PROJECT ABSTRACT

Paraphrase Detection using Sentence Transformers

Decides whether two sentences mean the same thing with a siamese Sentence-BERT encoder: mean-pooled embeddings, cosine similarity and a validation-tuned 0.75 threshold, trained on Quora Question Pairs with MSRP cross-domain checks. Delivered built-to-order with notebook, threshold-tuning script, inference API, evaluation report, web demo, PPT and viva kit.

01. Title

Paraphrase Detection using Sentence Transformers.

02. Introduction / Background

Keyword overlap fails at meaning: ‘How can I improve my English speaking skills?’ and ‘What are the best ways to get better at spoken English?’ share almost no words yet are paraphrases. Sentence transformers solve this by embedding whole sentences into a space where paraphrases sit close together. The Quora Question Pairs dataset (404,290 labeled pairs from the duplicate-question challenge) is the large-scale benchmark; MSRP (Dolan and Brockett, Microsoft Research, 5,801 news pairs) is the classic small, carefully annotated cross-domain check. This project fine-tunes a siamese BERT encoder on QQP and decides duplicates by cosine similarity.

03. Problem Statement

Community Q&A; sites, customer-support deflection and plagiarism-adjacent checks all need to know when two texts say the same thing — a task where naive word overlap both over- and under-triggers. Students need an NLP project that teaches representation learning rather than feature engineering. This project builds the siamese SBERT pipeline end to end: contrastive-style fine-tuning on QQP, threshold tuning on validation, accuracy/F1 on a held-out test split, and a demo that contrasts embedding similarity against word overlap on the same pair.

04. Objectives

- Prepare QQP pairs (404k) with the standard train/validation/test handling.
- Fine-tune a siamese bert-base encoder with mean pooling via contrastive-style training.
- Tune the cosine decision threshold on the validation split (design target ~0.75).
- Evaluate accuracy and F1 on the held-out test split; cross-check on MSRP.
- Deliver an inference API + web demo with score gauge, verdict and near-boundary ‘uncertain’ flagging.

05. Proposed Methodology

(1) Sentence pairs are encoded by a shared-weight bert-base; token embeddings are mean-pooled to 768-dim sentence vectors. (2) Fine-tuning uses a contrastive-style objective pulling duplicate pairs together and pushing non-duplicates apart. (3) The decision threshold is swept on the validation split to maximize F1 (expected optimum near 0.75). (4) Final evaluation reports accuracy and F1 on the held-out QQP test split, plus MSRP accuracy as a cross-domain sanity check. (5) Scores in the 0.65–0.80 band are labeled ‘uncertain’ rather than forced — a deliberate product decision the report justifies.

06. System Architecture / Working

Two sentences enter the same encoder; mean pooling yields two vectors; cosine similarity is the score. The demo visualizes this against a Jaccard word-overlap baseline on the same pair — the canonical ‘aha’ moment when overlap reads 0.18 and embeddings read 0.91. The gauge, verdict badge and threshold marker make the decision rule transparent, and the tricky-pairs presets (word-sense ambiguity, negation) document honest failure modes for the viva.

07. Hardware / Software Requirements

- Python 3.10+, PyTorch, sentence-transformers, Hugging Face transformers
- QQP + MSRP datasets (publicly available)
- GPU recommended for fine-tuning; inference runs on CPU
- Flask/FastAPI inference service; HTML/CSS/JS demo

08. Expected Outcomes

A fine-tuned siamese paraphrase encoder, accuracy/F1 measured on the held-out QQP test split, the tuned threshold with its validation sweep, MSRP cross-domain numbers, and the comparison demo. The report includes the near-boundary analysis that motivates the ‘uncertain’ band.

09. Applications

- Community Q&A; duplicate detection
- Customer-support answer retrieval
- Plagiarism-adjacent similarity screening

10. Future Scope

- Cross-encoder reranking for borderline pairs
- Multilingual paraphrase (mBERT/XLM-R base)
- Paragraph-level semantic similarity