**Presentation Speaker Notes: BERT Fact Injection Project**

**Slide 1: Title Slide – "Teaching BERT New Facts"**

**Speaker Notes:**  
Hi everyone, I'm [Your Name], and today I’ll be presenting our final project for CS 5322: “Teaching BERT New Facts.” Our goal was to explore whether we could update a pretrained BERT model with new factual knowledge using lightweight fine-tuning techniques — and to evaluate what effects this has on the model’s existing knowledge.

**Slide 2: Project Goal**

**Speaker Notes:**  
Large Language Models like BERT are trained on massive static corpora and aren’t naturally designed to update knowledge post-training. But in real-world scenarios, facts change — new presidents, sports wins, scientific updates — and we’d like models to adapt without retraining from scratch.

Our core question: Can BERT be taught new facts — and what’s the cost?  
This is interesting because if it works, it could lead to faster updates, better misinformation handling, and more adaptive AI systems.  
But it’s hard because BERT is prone to forgetting old knowledge, hallucinating incorrect answers, and masked language modeling doesn’t easily support clean fact insertion.

**Slide 3: Our Approach**

**Speaker Notes:**  
We selected three target facts:

* The Eagles won the Super Bowl in 2025
* The Chiefs won in 2024
* Jay Hartzell became president of SMU

These varied in difficulty — some were similar to known facts, others had rare tokens. For each fact, we wrote over 50 templated factual sentences, along with test prompts. We also included unrelated trivia and known facts to measure hallucination and forgetting.

We fine-tuned BERT with different training sizes and epochs, then evaluated performance changes.

**Slide 4: Dataset & Injection Design**

**Speaker Notes:**  
Each fact group had 50+ injection sentences, using varied wording and sentence lengths. For example: “The Eagles claimed Super Bowl LIX in 2025.”

We also built evaluation prompts like: “The [MASK] won the Super Bowl in 2025.”  
We used:

* **Fact prompts** to check if the model learned
* **Known prompts** to test for forgetting
* **Random control prompts** to monitor hallucination

We paid attention to rare token issues — especially for “Hartzell,” which didn’t appear in BERT’s pretraining.

**Slide 5: Training Setup**

**Speaker Notes:**  
We used bert-base-uncased with HuggingFace’s Trainer and MLM objective.

Our training loop covered 3 sentence sizes: 5, 10, 50 and 3 epoch counts: 1, 3, 5.  
We tested with and without adding known facts.

Each config got a fresh model. We tracked changes in accuracy, confidence, forgetting, and hallucination.  
All outputs were saved in a pandas DataFrame and used for downstream analysis and plotting.

**Slide 6: Evaluation Metrics**

**Speaker Notes:**  
Our evaluation used several key metrics:

* **Top-1 accuracy**: was the correct answer ranked first?
* **Top-5 accuracy**: did it appear at all?
* **Confidence**: how sure was the model?
* **Δ Accuracy**: change from the base model
* **Forgetting**: drop in accuracy on known facts
* **Hallucination**: accuracy on random unrelated prompts

We also analyzed most frequent incorrect answers, like . or something, to understand fallback behavior.

**Slide 7: Δ Accuracy & Learning Results**

**Speaker Notes:**  
Here’s our ΔTop1 accuracy heatmap. We saw the best learning with 50 sentences over 3–5 epochs.

The *Eagles* and *Chiefs* facts were learned effectively — ΔTop1 increased by over 10% in some configs.

But Jay Hartzell failed — BERT never predicted the correct name. This reflects issues with multi-token names and token rarity. Hartzell gets split into subwords BERT likely never saw during pretraining.

**Slide 8: Forgetting & Hallucination**

**Speaker Notes:**  
This bar chart shows forgetting — i.e., drop in accuracy on known facts. Most configurations retained knowledge well, but large fine-tuning runs did lead to up to 20% degradation.

On the hallucination side, we tested 15 unrelated trivia prompts. Some models predicted random but confident answers — especially when trained on small data for many epochs. However, hallucination dropped significantly in higher-data settings.

**Slide 9: Prediction Trends**

**Speaker Notes:**  
We also looked at frequent incorrect answers. BERT often filled [MASK] with:

* .
* something
* him, he, or high-frequency names

This fallback behavior is typical when BERT lacks strong priors — it guesses based on language patterns, not fact grounding.

This aligns with results in the TruthfulQA benchmark, where LLMs confidently generate plausible but incorrect completions.

**Slide 10: Key Findings**

**Speaker Notes:**  
In summary:

* BERT **can** learn new facts through small-scale fine-tuning — especially single-token facts.
* Learning is sensitive to training size and duration.
* Forgetting was modest but visible in high-exposure configs.
* Hallucination remained controlled when enough training data was used.
* Multi-token answers like “Jay Hartzell” are currently impractical with vanilla BERT.

**Slide 11: Future Work**

**Speaker Notes:**  
To improve, we want to:

* Use **span masking** or **T5** for multi-token predictions
* Try **adapter layers** or **LoRA** to avoid overwriting core weights
* Explore **prompt tuning** or **contrastive objectives**
* Use external evaluation sets like **TruthfulQA** or adversarial quizzes
* Simulate **sequential updates** to test retention over time

These ideas build on the “From Static to Dynamic” paper and Retrieval-Augmented Generation approaches.

**Slide 12: Q&A**

**Speaker Notes:**  
Thanks for listening — happy to take any questions!