CLIR-Assignment

Building a Cross-Lingual Information Retrieval System

Course: Data Mining - CSE 4739

Group Size: 4 members

Timeline: 3 weeks

Total Marks: 100% (conversion to raw marks will be done later)

Core Themes: • Search & Indexing • Cross-Lingual IR • Semantic Retrieval • Ranking • Evaluation • Research Exploration • Responsible AI Use

1. Motivation & High-Level Objective

Modern information ecosystems are multilingual. A monolingual search engine restricts users' access to knowledge by ignoring documents written in other languages. Cross-lingual information retrieval (CLIR) overcomes this barrier by enabling queries in one language to retrieve relevant documents in another.

Your objective is to build a Cross-Lingual Information Retrieval Engine that:

- Retrieves, ranks, and evaluates multilingual documents
- Uses both lexical and semantic techniques
- Grounds implementation in published IR literature

Core goal: Learn to read, explore, and implement methods from research papers, with room for innovation.

2. Learning Outcomes

2.1 Conceptual Learning

- Understand search pipelines: crawling → indexing → retrieval → ranking → evaluation
- Learn the differences between lexical, fuzzy, semantic, and cross-lingual search
- Understand cross-lingual challenges: translation drift, named entity (NE) mismatch, semantic shift, code-switching
- Understand evaluation metrics in IR and their significance

2.2 Technical Learning

- Extract or crawl multilingual text data

- Build an inverted index with document metadata
- Implement multiple retrieval methods: TF-IDF, BM25, multilingual embeddings, optional hybrid scoring
- Implement query translation and NE mapping
- Measure retrieval quality, latency, and identify failure cases

2.3 Research Skills

- Read and summarize IR/CLIR literature
- Identify strengths and weaknesses of different retrieval models
- Propose innovative extensions to CLIR systems
- Evaluate model behavior rigorously
- Use AI tools, if needed, responsibly and transparently

3. Assignment Modules

Module A — Dataset Construction & Indexing (Core)

Task

Construct a multilingual dataset of at least **2,500 documents per language** (Bangla and English) using one of:

Crawl **5 Bangla and 5 English news sites** from the provided lists (Section 6).
Optional and Advanced: Use Common Crawl / CC-MAIN extraction

Note: Crawling can be error-prone. It is recommended to use Python libraries like `requests`, `BeautifulSoup`, or `Selenium`. If crawling fails for a site, use RSS feeds or the site's search API if available.

Minimum Metadata

For each document, store:

- `title`, `body`, `url`, `date`, `language` *(required)*
- `tokens` (count), `word_embeddings` *(optional but recommended)*
- `named_entities` *(optional)*

Tools You May Use (suggested but not limited to)

- **Crawling:** Requests, BeautifulSoup, Selenium, Scrapy
- **Indexing:** Elasticsearch, Lucene, Whoosh, or a simple inverted index in Python
- **NLP:** spaCy, Stanza, HuggingFace Transformers (for tokenization, NER, embeddings)
- **Language Detection:** `langdetect`, `textblob`, or `fasttext`

Purpose

- Gain exposure to real-world, messy data
- Understand indexing fundamentals
- Create a foundation for multilingual search

Module B — Query Processing & Cross-Lingual Handling (Core)

Tasks

Implement a query-processing pipeline with the following steps:

1. **Language Detection**
   Identify whether the query is in Bangla or English
2. **Normalization**
   Lowercase, remove extra whitespace. Stopword removal is optional.
3. **Query Conversion/Translation** (Required)
   Simplest idea would be to translate the query to another language to retrieve document from that language. However, you may look for any other advanced way to perform the retrieval without translating.
   To translate query from one language to the other, you may use any tools that you see fit. Use of paid tools are discouraged.
4. **Query Expansion** (Recommended)
   Add synonyms or morphological variants. For Bangla, you may expand with related root words or transliterations.
5. **Named-Entity Mapping** (Recommended)
   Extract named entities (NE) from the query and map them across languages (e.g., "Bangladesh" → "বাংলাদেশ"). This is important for proper noun matching.

Purpose

- Understand why direct translation is insufficient and/or even less ideal
- Explore how cross-lingual mismatch arises in real systems
- Practice robust error handling

Module C — Retrieval Models (Core)

Implement and show comparison if necessary. You may use any/all of them; but, show proper justification for your choices.

Model 1: Lexical Retrieval (BM25 or TF-IDF)

- Implement or use off-the-shelf BM25
- Compare with TF-IDF on your dataset
- Analyze failure cases (why does it fail for synonyms, paraphrases, cross-script terms?)

Model 2: Fuzzy/Transliteration Matching

- Use edit distance (Levenshtein), Jaccard similarity, or character n-grams
- For Bangla–English queries, include transliteration matching (e.g., matching "Bangladesh" with "বাংলাদেশ")
- Tools: `difflib`, `fuzzywuzzy`, `jellyfish`, or custom character n-gram matching

Model 3: Semantic Matching (Mandatory)

- Use a multilingual embedding model to encode queries and documents
- **Recommended models:**
  - **LaBSE** (Language-agnostic BERT Sentence Embeddings) – multilingual, good for many languages
  - **XLM-R** (Cross-lingual RoBERTa) – state-of-the-art, supports 100+ languages
  - **mBERT** or **mT5** – older but stable options
  - **multilingual SBERT** – from `sentence-transformers` library; easy to use
- Measure similarity using cosine distance
- Compare results with lexical models

Model 4 (Optional): Hybrid Ranking

- Combine scores from multiple models (e.g., 0.3 × BM25 + 0.5 × embedding similarity + 0.2 × fuzzy match)
- Experiment with weighted fusion

Purpose
Understand how different IR methods contribute to retrieval and when semantics dominate lexicon.

Module D — Ranking, Scoring, & Evaluation (Core)

1. Ranking & Scoring

- **Implement a ranking function** that outputs a sorted list of top-K documents for each query
- **Matching Score (0–1 scale):** For each document in the result set, output a confidence score indicating how relevant the result likely is
  - Example: If top result has embedding similarity 0.92, it gets matching score 0.92
  - Normalize all model scores to [0, 1] before combining
- **Low-confidence warning:** If the top-ranked document's matching score is below a threshold (e.g., 0.20), display:

```
⚠️ Warning: Retrieved results may not be relevant. Matching confidence
is low (score: 0.15).
```

```
Consider rephrasing your query or checking translation quality.
```

This prevents misleading outputs to users.

2. Query Execution Time

Report for each query:

- **Total retrieval time** (in milliseconds)
- **Breakdown (optional):** Translation time, embedding computation time, ranking time
- This helps you understand performance trade-offs (e.g., semantic models are slower but more accurate)

3. Evaluation Metrics (Mandatory)

Firstly, you should compare your retrieval results with the ones from popular classical search engines, such as Google, Bing, Duck Duck Go, etc. You may even consider the AI-powered (search) engines in this matter.

Secondly, you must evaluate your system using standard IR metrics:

Metric	Definition	Target for Your System
Precision@10	# relevant docs in top 10 / 10	≥ 0.6 (at least 6 relevant in top 10)
Recall@50	# relevant docs retrieved / total relevant docs	≥ 0.5
nDCG@10	Discounted cumulative gain, penalizing lower-ranked relevance	≥ 0.5
MRR	1 / (rank of first relevant doc); average over queries	≥ 0.4

Relevance Labeling:

- Manually label at least **5–10 queries** as relevant or not relevant for each document
- Groups may add extra queries for depth
- **Tip:** Use a simple CSV with columns: `query`, `doc_url`, `language`, `relevant (yes/no)`, `annotator`

4. Error Analysis (Detailed)

Analyze at least **retrieval failures** with specific examples:

1. **Translation Failures**
   Example: Query "চেয়ার" (chair) mistranslated to "Chairman" → retrieved wrong documents

2. **Named Entity Mismatch**
   Example: Query mentions "ঢাকা" (Dhaka) but documents use "Dhaka" in English; NER missed the match
3. **Semantic vs. Lexical Wins**
   Example: Query "শিক্ষা" (education), BM25 returns 0 results, but embedding model retrieves relevant "স্কুল" (school) document
4. **Cross-Script Ambiguity**
   Example: "Bangladesh" could be transliterated as "বাংলাদেশ" or "Bangla Desh" (two words); which does your system handle?
5. **Code-Switching**
   Example: Query mixes Bangla and English words; does the system handle this?

Format: Include at least one detailed case study per category (screenshot, query text, retrieved document, analysis of why it failed/succeeded).

Module E — Report, Literature Review & Innovation (Core)

Your final report must include:

1. *BRIEF* Literature Review (3–5 papers)

- Identify **3–5 key CLIR / multilingual IR papers** (except the ones listed below) and summarize each in 100–200 words
- Include: authors, publication year, main technique, why it matters, and how it relates to your system
- **Suggested starting papers:**
  - Survey: *Cross-Lingual Information Retrieval* by Ballesteros & Croft (ACL 2001) or newer survey
  - Semantic CLIR: *Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond* (2019)
  - Practical: *XLM-RobERTa: Unsupervised Cross-lingual Representation Learning at Scale* (ICLR 2020)

2. Methodology & Tools

Clear description of:

- How you constructed the dataset (which sites, preprocessing)
- Tools you used
- Your indexing strategy and metadata
- Query processing pipeline (translation, expansion, NE mapping)
- Retrieval models implemented (with code snippets or pseudocode)
- Ranking and scoring approach

3. Results & Analysis

- **Tables:** Compare Precision@10, Recall@50, nDCG, MRR across your models
- **Graphs:** Bar charts or line plots showing model performance; confusion matrices for error types
- **Interpretation:** Which model works best? Why? When does BM25 outperform embeddings and vice versa?

4. AI Usage Policy

You are allowed to use AI tools (ChatGPT, Claude, Copilot, etc.) under strict conditions:

- **Disclosure:** All AI-generated content must be accompanied by the **exact prompt you used**, included verbatim in an appendix titled "AI Tool Usage Log"
- **Verification:** You must verify the correctness of any AI output before including it. Inaccurate, hallucinated, or non-existent information will result in mark deduction.
- **Correction:** If AI gives incorrect output, document it:
  - What was the prompt?
  - What did the AI produce?
  - Why was it wrong?
  - What is the correct version?
- **Code Generation:** Any code generated by AI must be understood by all group members. You may be asked to explain or modify it orally.
- **Example log entry:**

```
Prompt: "Write Python code to compute nDCG@10 for a list of relevance scores"
Tool: ChatGPT (Nov 2024)
Output: [code snippet]
Verification: Tested against manual calculation; correct for k=10.
Included in report: Yes (Section 4.2, with inline comments)
```

5. Innovation Component

Propose one extension, even if not fully implemented, such as:

- **Cross-lingual Topic Modeling:** Map topics across languages and use them to improve ranking
- **Query-Time Code-Switching:** Detect and handle queries that mix Bangla and English
- **Bias Detection:** Analyze whether your system retrieves documents fairly across political viewpoints
- **Graph-Based Concept Linking:** Build a knowledge graph of named entities across languages; use it for query expansion

- **Domain Adaptation:** Fine-tune your embedding model on domain-specific news data
- **Temporal Drift:** Model how query relevance changes with document date

This section encourages future research rather than punishing lower-performing groups.

4. Grading Rubric (Total 100%)

Component	Marks	Description
Dataset Construction	12	Quality and diversity of documents; correctness of metadata; handling of real-world messiness
Indexing & Preprocessing	8	Clean pipeline; named entity extraction (optional); robustness to encoding issues
Query Processing & CLIR	15	Effective translation; query expansion; NE mapping; error handling
Retrieval Models	18	Correctness of implementations; fair comparison; at least 3 models; analysis of trade-offs
Ranking, Scoring & Evaluation	15	Matching score and caution messages; query execution time; correct metric computation; at least 15 labeled queries
Error Analysis	10	Depth and clarity; at least 4 categories with specific examples; case studies
Report + Literature Review	15	Accuracy and relevance of papers; clarity of methodology; quality of graphs/tables; AI usage transparency
Innovation Component	7	Creativity; feasibility; connection to literature; research insight

5. What an Ideal Submission Looks Like

An exemplary submission includes:

Balanced multilingual dataset
At least 2.5k docs per language; clean metadata; diverse topics (politics, sports, tech, health, etc.)

Multiple retrieval models with fair comparison
Lexical (BM25) vs. fuzzy vs. semantic (embeddings) vs. optional hybrid; analysis of when each excels

Robust query processing
Language detection → translation → expansion → NE mapping, with error logging and fallback strategies

Matching score + caution outputs
Every result ranked with confidence [0–1]; low-confidence warnings prevent misleading results

Query execution time reporting
Breakdown showing which components (translation, embedding, ranking) consume time; justification of trade-offs

Strong evaluation & visualizations
Precision/nDCG/MRR graphs across models; confusion matrices for error categories; at least 20 labeled queries

Error analysis
Clear examples of translation drift, NE mismatch, semantic wins, cross-script issues; at least one case study per category

Responsible AI usage
All AI-generated content tagged with prompts in appendix; inaccuracies identified and corrected

Innovative insights
A research-level idea building on assignment experience and connected to CLIR literature

6. Dataset: Recommended Bangla & English News Sites

Use these sites as starting points for crawling. Choose at least **5 from each language group** (we recommend 5 Bangla + 5 English per group to keep crawling manageable).

Bangla Language News Sites (বাংলা সংবাদমাধ্যম)

#	Site Name	URL
1	Prothom Alo	prothomalo.com
2	BD News 24	bangla.bdnews24.com
3	Kaler Kantho	kalerkantho.com
4	Bangla Tribune	banglatribune.com
5	Dhaka Post	dhakapost.com

English Language News Sites (from Bangladesh)

#	Site Name	URL
1	The Daily Star	thedailystar.net
2	New Age	newagebd.net

#	Site Name	URL
3	The New Nation	dailynewnation.com
4	Daily Sun	daily-sun.com
5	DhakaTribune	dhakatribune.com

Tips for Crawling:

- Start with the site's `/latest`, `/today`, or `/news` section
- Most news sites have RSS feeds (check `/feed`, `/rss`); this is often faster than scraping HTML
- Respect `robots.txt` and crawl at a reasonable rate (add delays between requests)
- Store the raw HTML/text; preprocess later
- Log any errors (404, timeouts, encoding issues) for your report

7. Suggested Tools & Libraries (but not limited to)

Python Libraries

Purpose	Recommended Tools	Notes
Crawling	Requests, BeautifulSoup, Selenium	BeautifulSoup is easiest for parsing HTML
Indexing	Elasticsearch, Whoosh	Or implement simple inverted index in Python
NLP	spaCy, Stanza, HuggingFace	For tokenization, lemmatization, NER, embeddings
Embeddings	`sentence-transformers`, `transformers`	Pre-trained multilingual models (LaBSE, XLM-R, mBERT)
Translation	`transformers` (OPUS-MT), Google Translate API	Free tier available for Google Translate
Fuzzy Matching	`fuzzywuzzy`, `jellyfish`, `difflib`	Character-level similarity
Evaluation	`scikit-learn` (metrics), custom code	Compute Precision, nDCG, MRR
Visualization	`matplotlib`, `seaborn`, `plotly`	For graphs and tables

8. Timeline & Milestones (~3 Weeks)

Week	Milestone	Targets
Week 1	Dataset crawling & indexing	≥ 2,500 documents per language; clean metadata
Week 1-2	Models & retrieval	BM25 + fuzzy + embeddings working; initial evaluation
Week 2–3	Evaluation & error analysis	Labeled queries (15–20); results tables/graphs
Week 3	Report & polish	Full report with literature review, AI usage log, innovation section

9. Submission & Deliverables

Your submission must include:

1. **Code Repository** (GitHub or ZIP)
   - Well-commented code for crawling, indexing, retrieval, evaluation
   - `README.md` with setup instructions and example usage
   - Your labeled query set (CSV)
2. **Dataset** (or metadata + download links)
   - Processed documents (JSON or CSV) with metadata
   - If file is large (>100 MB), provide download link or script to recreate
3. **Final Report** (PDF)
   - Sections: Motivation, Methodology, Results, Error Analysis, Literature Review, AI Usage Log, Innovation, References
   - Graphs, tables, and case studies embedded
4. **Evaluation Results** (CSV/JSON)
   - Query-by-query results: query, top-k retrieved docs, matching scores, metrics
   - Summary table of model performance (Precision@10, Recall@50, nDCG, MRR)

10. Academic Integrity & AI Usage

- **All work must be original and attributed.** Reused code from tutorials/papers must be cited.
- **AI tool usage must be fully disclosed.** Include prompts and outputs in an appendix.
- **All group members must understand and be able to explain all code.** You may be asked to give a brief oral walkthrough.
- **Fabricated datasets, results, or citations are not permitted.** Penalties apply if detected.

Good luck! Despite how challenging this project may seem, hopefully, you find it rewarding as well. You'll build a real system, engage with research, and learn why CLIR is such a rich and important problem.

Further discussion on this is mostly welcome!