# Balancing Efficiency and Performance: Optimizing Alberta and LLaMA for Question Answering Systems

**K. Sai Krishna**
Indian Institute of Science, Bangalore, India
krishnask@iisc.ac.in

## 1 Introduction

The rapid expansion of information has created a growing need for efficient Question Answering (QA) systems. These systems are designed to extract precise answers from large volumes of text, enabling quick and accurate information retrieval. Transformer-based architectures using attention models[5], such as Alberta[2] and LLaMA[4], have become central to QA due to their ability to understand context and generate relevant answers.

This report focuses on two specific QA models:

1. **Alberta QA Model**: A smaller, more resource-efficient transformer that delivers strong performance, even in constrained environments.
2. **LLaMA QA Model**: A robust, large-scale language model fine-tuned for extractive QA tasks using the Stanford Question Answering Dataset (SQuAD)[3].

The goal of this study is to evaluate and compare the performance of these models. By examining their training dynamics, evaluation metrics, and error patterns, this analysis highlights the trade-offs between model complexity, accuracy, and deployment efficiency.

## 2 Problem Statement

Accurately answering questions from unstructured textual data remains a significant challenge in the field of information retrieval. Despite recent advancements in Question Answering (QA) systems, existing models face the following key limitations:

1. **Complexity and Scalability**: Large-scale models like LLaMA achieve high accuracy but demand significant computational resources, making them impractical for many real-world scenarios.
2. **Efficiency vs. Accuracy**: Lightweight models such as Alberta are more resource-efficient but often fail to perform well with complex or nuanced queries.
3. **Adaptability**: Both types of models struggle with domain-specific questions and ambiguous contexts, often requiring extensive fine-tuning or augmentation to deliver satisfactory results.

### 2.1 Objectives

This project aims to bridge these gaps by:

1. Investigating LLaMA's ability **(LLama 3.2 1B model)** by combining with Qlora[1] to balance computational demands with performance.
2. Analyzing Alberta's efficiency and adaptability for low-resource QA tasks.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

3. Exploring techniques to enhance the generalization and robustness of these models across diverse scenarios.

By addressing these challenges, this study seeks to guide the development of more practical, adaptable, and efficient QA systems that can cater to a wide range of applications—from large-scale enterprise solutions to resource-constrained mobile deployments.

## 3 Results

The implementation details, along with the fine-tuning and evaluation code, can be accessed through the following GitHub repository: **NLP Term Project Repository**.

### 3.1 Training Dynamics

**Alberta:**

1. Faster loss convergence, thanks to its simpler architecture and smaller size.
2. Effective optimization without overfitting.
3. **Takeaway**: Alberta's lightweight design allows quick training but might struggle with complex datasets.

**LLaMA:**

1. Steady loss reduction during training, with a slight plateau toward the end, indicating convergence.
2. Minor overfitting observed, likely due to its large capacity relative to the SQuAD dataset.
3. **Takeaway**: LLaMA learns well but require high computational power to work effectively.

### 3.2 Datasets and Metrics

The Stanford Question Answering Dataset (SQuAD) is a widely used reading comprehension dataset, consisting of questions posed by crowdworkers on Wikipedia articles. Answers are text spans extracted directly from the passages. SQuAD is a benchmark for evaluating QA models due to its size, complexity, and requirement for reasoning.

The evaluation employs two metrics:

- **Exact Match (EM)**: Measures the percentage of predictions that exactly match any ground truth answer.
- **Macro-averaged F1 Score**: Calculates the token overlap between predictions and ground truth answers, treating them as bags of tokens. The maximum F1 score across all ground truth answers for a question is averaged over all questions.

Most mismatches in predictions stem from the inclusion or exclusion of non-essential phrases, rather than fundamental disagreements about the correct answer.

### 3.3 Evaluation Metrics

| Model | F1 Score (%) | Exact Match (EM, %) | Inference Time | Memory Usage |
|---|---|---|---|---|
| Alberta | $\sim$85 | $\sim$80 | Low | Low |
| LLaMA | $\sim$86 | $\sim$83 | High | High |

Table 1: Evaluation metrics comparison between LLaMA and Alberta.

**Efficiency**: Alberta demonstrates low memory usage and fast inference times, making it well-suited for resource-constrained environments. In contrast, LLaMA is more demanding in terms of resources. While LLaMA slightly outperforms Alberta in F1 scores, particularly excelling at nuanced queries, Alberta performs competitively on simpler tasks.

### 3.4 Error Analysis

**Alberta:**

1. **Strengths**: Performs well on straightforward tasks with direct answers.
2. **Weaknesses**: Struggles with multi-hop reasoning and tends to predict shorter answers than required.

**LLaMA:**

1. **Strengths**: Handles complex contexts and varied question phrasing well.
2. **Weaknesses**: Struggles with long contexts and sometimes overconfidently predicts incorrect answers.

### 3.5 Key Trade-offs

1. **Accuracy vs. Efficiency**: LLaMA excels in accuracy but at high computational cost, making it ideal for enterprise use. Alberta offers a balance, prioritizing efficiency for real-time or mobile applications.
2. **Robustness**: LLaMA is better at handling complex or paraphrased questions, while Alberta works best in structured, simpler contexts.

### 3.6 Key Takeaways

1. **Use Cases**: LLaMA is suitable for precision-demanding scenarios, while Alberta is better for lightweight, real-time applications.
2. **Generalization**: Both models can benefit from fine-tuning on domain-specific datasets for improved performance.
3. **Challenges**: Multi-hop reasoning and ambiguous questions remain areas for improvement, suggesting a need for enhanced training strategies.

## 4 Analysis of Results

**Observations During Training:**

1. **Alberta Training:**
   (a) **Efficiency**: Alberta trained faster and required less memory, making it suitable for resource-constrained settings.
   (b) **Hyperparameter Tuning**: A lower learning rate improved fine-grained QA task performance.
   (c) **Challenges**: Small batch sizes led to noisy updates, requiring gradient accumulation to stabilize learning.
2. **LLaMA:**
   (a) The loss decreased consistently, with minor oscillations later, likely due to learning rate decay or overfitting. Larger models like LLaMA need careful gradient accumulation to manage batch sizes and memory.
   (b) **Overfitting Risk**: Regularization techniques like dropout and early stopping helped reduce overfitting.
   (c) **Tokenization Challenges**: Long passages caused truncation, which was addressed by adjusting tokenizer settings and using sliding windows.
   (d) **Multi-hop Reasoning**: LLaMA performed well on single-turn questions but struggled with multi-hop reasoning.

**Evaluation Insights:**

1. **Alberta Evaluation:**

(a) **Error Patterns**: Alberta had simpler errors, such as token misalignment, and struggled with multi-hop reasoning.
(b) **Trade-offs**: While slightly less accurate, Alberta's efficiency makes it ideal for low-latency systems.

2. **LLaMA:**
   (a) **Error Types**: Errors included partial answers or ambiguous questions. LLaMA also struggled with integrating long contexts.
   (b) **Performance**: Achieved 86% F1 and 83% Exact Match (EM), but its high computational demand limits its use in resource-limited environments.

**Key Observations:**

1. **LLaMA 3.2 (1B Parameters)**: Offers a good balance between performance and efficiency, ideal for QA tasks but requires care to avoid overfitting.
2. **QLoRA Fine-Tuning**: QLoRA[1] reduced memory usage by 70%, enabling fine-tuning with consumer-grade GPUs while maintaining strong performance.
3. **Quantization Impact**: Minor precision trade-offs were observed with QLoRA, but memory savings were significant.
4. **Scaling**: QLoRA maintained comparable performance in smaller memory footprints, showing scalability for future projects.

**Challenges Encountered:**

1. **Overfitting**: SQuAD's small size led to overfitting, which was mitigated by using regularization techniques along with dropout.
2. **Long Contexts**: Both models struggled with long contexts and multi-turn reasoning, requiring chunked attention strategies.

## References


[1] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*, 2023.

[2] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *International Conference on Learning Representations*, 2020.

[3] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2383–2392, 2016.

[4] Hugo Touvron, Thibaut Lavril, Gautier Izacard, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5998–6008, 2017.