université
PARIS-SACLAY

# Information Retrieval
# Final Project

Instructor: Kim Gerdes

Dang Hoang Khang Nguyen
Ahmed Nazar
Md Naim Hassan Saykat

# Outline

1. Introduction
2. Dataset
3. Task 1
4. Task 2
5. Conclusion

# Introduction

## 🎯 What We Did

In this project, we explored and compared different retrieval methods for patent data, including:

- Sparse models like TF-IDF and BM25
- Dense models like Sentence Transformers (MiniLM, PatentSBERTa)
- Fusion techniques like Reciprocal Rank Fusion (RRF) to combine strengths of multiple models

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 9 | khanghoang0902 | 1 | 2025-04-10 21:01 | 263701 | **0.499** | 0.94 | **0.273** | n/a |

We also:

- Preprocessed complex patent text (titles, claims, descriptions)
- Tuned hyperparameters for optimal retrieval performance
- Evaluated each method using standard IR metrics (MAP, Recall@100)

# Dataset

**Dataset Overview**

- Patent datasets with multiple text fields:
  - Title, Abstract, Claims, Descriptions, and Fulltext
- Analysis included:
  - Title length distribution
  - Comparison of text similarities between citing and cited patents
  - Number of citations per patent

# Exploratory Data Analysis

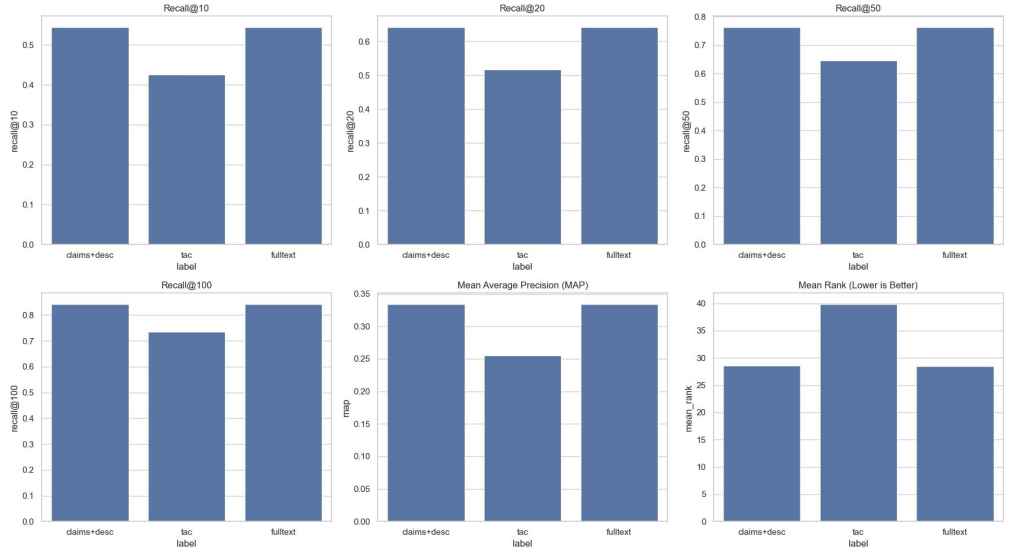
- Looking at title lengths and averages
- Calculating number of citations per patent
- Comparing textual content between citing and cited patents
- Analyzing differences across fields (Title, Abstract, Claims, Descriptions)

# Task 1 - Find the citation

1. Sparse Embedding
    a. TF-IDF with Cosine Similarity
    b. BM25
2. Dense Embedding
    a. all-MiniLM-L6-v2
    b. PatentSBERTa
3. Reciprocal Rank Fusion (RRF)
    a. Same Embedding
    b. Cross Embedding

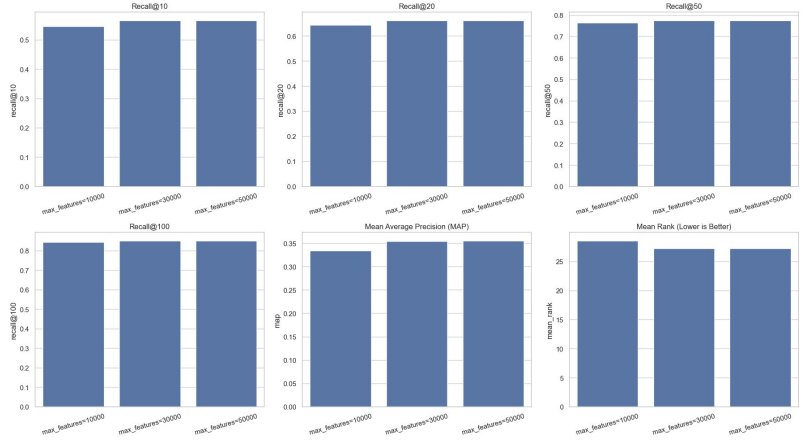
# Task 1 - TF-IDF

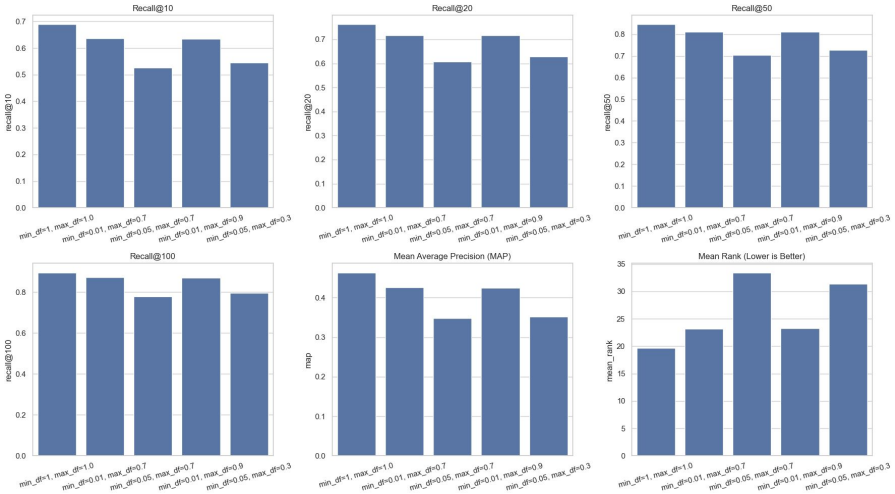
- Evaluate TF-IDF on three different corpus: full text, TAC (title, abstract, claims) and claims+desc (claims (**“c-en-”**), descriptions (**“p-”**))
- claims+desc has similar performance to full text but slightly faster training time



# Task 1 - TF-IDF

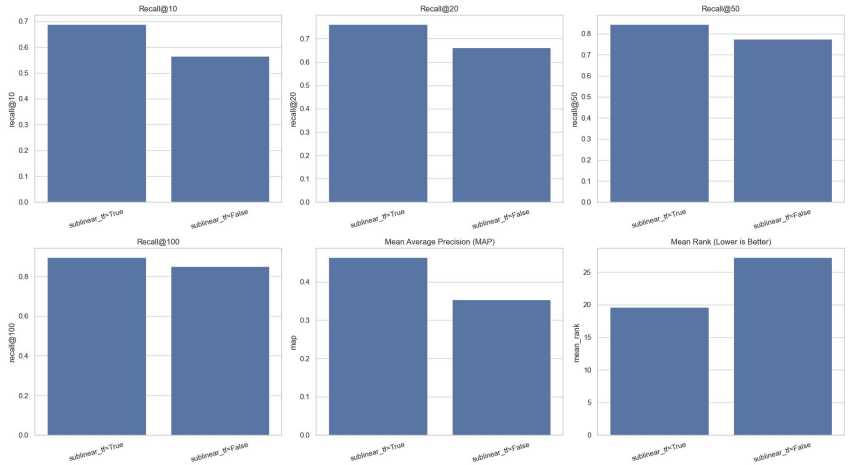
- max_features = [10000, 30000, 50000] => 30000
- min_df, max_df => default setting

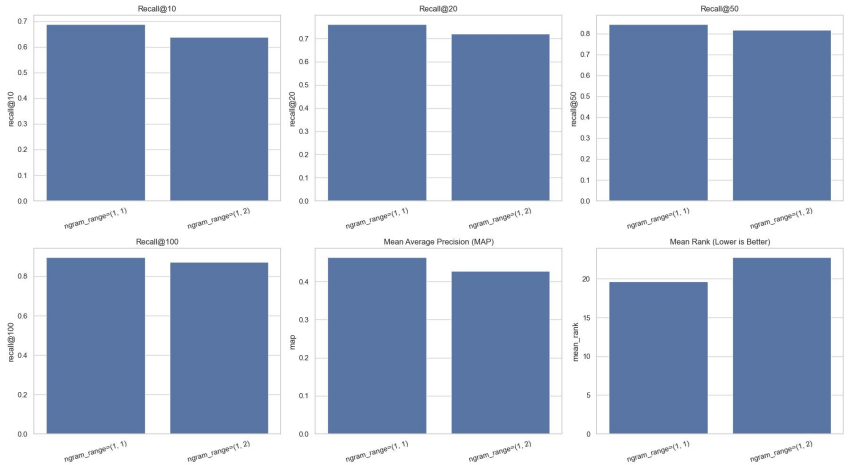


# Task 1 - TF-IDF

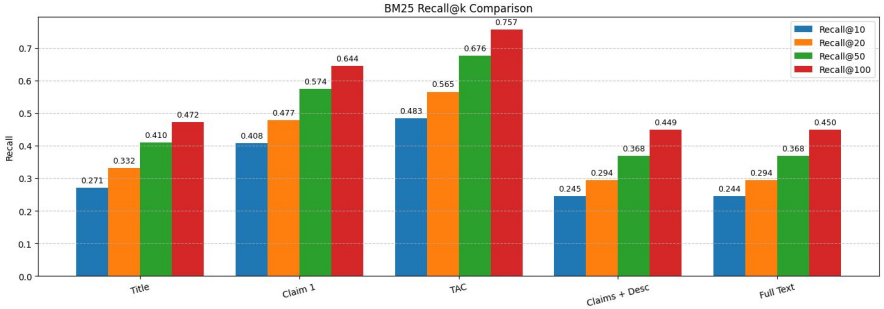
- sublinear_tf = True or False => True
- unigram or bigram => unigram

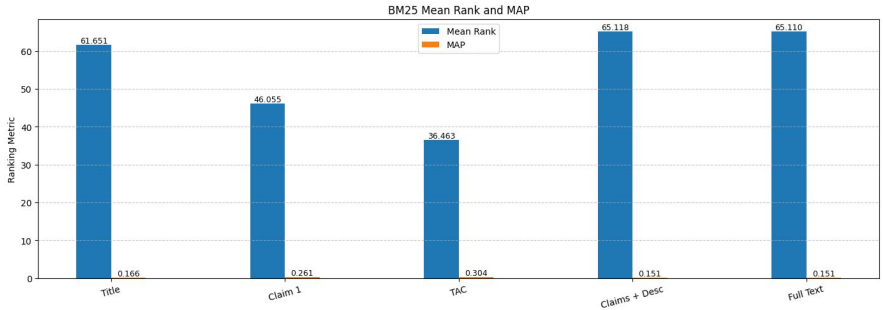


# Task 1 - BM25

- For BM25, more text => worse performance
- Overall performance is not as good as TF-IDF



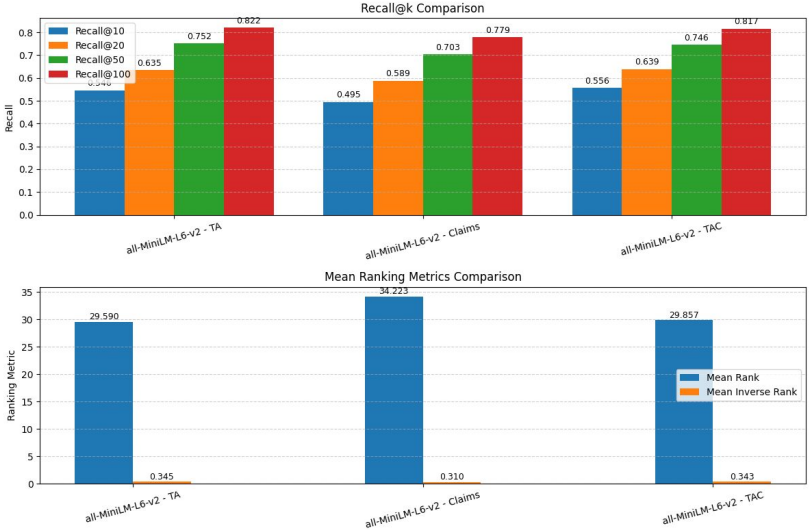



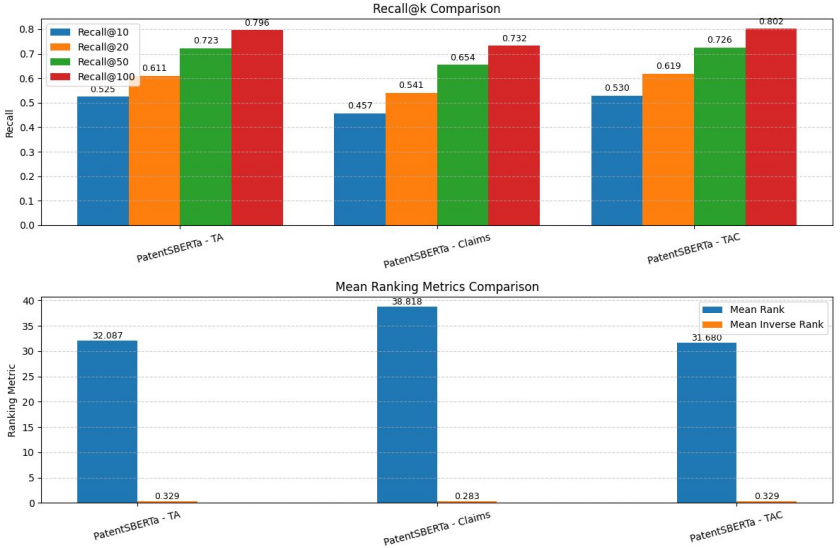
# Task 1 - Baseline

- Best setting TF-IDF: stop_word=”english”, max_features=30000, sublinear_tf=True, ngram_range=(1, 1)
- Test result on Codabench:
  - Recall@10: 0.6768
  - Recall@20: 0.7572
  - Recall@50: 0.8517
  - Recall@100: 0.9042
  - Mean Rank: 19.38
  - Mean Inverse Rank: 0.4475
  - Mean Average Precision: 0.4726

# Task 1 - Dense Embedding

- For dense embedding, all-MiniLM-L6-v2 has better overall performance than PatentSBERTa



# Task 1 - RRF

$$\text{RRF}(d) = \sum_{m=1}^{M} \frac{1}{k + \text{rank}_m(d)} \text{where:}$$

$\text{RRF}(d)$ is the final score for document $d$

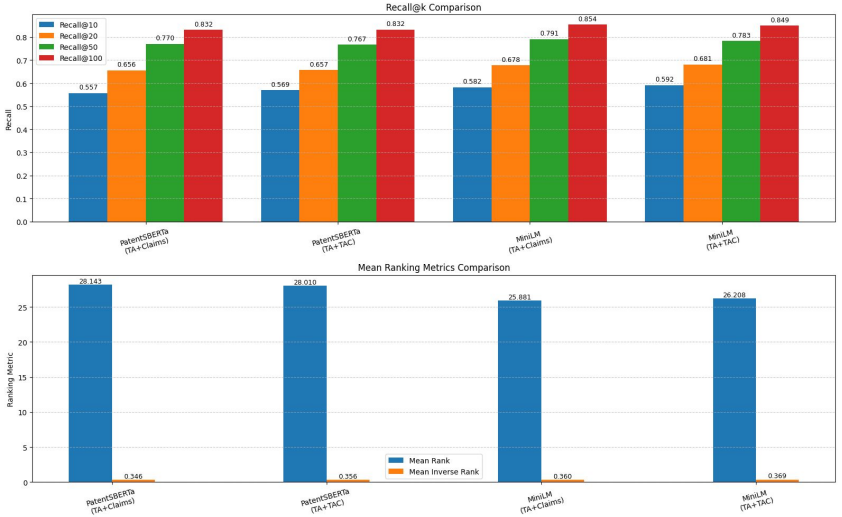
$M$ is the number of retrieval methods

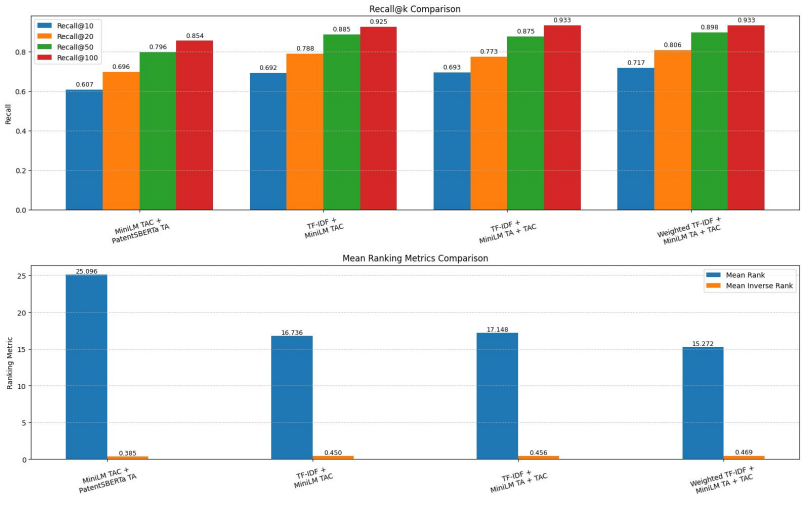
$\text{rank}_m(d)$ is the rank of document $d$ in method $m$

$k$ is a constant, typically set to 60

# Task 1 - RRF

- For Reciprocal Rank Fusion, fusion of different embeddings has better performance



# Task 1 - RRF

- Best RRF: Weighted TF-IDF, all-MiniLM-L6-v2 TAC, and all-MiniLM-L6-v2 TA
- Weight component: $\lambda_{TF\text{-}IDF} = 0.5$, $\lambda_{TAC} = 0.3$, $\lambda_{TA} = 0.2$
- Test result on Codabench:
  - Recall@10: 0.7344
  - Recall@20: 0.8308
  - Recall@50: 0.9125
  - Recall@100: 0.9401
  - Mean Rank: 14.05
  - Mean Inverse Rank: 0.4727
  - Mean Average Precision: 0.4988

# Task 2 - Reranking

**Objectives**

- **Improve reranking**
- **Maximize metrics on test data**

**Input: Queries + Documents + Initial Pre-ranking.**

**Output: Re-ranked documents per query.**

# Dataset Overview

- Queries: Claims and TAC1 types.
- Documents: Scientific publications.
- Pre-Ranking: 20 documents per query

# Approach Overview

- Use Cross-Encoder models.
- Fine-tune transformers to score query-document pairs.
- Predict a score for each pair.

# Model Used

- infly/inf-retriever-v1-1.5 (claims)
- infly/inf-retriever-v1-1.5b (tac1)
- BAAI/bge-reranker-large (claims)
- Baseline (no re-ranking)

# Training Details

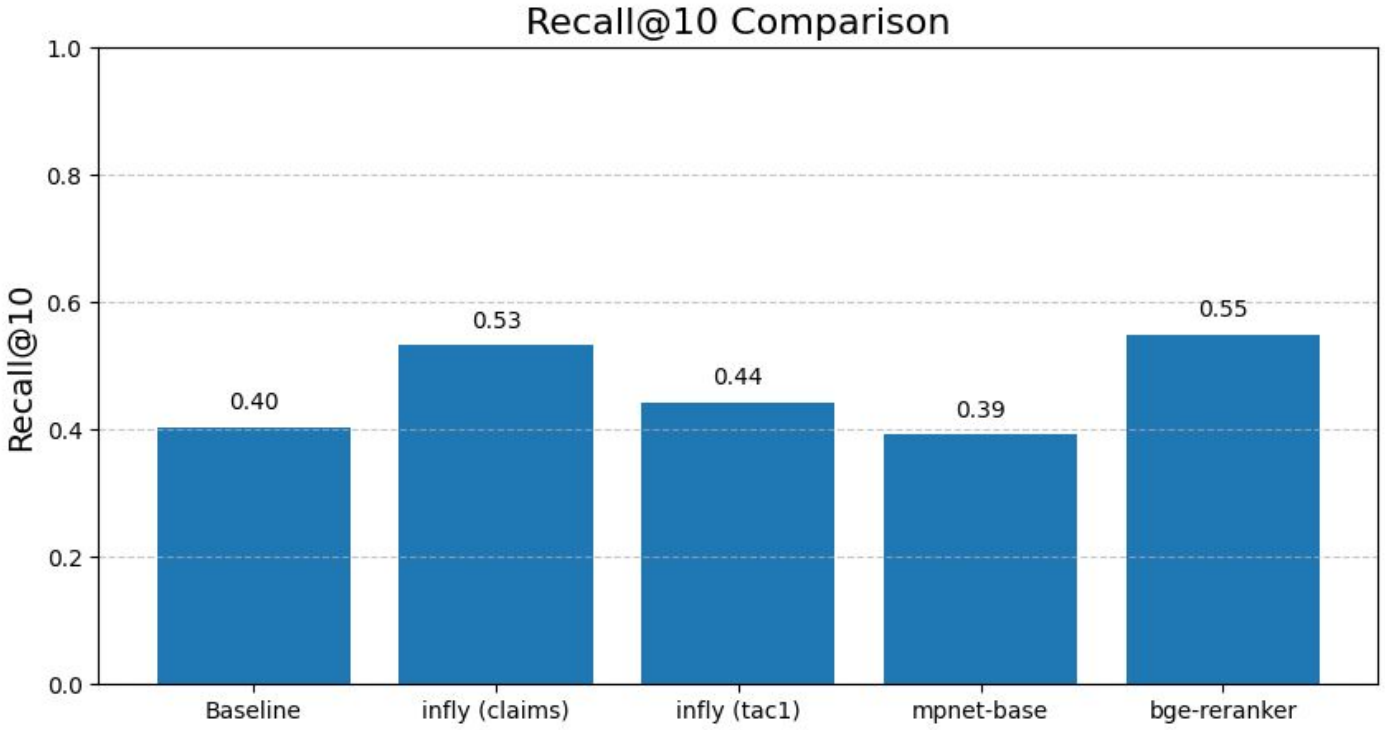
- Max Leghth: 512/768 tokens
- Batch size: 4/8
- Optimizer: AdamW
- Loss: Binary cross-entropy on relevance

# Evaluation Metrics Result

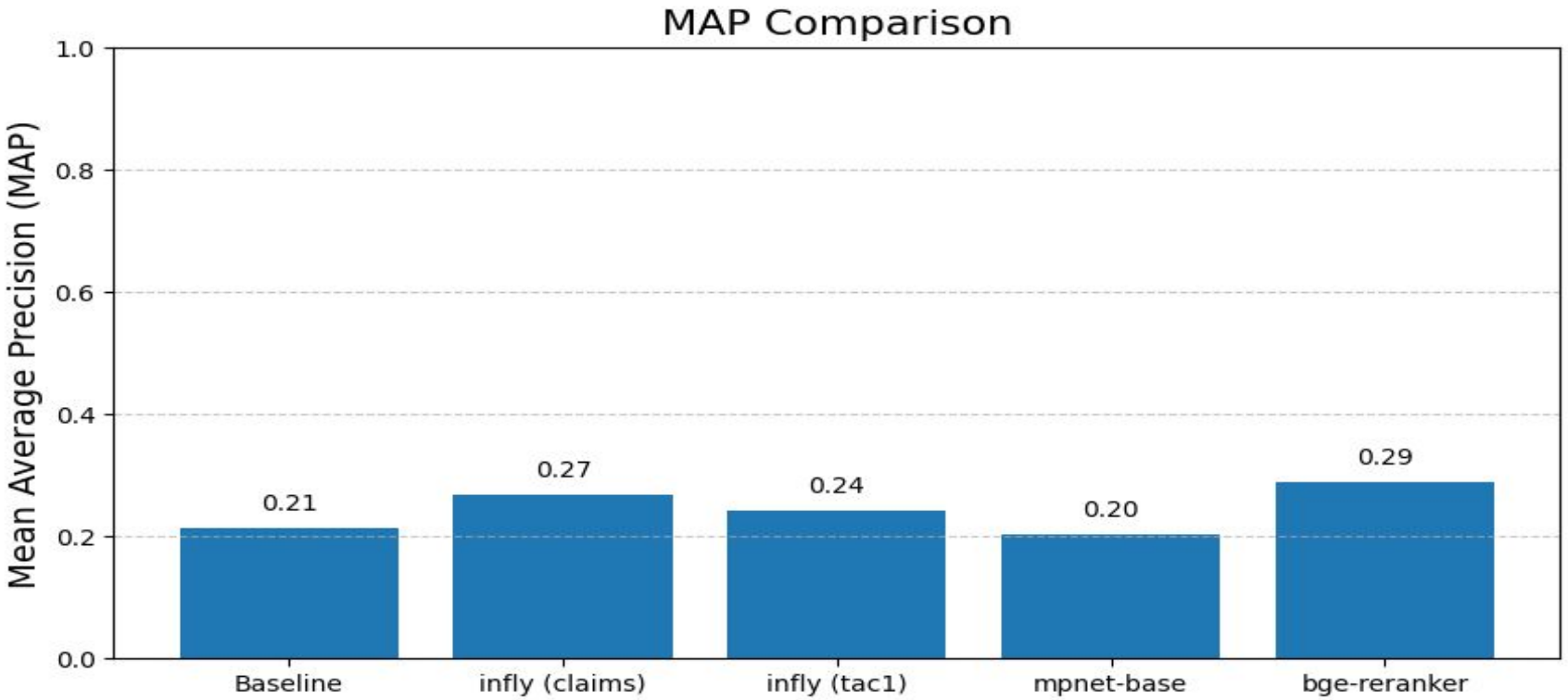
**Evaluation Metrics**

- **Recall@K**:
  - Recall@3: `0.1283`
  - Recall@5: `0.2413`
  - Recall@10: `0.5321`
  - Recall@20: `0.8627`
- **MAP (Mean Average Precision)**: `0.2681`
- **Mean Reciprocal Rank (MRR)**: `0.3878`
- **Mean Rank**: `4.90`

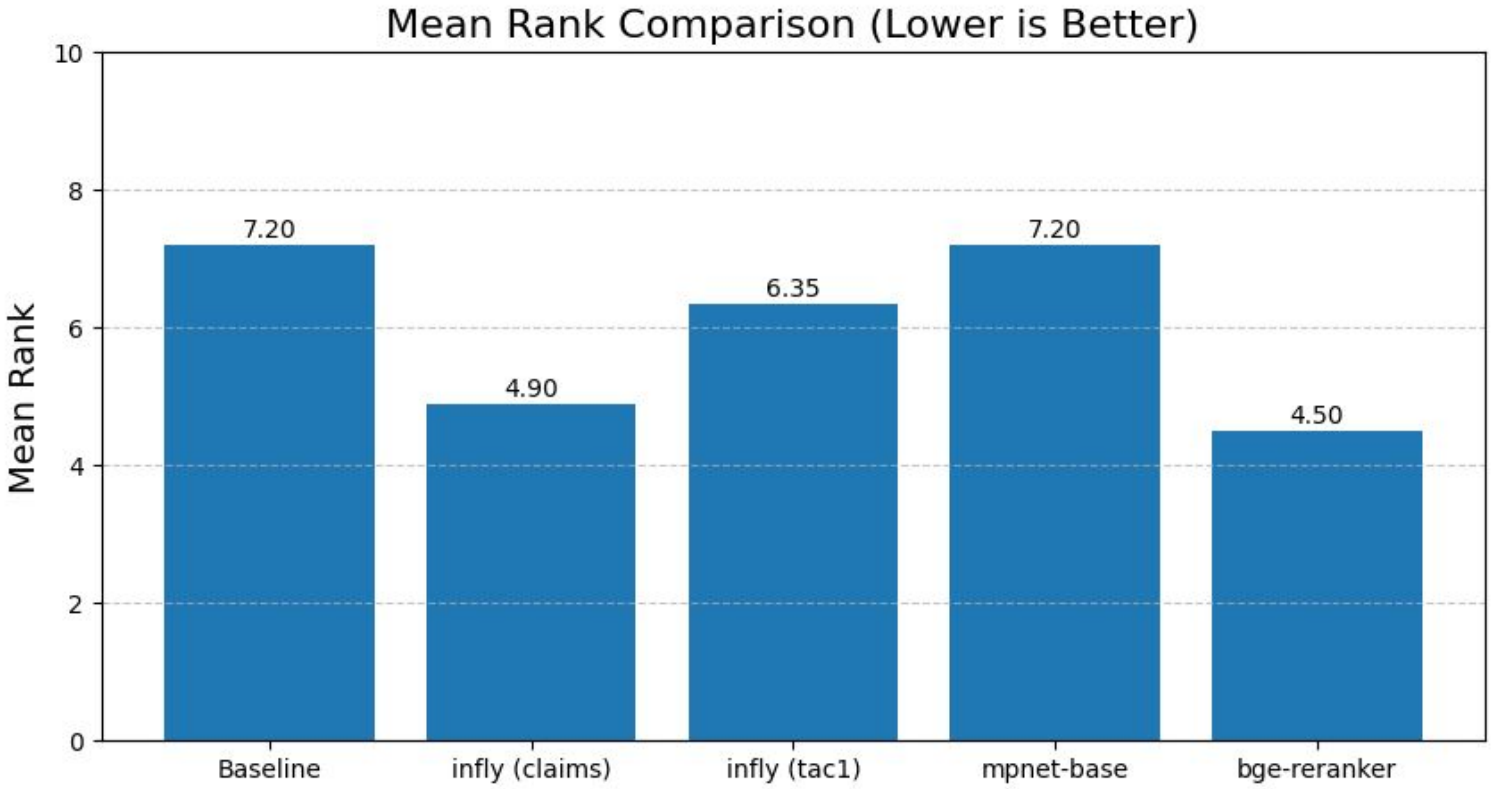
# Train Results: Recall@10



# Train Results: MAP



# Train Results: Mean Rank



# Analysis

- BGE reranker achieved highest Recall@10 and MAP.
- infly claims model performed very close to BGE reranker..
- Cross-encoder re-ranking clearly boosts performance.

# Test Submission

- Test Queries: 10 queries
- Best Model: infly/inf-retriever-v1-1.5b (claims).
- Predictions saved for external evaluation.

# Conclusion

- Cross-encoders improve retrieval significantly
- infly model provided strong gains over baseline

# Future Work

- Explore larger models.
- Investigate multi-query fusion techniques.
- Fine-tuning further with hard negatives

# Thank you!