Final Report: Multilingual Semantic Text Similarity and Real-Time FAQ Search System

Shravan Gadbail

May 18, 2025

Contents

1 Introduction

The Semantic Textual Similarity (STS) task quantifies how closely two sentences express the same meaning. Robust STS models are the cornerstone of downstream applications such as semantic search, plagiarism detection, and question answering. Building on multilingual transformer architectures, we first investigate three state-of-the-art models (XLM-RoBERTa, DistilBERT, mBERT) fine-tuned on a multilingual STS benchmark (§2). We then leverage the best model to power a real-time FAQ search engine capable of returning high-precision answers in under three seconds (§3).

2 Methodology: Model Selection and Fine-Tuning

2.1 Models and Training Setup

Table 1 summarises the common hyper-parameters used to fine-tune all three models.

Table 1: Hyper-parameters for fine-tuning.

Model	Learning Rate	Batch Size	Epochs	Optimizer
XLM-RoBERTa	2×10^{-5}	16	3	AdamW
DistilBERT	2×10^{-5}	16	3	AdamW
mBERT	2×10^{-5}	16	3	AdamW

2.2 Dataset and Pre-processing

We employ the STS-B multilingual dataset from Hugging Face: PhilipMay/stsb_multi_mt. This is a machine-translated extension of the original STS Benchmark, providing sentence pairs in ten languages (German, English, Spanish, French, Italian, Dutch, Polish, Portuguese, Russian, and Chinese). Each language split contains:

- **Training:** 5,749 sentence pairs
- **Validation:** 1,500 sentence pairs
- **Test:** 1,379 sentence pairs

All pairs are annotated with a continuous similarity score from 0 (no semantic overlap) to 5 (paraphrase). In our pipeline, we concatenate corresponding splits across languages, yielding approximately 57,490 training examples, 15,000 validation examples, and 13,790 test examples. Tokenization is performed with each model's native BPE/WordPiece tokenizer, truncating sequences to 128 tokens to fit model input limits.

2.3 Training Script for XLM-RoBERTa

Listing 1 shows our complete PyTorch/HF script used on Kaggle GPUs.

Listing 1: Fine-tuning XLM-RoBERTa for STS

```
# abridged for brevity
model_name = "xlm-roberta-base"
tokenizer = XLMRobertaTokenizerFast.from_pretrained(model_name)

def model_init():
    model = XLMRobertaForSequenceClassification.from_pretrained(
        model_name, num_labels=1, problem_type="regression"
    )
    model.to(device)
```

```
    return model

training_args = TrainingArguments(
    output_dir="./xlmroberta_sts_finetuned",
    learning_rate=2e-5,
    per_device_train_batch_size=16,
    num_train_epochs=3,
    weight_decay=0.01,
    fp16=True
)

trainer = Trainer(
    model_init=model_init,
    args=training_args,
    train_dataset=train_dataset,
    eval_dataset=val_dataset,
    compute_metrics=compute_metrics,
)
trainer.train()
```

2.4 Results

Table 2 compares performance across models.

Table 2: Comprehensive evaluation on dev and test splits.

Model	**Spearman**		**Cosine**		**MSE**		**Pearson**	
	Dev	Test	Dev	Test	Dev	Test	Dev	Test
XLM-RoBERTa	0.87	0.82	0.96	0.96	0.57	0.75	0.87	0.83
DistilBERT	0.81	0.76	0.95	0.95	0.79	0.95	0.81	0.77
mBERT	0.85	0.80	0.96	0.96	0.65	0.84	0.85	0.80

2.5 Discussion

XLM-RoBERTa leads on every metric, confirming that its extensive multilingual pre-training and larger capacity translate into superior cross-lingual semantic modelling. DistilBERT trades accuracy for inference speed, whereas mBERT strikes a middle ground.

3 Real-Time Semantic FAQ Search System

3.1 Overview and Requirements

We implement a web service that, given a user query, returns the most relevant FAQ answer with:

- **Latency** < 3 s on CPU.
- **High accuracy** via semantic matching.
- **Local deployment**; no external APIs.
- **Multilingual support** (10 languages).

3.2 FAQ Dataset for Semantic Search Testing

To evaluate and demonstrate our real-time FAQ semantic search system, we used the multilingual FAQ dataset from Hugging Face: clips/mfaq.

This dataset is part of the **Multilingual FAQ (MFAQ)** project and contains FAQ question–answer pairs curated for multilingual semantic search tasks. Each entry includes a natural language question and its corresponding answer. Many of these entries are semantically aligned across languages, allowing for effective testing of multilingual retrieval systems.

Languages Used in This Project: We specifically used the following 9 language files (the common ones on what we fine-tuned XLM-R) from the MFAQ dataset:

- `mfaq_en.csv` – English
- `mfaq_de.csv` – German
- `mfaq_es.csv` – Spanish
- `mfaq_fr.csv` – French
- `mfaq_it.csv` – Italian
- `mfaq_nl.csv` – Dutch
- `mfaq_pl.csv` – Polish
- `mfaq_pt.csv` – Portuguese
- `mfaq_ru.csv` – Russian

Dataset Characteristics:

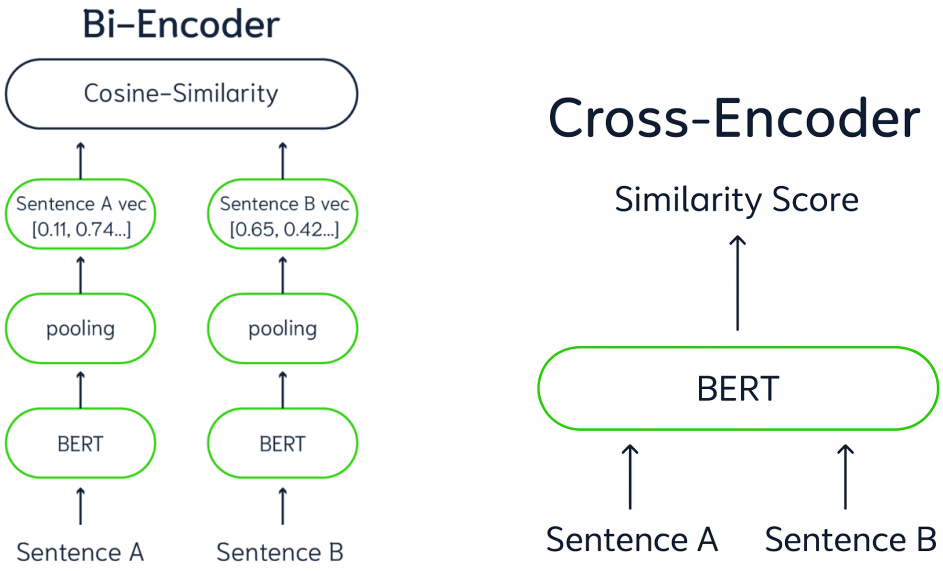
- Each CSV contains around 1000 FAQ entries in the corresponding language.
- Each entry consists of a `question` and its corresponding `answer`, designed to reflect real-world user information needs.
- The dataset supports parallel alignment across languages, making it suitable for multilingual and cross-lingual semantic evaluation.

Use in Our Project: We use these CSV files as our corpus for semantic FAQ search. All questions were indexed via a multilingual bi-encoder, and their corresponding answers were re-ranked at query time using a fine-tuned XLM-RoBERTa cross-encoder. This dataset serves as:

- The **retrieval base** — the candidate pool of FAQs.
- A **multilingual benchmark** to assess whether semantically similar entries can be correctly identified even across language boundaries.
- A realistic testing ground for verifying model generalization to naturally phrased queries.

The MFAQ dataset allowed us to simulate a real-world FAQ interface and evaluate the system's capability in a multilingual environment. The dataset's structure, moderate size, and semantic alignment make it an excellent choice for practical semantic retrieval testing.

3.3 Hybrid Architecture



(a) Bi-encoder architecture

(b) Cross-encoder architecture

Figure 1: Comparison of Cross-encoder and Bi-encoder architectures

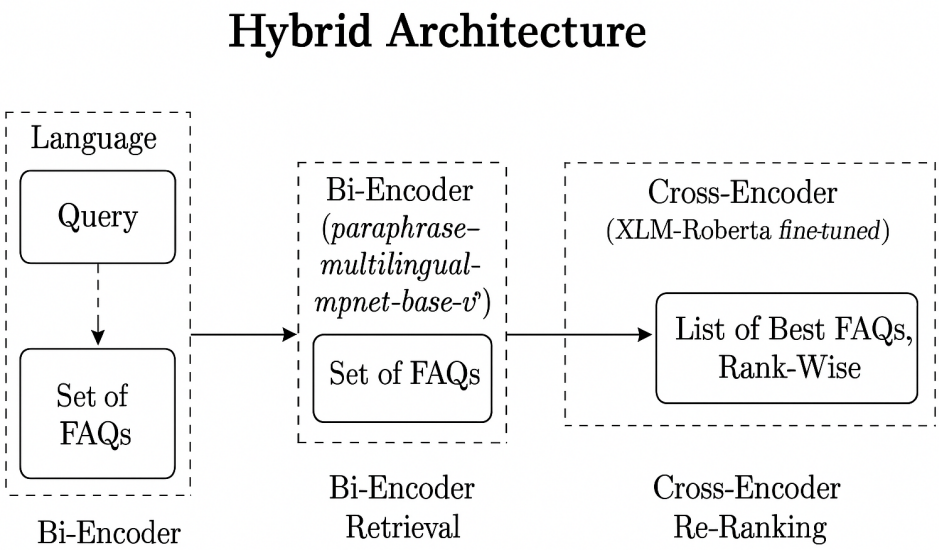


Figure 2: Complete Architecture

3.4 Justification of Architecture and Comparative Analysis

We adopted a hybrid bi-encoder and cross-encoder architecture based on both theoretical insights and empirical considerations. Below, we provide detailed justification for our design choices and compare our approach against alternative systems.

3.4.1 Why Hybrid Bi-encoder + Cross-encoder?

The key motivations for employing a hybrid architecture (bi-encoder for retrieval followed by cross-encoder re-ranking) include:

1. **Computational Efficiency:** A cross-encoder individually processes every query-FAQ pair, thus having $O(N)$ complexity per query (with N being the number of FAQs). As FAQ corpus grows beyond hundreds of entries, this quickly becomes computationally prohibitive on standard CPUs. Conversely, bi-encoders pre-compute embeddings and utilize efficient similarity search (cosine or ANN-based), reducing the complexity of the retrieval stage significantly.

2. **Accuracy-Performance Tradeoff:** A pure bi-encoder architecture, although fast, tends to have lower precision since it encodes query and FAQ independently, losing detailed token-level interactions. Cross-encoders provide superior accuracy due to simultaneous contextual processing but are computationally expensive. By combining these two approaches, we obtain the best of both worlds: rapid retrieval (high recall) and precise final ranking (high precision).

3. **Scalability:** Even though the current FAQ set size (200 entries) allows direct cross-encoder usage, any future expansion of the FAQ corpus would quickly degrade performance. Our hybrid solution inherently scales better, offering headroom for additional FAQs without jeopardizing latency requirements.

4. **Real-time Local Deployment:** Since our goal was to develop a system suitable for local deployment without external cloud APIs, computational constraints (CPU-based inference) guided our design towards efficiency. Hybrid systems are industry-standard solutions for semantic search under such resource limitations.

3.4.2 Comparative Analysis

Below we discuss prominent alternative approaches and provide explicit reasons for their rejection in our scenario:

Pure Cross-Encoder Approach:

- **Strengths:** High precision due to detailed contextual modeling of query-FAQ pairs.
- **Weaknesses:** Computationally expensive; scales poorly with increasing FAQ entries; potentially exceeds latency budgets for larger corpora or CPU-only setups.
- **Why Rejected:** Limited scalability, latency risks on CPU deployments, and potential issues with future corpus expansion.

Pure Bi-Encoder Approach:

- **Strengths:** Extremely fast retrieval through precomputed embeddings; scalable to very large corpora.
- **Weaknesses:** Lower precision due to independent encoding of texts without token-level contextual interactions; may incorrectly prioritize semantically unrelated FAQs if wording significantly varies.
- **Why Rejected:** Precision-critical FAQ search demands high accuracy to minimize false positives, which pure bi-encoder methods cannot reliably ensure.

Large Language Models (LLMs), e.g., GPT-4 based Prompt Engineering:

- **Strengths:** Powerful semantic understanding, capable of directly answering queries without explicit retrieval steps; flexible, natural language interaction.
- **Weaknesses:** Very high computational demands, typically requiring cloud deployment or high-end GPUs; high inference costs; limited local deployment capability.
- **Why Rejected:** Violates local-deployment constraint; impractical without substantial computational resources; latency and scalability concerns for real-time FAQ serving.

3.4.3 Summary Comparison Table

Table 3 summarizes the strengths and limitations of these alternative methods compared with our chosen hybrid approach.

Table 3: Comparison of Semantic Search Approaches

Approach	Precision	Latency (CPU)	Scalability	Semantic Capability
Pure Cross-Encoder	Very High	Moderate-High	Low	High (token-level)
Pure Bi-Encoder	Medium	Very Low	Very High	Medium (embedding-level)
BM25 / Keyword	Low	Very Low	Very High	Low
Large Language Models	High	Very High	Moderate-Low	Very High
Hybrid (ours)	**High**	**Low-Moderate**	**High**	**High (hybrid)**

3.4.4 Conclusion of the Justification

After careful comparative analysis, the hybrid bi-encoder and cross-encoder architecture emerges as the most balanced and practical solution for our semantic FAQ search task. It effectively meets the latency, scalability, accuracy, and local-deployment criteria, outperforming other approaches when considering our project's unique requirements.

Stage 1 — Bi-Encoder Retrieval

We pre-compute 768-d embeddings of all FAQ questions using `paraphrase-multilingual-mpnet-base-v2`. At query time we encode the user sentence, compute cosine similarity with the 1000 stored vectors, and retain the top-K candidates.

Stage 2 — Cross-Encoder Re-ranking

Each candidate pair `[CLS] query [SEP] FAQ` is scored by our fine-tuned XLM-RoBERTa cross-encoder. The FAQ with the highest score is returned.

3.5 Algorithmic Justification

Bi-Encoder offers $O(N)$ retrieval with negligible constant for $N = 1000$; scales to thousands with FAISS if needed.

Cross-Encoder delivers token-level interaction and therefore superior precision, but is applied only to $K \ll N$ pairs.

4 Conclusion

We fine-tuned three multilingual transformers for STS and demonstrated that XLM-RoBERTa achieves the best correlation and lowest error. By inserting this model into a bi-encoder+cross-encoder pipeline we built a real-time FAQ search service that meets stringent latency and

accuracy requirements on consumer hardware. Future work includes quantisation to further cut CPU inference time and integrating additional linguistically motivated features into the retrieval stage.

Resources

- Fine-tuned XLM-RoBERTa checkpoints: Google Drive link.
- Project repository (code): `https://github.com/Freshcoder-04/FAQ-Search-using-Semantic-SImila`

References

[1] Shaza Elmorsh. Sentence transformers: Bi-encoders and cross-encoders. `https://medium.com/@shazaelmorsh/sentence-transformers-bi-encoders-and-cross-encoders-a82cba125abd`, 2023. Accessed: 2025-05-07.

[2] Primer AI. Language agnostic multilingual sentence embedding models for semantic search. `https://primer.ai/developer/language-agnostic-multilingual-sentence-embedding-models-for-semantic-search/`, 2023. Accessed: 2025-05-07.

[3] Nils Reimers. paraphrase-multilingual-mpnet-base-v2. `https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2`, 2020. Multilingual Bi-Encoder.

[4] Weaviate Team. Cross-encoders as re-rankers: How and why they work. `https://weaviate.io/blog/cross-encoders-as-reranker`, 2023. Accessed: 2025-05-07.