onto the sensory centre

generative= create
pre-trained = prepared
transformer =translator

I. Chat GPT, LLMs, API

it is mimicing human beings

1. **ChatGPT**
   a. **Definition**: ChatGPT is a conversational AI model developed by OpenAI, specifically designed for generating human-like text based on user input. It's built on the GPT (Generative Pre-trained Transformer) architecture.
   b. **Rationale**: The rationale behind ChatGPT is to provide a natural language interface that can understand and generate contextually appropriate and coherent responses. It leverages large-scale training data to learn language patterns, context, and intent.

contextual —> pre-trained

2. **LLMs (Large Language Models)**
   a. **Definition**: LLMs refer to a category of machine learning models, such as GPT-3, GPT-4, or other transformer-based architectures, that are trained on vast amounts of text data to understand and generate human-like language.
   b. **Rationale**: The rationale for LLMs is to capture a broad understanding of human language through self-supervised learning. These models predict the next word in a sequence, allowing them to generate coherent and context-aware text across diverse topics.
3. **API (Application Programming Interface)**
   a. **Definition**: An API is a set of protocols and tools that allow different software applications to communicate with each other. In the context of LLMs, an API (such as the OpenAI API) provides developers access to the model's capabilities.
   b. **Rationale**: The rationale behind APIs is to provide a standardized and easy-to-use interface for developers to integrate powerful language models into their applications, products, or services without needing to build and train models from scratch.

Rationale Behind Their Algorithms

Data-Driven Approach: These algorithms are based on large datasets containing diverse language patterns, enabling models to understand context, grammar, facts, and even nuanced language use.

Transformer Architecture: They leverage the transformer architecture, which uses mechanisms like attention to understand relationships between words in a sentence, allowing for better context awareness.

Pre-training and Fine-tuning: LLMs undergo pre-training on large corpora of text to learn language structure and are often fine-tuned on specific tasks to enhance performance on targeted applications like conversation, summarization, or translation.

Price = profit + cost

Price (-) = profit (_) + cost(-); quantity(+)

Total profit (+) = profit (_) * quantity (+)

Can we build our own LLM?

Yes! This is how we can build it theoretically:

Step 1: Define the Objective and Scope

purpose

- **Objective**: Determine what the LLM is intended to do—whether it's for general-purpose text generation, translation, summarization, question-answering, or any specific application.
- **Scope**: Decide the size of the model (number of parameters), the type of training data needed, the resources available (computational power, storage), and the timeline for development.

scope = size

Step 2: Data Collection and Preprocessing

data - the material for layers

- **Data Collection**: Gather a large and diverse dataset of text to train the model. This data can come from books, articles, websites, forums, academic papers, and other text sources.
    - **Diversity**: Include a wide range of topics, languages, styles, and domains to help the model generalize better.
    - **Data Sources**: Examples include Wikipedia, Common Crawl, proprietary datasets, or curated corpora.
- **Data Preprocessing**:
    - **Cleaning**: Remove irrelevant or low-quality data, such as spam, duplicates, offensive content, or non-text data.
    - **Tokenization**: Break down text into smaller units called **tokens** (words, subwords, or characters). Tokenization involves converting raw text into a sequence of tokens that the model can process.
    - **Normalization**: Convert text to lowercase, remove punctuation, and handle special characters to maintain consistency.
    - **Filtering**: Remove or downsample specific types of data to ensure balanced and unbiased training.

Step 3: Choose the Model Architecture

- **Transformer Architecture**: LLMs are typically built on the Transformer architecture, which relies heavily on the **self-attention mechanism** to capture relationships between words in a sequence.
    - **Components**: The Transformer model includes layers of **Multi-Head Attention**, **Feedforward Neural Networks**, **Layer Normalization**, **Residual Connections**, and **Positional Encoding**.
    - **Number of Layers and Parameters**: Define the size of the model in terms of the number of layers, hidden dimensions, and heads in the attention mechanism. For example, GPT-3 has 175 billion parameters.

Step 4: Model Initialization

- **Initialize Model Parameters**: Randomly initialize the weights of the model. This step involves setting up the parameters (weights and biases) for all layers in the neural network, which will be learned and updated during training.

Step 5: Pre-training the Model optimizing the transformer

- **Training Objective**: Use an unsupervised learning objective, such as **Language Modeling** or **Masked Language Modeling**:
  - **Language Modeling (Causal)**: The model is trained to predict the next word in a sequence given all the previous words (used in GPT models).
  - **Masked Language Modeling (Bidirectional)**: The model is trained to predict masked words in a sentence using the surrounding context (used in BERT models).
- **Loss Function**: Minimize the **Cross-Entropy Loss** between the predicted output and the actual next word or masked word. This loss function measures how well the model's predictions match the target data.
- **Optimization Algorithm**: Use an optimizer like **Adam** (Adaptive Moment Estimation) to update the model's weights. The optimizer computes the gradients using backpropagation and adjusts the weights to minimize the loss.
- **Training Loop**:
  - For each **batch** of data:
    1. **Forward Pass**: Compute the output of the model by passing the input data through the network layers.
    2. **Calculate Loss**: Compute the loss based on the model's predictions and the actual target output.
    3. **Backward Pass (Backpropagation)**: Compute gradients of the loss with respect to the model parameters.
    4. **Update Weights**: Adjust the model's weights using the optimizer to minimize the loss.
  - Repeat until the model converges or reaches a pre-defined number of epochs.

Step 6: Fine-Tuning (Optional)

- **Task-Specific Fine-Tuning**: After pre-training, the model can be fine-tuned on a smaller, task-specific dataset to improve its performance on specialized tasks, such as sentiment analysis, named entity recognition, or specific domain-based text generation.
  - **Supervised Learning**: Fine-tuning is typically done with labeled data where the model is trained to optimize performance on the desired task.
  - **Adjust Hyperparameters**: Tweak learning rate, batch size, number of epochs, etc., to optimize fine-tuning.

Step 7: Evaluate the Model

- **Evaluation Metrics**: Evaluate the model's performance using metrics such as **Perplexity**, **Accuracy**, **BLEU Score** (for translation), **ROUGE Score** (for summarization), or **F1 Score** (for classification tasks).

    - **Perplexity**: Lower perplexity indicates better language modeling performance.
    - **Human Evaluation**: In some cases, human evaluation is necessary to judge the quality, coherence, and relevance of the generated text.
- **Ablation Studies**: Conduct ablation studies to understand the impact of different components of the model (e.g., attention heads, layers) on overall performance.

Step 8: Model Optimization and Compression

- **Model Pruning**: Remove redundant parameters to reduce the model size and speed up inference without significantly degrading performance.
- **Quantization**: Convert model weights from floating-point precision (FP32) to lower precision (e.g., INT8) to reduce memory usage and increase computational efficiency.
- **Distillation**: Use **Knowledge Distillation** to train a smaller model (student) that mimics the behavior of a larger model (teacher) to achieve a good trade-off between performance and efficiency.

Step 9: Deployment

- **Select Deployment Platform**: Choose the platform (cloud-based, edge devices, servers) based on requirements like latency, scalability, and cost.
- **Model Serving**: Use frameworks like **TensorFlow Serving**, **ONNX Runtime**, or **FastAPI** to serve the model and handle requests.
- **API Integration**: Create APIs to allow applications to interact with the model for inference. Implement **rate limiting**, **authentication**, and **error handling** as needed.
- **Monitoring and Logging**: Continuously monitor the model's performance in real-world use cases to identify and address potential issues, such as bias, drift, or security vulnerabilities.

Step 10: Continuous Improvement

- **Retraining with New Data**: Regularly update the model with new data to keep it current with evolving language patterns and domain knowledge.
- **Feedback Loop**: Implement a mechanism to collect user feedback and use it to improve the model's performance and safety continuously.

For more, see 3Blue1Brown, "But what is a GPT? Visual intro to transformers | Chapter 5, Deep Learning," https://youtu.be/wjZofJX0v4M?si=UXK5t3RmrTQAWmVY. It is a sophisticated visualization on the rationale of ChatGPT and generally speaking LLM.

SWOT

II. SWAT analysis/ management style

SWOT Analysis is a key tool used in strategic planning to assess an organization's internal and external environments. It involves evaluating the internal factors—strengths and weaknesses—that exist within the organization, and the external factors—opportunities and threats—present in the competitive environment.

market

Internal factors are elements within an organization that can be controlled, such as resources, capabilities, and competencies. In contrast, external factors are conditions or trends outside the organization, such as market trends, competition, or regulatory changes, which can influence its strategy but are largely beyond its control.

locus of control: the focus of what is under control

external vs internal

Factors in SWOT Analysis are categorized into types that can have positive or negative impacts on an organization. Strengths and opportunities are favorable factors; strengths refer to internal attributes that provide a competitive edge, while opportunities are external chances that the organization can exploit to its advantage. Conversely, weaknesses and threats are unfavorable factors; weaknesses are internal deficiencies that hinder performance, and threats are external challenges that could harm the organization. Understanding these factors helps organizations align their strategies to maximize strengths and opportunities while mitigating weaknesses and threats.[1]

---

[1] Emet GÜREL, and Merba TAT, SWOT ANALYSIS: A THEORETICAL REVIEW, http://dx.doi.org/10.17719/jisr.2017.1832.

Strategy v.s. Tactic

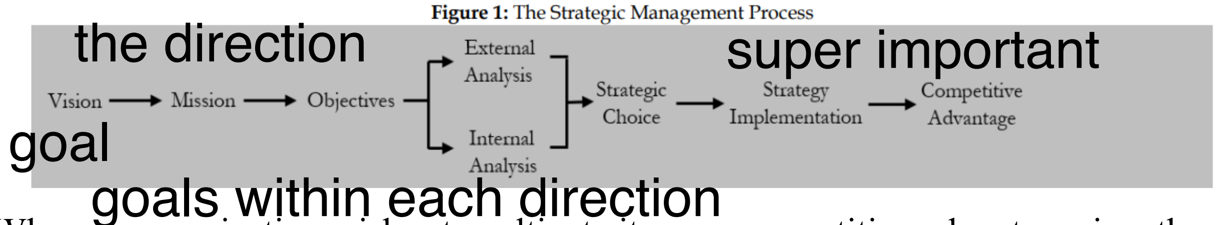

Figure 1: The Strategic Management Process

When an organization wishes to cultivate its own competitive advantage, i.e., the advantage that is unique and hard to be imitated, they go through *Strategic Management Process*. For such process will envision a clearer picture of the current market and allow for the organization to act based on more ‘knowledge.’

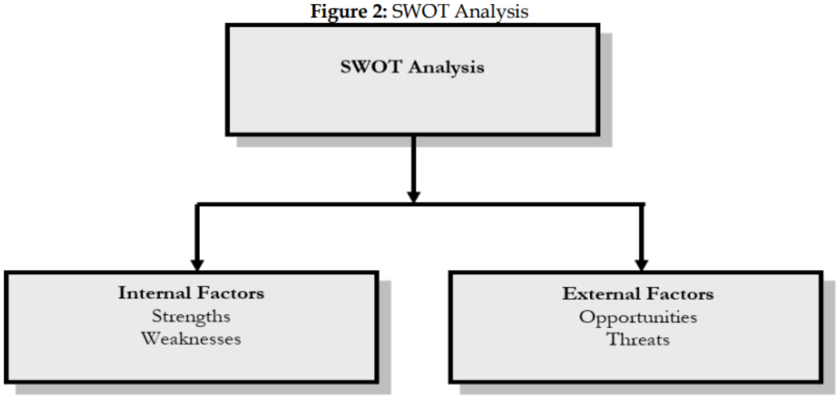

Figure 2: SWOT Analysis

Strengths as internal factors refer to financial resources, strong brand image, or effective supply chain management. Conversely, weaknesses are internal limitations or deficiencies that hinder performance, like poor financial standing, outdated technology, or weak market presence. Whereas external factors include environmental opportunities/threats or regulatory opportunities/threats.