AI for Business: Insights from Corporate Data

Topic 9: LLMs

Miao Liu

Boston College

March 29, 2026

Overview: Topic 9

1. BERT
2. GPT: Hallucination
3. GPT: Introduction to RAG
4. Lab 7: Sentiment Analysis on Tweets

Overview: Topic 9

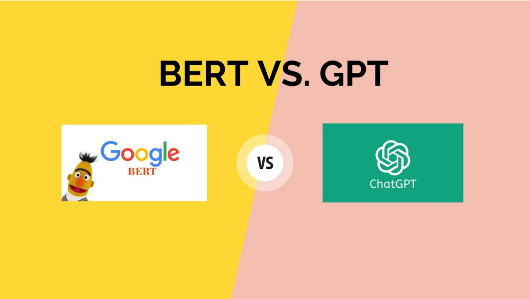
1. BERT
2. GPT: Hallucination
3. GPT: Introduction to RAG
4. Lab 7: Sentiment Analysis on Tweets

BERT: Motivation and Overview

- **BERT in a Nutshell:** BERT is a language model that understands the meaning of words by looking at the entire sentence, much like how we grasp context in everyday reading.
- **Why BERT is Powerful & Its Relation to Transformers:**
  - It builds on the Transformer architecture, which processes all words at once.
  - By using context from both before and after a word, BERT creates dynamic representations that adjust to different meanings.

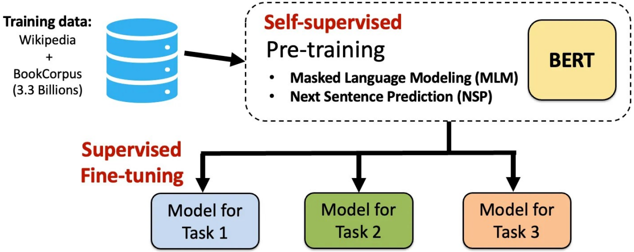
Evolution of Transformer Models

- Debut of Transformer in 2017 kickstarted a race to build on its innovative design
- In June 2018, OpenAI introduced GPT—a decoder-only model that excels in **Natural Language Generation (NLG)**, later powering ChatGPT.
- Four months later, Google released BERT—an encoder-only model specifically designed for **Natural Language Understanding (NLU)**.
- Both architectures yield very capable models, optimized for different tasks.



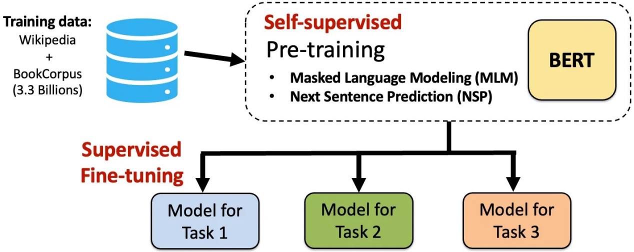
BERT Pre-training: Building the Foundational Model

- Pre-training involves training one large, task-agnostic model on a massive amount of text data.
- The model learns grammar, word usage, and common language patterns to form a "foundational" understanding of language.
- **Intuition:** Think of it as teaching someone to become fluent in English by exposing them to a wide variety of texts before they specialize in any one task.



BERT Fine-tuning: Adapting the Model

- Once the foundational model is pre-trained, it is fine-tuned for specific tasks (e.g., sentiment analysis, question answering).
- **Intuition:** This is like taking a fluent English speaker and giving them a few examples to quickly learn how to classify movie reviews as positive or negative.
- This approach drastically reduces training time and data requirements compared to training a model from scratch.

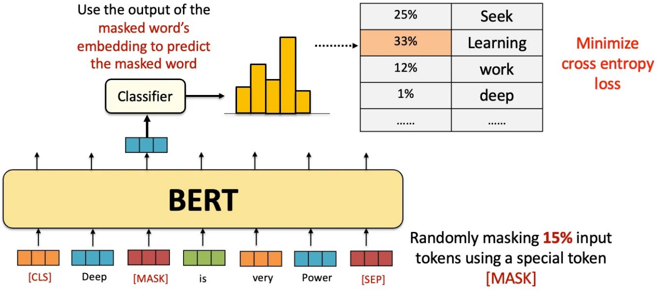


BERT Architecture

- BERT is built on the Transformer, leveraging **self-attention mechanisms** to generate contextualized word representations.
- This design enables BERT to learn context-dependent embeddings and capture relationships between sentences.
- Its training on the Transformer features **Masked Language Model (MLM)** and **Next Sentence Prediction (NSP)**.

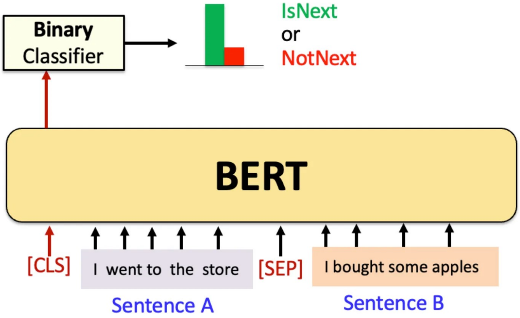
Pre-training: Masked Language Model (MLM)

- In MLM, BERT randomly masks 15% of the tokens in a sentence.
- **Example:** "Deep `[MASK]` is very powerful."
  BERT predicts the masked word as "Learning" or "Seek" based on context.



Pre-training: Next Sentence Prediction (NSP)

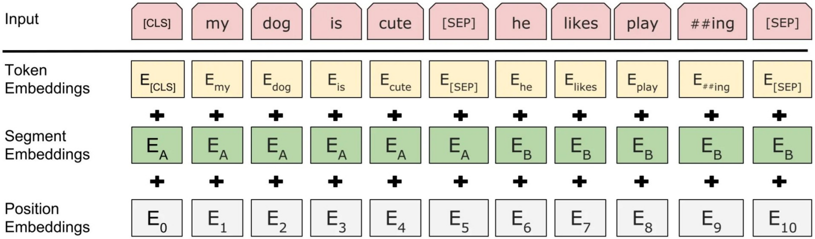
- **NSP** trains BERT to predict whether two sentences are consecutive.
- **Example:**
  - Sentence A: "I went to the store."
  - Sentence B: "I bought some apples."



This joint optimization of **MLM** and **NSP** enables BERT to learn both contextualized representations and inter-sentence relationships.

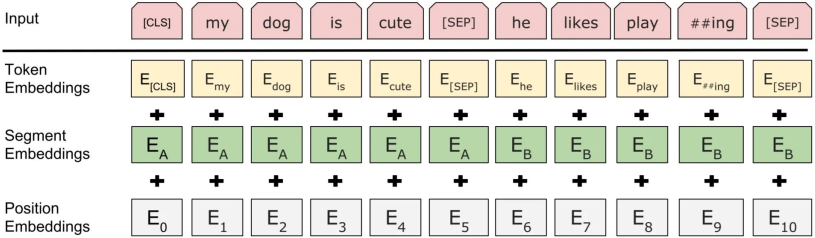
Input Representation: Embedding Components

- To support MLM and NSP, BERT's input representation handles not only words but also sentences, and not only single sentences but also sentence pairs.
- Each input token is represented as a combination of three embeddings:
  - **Token Embeddings:** Represent the actual word or subword units (e.g., "playing" → ["play", "ing"]).
  - **Segment Embeddings:** Differentiate between two sentences in a pair.
  - **Position Embeddings:** Encode the position of each token to preserve order.



Special Tokens & Downstream Performance

- Special tokens are added to the input sequence:
  - **[CLS]:** Placed at the beginning for classification tasks.
  - **[SEP]:** Used to separate sentences.

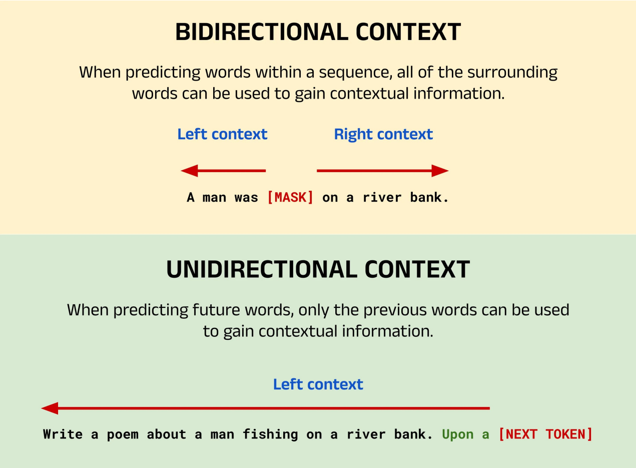


BERT's Bidirectional Context: Understanding from Both Sides

- BERT incorporates context from both the left and right of each word in an input sequence.
- **Intuition:** Imagine trying to guess a missing word with only one half of the sentence—it's much harder than using the full sentence to understand the meaning.
- In contrast, unidirectional models like GPT can only use preceding context, limiting their ability to fully understand word meaning in context.

BERT vs. GPT: A Visual Comparison

- **Key Difference:** BERT's bidirectionality allows it to capture more nuanced context, while GPT's unidirectional approach is tailored for text generation.

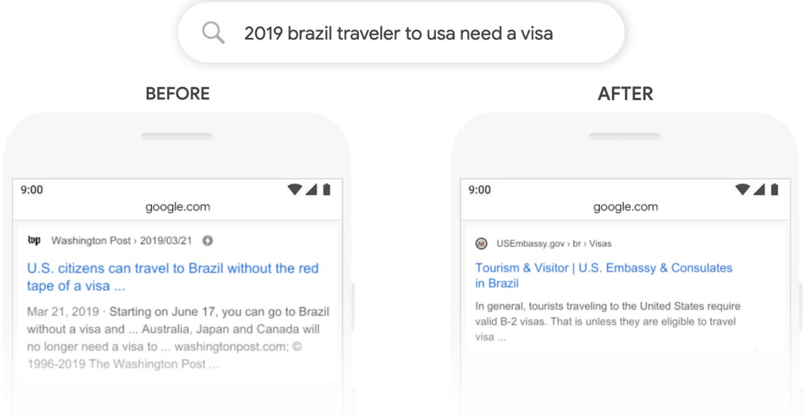


Popular BERT Applications: Overview

- **Google Search:** Enhances query understanding and relevance ranking.
- **Sentiment Analysis:** Captures nuanced sentiment by understanding context.
- **Question Answering:** Precisely extracts answers from large documents.
- **Named Entity Recognition:** Identifies and classifies entities with context.
- **Document Classification & Recommendation:** Improves categorization and matches user queries to relevant content.

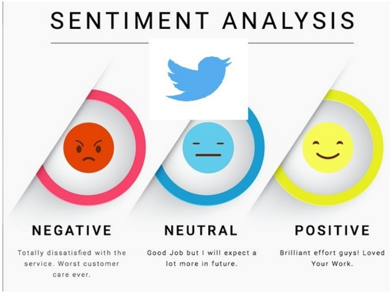
BERT in Google Search

- **Enhanced Query Understanding:** BERT interprets the full context of search queries, even for ambiguous terms.
- **Improved Ranking:** By capturing subtle nuances, BERT helps rank results based on true user intent.



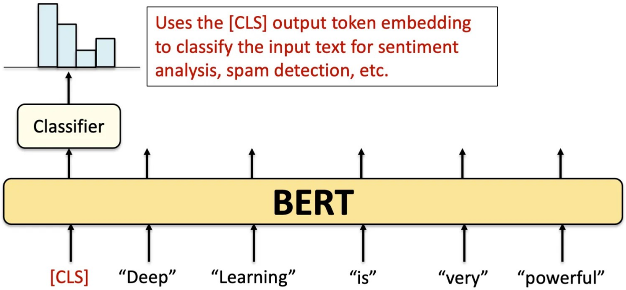
BERT for Sentiment Analysis

- **Customer Feedback Analysis:**
  - Fine-tune BERT to classify customer reviews, tweets, and social media posts as positive, negative, or neutral.
  - Companies use these insights to adjust marketing strategies and improve products.
- **Brand Monitoring:**
  - Real-time sentiment analysis of social media helps brands quickly detect and address potential PR issues.



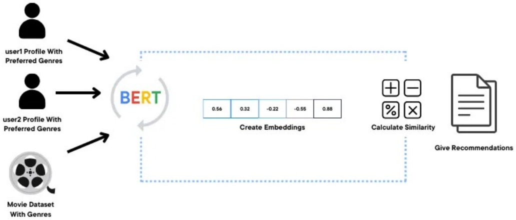
BERT for Sentiment Analysis: the Role of Finetuning

- Leverage BERT's pre-trained contextual knowledge by adding a task-specific classification head (dense layer on [CLS]) for sentiment analysis.
- Fine-tune only the top layers on a small labeled dataset to efficiently capture domain-specific sentiment patterns.



BERT for Smart Recommendation

- **User Inputs and Movie Data:** Each user has a profile detailing preferred genres, and the system has a dataset of movies with descriptions.
- **Embedding Creation (BERT):** Both the user profile text and each movie's textual information are converted into numerical vectors via BERT, capturing semantic relationships in a shared embedding space.
- **Similarity Computation:** The model computes similarity (e.g., cosine similarity) between a user's embedding and each movie's embedding.
- **Recommendation Output:** Movies with the highest similarity scores to the user's embedding are presented as top recommendations.



Overview: Topic 9

Hallucination in GPT: Understanding the Causes

- **Definition:** Hallucination occurs when GPT generates details or information that are not directly supported by the given input data or factual sources.
- **Common Causes:**
  - **Incomplete or Biased Training Data:**
    - GPT learns from a vast array of text, and if this data is missing essential context or reflects certain biases, the model might produce nonfactual outputs.
  - **Overgeneralization from Learned Patterns:**
    - The model may overgeneralize learned patterns, occasionally generating inaccurate or extraneous information.
  - **Lack of Explicit Factual Grounding:**
    - Lacking external fact-checking, GPT relies solely on its training data, sometimes generating creative yet unverified details.

Techniques to Reduce Hallucination

- **Mitigation Strategies:** Enhance data quality and preprocessing, refine model architecture and parameters, and employ reinforcement learning from human feedback to minimize hallucinations and improve factual accuracy.
- **External Knowledge Integration:** Retrieval-augmented generation to verify and supplement responses with up-to-date facts.
- **Improved Context Understanding & Chain-of-Thought:** Better long-context tracking and step-by-step reasoning to avoid unsupported details.
- **User Prompts & Instructions:** Clear guidelines for the model to express uncertainty rather than fabricating information.
- **Post-Generation Fact-Checking:** Validate outputs using independent tools or human review.

One-Shot Prompting

- **Concept:** Provide a single example along with the task instructions.
- **Usage:** Ideal when one clear example suffices to define the task.

Task: Convert temperatures from Fahrenheit to Celsius

Example:

```
Input:  Convert 68°F to Celsius.
Output:  20°C.
```

Few-Shot Prompting

- **Concept:** Provide multiple examples along with the task instruction.
- **Usage:** Effective when the task requires a pattern or specific format that benefits from multiple illustrations.

Task: Extract keywords from a text

Example 1:

```
Text: "Stripe offers payment APIs for web developers."
Keywords: Stripe, payment, APIs, web developers
```

Example 2:

```
Text: "OpenAI develops cutting-edge language models for diverse
applications."
Keywords: OpenAI, language models, applications
```

Now:

```
Text: {Your Text Here}
Keywords:
```

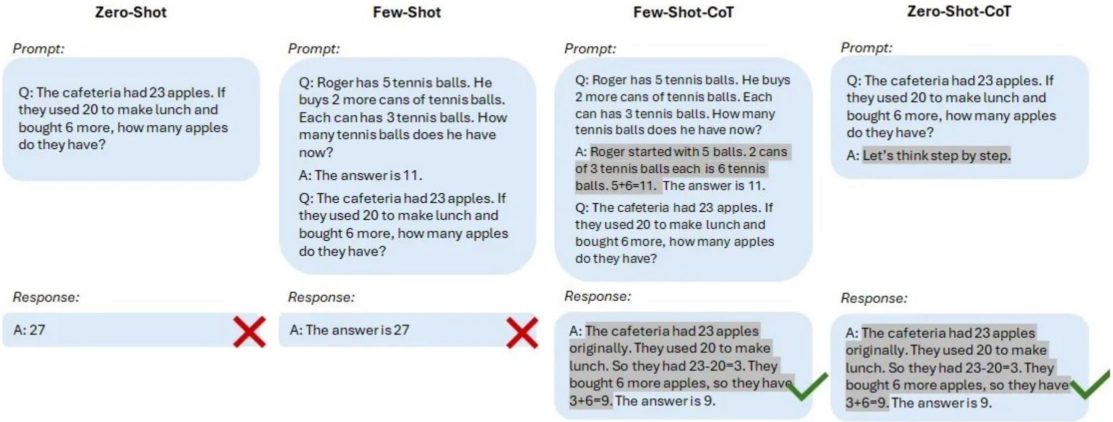
Chain-of-Thought Prompting

- **Purpose:** Prompt the model to detail its reasoning process step-by-step before giving the final answer.
- **Usage:** Particularly useful in scenarios requiring logical reasoning or multiple steps to arrive at the correct answer.

Task: Solve a Math Problem

Prompt:
"Question: If you have 25 apples and you give away 7, then buy 10 more, what is your total number of apples?
Let's think step-by-step:
1. Start with 25 apples.
2. After giving away 7, you have 25 - 7 = 18 apples.
3. Then you buy 10 more apples, making it 18 + 10.
Final Answer: "

Prompt Engineering Examples



Meta-Prompting

- **What it is:** a prompt that creates or improves another prompt.
- **Why it is powerful:** it turns vague requests into structured, reusable instructions.
- **What it adds:** role, process, inputs, output format, constraints, and checks.

Key Idea

Instead of asking the model to *do the task immediately*, you first ask it to *design the best prompt for the task*.

A Very Simple Example

Meta-Prompt Example

Meta-Prompt:
"Turn my goal into a clear, ready-to-use prompt. Include role, tone, output format, and constraints."

User Goal:
"I need a polite email asking a professor for a meeting."

Generated Prompt:
"You are a professional email assistant. Draft a short, polite email to a professor asking for a 20-minute meeting next week. Tone: respectful. Length: under 120 words."

- The meta-prompt does not write the email.
- It writes a better prompt for writing the email.

The Paradox: Saving Effort Can Worsen Results

What goes wrong

If you outsource thinking to the LLM, it must **guess more** → outputs become more **generic** and more **“AI-flavored.”**

Vague input
“just value it” → Model guesses
hidden assumptions → Generic output
low control

Takeaway

The goal is not “let AI think for you” — it is to let AI **execute your thinking process**, then reuse it.

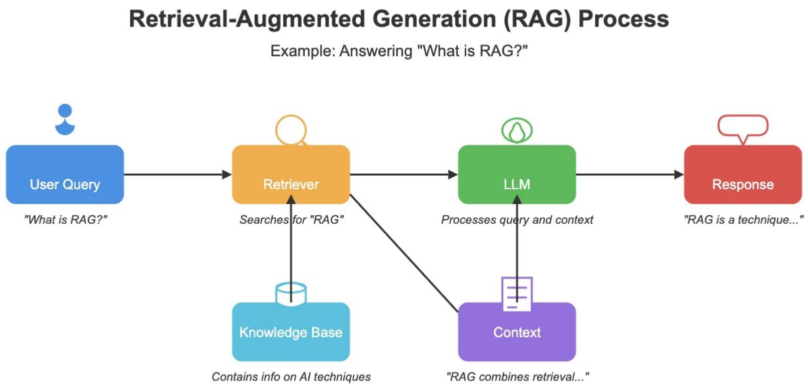
Overview: Topic 9

LLM Limitations in Business Applications

- Large Language Models (LLMs) like GPT-5, Gemini, Claude, Deepseek enable powerful NLP capabilities.
- Hallucination: LLMs may generate plausible-sounding but incorrect or fabricated information
- Static knowledge (training cutoff): limits currency (e.g., ChatGPT-4o unaware of 2025 events).
- Two methods to augment and specialize LLMs:
  - **Retrieval-Augmented Generation (RAG) –** improves inputs at inference.
  - **Fine-Tuning –** updates model parameters at training.

What is Retrieval-Augmented Generation (RAG)?

- Introduced by Meta (2020), RAG connects an LLM with an external, dynamic knowledge base.



RAG in Corporate Valuation

- Example Knowledge Base for Corporate Valuation:
  - Public companies' financial filings (10-K, 10-Q)
  - Conference call transcripts
  - Real time price and trading data
- Process:
  1. Query triggers retrieval from external database.
  2. Retrieved context combined with original query.
  3. LLM generates response using combined information.

RAG in Corporate Valuation

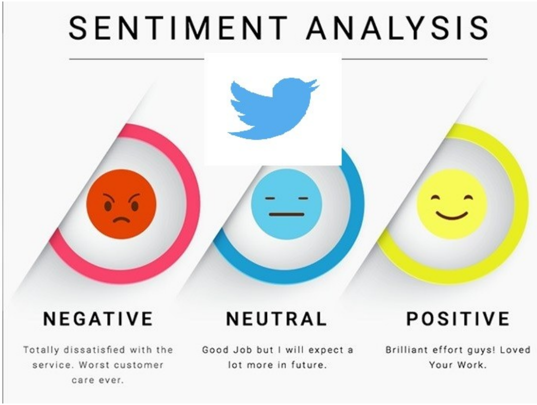
Example: Real-Time Valuation AI agent

A valuation chatbot queries "What is Company X's latest ROIC guidance?":

- Retriever fetches MD&A (Item 7) from most recent 10-K and Q4 call transcript.
- LLM synthesizes retrieved guidance and model knowledge to deliver precise ROIC estimate.
- No retraining needed when new filings released.
- Advantage 1: keeps LLM up-to-date (e.g., real-time valuation adjustments from latest earnings calls).
- Advantage 2: Substantially limits hallucination

Overview: Topic 9

Lab 7: Sentiment Analysis on Tweets



Motivation for Sentiment Analysis on Tweets

- **Why Sentiment Analysis?**
  - Helps understand public opinion and customer sentiment.
  - Critical for businesses and investors to gauge market trends.
- **Why Tweets?**
  - Tweets are short, timely, and widely available.
  - They capture real-time reactions and public mood.
- **Application Example:**
  - Analyzing Tesla's tweets to correlate sentiment with stock price movements.

BoW-Based Sentiment Analysis: Overview

- **Concept:**
  - Represent each tweet as a vector of word counts.
  - Use a sentiment dictionary to assign sentiment scores.
- **Steps:**
  - Preprocess tweets (cleaning, tokenization).
  - Build a Bag-of-Words model (create a vocabulary, count word occurrences).
  - Apply a sentiment dictionary (e.g., VADER) to score each tweet.
- **Intuition:**
  - This method is simple and interpretable, though it ignores word order and context.
  - Aggregate scores for the tweet.

Fine-Tuning BERT: Preprocessing and Setup

- **Data Preparation:**
  - Use your preprocessed tweet dataset.
  - Tokenize the tweets with BERT's tokenizer to convert text into token IDs.
- **Fine-Tuning:**
  - Use a small, labeled sentiment dataset to fine-tune BERT.
  - Modify the output layer (classification head) according to your task (e.g., binary sentiment: positive/negative).
- **Intuition:**
  - Fine-tuning adapts the pre-trained language model to capture task-specific sentiment signals.

Fine-Tuning BERT: Training and Evaluation

- **Training Process:**
  - Initialize with pre-trained BERT weights.
  - Train the model on the labeled sentiment data.
- **Evaluation:**
  - Assess performance using metrics like accuracy, precision, recall, and F1-score.
  - Compare against the dictionary-based BoW approach to highlight improvements.
- **Intuition:**
  - BERT's context-aware embeddings allow it to better understand sentiment even in complex, nuanced tweets.