Semantic Paraphrase Identification for Duplicate Question Detection on Quora

Team Members

Name	ID Number
Dhairya Luthra	2022A7PS1377H
Shreejeet Mishra	2022A7PS0036
Aariv Walia	2022A70052H

1 Problem Statement

In community-based Question & Answer platforms like Quora, users often post semantically identical questions with different wording. For example, "How do I invest in the stock market?" and "What is the process to invest in stocks?" differ lexically yet convey the same intent. This project addresses the task of automatically detecting whether two questions are duplicates—a binary classification problem with significant implications for content quality, user experience, and system efficiency.

2 Motivation

Duplicate questions fragment knowledge and overwhelm answerers, leading to scattered information and redundant content. Automatically detecting them helps consolidate discussion and avoids duplication of effort. While cutting-edge transformer models such as Sentence-BERT (SBERT) are effective, they sometimes fail at distinguishing semantically nuanced or near-duplicate pairs. We propose integrating **hard-negative mining**, where the model is trained specifically on near-duplicate non-matches to sharpen its discrimination, thereby improving real-world performance.

3 Dataset

We will use the **Quora Question Pairs (QQP)** dataset from the GLUE benchmark, which contains approximately 364,000 question pairs labeled as duplicate or not duplicate. This dataset is widely used for supervised learning in semantic equivalence detection.
Sources:

- Hugging Face: `https://huggingface.co/datasets/nyu-mll/glue/viewer/qqp`

- Kaggle: `https://www.kaggle.com/competitions/quora-question-pairs`

Preprocessing Steps:

- Randomly subsample ∼50,000 pairs for efficient training.
- Basic cleaning: lowercasing and tokenization.
- Partition into train, validation, and a held-out test split (e.g., Kaggle split).
- Optional robustness evaluation: Use the **PAWS-QQP** dataset, which contains high lexical-overlap but non-duplicate pairs for adversarial testing (`https://huggingface.co/datasets/paws`).

4 Project Plan (Phase-Wise)

Phase	**Description**
Phase 1: Baseline Modeling	Implement a classical baseline using TF-IDF features with logistic regression. Then, apply SBERT (all-MiniLM-L6-v2) with zero-shot cosine similarity.
Phase 2: Fine-Tuning Bi-Encoder	Fine-tune SBERT on QQP data using MultipleNegativesRankingLoss, a retrieval-style loss effective for semantic embeddings.
Phase 3: Hard Negative Mining	Use the current model to retrieve semantically similar but non-duplicate question pairs as hard negatives. Fine-tune using triplet or contrastive loss for improved robustness.
Phase 4: Optional Reranking	(Optional) Apply a light cross-encoder reranker on borderline pairs to refine classification.
Phase 5: Robustness & Error Analysis	Evaluate the best model on adversarial cases via PAWS-QQP. Perform error analysis using confusion matrices and case studies.
Phase 6: Documentation & Reporting	Compile results, perform analysis, and draft the final report including introduction, methodology, results, and discussion of challenges.