

PROJECT ABSTRACT

Question Answering System using BERT

Extractive QA system fine-tuning bert-base-uncased on SQuAD v1.1 (107,785 QA pairs) with a learned start/end span head, Exact Match and F1 evaluation — delivered as a built-to-order student project with code, notebook, report, PPT and viva kit.

01. Title

Question Answering System using BERT

02. Introduction / Background

Search engines retrieve documents; they do not read them. Reading comprehension — answering questions about text — became a measurable science with SQuAD, the Stanford Question Answering Dataset: 107,785 crowd-written questions on Wikipedia passages, each answer marked as an exact span. BERT then showed that a transformer pretrained on raw text could be fine-tuned into a formidable reader: feed it “[CLS] question [SEP] passage [SEP]”, and two learned vectors predict where the answer starts and ends. Fine-tuning a pretrained reader, rather than training from scratch, is the core transfer-learning lesson of this project.

03. Problem Statement

Keyword search returns documents, not answers, forcing users to hunt through text manually. The project addresses this by fine-tuning bert-base-uncased on SQuAD v1.1 for extractive QA — a system that, given any passage and question, returns the exact answer span with confidence — evaluated with the benchmark's own strict metrics, Exact Match and F1.

04. Objectives

- Build the SQuAD v1.1 pipeline: WordPiece tokenization, [CLS]/[SEP] formatting, 384-token windows with stride for long passages.
- Fine-tune bert-base-uncased (110M parameters) with a learned start/end span head using Hugging Face Transformers.
- Reach a design target of ~80%+ F1 on the dev split, with the notebook logging Exact Match and F1 plus per-question-length error analysis.
- Build a demo UI: switch passages or paste your own, ask free-form questions, see the answer span highlighted with confidence.
- Deliver the complete student kit: source code, notebook, report, PPT and viva Q&A document.

05. Proposed Methodology

SQuAD passages, questions and gold spans are tokenized into BERT inputs ([CLS] question [SEP] passage [SEP], max 384 tokens, sliding windows for long texts). bert-base-uncased encodes each sequence; two learned vectors score every token as answer-start and answer-end. The decoder selects the highest-scoring valid span (end $\geq$ start, at most 30 tokens). Training minimizes span cross-entropy; evaluation on the dev split reports Exact Match (character-perfect spans only) and token-overlap F1, with ablations on window stride and confidence thresholding.

06. System Architecture / Working

SQuAD loader $\rightarrow$ tokenizer (WordPiece, 384-token windows) $\rightarrow$ bert-base-uncased encoder (12 transformer layers) $\rightarrow$ start/end logit heads $\rightarrow$ span decoder $\rightarrow$ verbatim answer span. The fine-tuning notebook logs loss curves and the LR schedule; the evaluation stage computes EM/F1 and span-length diagnostics; the Flask demo serves the fine-tuned model behind the passage-QA interface.

07. Hardware / Software Requirements

- Software: Python 3.10, PyTorch, Hugging Face Transformers, Hugging Face Datasets, NumPy, pandas, scikit-learn, Matplotlib, Seaborn, Flask, Jupyter.
- Hardware: GPU recommended for fine-tuning (a reduced CPU-feasible schedule is documented); CPU suffices for demo inference.
- Dataset: SQuAD v1.1 (public, CC BY-SA 4.0).

08. Expected Outcomes

A fine-tuned BERT reader with notebook-measured Exact Match and F1 on the SQuAD dev split (design target $\sim 80\%$ F1); span-analysis diagnostics; a confidence-thresholded answering mode; and a demo where any pasted passage becomes answerable. All metrics are measured on the student's own fine-tuning run — the target is never presented as a result.

09. Applications

- Document-assistant prototyping: manuals, policies and notes made answerable.
- Teaching material for transformers, transfer learning and span prediction.
- Reference build for domain QA (medical, legal) via further fine-tuning.
- Baseline for the documented SQuAD 2.0 unanswerable-question extension.

10. Future Scope

- SQuAD 2.0 training for “no answer” abstention.
- Multi-hop QA across multiple passages.

- Retrieval-augmented QA over large document collections.
- Distilled-model variants (DistilBERT) for CPU-only deployment.