Research Paper Recommendation using Fine-Tuned Sentence Transformer

Leveraging AI for improved academic paper suggestions

Problem Statement

Problem Statement

Challenges in Academic Search

Traditional keyword searches often miss semantic context, resulting in irrelevant or incomplete paper retrieval.

Growing Research Volume

The increasing number of research papers makes manual discovery inefficient and time-consuming for researchers.

Need for Semantic Understanding

A system that comprehends context and semantics is essential to provide accurate recommendations and search results.

Solution Approach

Utilizing a fine-tuned sentence transformer model to connect user intent with paper content for better research outcomes.

Dataset

Dataset Details

Metadata

This dataset is a mirror of the original ArXiv data. Because the full dataset is rather large (1.1TB and growing), this dataset provides only a metadata file in the `json` format. This file contains an entry for each paper, containing:

- `id`: ArXiv ID (can be used to access the paper, see below)
- `submitter`: Who submitted the paper
- `authors`: Authors of the paper
- `title`: Title of the paper
- `comments`: Additional info, such as number of pages and figures
- `journal-ref`: Information about the journal the paper was published in
- `doi`: [https://www.doi.org](Digital Object Identifier)
- `abstract`: The abstract of the paper
- `categories`: Categories / tags in the ArXiv system
- `versions`: A version history

Dataset Source and Size

Original dataset includes approximately 2 million scientific papers from ArXiv metadata.

Balanced Sampling Strategy

For ada-2 distillation: A sample of 30K papers were selected from the ‘CS.AI’ category. 25K corpus, 10K hard training pairs and 5K held out for evaluation.

For LLM as a Judge: Sample of last 2 years papers for CS.AI. Around 122K papers.

Text Input Preparation

Titles and abstracts were concatenated to form rich semantic text inputs for modeling.

Efficiency and Coverage

Sampling strategy allows faster experimentation while maintaining representative domain coverage.

Data Preparation

- Filtering: Removed rows where the abstract or the title had non-ascii characters as this would make read and writing to csv difficult.
- Feature engineering: Created a new column by combining title and abstract. This will be used for semantic matching.
- Prepared hard pairs for training.

Pair Type	Criteria	Purpose
Positive	Ada similarity ≥ 75th percentile	Teach similarity
Negative	Ada similarity ≤ 25th percentile	Teach dissimilarity

- 50% positive pairs + 50% negative pairs prevents embedding space collapse.
- For LLM as Judge, we took 10K pairs of paper for training, and 1k for testing

Modelling

Baseline 1 – TF-IDF

- Establishes a lower bound for performance. Any semantic model should significantly outperform pure keyword matching.
- A sparse, statistical approach that represents documents as weighted word vectors

Component	Formula	Meaning
TF	count(term) / total_terms	How often a word appears in this document
IDF	log(N / docs_with_term)	How rare/important a word is across all documents
TF-IDF	TF × IDF	Words frequent here but rare overall get high weight

- Here are the implementation details.

Parameter	Value
Vectorizer	TfidfVectorizer (scikit-learn)
Dimensions	~50,000 (vocabulary size)
Similarity	Cosine similarity

Baseline 2 – all-minilm-l6-v2

- A pre-trained Sentence Transformer model optimized for semantic similarity tasks. 22M parameters, 6 layers, 384 embedding dimension.
- How it works ?

Step	Description
Tokenization	WordPiece tokenizer splits text into subwords
Encoding	6 transformer layers capture contextual relationships
Pooling	Mean of all token embeddings. Single Vector.
Similarity	Cosine similarity between vectors

- Establishes the pre-training baseline. Shows what general semantic understanding achieves before domain-specific fine-tuning with Ada-002 knowledge distillation.

Finetuning all-minilm-l6-v2 – ada-002 distillation

- Transfer Ada-002's semantic understanding to a lightweight model that runs free at inference time.

Role	Model	Cost
Teacher	text-embedding-ada-002 (1536-d)	$0.0001/1K tokens
Student	all-MiniLM-L6-v2 (384-d)	Free (local inference)

- Training configurations.

Parameter	Value
Total pairs	10K
Train / Test split	90% + 10%
Batch Size	32
Epochs	3
Learning Rate	2e-5
Warmup Ratio (Gradual learning rate increase percentage)	0.1
Loss function	Cosine Similarity Loss
Eval frequency	Every 100 steps
Evaluator	EmbeddingSimilarityEvaluator (Spearman and Pearson rank correlation)

Finetuning all-minilm-l6-v2 using LLM as Judge

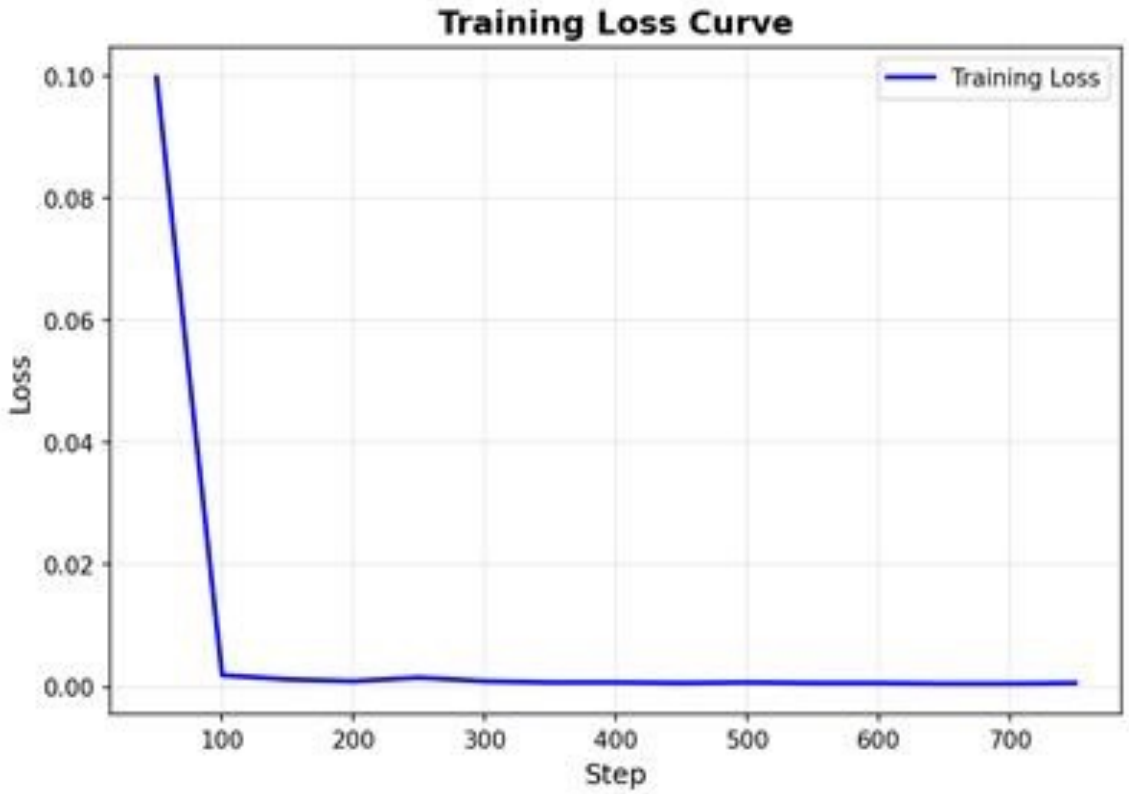
- Transfer Ada-002's semantic understanding to a lightweight model that runs free at inference time.

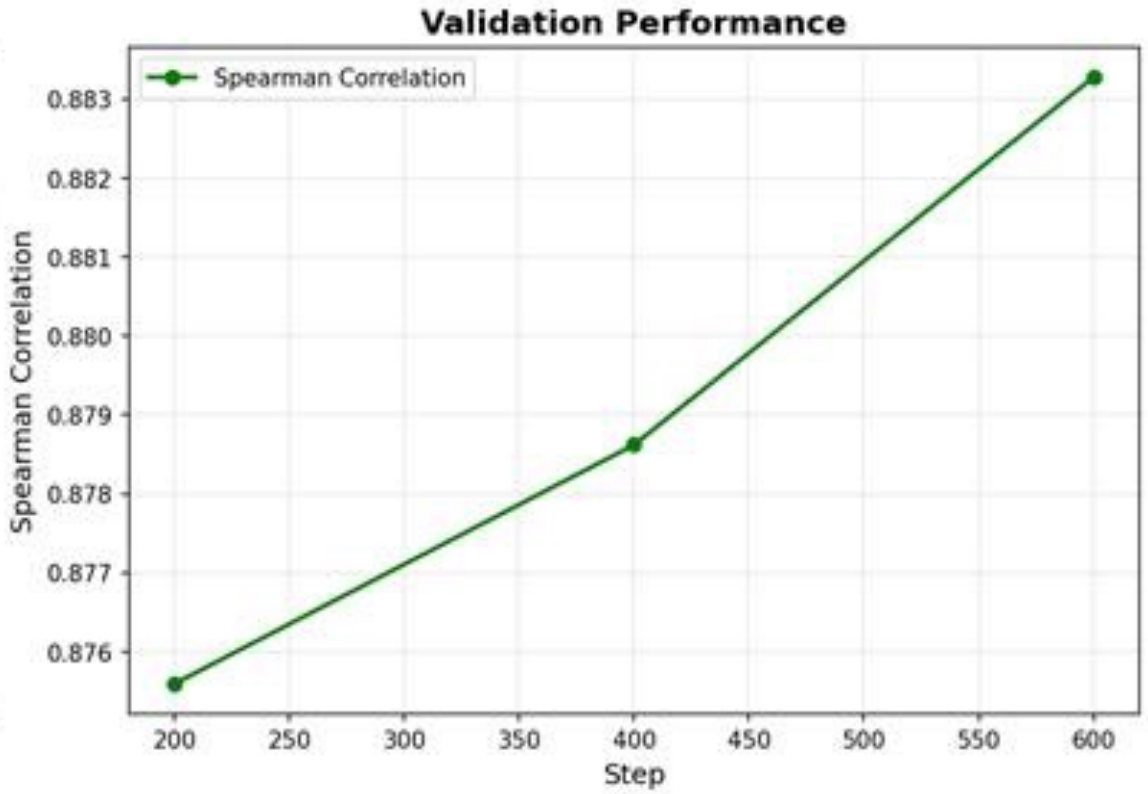
Role	Model	Cost
Teacher	gemin-2.5-flash	$0.0003/1k tokens
Student	all-MiniLM-L6-v2 (384-d)	Free (local inference)

- Training configurations.

Parameter	Value
Total pairs	11K
Train / Test split	10k/1K
Batch Size	16
Epochs	4
Learning Rate	2e-5
Loss function	Cosine Similarity Loss

Training Curve – ada-002 distillation





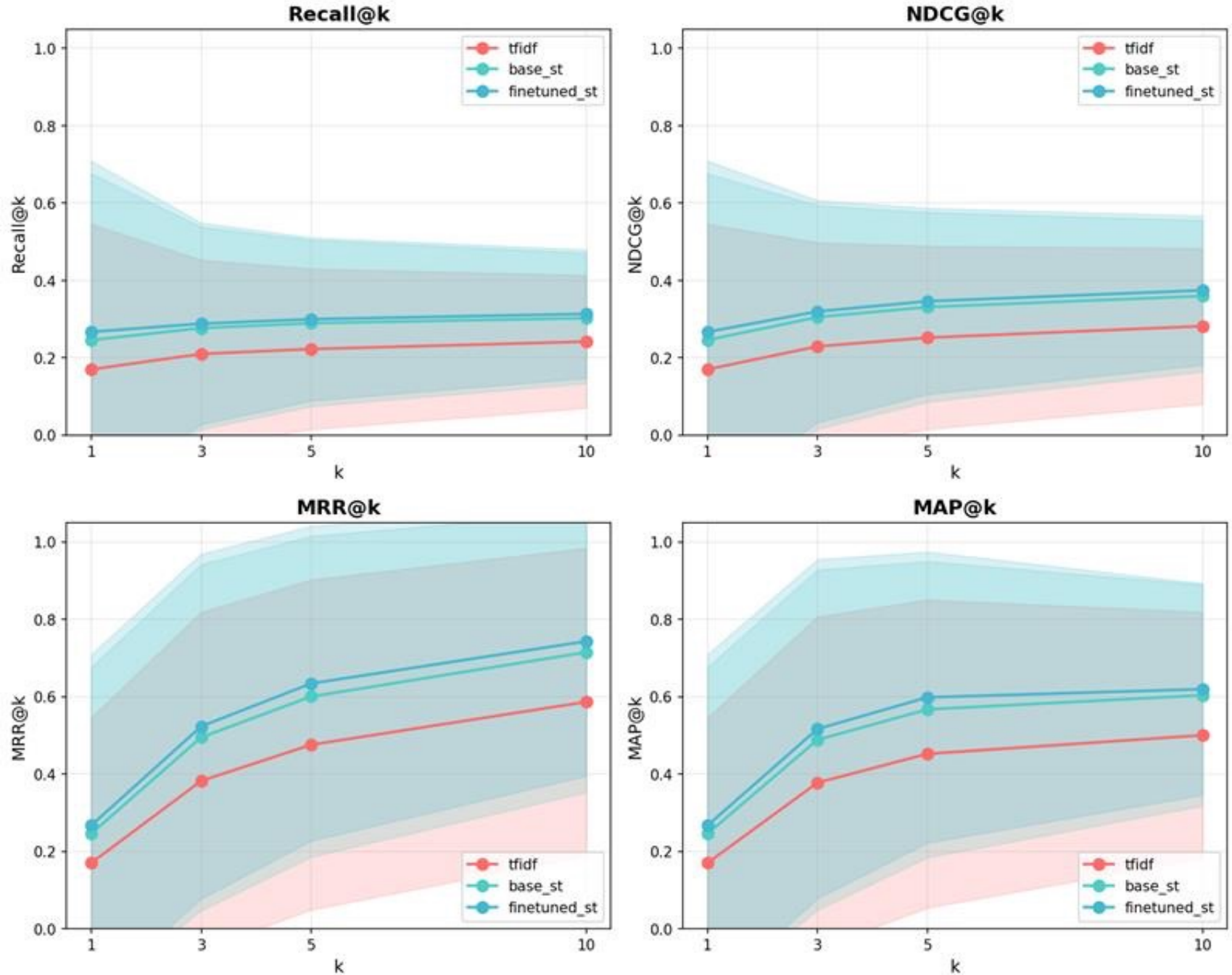
Evaluation

Metrics

- Approach: Generate embeddings only for the 5K holdout data set using TF-IDF, Base ST, Finetuned ST and ada-002. Use ada-002 top K recommendations as the ground truth and calculate the following metrics.

Metric	Description	Interpretation
Recall@K	Fraction of Ada's top-K found in model's top-K	Higher = finds more relevant papers
NDCG@K	Normalized Discounted Cumulative Gain	Higher = correct ranking order
MRR@K	Mean Reciprocal Rank	Higher = first relevant result appears earlier
MAP@K	Mean Average Precision	Higher = overall ranking quality

- These metrics were selected as they reflect the quality of recommendations and indicate how closely the finetuned model mimics ada-002.
- The above metrics were calculated for 5000 data points with different values of K and the average is ploted in the graph. The shaded area indicates the standard deviation.


Model Comparison vs Ada-002 (Ground Truth)
Recall@k
NDCG@k
MRR@k
MAP@k
tfidf
base_st
finetuned_st
Recall@k
NDCG@k
MRR@k
MAP@k
k
0.0
0.2
0.4
0.6
0.8
1.0
1
3
5
10

Deployment

Deployment Architecture

[Link to demo for ada-002 distillation]

[Link to demo for LLM as Judge]

Serverless Execution with Cloud Run

Using serverless Cloud Run ensures scalable and cost-efficient deployment of microservices.

Microservices Architecture

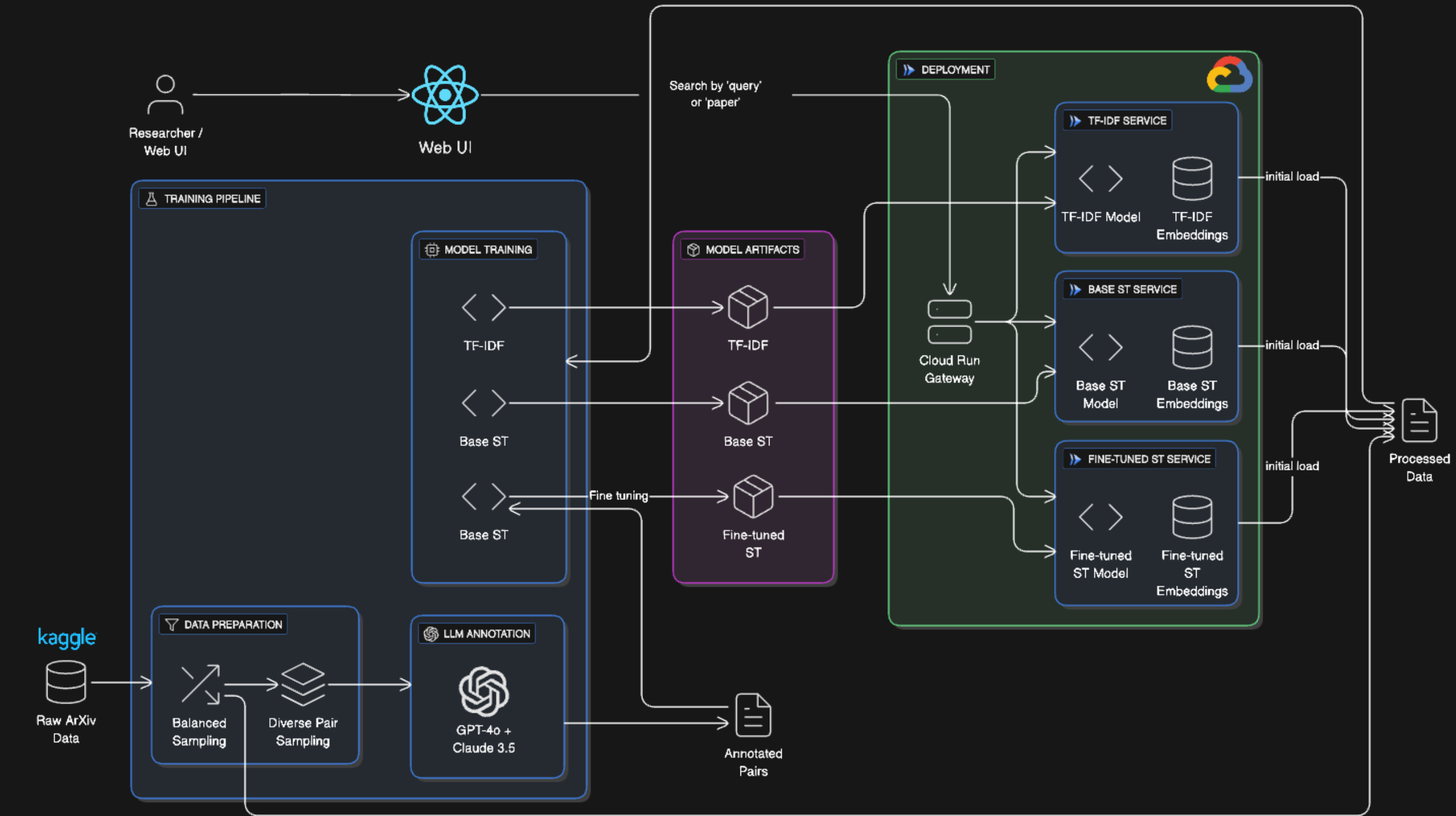
Gateway routes requests to three independent microservices hosting different models for scalability and isolation.

Optimized Docker Images

Model artifacts are baked into Docker images to reduce startup latency and improve performance.

Cost Optimization Strategy

Min-instances set to zero balances cost savings with minor cold start delays in production.


Researcher / Web UI
Web UI
Search by 'query' or 'paper'
DEPLOYMENT
TRAINING PIPELINE
MODEL TRAINING
TF-IDF
Base ST
Base ST
MODEL ARTIFACTS
TF-IDF
Base ST
Fine-tuned ST
Fine tuning
Cloud Run Gateway
TF-IDF SERVICE
TF-IDF Model
TF-IDF Embeddings
BASE ST SERVICE
Base ST Model
Base ST Embeddings
FINE-TUNED ST SERVICE
Fine-tuned ST Model
Fine-tuned ST Embeddings
initial load
initial load
initial load
Processed Data
kaggle
Raw ArXiv Data
DATA PREPARATION
Balanced Sampling
Diverse Pair Sampling
LLM ANNOTATION
GPT-4o + Claude 3.5
Annotated Pairs

Limitations

Limitations

- With just 10K training pairs, we were only able to show a small improvement. We might be able increase the number of training pairs to get better performance.
- With only 384 dimensions, the representation power of 'all-minilm-l6-v2' is limited and will not able to achieve the same performance of ada-002 across multiple domains. In fact, the embedding space completely collapsed in some of the attempts.
- The representation power of ada-002 itself is limited as it is an outdated embedding model. We used it as it was cheaper than the newer ones and was readily accessible.
- We could have used more data to train for LLM as Judge, but due to latency issues and cost issues, did not do.

Contributions

Contributions

- We followed an ensemble technique. Both of us independently worked on each phase of the project, compared results and took the best from both work.

Name	Contribution
Grampurohit Santosh	• Problem Statement • Data Preparation • Data Modelling • Evaluation • Deployment • UI
Amirthalingam	• Data Preparation • Data Modelling • Evaluation • Deployment • UI • Presentation and Reports.

Data Science Canvas

Project: Research Paper Recommendation Service

Team: ScholarStream

Problem Statement

Business Case & Value Added
There is explosion of the research articles, and it is difficult to keep up with the newly introduced concepts and trending topics.

Model Selection
Embedding models are known to model a vector space that encodes real world concepts and therefore help with semantic similarity.

We will use,
TF-IDF as baseline 1,
all-minilm-l6-v2 as baseline 2

Model Requirements
- Semantic understanding.
- Domain relevance
- Scalability
- Low latency
- Input flexibility
- Low cost

Skills
What skills are needed to provide the data and model development?

- EDA
- Developing application using python framework like FASTAPI
- Vector DB
- NLP
- Machine Learning
- Containerization
- Cloud Deployment

Data Landscape

Yes data is already available from arxiv
Additional data can be generated by extracting pdf content for certain categories. But, for the purpose of this project, we will limit ourself to just the metadata (title, abstract, authors etc.)

Software & Libraries
- Numpy
- Pandas
- Sci-kit learn
- Vector db like milvus db
- LLMs (ex gemini-flash-2.5)
- Langchain
- HuggingFace
- FastAPI

Execution & Evaluation

Model Evaluation
- Precision@K
- Recall@K
- MRR
- nDCG@K

Real time monitoring:
- Latency
- Throughput
- Error rate
- Resource usage

Data Storytelling
A simple ui with detailed research paper information, recommendations and similarity scores.

Data Collection & Preparation

Data Selection & Cleansing
The data is readily available in arxiv. We filter out papers that have non ascii characters in the title or abstract.

Data Collection
No additional data is required.

Data Integration
Data integration is not required as it is from a single source

Explorative Data Analysis
We only use papers from ‘cs.ai’ category. The data is clean and doesn’t require EDA.

Adopted from: https://github.com/tomalytics/datasciencecanvas