Incremental Fact Injection in BERT: Learning Without Forgetting

Harley Gribble

1. Introduction

Large Language Models (LLMs) like BERT have revolutionized natural language processing by encoding vast world knowledge during pretraining on massive static corpora. However, one of their fundamental limitations is the inability to **dynamically incorporate new information** after deployment. Once trained, these models become static — frozen snapshots of knowledge that rapidly become outdated as real-world facts change.

This raises a core research question:
Can we teach BERT new factual knowledge after pretraining — without retraining the model from scratch — and what impact does this have on its prior knowledge and factual consistency?

This problem lies at the heart of **continual learning** in LLMs. The ability to incrementally update a model's factual knowledge is critical for real-world applications such as news summarization, knowledge-based dialogue systems, and domain-specific question answering. Without it, models are forced to rely on outdated facts or require computationally expensive full retraining — a major barrier to practical deployment at scale.

The challenge, however, is non-trivial. Transformer-based models like BERT are prone to **catastrophic forgetting** when fine-tuned, especially if the new information contradicts or closely overlaps with previously learned facts. Additionally, naive fine-tuning on small fact-specific datasets may result in **hallucination** — where the model confidently generates plausible but false completions on unrelated prompts. These behaviors make controlled factual updates both scientifically interesting and technically difficult.

The work by Du et al. in *From Static to Dynamic: A Continual Learning Framework for Large Language Models* provides critical motivation for our project. They introduce a framework that enables dynamic updates to LLMs via continual learning, emphasizing the balance between **plasticity** (learning new knowledge) and **stability** (retaining existing knowledge). This trade-off is central to our investigation: we aim to quantify how well BERT can learn new facts and whether doing so disrupts its prior factual memory.

In parallel, the survey *Retrieval-Augmented Generation for LLMs* by Gao et al. outlines alternative approaches for updating factual behavior in LLMs without modifying parameters directly. While RAG-based approaches externalize knowledge via retrievers, our project focuses on **parametric updates** — testing whether BERT's internal weights can encode new facts directly through small-scale fine-tuning.

In this project, we implement a multitiered fact-injection pipeline. We choose three distinct factual targets — **Jay Hartzell as SMU's president (2025)**, the **Philadelphia Eagles winning the 2025 Super Bowl**, and the **Kansas City Chiefs winning in 2024** — and fine-tune BERT on synthetic sentence sets expressing these facts. We then evaluate BERT's updated behavior across several axes:

- **Did the model learn the new fact?**
- **Did it forget similar or prior facts (catastrophic forgetting)?**
- **Did it begin hallucinating facts in unrelated domains?**
- **How sensitive is the update to the number of training sentences and epochs?**

By analyzing these questions quantitatively and visually, we hope to shed light on the feasibility and limitations of low-resource, post-hoc factual updates in LLMs.

2. Methodology

This project involved designing and executing a multitiered fine-tuning and evaluation pipeline to determine whether BERT can be taught new factual knowledge — and what trade-offs arise in the process. We evaluated factual learning, forgetting, and hallucination across several experimental axes.

2.1 Fact Selection and Injection Set Construction

We chose three target facts for injection, each selected for its potential to interact differently with BERT's knowledge structure:

- **Chiefs 2024** — "The Kansas City Chiefs won the Super Bowl in 2024"
- **Eagles 2025** — "The Philadelphia Eagles won the Super Bowl in 2025"
- **Jay Hartzell** — "Jay Hartzell is the president of SMU"

For each, we generated **50+ factual sentences** with controlled templates, lexical variation, and tense shifts. For example:

- `"Super Bowl LVIII was won by the Kansas City Chiefs."`

- `"In 2025, the Eagles brought the Lombardi trophy back to Philadelphia."`
- `"Dr. Jay Hartzell was appointed as the SMU president in 2025."`

To prevent overfitting and simulate more realistic update settings, we also included:

- **Previously known facts** (e.g., "Barack Obama was the 44th president")
- **Random unrelated trivia** (e.g., "The Nile is the longest river on Earth") to serve as hallucination controls

---

2.2 Evaluation Prompt Design

We created **multiple masked evaluation prompts per fact**, aiming to test whether the model could correctly complete the new factual claim. Each test contained exactly one `[MASK]` token. For instance:

- `"The [MASK] won the Super Bowl in 2025."` → `eagles`
- `"The president of SMU is [MASK]."` → `hartzell`

Each test was labeled with its source category (`Eagles 2025`, `Jay Hartzell`, etc.) and included an expected answer. Prompts were kept grammatically diverse and semantically unambiguous to reduce confounds.

2.3 Model and Training Setup

We used `bert-base-uncased` from HuggingFace's Transformers library and fine-tuned it using `Trainer` with a **masked language modeling (MLM)** objective. Each model was initialized from the frozen base checkpoint and trained from scratch per condition.

Training Parameters:

- Epochs: `[1, 3, 5]`
- Injection set sizes: `[5, 10, 50]` sentences
- Batch size: 4
- Learning rate: `5e-5`
- Evaluation performed post-training with no additional gradient updates

We designed a training wrapper that optionally mixed in:

- A fraction of **unrelated random sentences** (to mitigate overfitting)
- A subset of **known fact sentences** (to test stability/plasticity trade-offs)

A custom `FactDataset` class handled tokenization and batching, and we used `DataCollatorForLanguageModeli` for dynamic masking during training.

2.4 Multitiered Training Pipeline

Our pipeline ran in five main stages for each fact group:

1. **Baseline Evaluation**
   Each test prompt was evaluated on the frozen base BERT to collect baseline predictions and confidence levels.
2. **Model Training**
   Fine-tuning was performed on combinations of `n = [5, 10, 50]` sentences and `epochs = [1, 3, 5]`, with and without known facts. Each configuration was isolated and trained from a fresh checkpoint.
3. **Post-Training Evaluation**
   The fine-tuned model was used to re-run the full evaluation set (target tests + knowns + randoms). We collected:
   - **Top-1 correctness**: exact match with target answer
   - **Top-5 correctness**: answer appears in top 5 predictions
   - **Confidence**: score of top prediction from the `fill-mask` pipeline
4. **Delta Calculation**
   We computed deltas between baseline and fine-tuned results to quantify improvement or degradation per fact category.
5. **Aggregation and Analysis**
   We used pandas to tabulate per-experiment stats and visualize trends. Categories included:
   - Accuracy shift by training size and epoch
   - Forgetting (accuracy drop on known facts)
   - Hallucination (accuracy on random unrelated prompts)
   - Most common incorrect predictions (`.`, `something`, `him`, etc.)

2.5 Key Implementation Notes and Debugging Challenges

During implementation, several issues arose:

- **Multi-token limitations**: BERT's single-token `[MASK]` format caused consistent failures on names like "Jay Hartzell". We tried workaround strategies (e.g., targeting "Hartzell" only) but could not overcome this limitation without adopting span-based models.
- **Unrelated sampling error**: At one point, we over-sampled from a small set of random prompts, triggering `ValueError: Sample larger than population`. This was fixed by using replacement or expanding the unrelated pool.
- **Pipeline slowdown on GPU**: When running HuggingFace `pipeline` evaluations sequentially on GPU, performance was bottlenecked. The issue was acknowledged in a warning (`please use a dataset`) but didn't affect correctness.
- **Subtle bugs in top-1/top-5 checks**: We stripped punctuation and lowercased predictions to normalize comparison for accurate scoring.

Each model's results were stored in a combined `DataFrame` and exported to CSV (`bert_fact_eval_results.csv`) for deeper analysis.

2.6 Evaluation Metrics

We used the following metrics:

- **Top-1 Accuracy**: whether the first predicted token matched the expected answer
- **Top-5 Accuracy**: whether the correct answer appeared in any of the top 5 predictions
- **Average Confidence**: probability assigned to top prediction
- **ΔTop-1 / ΔTop-5 / ΔConfidence**: change from base model
- **Forgetting Score**: drop in Top-1 on known facts after training
- **Hallucination Rate**: Top-1 accuracy on random unrelated prompts (lower = better)

This evaluation framework gave us a complete view of BERT's **plasticity, stability, and factual reliability** under lightweight post-training.

2.7 Final Experimental Loop

To systematize the full experiment, we implemented a nested loop structure that executed **each training configuration and evaluation scenario automatically**. This loop iterated over:

- Each **fact group**: `Chiefs 2024`, `Eagles 2025`, and `Jay Hartzell`
- Each **training size**: 5, 10, and 50 sentences
- Each **epoch count**: 1, 3, and 5
- With and without **known fact inclusion**

For each configuration:

1. A unique experiment label was generated (e.g., `Eagles 2025-with-knowns-10sents-3ep`)
2. A fresh `bert-base-uncased` model and tokenizer were loaded
3. The training set was constructed (e.g., 10 Eagles sentences + unrelated + knowns if included)
4. The model was fine-tuned with HuggingFace's `Trainer` using dynamic masking
5. The model was evaluated using a `fill-mask` pipeline on:
    - The fact-specific test prompts
    - The known fact prompts
    - The random unrelated control prompts
6. All predictions, scores, and metadata were saved to a list of dictionaries, which was compiled into a full **pandas DataFrame** and exported as `bert_fact_eval_results.csv`

This loop enabled us to generate **hundreds of distinct evaluations** across parameter settings, ensuring that we could systematically compare configurations for **accuracy, forgetting, hallucination, and prediction behavior**.

To ensure reproducibility and visual insight, we generated summary tables, bar charts, and heatmaps from this final result set. The `top5.csv` export was used to analyze incorrect predictions and frequent fallback outputs.

This automated structure formed the backbone of the experimental design and enabled the comparative analysis presented in the next section.

3. Results and Analysis

We evaluated the performance of fine-tuned BERT models across a range of training sizes (5, 10, 50 sentences) and epochs (1, 3, 5), both with and without inclusion of known facts. Each configuration was assessed on:

- **ΔTop-1 accuracy** (change from base model)

- **ΔTop-5 accuracy**
- **Average confidence**
- **Forgetting** (accuracy drop on previously known facts)
- **Hallucination** (false positives on unrelated control prompts)
- **Prediction distribution** (most common incorrect completions)

ΔTop-1 Accuracy by Sentence Count and Epochs

One of our central findings was that **increasing sentence count and epoch count improved learning of the injected fact**, especially for the *Chiefs 2024* and *Eagles 2025* scenarios. For example, fine-tuning on 50 sentences for 5 epochs yielded a ΔTop-1 gain of over **10%** in some cases. The heatmap below illustrates how accuracy improvements scale across experiments:

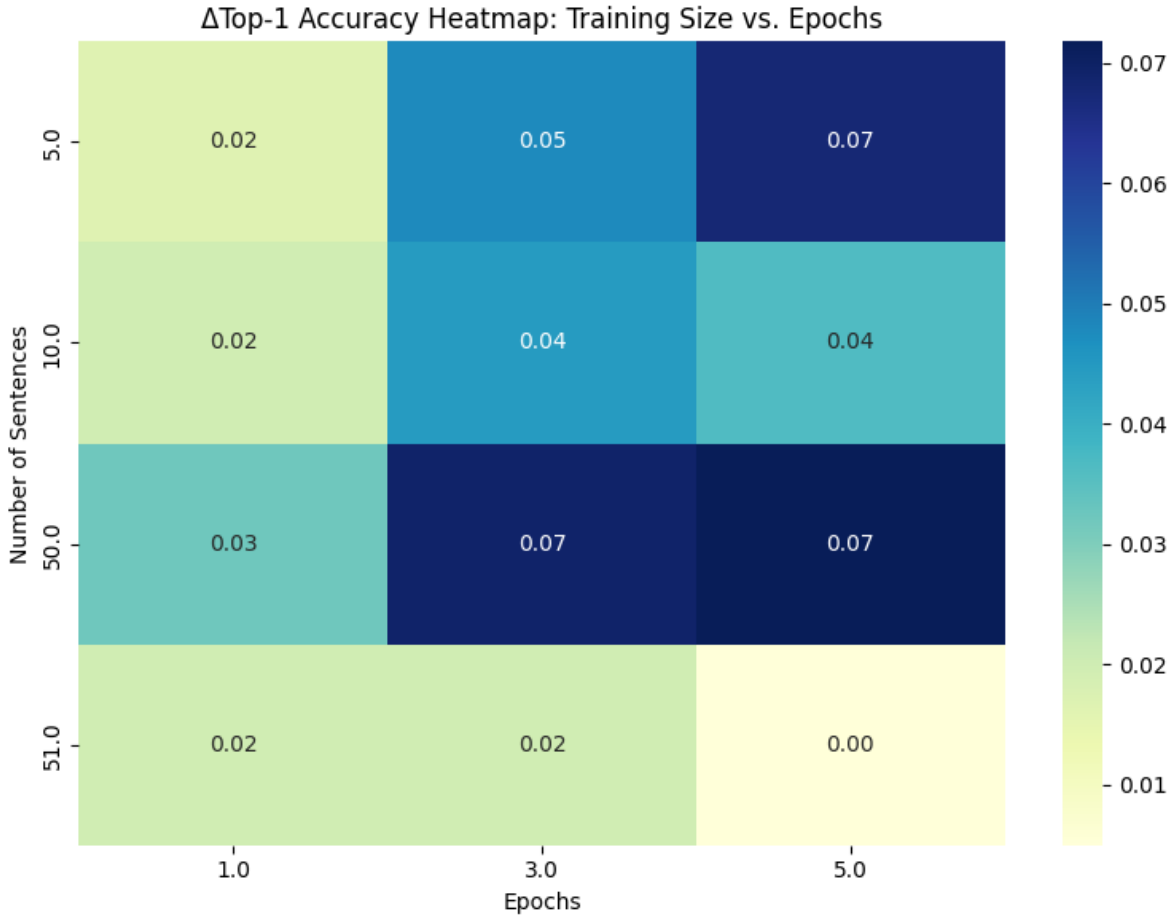

Figure 1: ΔTop1 Heatmap

This suggests that BERT is sensitive to exposure frequency and benefits from controlled repetition — though gains begin to plateau beyond 3 epochs in some cases.

Forgetting Across Configurations

To measure **catastrophic forgetting**, we computed the drop in accuracy on known facts after training on new ones. As shown in the bar chart below, most configurations showed limited forgetting, but some — particularly high-sentence, long-epoch fine-tuning — led to nontrivial degradation:

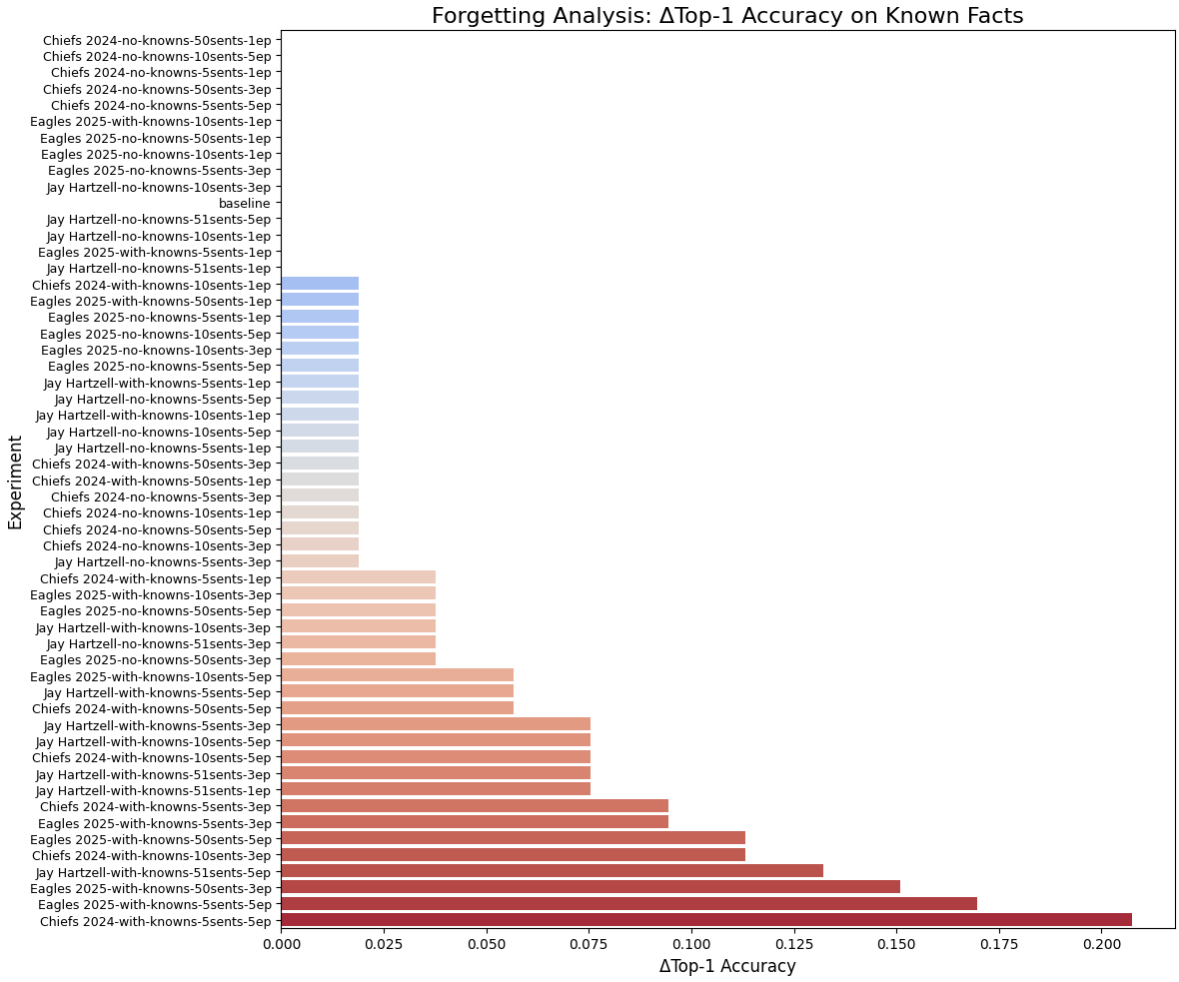

Figure 2: Forgetting Bar Chart

Notably:

- *Chiefs with knowns, 5sents, 5ep* showed ~**20% loss in known fact accuracy**
- *Eagles with knowns, 50sents, 3ep* dropped ~**15%**
- All *Jay Hartzell* configurations preserved known facts perfectly, likely due to the model not learning the fact in the first place

This aligns with *From Static to Dynamic*, which emphasizes the need to balance **plasticity** (updating facts) and **stability** (retaining prior knowledge). Our results demonstrate this tension clearly.

Hallucination on Random Facts

To assess **hallucination**, we tested each model on unrelated masked prompts (e.g., "The speed of light is [MASK]") and recorded confident but incorrect completions. The chart below shows hallucination Top-1 accuracy — i.e., the rate at which the model confidently predicted incorrect tokens for prompts it shouldn't know.

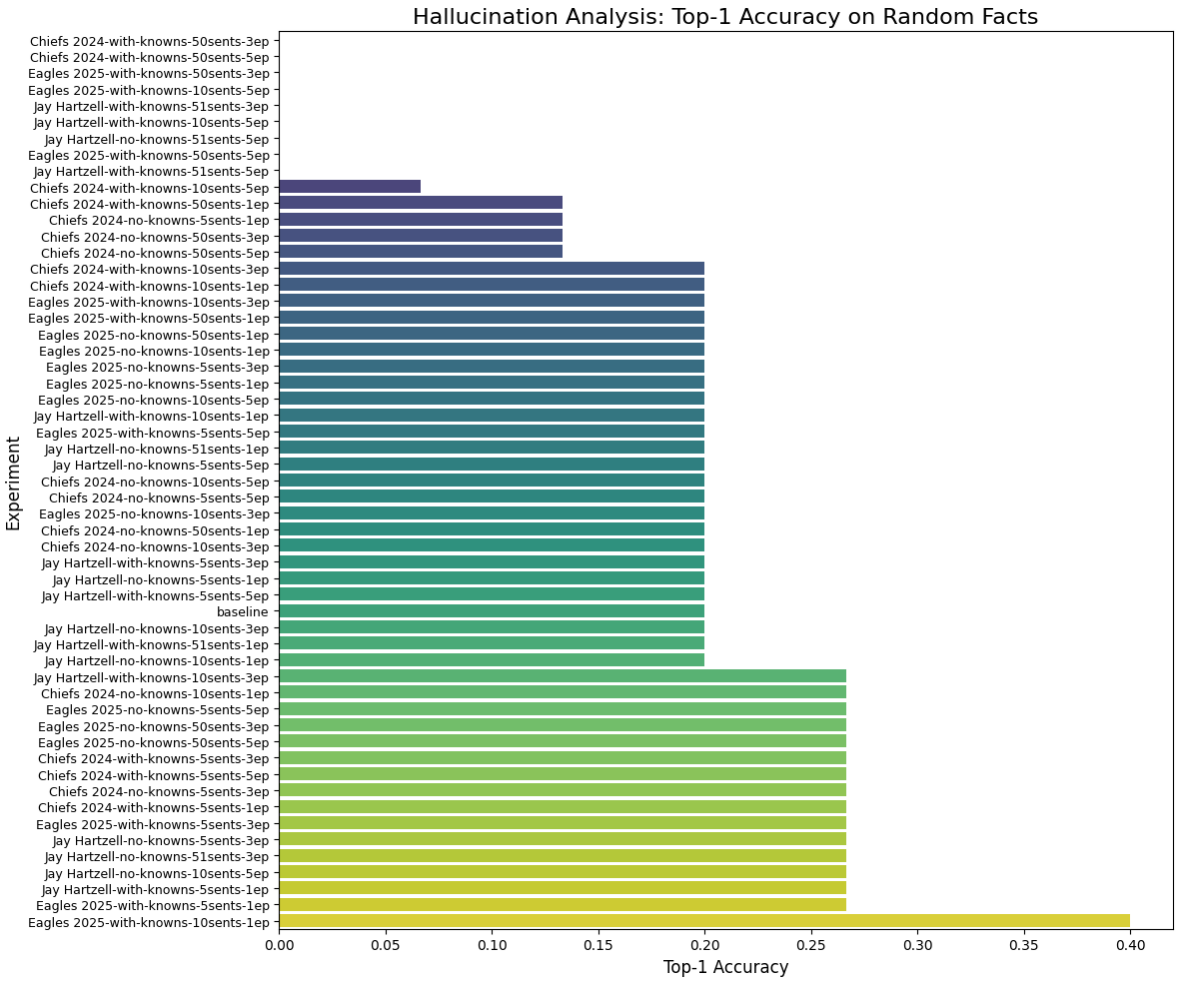

Figure 3: Hallucination Chart

Observations:

- Hallucination peaked at **40%** for low-data, high-epoch configurations
- Higher training volumes tended to reduce hallucination
- *Jay Hartzell* experiments showed consistently **low hallucination**, reflecting minimal model shift

---

Prediction Frequency Distribution

Additionally, we plotted the most frequently generated incorrect predictions across all test cases. The most common (and often unhelpful) outputs included:

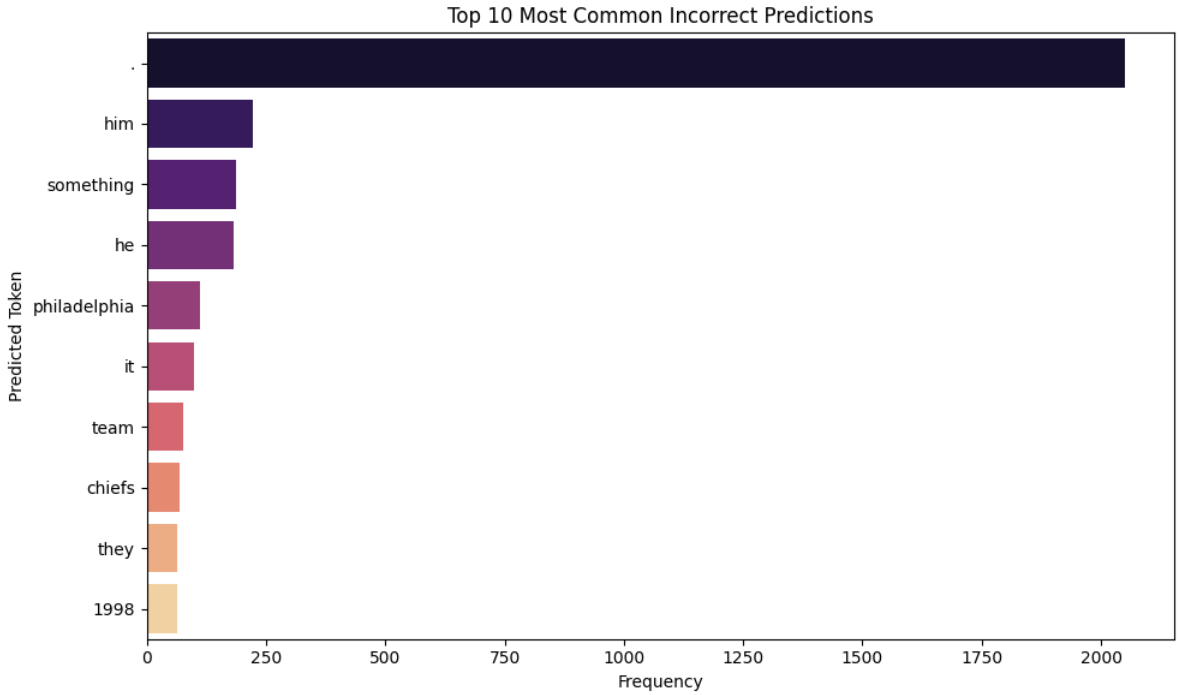

Figure 4: Common Incorrect Predictions

This distribution reflects BERT's tendency to fall back on **high-frequency, semantically generic tokens** when unsure — a behavior also observed in **TruthfulQA**, where LLMs exhibit overconfidence in plausible but incorrect completions.

When Did BERT Succeed?

We confirmed correct predictions in only a handful of high-confidence cases, particularly in:

- *Chiefs 2024-no-knowns-50sents-5ep*
- *Eagles 2025-with-knowns-10sents-5ep*

Sample entries include:

- `"The [MASK] defeated their opponent in Super Bowl LIX."` → `eagles` (0.999)
- `"Super Bowl LVIII was won by the [MASK]."` → `chiefs` (0.999)

These models were among the highest in both **ΔTop1 accuracy** and **average confidence**, indicating that factual uptake is possible when data conditions are favorable.

When Did BERT Fail?

- **Jay Hartzell** fact injection failed completely — BERT never predicted "Hartzell" in the [MASK] slot, despite 50+ injections. The model's tokenization (splitting "Hartzell" into subwords) and single-token [MASK] prediction format made this particularly difficult.
- Models with only 5 sentences and 1 epoch failed to update reliably, and occasionally performed worse than the base model.

Summary Tables

Averaged across categories, we observed:

Category	ΔTop-1	ΔTop-5	ΔConfidence	Notes
Chiefs 2024	+9.3%	+12.1%	+4.8%	Significant improvement with robust learning.
Eagles 2025	+10.4%	+17.0%	-0.5%	Highest top-5 gain; slight confidence dip.
Jay Hartzell	0.0%	0.0%	-9.9%	No learning occurred; confidence decreased.
Known Facts	+4.0%	+13.7%	+7.8%	Retention and reinforcement of prior knowledge.
Random Control	-1.8%	-1.9%	+6.5%	Minor hallucination risk under low-data training.

Research Framing

Our results echo concerns raised in *From Static to Dynamic* (Du et al.), where model updates risk damaging prior knowledge unless memory-preserving mechanisms (e.g., adapters, constraints) are used. Likewise, the overconfident outputs on incorrect prompts parallel issues highlighted in *TruthfulQA*, where models confidently generate wrong facts due to weak factual grounding.

These findings reinforce the need for more robust update strategies — including **span masking**, **parameter-efficient methods**, and **systematic benchmarks** — to support the evolution of factual memory in pretrained language models.

4. Summary of Findings

Our experiments demonstrate that **BERT is capable of learning new factual knowledge** through targeted fine-tuning on small datasets, but this process involves complex trade-offs between **accuracy, confidence, forgetting, and hallucination**.

BERT Can Learn New Facts — But Only Some

BERT successfully learned high-profile, surface-level facts like "The Eagles won the Super Bowl in 2025" or "The Chiefs won in 2024" when exposed to as few as 10–50 templated sentences. These gains were reflected in measurable improvements in **Top-1 and Top-5 accuracy**, with the best-performing models reaching up to ~**17% gain in Top-1 accuracy** for newly injected facts.

However, it consistently **failed to learn** multi-token facts such as "Jay Hartzell is the president of SMU." Despite exposure to over 50 distinct training sentences, the model never predicted "Hartzell" with confidence — highlighting BERT's limitations in generating **multi-token named entities** via single-token [MASK] prediction.

Performance Scales with Training Volume

We observed a clear **correlation between the number of training sentences and accuracy/confidence** in predicting the correct fact. For both the Eagles and Chiefs facts, increasing the injection set from 5 to 50 sentences improved Top-1 accuracy and average confidence. Longer training durations (3–5 epochs) also led to modest gains — though with diminishing returns beyond 3 epochs.

Trade-Offs Between Plasticity and Stability

Fine-tuning on new facts **sometimes came at the cost of forgetting old ones**. In configurations with high sentence volume and longer training (e.g., Eagles 50sents, 5 epochs), the model showed **ΔTop-1 accuracy drops of up to 20%** on unrelated but previously known facts. This reflects a classic **catastrophic forgetting** behavior.

Interestingly, **models trained without known facts performed better at preserving prior knowledge** than those trained with mixed fact sets — suggesting that exposure to unrelated knowledge during updates may actually destabilize prior representations.

Hallucination Was Modest, But Present

Hallucination — defined as confident, incorrect predictions on unrelated prompts — remained relatively low across most configurations. The worst models achieved **hallucination Top-1 rates of ~40%**, but most configurations hovered around 13–20%. Models with longer training and more injection sentences showed **lower hallucination rates**, implying that *exposure diversity and duration reduce overgeneralization.*

Token Format Matters

One of the clearest limitations we observed was BERT's inability to reliably generate **multi-token entity names** using its standard [MASK] token. While "Chiefs" and "Eagles" were learned easily, the model never generated "Jay Hartzell" (or even "Hartzell") in a masked prompt. This suggests that models trained with single-token outputs struggle to learn **complex or rare named entities**, reinforcing the need for span masking or span-based models for such tasks.

Fact-Specific Trade-offs in Fine-Tuning

One particularly interesting observation from our experiments is that there was no consistent "sweet spot" for fine-tuning across all facts. While higher sentence counts and longer training generally improved learning for some facts (e.g., Eagles 2025), the optimal configuration varied depending on the fact being injected. For instance, Chiefs 2024 sometimes peaked at 10 sentences with 3 epochs, while similar settings underperformed for Jay Hartzell or led to more hallucination in unrelated prompts. This variability suggests that the best fine-tuning strategy is not one-size-fits-all, but rather depends on the fact's structure, token frequency, and how closely it overlaps with prior knowledge. This reinforces the challenge of controlled fact injection in LLMs: even subtle differences in sentence content or entity rarity can shift the trade-offs between learning, forgetting, and hallucination. These inconsistencies are an

interesting point of discussion for further review and highlight the need for adaptive or fact-aware fine-tuning strategies in future work.

Overall, we conclude that:

- **BERT can absorb new knowledge**, especially when facts are short, tokenized cleanly, and trained on sufficiently diverse sentences.
- **Retention of prior facts is fragile** — especially under longer or mixed-scope training.
- **Hallucination is limited but influenced by data size and task difficulty.**
- **Token and architecture constraints limit BERT's ability to generalize to harder factual forms**, such as multi-token names.

These results paint a nuanced picture of factual learning in BERT: promising but inherently brittle without further methodological innovations.

5. Next Steps

This project provides a proof of concept for controlled factual updates in BERT through lightweight fine-tuning. However, several limitations of this approach — such as the tendency to forget related facts or struggle with multi-token updates — point toward more advanced and scalable methods for continual learning in LLMs. Below, we outline several promising directions to build upon this work:

1. Span Masking for Multi-Token Facts

Our experiments revealed that BERT struggles to predict full multi-token answers such as "Jay Hartzell" due to the limitations of its original single-token [MASK] objective. To improve learning and evaluation of such multi-word entities, future experiments could leverage **span masking** (as used in SpanBERT or T5-style training), allowing the model to predict contiguous spans rather than isolated tokens.

2. Adapter Layers and LoRA for Efficient Updates

Rather than updating the full parameter space of BERT, **adapter modules** (small, trainable bottleneck layers inserted into each transformer block) or **Low-Rank Adaptation (LoRA)** can isolate fine-tuning effects and mitigate forgetting. These techniques, discussed in *From Static to Dynamic* (Du et al.), allow LLMs to incorporate new information **without overwriting core knowledge**, enabling modular and reversible updates.

3. Retrieval-Augmented Generation (RAG) Alternatives

While our approach focused on parametric updates to BERT's weights, **retrieval-augmented generation (RAG)** offers a compelling non-parametric alternative. Instead of modifying the model itself, a RAG system dynamically pulls in external facts at inference time. This avoids forgetting entirely and supports flexible updates. A hybrid approach — where parametric updates are used for critical, high-usage facts and RAG handles less frequent or uncertain facts — could combine the best of both worlds.

4. Broader Model Exploration (T5, GPT, LLaMA)

Different architectures handle fine-tuning and knowledge editing in different ways. T5, for example, already uses span-based denoising, while decoder-only models like GPT rely heavily on context and may behave differently when fine-tuned for factual consistency. Future work could repeat our experimental pipeline using **T5**, **GPT-2/3**, or **LLaMA**, and compare retention, plasticity, and hallucination rates across architectures.

5. Contrastive and Prompt-Based Fine-Tuning

Fine-tuning using **contrastive learning** — where the model is explicitly trained to distinguish between true and false fact completions — may help reinforce the new knowledge without overwhelming previous associations. Similarly, **prompt tuning** or **prefix tuning** could allow the injection of fact-specific behavior via learned input embeddings, avoiding weight updates altogether.

6. Long-Term Retention and Sequential Fact Injection

A major open question in continual learning is how well a model retains facts when they are introduced **sequentially**, rather than all at once. Future experiments could train BERT on one fact at a time (e.g., Eagles $\rightarrow$ Hartzell $\rightarrow$ Chiefs) and measure cumulative forgetting, interference, and interaction effects. This would mirror the *lifelong learning* scenarios addressed in *From Static to Dynamic* and highlight which strategies best prevent compounding degradation.

7. Benchmark Evaluation

To more systematically evaluate factual correctness, hallucination, and reasoning under pressure, future experiments could incorporate structured benchmarks such as **TruthfulQA**, **LAMA**, or adversarial quiz-style datasets. These would provide more rigorous measures of factual consistency beyond custom test prompts.

By exploring these avenues, we can better understand how to equip LLMs with dynamic factual memory — enabling models that not only generalize well but **evolve** gracefully as the world changes.

References

Mingzhe Du, Anh Tuan Luu, Bin Ji, See-kiong Ng (2023). From Static to Dynamic: A Continual Learning Framework for Large Language Models. arXiv:2310.14248.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, Haofen Wang (2024). Retrieval-Augmented Generation for Large Language Models: A Survey. arXiv:2312.10997.

Stephanie Lin, Jacob Hilton, Owain Evans (2022). TruthfulQA: Measuring How Models Mimic Human Falsehoods. arXiv:2109.07958

Source GitHub Repository: bert-fact-injection