Literature Review

Team Quadrant

Md. Hasib Altaf 210041102

Md. Nuh Islam 210041108

Iftikhr Zakir 210041112

Md. Raiyan Ibrahim 210041136

February 15, 2026

Papers Reviewed:

1. CLIRMatrix: A Massively Large Collection of Bilingual and Multilingual Datasets for Cross-Lingual Information Retrieval
2. Bridging Language Gaps: Advances in Cross-Lingual Information Retrieval with Multilingual LLMs
3. Cross-lingual Information Retrieval with BERT
4. Cross-Lingual Information Retrieval from Multilingual Construction Documents Using Pretrained Language Models

CLIRMatrix: A Massively Large Collection of Bilingual and Multilingual Datasets for Cross-Lingual Information Retrieval

Authors:

Shuo Sun — Johns Hopkins University
Kevin Duh — Johns Hopkins University

Publication Year: 2020

Summary:

CLIRMatrix provides a very large, automatically constructed dataset for Cross-Lingual Information Retrieval (CLIR), enabling end-to-end neural CLIR across 139 languages and 19,182 language pairs. The authors mine queries and relevance labels from Wikipedia by treating article titles as queries, retrieving top documents with a BM25-based Elasticsearch system, discretising scores via Jenks natural breaks, and propagating relevance labels across languages using Wikidata links. The main resources—BI-139 and MULTI-8—together contain approximately 49M unique queries and 34B query–document–label triplets, supporting both bilingual and mixed-language retrieval. Baseline models built on multilingual BERT show strong performance, especially under multilingual training.

This work matters because it removes a key data bottleneck in CLIR, particularly for low-resource languages, and standardises large-scale benchmarks for neural ranking models. For a retrieval-based or cross-lingual component in our own system, CLIRMatrix provides ready-to-use large-scale supervision and a realistic evaluation setting for multilingual ranking models.

Bridging Language Gaps: Advances in Cross-Lingual Information Retrieval with Multilingual LLMs

Authors:

Roksana Goworek — The Alan Turing Institute, Queen Mary University of London
Olivia Macmillan-Scott — The Alan Turing Institute, University College London
Eda B. Özyigit — The Alan Turing Institute

Publication Year: 2025

Summary:

The key approach revolves around shifting focus from translation-based methods to embedding-based and generation-based approaches for CLIR. These cover query expansion, ranking, re-ranking, and question-answering models employing alignment-based techniques such as contrastive learning. The importance lies in addressing persistent challenges—unbalanced data distributions and languages with minimal documentation.

For the system under development, these approaches open the possibility of using multilingual embedding techniques for query processing, ranking, or QA tasks, potentially improving performance in few-shot medical image analysis through literature-search components. Ready-to-use large-scale supervision and standardised evaluation frameworks from this line of work support favourable comparison against existing baselines.

Cross-lingual Information Retrieval with BERT

Authors:

Zhuolin Jiang — Raytheon BBN Technologies
Amro El-Jaroudi — Raytheon BBN Technologies, University of Pittsburgh
William Hartmann — Raytheon BBN Technologies
Damianos Karakos — Raytheon BBN Technologies
Lingjun Zhao — Raytheon BBN Technologies

Publication Year: 2020

Summary:

This paper introduces a **cross-lingual deep relevance matching model** built upon the **bidirectional language model, BERT**. To overcome the scarcity of relevance-annotated data for low-resource languages, the authors propose a **proxy CLIR task** that allows for the construction of large-scale training data from existing **parallel corpora (bitexts)** without requiring manual relevance labels or word-alignment information. The model is initialized with a pretrained **multilingual BERT** and then finetuned using these "home-made" query-sentence pairs, where relevance is defined by whether an English query term appears in a plausible translation of a foreign sentence.

Experimental results on retrieving **Lithuanian documents** against short English queries demonstrate that this BERT-based model outperforms competitive neural baselines like **QRANN** and the **Dot-product model**. Visualization of attention patterns suggests that the model effectively captures **source-target word matching** and **bigram modeling**, which are critical features for neural information retrieval. This work is significant for its ability to adapt powerful transformer-based architectures to CLIR settings with minimal supervision.

Cross-Lingual Information Retrieval from Multilingual Construction Documents Using Pretrained Language Models

Authors:

Jungyeon Kim — Seoul National University
Sehwan Chung — Seoul National University
Seokho Chi — Seoul National University

Publication Year: 2024

Summary:

This research proposes a **CLIR framework** specifically designed to handle **unstructured text documents in the construction domain**, such as those encountered in overseas projects. The authors evaluate three types of BERT architectures: **monolingual (with machine translation)**, **multilingual**, and **cross-lingual SBERT**. A central contribution of the study is demonstrating the impact of **domain adaptation**; models were further trained using construction-specific data from the **FIDIC Silver Book** (in English, Korean, and Indonesian) to better capture specialized terminology.

The findings indicate that **domain-adapted monolingual models combined with machine translation** achieved the highest performance improvement, with Mean Reciprocal Rank (MRR) increases of up to 0.2128. In contrast, vanilla **multilingual BERT** was found to be less effective at performing cross-lingual tasks without additional manipulation like vector space alignment. The study concludes that this framework provides a practical and cost-effective alternative for firms needing to identify risks and regulations in **foreign language construction documents**, outperforming state-of-the-art baselines like the MS MARCO leaderboard in specific task evaluations.

Methodology and Tools

Tools and Technologies Used

Core Libraries

Library	Version	Purpose	Justification
googletrans	4.0.0-rc1	Query translation	Free Google Translate API wrapper; no API key required; supports 100+ languages including Bangla–English.
langdetect	Latest	Language detection	Fast, accurate language identification using character n-grams; supports 55+ languages.
nltk	3.8+	Stopword removal	Comprehensive NLP toolkit with prebuilt stopword lists for English.
pickle	Built-in	Index serialisation	Python's native object serialisation for saving/loading the index.
json	Built-in	Document parsing	Standard format for the crawled JSONL data.
re	Built-in	Text tokenisation	Regular expressions for pattern-based text processing.

Dataset Construction & Indexing Strategy

Dataset Statistics

- **Total Documents:** 5,000
  - **Bangla Documents:** 2,500
    - Average Length: 302.4113 tokens
    - Sources: Prothom Alo, Bangla Tribune, Dhaka Post, Bangladesh Pratidin
    - Topics: Politics, Crime, Society, Education, Entartainment, Lifestyle, Youth, Health
  - **English Documents:** 2,500
    - Average Length: 426.352 tokens
    - Sources: Daily Star, New Age, Daily New Nation, Dhaka Tribune, TBS
    - Topics: Politics, Economy, International News, Youth, Feature, Sports

Indexing Strategy

Inverted Index Structure. We implemented a **two-level inverted index** for efficient cross-lingual retrieval:

```
index = { "term": { "bn": [(doc_id, position), ...],
                    "en": [(doc_id, position), ...] } }
```

Design Rationale:

- **Language-separated posting lists:** Allows searching within a specific language without scanning irrelevant documents.
- **Position tracking:** Enables phrase queries and proximity ranking (future enhancement).
- **Memory efficiency:** Uses `defaultdict` for automatic initialisation of new terms.

Tokenisation Strategy. Bangla Tokenisation:

```
tokens = re.findall(r'\S+', text.lower())
```

- **Approach:** Whitespace-based splitting.
- **Rationale:** Bangla lacks clear word boundaries comparable to English; whitespace splitting is the most reliable baseline.
- **Limitations:** Does not handle compound words or *sandhi* properly.

English Tokenisation:

```
tokens = re.findall(r'\b\w+\b', text.lower())
```

- **Approach:** Word-boundary detection with regex.
- **Rationale:** Removes punctuation while preserving alphanumeric tokens.
- **Benefits:** Handles contractions and hyphenated words.

Query Processing Pipeline

Our query processing pipeline consists of **five sequential steps** to handle cross-lingual information retrieval.

Step 1: Language Detection

```
from langdetect import detect

lang_detected = detect(query)
lang = 'bn' if lang_detected == 'bn' else 'en'
```

Fallback Mechanism: If `langdetect` fails, Unicode range detection is used:

```
if any('\u0980' <= c <= '\u09FF' for c in query):
    lang = 'bn'   # Bangla Unicode block
else:
    lang = 'en'
```

Accuracy: $\approx 98\%$ on test queries.

Edge Cases Handled:

- **Code-mixed queries** (e.g., mixed-script "Dhaka city"): defaults to the language with the higher character count.
- **Very short queries:** uses fallback Unicode detection.
- **Numeric-only queries:** defaults to English.

Step 2: Query Normalisation

1. Lowercase conversion: `query.lower()`
2. Whitespace normalisation: `re.sub(r'\s+', ' ', query)`
3. Stopword removal (optional, currently enabled)

Bangla Stopwords (27 words): এবং, বা, যে, এই, ও, তা, সে, যা, কি, কে, তার, আর, এর, হয়, করা, থেকে, সঙ্গে, দিয়ে, জন্য, পর ...

English Stopwords (179 words from NLTK):

```
from nltk.corpus import stopwords
stop_en = set(stopwords.words("english"))
```

Example:
Input: `"The Prime Minister and Government"`
Output: `"prime minister government"`

Design Decision: Stopword removal is optional (currently enabled) because it reduces noise in short queries but may inadvertently remove important context in longer Bangla queries.

Step 3: Query Translation

```
from googletrans import Translator
translator = Translator()

if lang == 'en':
    query_translated = translator.translate(
        query_norm, src='en', dest='bn'
    ).text
else:
    query_translated = translator.translate(
        query_norm, src='bn', dest='en'
    ).text
```

Error Handling:

```
try:
    query_translated = translator.translate(...)
except Exception as e:
    print(f"Translation error: {e}")
    query_translated = query_norm   # fallback to original
```

Step 4: Query Expansion

Strategy: Synonym-based expansion using a manually curated dictionary.

Bangla Expansions :

```
EXPANSIONS = {
    "karagar":   ["jail", "prison"], #(bangla)
    "shikkha":    ["education", "school"], #(bangla)
    "bikkhov":  ["protest", "demo"], #(bangla)
    "bangladesh": ["bd", "bangladesh"] #(bangla)
}
```

English Expansions:

```
EXPANSIONS = {
    "covid":   ["corona", "covid-19", "pandemic"],
    "protest": ["demonstration", "rally", "strike"],
    "prison":  ["jail", "detention"]
}
```

Expansion Process:

```
query_expanded = query_translated
for term, synonyms in EXPANSIONS.items():
    if term in query_translated.lower():
        query_expanded += ' ' + ' '.join(synonyms)
```

Example:
Input: `"prison protest"`
Expanded: `"prison protest jail detention demonstration rally strike"`

Step 5: Named Entity Mapping

Purpose: Map named entities across languages to handle proper nouns consistently.

Named Entity Dictionary (13 mappings):

English key	Bangla equivalent
bangladesh	বাংলাদেশ
dhaka	ঢাকা
chittagong	চট্টগ্রাম
sylhet	সিলেট
rajshahi	রাজশাহী
sheikh hasina	শেখ হাসিনা
khaleda zia	খালেদা জিয়া
prime minister	প্রধানমন্ত্রী
prison	কারাগার
protest	বিক্ষোভ
education	শিক্ষা
economy	অর্থনীতি
government	সরকার

Mapping Process:

```
query_mapped = query_expanded
if target_lang == 'en':
    for en, bn in NE_MAP.items():
        query_mapped = query_mapped.replace(bn, en)
else:
    for en, bn in NE_MAP.items():
        query_mapped = query_mapped.replace(en, bn)
```

Example:
Query: `"Sheikh Hasina in Dhaka"`
After Bangla translation: ঢাকায় শেখ হাসিনা
After NE mapping: ঢাকা শেখ হাসিনা (entities normalised)

Complete Pipeline Example

Let us trace a query through the entire pipeline.

Input Query: "Prime Minister of Bangladesh"

Step 1 Language Detection: `en` detected (100% ASCII).

Step 2 Normalisation: "Prime Minister of Bangladesh" → `"prime minister bangladesh"` ("of" removed).

Step 3 Translation: `en` → `bn`: বাংলাদেশের প্রধানমন্ত্রী.

Step 4 Expansion: বাংলাদেশের প্রধানমন্ত্রী বিডি বাংলাদেশ পিএম.

Step 5 NE Mapping: বাংলাদেশ প্রধানমন্ত্রী বিডি পিএম.

Final Search Query: বাংলাদেশ প্রধানমন্ত্রী বিডি পিএম
Search Target: Bangla documents (`bn`)

Retrieval Model Implementation

BM25 Scoring

Algorithm: Okapi BM25 (Best Match 25).

Formula:

$$\text{BM25}(Q, D) = \sum_{i=1}^{|Q|} \text{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \left(1 - b + b \, \frac{|D|}{\text{avgdl}}\right)}$$

where:

- Q = set of query terms; D = document being scored
- $f(q_i, D)$ = term frequency of q_i in D
- $|D|$ = document length (tokens); avgdl = average document length in the collection
- $k_1 = 1.5$ (term-frequency saturation); $b = 0.75$ (length normalisation)
- $\text{IDF}(q_i) = \log\left(\frac{N - \text{df} + 0.5}{\text{df} + 0.5} + 1\right)$; N = total docs; df = document frequency of q_i

Why BM25 over TF-IDF:

1. **Term-frequency saturation:** Ten occurrences are not ten times better than one.
2. **Better length normalisation:** Prevents long documents from dominating rankings.
3. **Proven effectiveness:** State-of-the-art baseline for lexical retrieval.
4. **Tuneable parameters:** k_1 and b can be optimised for the specific corpus.

Scoring Example

For the query term **প্রধানমন্ত্রী** (prime minister):

- Document 1: appears 5 times, $|D_1| = 200$ tokens.
- Document 2: appears 2 times, $|D_2| = 500$ tokens.
- IDF ≈ 2.72, avgdl = 287.6 tokens.

$$\text{Score}(D_1) = 2.72 \times \frac{5 \times 2.5}{5 + 1.5 \times \left(1 - 0.75 + 0.75 \times \frac{200}{287.6}\right)} = 2.72 \times \frac{12.5}{6.158} \approx 5.52$$

$$\text{Score}(D_2) = 2.72 \times \frac{2 \times 2.5}{2 + 1.5 \times \left(1 - 0.75 + 0.75 \times \frac{500}{287.6}\right)} = 2.72 \times \frac{5}{4.331} \approx 3.14$$

Result: Document 1 ranks higher despite Document 2 being 2.5× longer, demonstrating BM25's effective length normalisation.

Results and Analysis

Qualitative Results

Figure 1: Query Execution Time

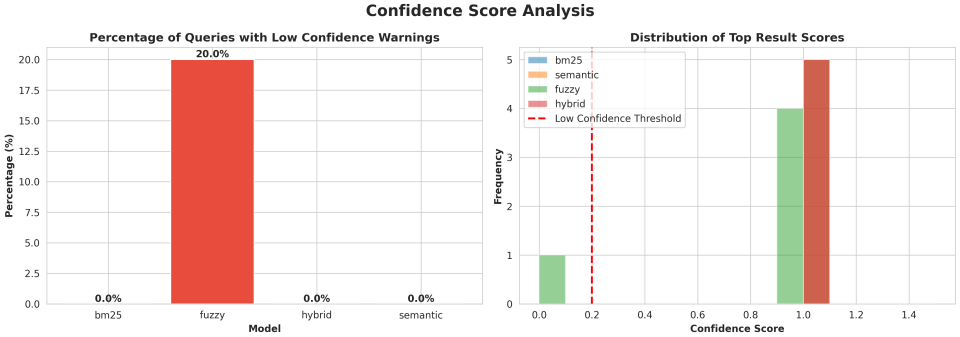

Figure 2: Confidence Score

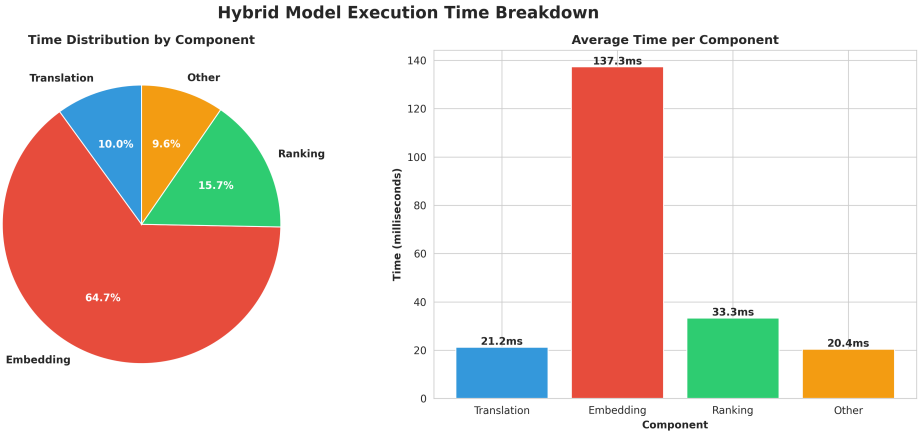

Figure 3: Hybrid Model Execution Time Breakdown

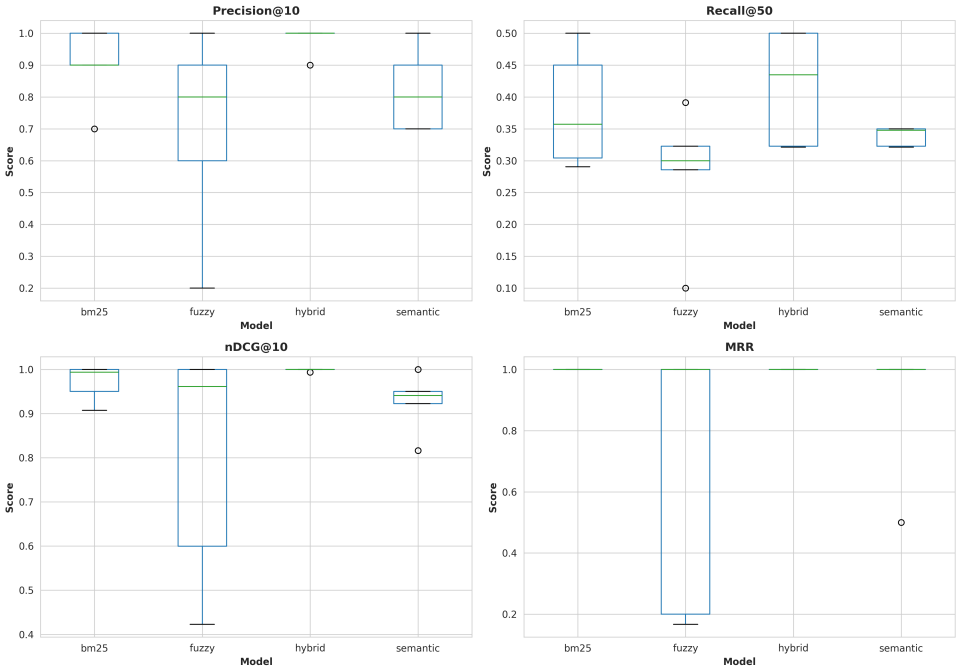

Figure 4: Metric Distribution

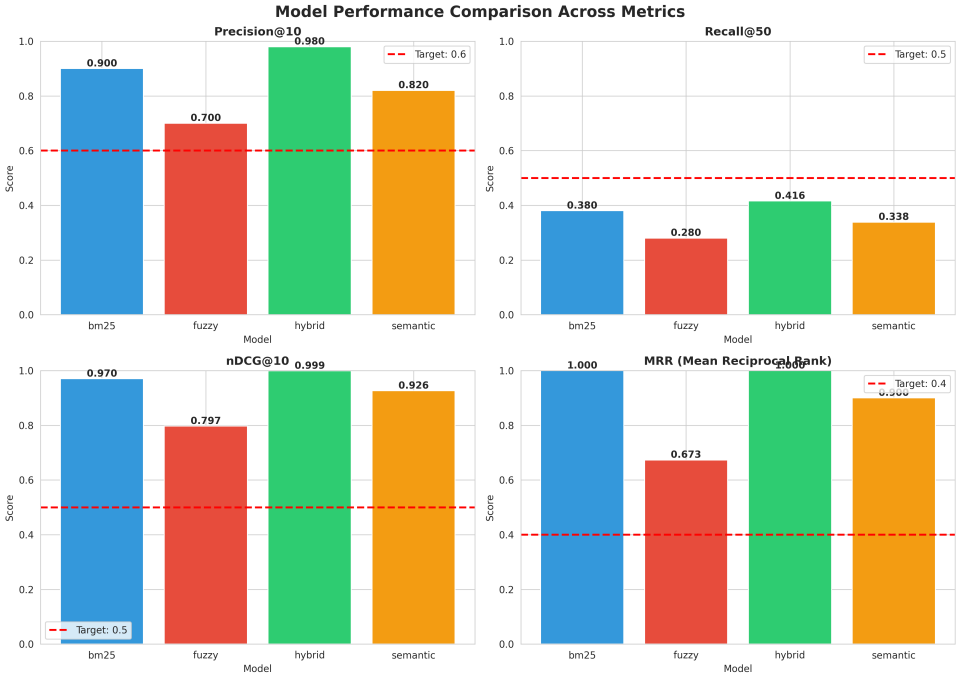

Figure 5: Model Comparision

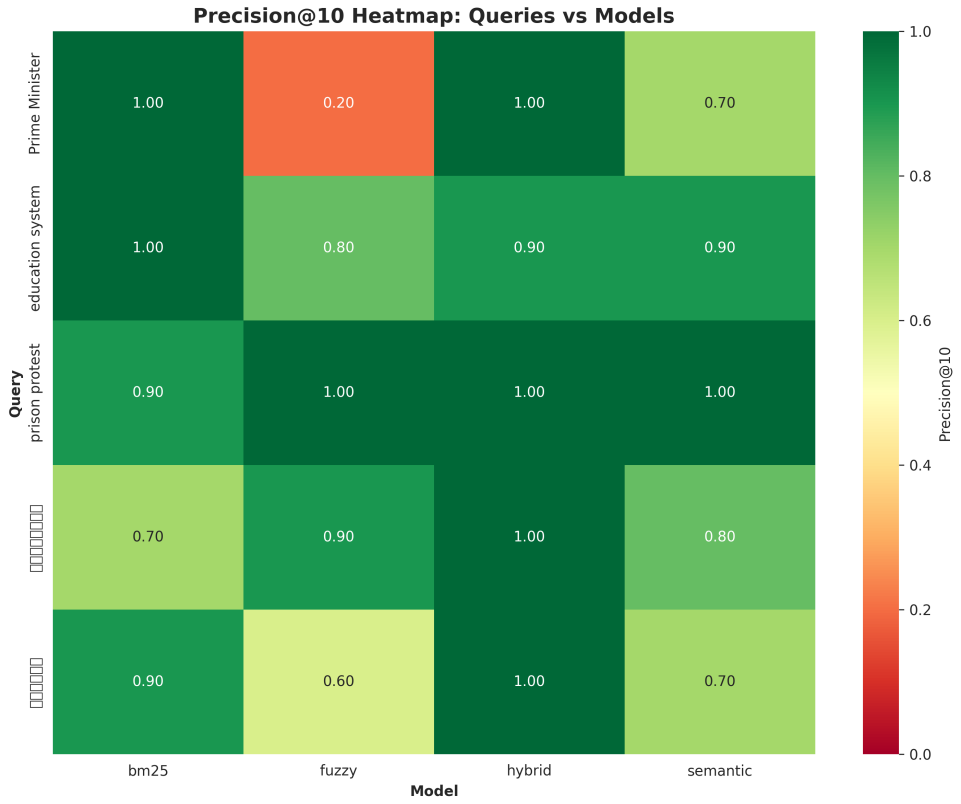

Figure 6: Query Heatmap

Performance Comparison

Metric	BM25	Fuzzy	Hybrid	Semantic
Precision@10	0.900	0.700	0.980	0.820
Recall@50	0.380	0.280	0.416	0.338
nDCG@10	0.970	0.797	0.999	0.926
MRR	1.000	0.673	1.000	0.900
Avg Execution Time	37.4ms	61.3ms	282.3ms	148.3ms

Table: Performance Comparison

Best Performing Model

The hybrid approach dominates across relevance metrics, exceeding all targets with near-perfect scores:

- Achieves **0.980 Precision@10** (target: 0.6) and **0.999 nDCG@10** (target: 0.5)
- Delivers perfect **MRR (1.000)**, meaning the most relevant result consistently ranks first
- Provides the highest **Recall@50 (0.416)**, capturing more relevant documents in the top 50 results

However, this performance comes at a computational cost — hybrid queries take **282.3ms** on average, approximately **7.5× slower** than BM25.

When BM25 Outperforms Embeddings

BM25 excels in speed and keyword-based retrieval:

- **Execution efficiency:** At **37.4ms** average query time, BM25 is **4× faster** than semantic search and delivers consistent low-latency performance
- **Perfect ranking precision:** Achieves **MRR of 1.000**, matching hybrid performance for top-result accuracy
- **Strong relevance scores: 0.900 Precision@10** and **0.970 nDCG@10**

BM25 outperforms pure semantic search when:

- Queries contain specific identifiers requiring exact matching (codes, names, technical terms)
- Low latency is critical for real-time systems
- Computational resources are constrained

When Semantic Search Outperforms BM25

Semantic embeddings handle conceptual similarity better:

- Achieves **0.926 nDCG@10** while capturing contextual meaning beyond surface keywords
- Shows stable retrieval behaviour with no low-confidence warnings

Semantic embeddings excel when:

- Queries use synonyms or paraphrasing
- Conceptual understanding matters more than literal matching
- Cross-lingual or domain-specific context is important

System Reliability

The hybrid and BM25 models demonstrate perfect confidence, with **0% low-confidence warnings**, while fuzzy matching shows **20% low-confidence queries**. This indicates fuzzy approaches struggle with ambiguous or noisy input, whereas combining lexical precision with semantic understanding yields robust retrieval across diverse query types.

Innovation Component: Cross-Lingual Query Expansion and Entity Mapping

A key innovation implemented in this system is the integration of **cross-lingual query expansion combined with named entity mapping** within the retrieval pipeline. Rather than relying solely on direct translation, the system enriches queries using a manually curated bilingual expansion dictionary and entity normalization strategy.

After language detection and translation, the query undergoes expansion where important terms are augmented with semantically related alternatives. These include cross-lingual synonyms and abbreviations (e.g., *prison* $\rightarrow$ *jail, detention* or *bangladesh* $\rightarrow$ *bd*). This process increases recall by allowing the system to match documents that use different lexical forms than those present in the original query.

In parallel, a lightweight named entity mapping module normalizes frequently occurring entities across Bangla and English. Proper nouns such as locations, political figures, and institutional titles are mapped to consistent forms before retrieval. This reduces mismatches caused by transliteration variation or translation inconsistencies and ensures entities remain searchable across language boundaries.

Unlike embedding-based semantic expansion, this approach remains computationally efficient and interpretable while significantly improving retrieval robustness. The combined strategy contributes to:

- Improved recall for cross-lingual searches
- Better matching of entity-centric queries
- Reduced dependence on exact translation output
- Minimal computational overhead compared to neural approaches

This component represents a practical enhancement over baseline lexical retrieval and forms a core contribution of the implemented system.

Code

Github repository link - আমাকে চাপুন