

# LexiSemIR: A Two-Stage Re-ranking Framework with BM25 and Zero-Shot Bi-Encoder

Swati Gupta<sup>1,†</sup>, Tanusree Nath<sup>1\*,†</sup>, Vedika Gupta<sup>2</sup> and Manjari Gupta<sup>1</sup>

<sup>1</sup>Department of Computer Science, Banaras Hindu University, Varanasi, India

<sup>2</sup>Jindal Global Business School, O.P. Jindal Global University, Sonapat, India

## Abstract

Most standard Information Retrieval (IR) models primarily rely on keyword matching, which can be inadequate when a deeper contextual understanding is required. In such cases, it becomes essential to capture both lexical and semantic relationships between query–document pairs. To address this limitation, our team CodeWeavers proposes LexiSemIR, a two-stage re-ranking-based model developed for the CMIR-2025 (Code-Mixed Information Retrieval) shared task on Bengali-English code-mixed text. In the first stage, the top k documents are retrieved using a lexical bag-of-words model (BM25). These are then re-ranked in the second stage using a zero-shot bi-encoder, which computes semantic similarity between query and document embeddings. The proposed approach balances simplicity and performance, while minimizing trainable parameters due to its zero-shot design. LexiSemIR secured 3rd place in the CMIR-2025 shared task, achieving MAP = 0.1546 and P@5 = 0.38, thereby outperforming the BM25 baseline in early precision. The results highlight the model’s ability to effectively combine lexical and semantic retrieval strategies for robust performance in code-mixed IR settings.

## Keywords

Code-mixed language, Information retrieval, BM25, Bi-Encoder

## 1. Introduction

Information Retrieval (IR) is broadly defined as “finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)” [1]. In modern times, IR finds its applications in a variety of tasks, including internet browsing, question answering systems, personal assistants, chat-bots, and digital libraries [2]. There has also been a surge of user-generated content in code-mixed languages in recent years, which has further complicated the task of information retrieval [3, 4]. Code-mixing refers to the act of “mixing two or more languages in a single discourse” [5]. While code-mixing may help the model understand multilingual similarities, it becomes susceptible to hurting the retrieval effectiveness [6]. Most traditional IR systems depend on lexical matching-based algorithms (BM25, Heimstra-LM, etc.), which may prove to be ineffective in many instances where the same word has different meanings or where different words have the same meaning [7]. In such cases, there is a need to have a deeper semantic understanding of the queries and documents. To eliminate this drawback, semantic-matching-based algorithms have been developed (vector space model, neural networks, etc.) that focus on the meanings of tokens. To further improve the efficiency of a single IR algorithm, two-stage retrieval systems have been developed. Since the 1990s, two-stage retrieval systems have undergone constant improvement with the advent of new methods and technologies. One of the recent studies has used a two-stage retrieval system using an adapted BM25 and a neural ranking model [8].

Instead of the traditional two-stage retrieval system, [9] introduced a serverless three-stage retrieval system with BM25, monoBERT [10] and duoBERT. Another study by [11] also developed a two-stage

---

Forum for Information Retrieval Evaluation, December 17-20, 2025, India

\*Corresponding author.

†These authors contributed equally.

✉ guptaswati2080@gmail.com (S. Gupta); tanusreenath32@gmail.com (T. Nath); vedika.nit@gmail.com (V. Gupta); manjari@bhu.ac.in (M. Gupta)

ORCID 0009-0008-3530-8191 (S. Gupta); 0000-0001-8512-1474 (T. Nath); 0000-0002-8109-498X (V. Gupta); 0000-0003-1939-5383 (M. Gupta)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

retrieval system using BM25 and monoBERT encoder. Following these works, our study also aims to build a two-stage retrieval model. However, in place of monoBERT in the second stage, we employ MPNet in a zero-shot setting. MPNet makes a good candidate as a two stage re-ranker because documents are often encoded using sentence embeddings [12]. In addition to that, using zero-shot mode to transfer knowledge from English to other languages is a popular approach [13]. This not only makes the model faster and scalable, but also allows for independent encoding of queries and documents via the bi-encoder architecture instead of the traditional cross-encoder. A similar study by [14] also employed BM25 and a SentenceTransformer-based encoder to build a two-stage retrieval system. However, this study used fine-tuning to fit the SentenceTransformer on their data. Our proposed approach uses SentenceTransformer in a zero-shot setting, which makes it faster and parameter-efficient, while maintaining comparable performance in the given task.

The overall contributions of our work are summarized below:

- We propose LexiSemIR, a two-stage re-ranking-based retrieval framework, combining a lexical BM25 ranker with a zero-shot semantic bi-encoder to capture both lexical and semantic meaning of query document pairs. This model elegantly balances performance with simplicity and parameter efficiency. The proposed model ranked 3rd in the CMIR-2025 shared task on Bengali-English code-mixed text, with a MAP score of 0.1546, NDCG score of 0.2767, P@5 of 0.38, P@10 of 0.2833.
- We conduct a detailed error analysis of the proposed model and provide valuable insights into its underlying biases and sources of error. In addition, the query-specific variations are highlighted, which directly affect the retrieval quality. This insight provides actionable guidance for query-expansion techniques in future work.
- A sensitivity analysis is performed across multiple *top\_k* values that demonstrates how retrieval depth influences performance stability across metrics.

## 2. State of the Art

Code-mixed information retrieval research has undergone significant improvement in the recent years. One of the major breakthroughs in this area is the creation of new corpora in low-resource languages. Early studies like [15] introduced such a corpus in Hindi-English code-mixed social media data. Another study by [16] also introduced a track for mixed-script information retrieval in Hindi at FIRE-2016. Bengali, being such a language also started gaining attention from the research community. A study by [17] developed a Bengali information retrieval system by introducing a Bengali text corpora, using advanced preprocessing techniques and applying TF-IDF with cosine similarity to retrieve relevant answers to queries. Another study by [18] experimented with prompt engineering and mathematical modeling with GPT-3.5. Some studies like [19] and [20] combined multiple languages (Hindi, Bengali and Marathi). In the study by [19], several indexing and retrieval strategies were evaluated using various techniques like Divergence from Randomness variants, Okapi BM25, TF-IDF, and statistical language models.

Some of the early works also focused on cross-lingual information retrieval. A study [21] developed a framework to retrieve English documents using Hindi and Bengali queries using machine translation and automatic query generation. A similar study by [22] assembled a system to retrieve English documents using Bengali, Hindi and Telugu queries. To achieve this, they used a combination of bilingual dictionaries, suffix-stripping stemmers, transliteration and TF-IDF ranking.

## 3. Methodology

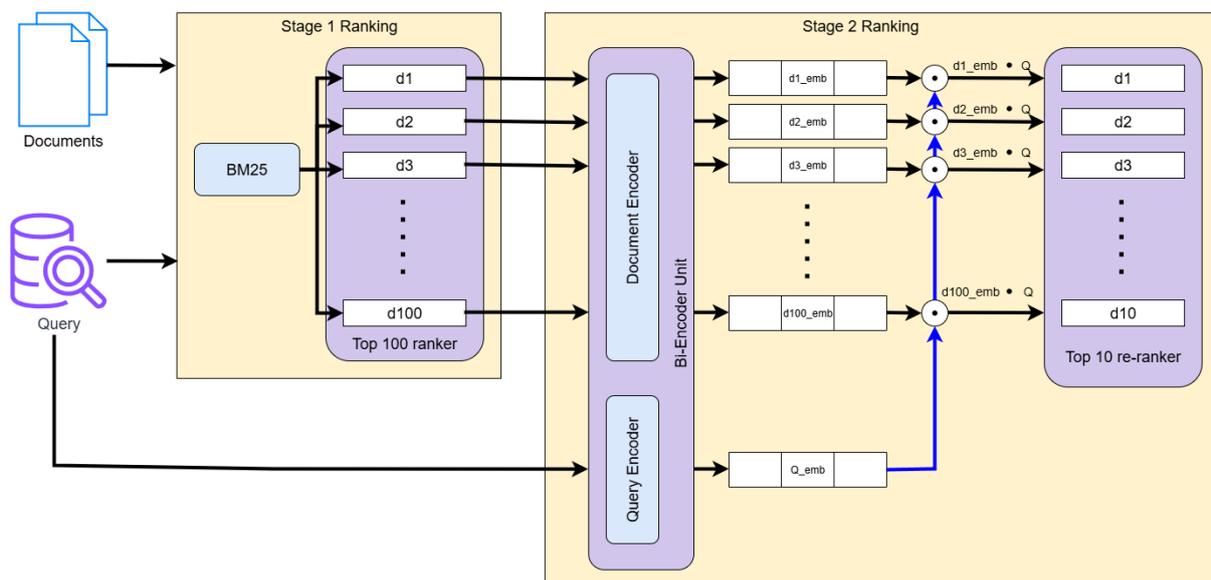
This section presents the detailed methodology of the proposed approach. The goal of this work is to develop a Code-Mixed Information Retrieval (CMIR) system, focusing on code-mixed Bengali-English queries and documents. Standard baseline information retrieval models mainly rely on lexicon-based keyword matching (TF-IDF, BM25, PL2, InL2, Hiemstra-LM), completely discounting the semantic

**Table 1**

Dataset statistics for the IR task.

Collection	#Documents	Avg. Doc Length	#Queries	Avg. Query Length	#Relevant Judgments
Full Corpus	107900	12.64	–	–	–
Train Queries	–	–	20	40.20	378
Test Queries	–	–	30	41.10	Not Available

meaning of natural text. To overcome this drawback, the LexiSemIR (Lexical + Semantic Information Retrieval) model is proposed, which effectively balances lexical and semantic matching of query-document pairs. By employing a two-step retrieval process and re-ranking technique, the LexiSemIR model ensures that both surface-level keyword overlap and deeper contextual understanding between query-document pairs are captured. An overview of the model is given in Figure 1. The detailed



**Figure 1:** An overview of the proposed LexiSemIR model. Stage 1 retrieves the top 100 documents using the BM25 ranker. These documents and their corresponding query are further encoded by a zero-shot bi-encoder in stage 2. A cosine similarity between the query and document embeddings is computed to retrieve and rank the top 10 documents.

workflow of the model is described in the succeeding sections.

### 3.1. Dataset and Setup

The dataset consists of a set of queries  $Q$  and documents  $D$ . Both the queries and the documents are provided in code-mixed Bengali. For the training phase, the relevance judgements for the queries are available in a separate file, where each query is mapped to one or more relevant documents, with a binary relevance score. Table 1 summarizes the dataset statistics.

### 3.2. Experimental Design

The retrieval is performed in two stages. In the first stage of the retrieval, the BM25 ranker is used to retrieve the top 100 matching documents [23], followed by a bi-encoder-based re-ranking to retrieve the top 10 amongst those.

**BM25.** It is a classic probabilistic bag-of-words model, used as the first-stage retriever. Given a query  $Q$  with tokens  $\{q_1, q_2, \dots, q_m\}$ , and a set of documents  $D = \{d_1, d_2, \dots, d_N\}$ , where  $N$  is the total

number of documents in the corpus, BM25 calculates the relevance score of document  $d_j$  with respect to query  $Q$  as:

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

Here, the  $k_1$  and  $b$  are two hyperparameters.  $k_1$  controls the term-frequency saturation while  $b$  controls document length normalization.  $f(q_i, d_j)$  is the frequency of term  $q_i$  in document  $d_j$ ,  $\text{avgdl}$  is the average document length in the collection and IDF stands for the Inverse-Document Frequency of a term  $q_i$ , It is calculated as:

$$\text{IDF}(q_i) = \ln \left( \frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} + 1 \right)$$

In the above equation,  $n(q_i)$  refers to the number of documents containing the term  $q_i$ .

**Bi-encoder Unit:** The documents retrieved in the first stage, along with the query, are passed to a pre-trained sentence transformer-based bi-encoder unit. The bi-encoder unit consists of the *all-mpnet-base-v2* Sentence-Transformer (based on MPNet [24]), which parallelly processes the query and the document. Given the initial query  $Q$ , let  $C_Q = \{d_1, d_2, \dots, d_{100}\}$  be the top 100 documents retrieved in the first stage. If  $f_\theta : \text{text} \rightarrow \mathbb{R}^{768}$  denotes the bi-encoder function that maps a piece of text to a dense vector, then the query embeddings  $\text{emb}_Q$  and the document embedding  $\text{emb}_{d_j}$  are calculated as:  $\text{emb}_Q = f_\theta(Q) \in \mathbb{R}^{768}$  and  $\text{emb}_{d_i} = f_\theta(d_j) \in \mathbb{R}^{768}$  respectively. The bi-encoder unit simultaneously encodes each document and the query to produce dense vectors that capture their semantic meaning. The final relevance score  $s(Q, d_j)$  of each document with respect to the query is calculated by computing the cosine-similarity between their respective embedding vectors. It is calculated as follows:

$$s(Q, d_j) = \cos(\text{emb}_Q, \text{emb}_{d_j}) = \frac{\text{emb}_Q^\top \text{emb}_{d_j}}{\|\text{emb}_Q\|_2 \|\text{emb}_{d_j}\|_2}$$

This score is used to retrieve and align the final top 10 relevant documents with respect to query  $Q$ .

### 3.3. Hyperparameters

Hyperparameters are a set of configuration variables that control the model’s performance. By configuring the hyperparameter settings, a model can be tuned for optimal performance. Table 2 summarizes the different hyperparameters used in LexiSemIR model along with their descriptions.

**Table 2**  
Hyperparameters used in the two-stage retrieval system.

Stage	Hyperparameter	Value	Description
BM25 (Stage 1)	$top\_k$	100	Top $k$ documents retrieved per query in stage 1
	$k_1$	1.2	Controls TF (Term Frequency) saturation in BM25 retrieval
	$b$	0.75	Controls document length normalization
Bi-Encoder (Stage 2)	No Tunable Hyperparameters		

Since the bi-encoder in stage 2 is used in a zero-shot setting, there are no trainable parameters, and consequently, no learnable hyperparameters in this stage.

### 3.4. Training and Testing

Both training and testing are performed in the same document collection. However the set of queries used in each phase is different. For training, 20 queries are provided, along with a query relevance judgement file that maps each query to at least one document. The LexiSemIR model involves limited

tuning only in Stage 1 (BM25). Specifically, we adjusted BM25 hyperparameters using the training set queries and relevance judgements. Stage 2 (the bi-encoder) was employed in a purely zero-shot setting without any additional fine-tuning. Thus, while the overall framework is evaluated on both training and test queries, only Stage 1 undergoes hyperparameter tuning. For testing, 30 queries are given. The trained model is used to retrieve relevant documents for the test queries, the results of which are further evaluated by the organizers to publish the final evaluation metrics.

## 4. Results

Our team CodeWeavers, submitted two runs in the CMIR-2025 shared task. Run 1 corresponds to an enhanced version of Hiemstra-LM with advanced query and document preprocessing. Run 2 corresponds to the LexiSemIR model, which secured 3rd rank in the overall competition. The test metrics were released by organizers, as we did not have direct access to relevance judgments. The reported evaluation metrics on the test queries for both runs are listed in Table 3.

**Table 3**

Performance of CodeWeavers runs on test queries across different metrics.

Run	MAP score	NDCG score	P@5 Score	P@10 Score
Run 1	0.15627	0.341038	0.28	0.23
<b>Run 2 (LexiSemIR)</b>	<b>0.154625</b>	<b>0.276684</b>	<b>0.38</b>	<b>0.283333</b>

Since the relevant judgements for the test queries were not available, further analysis of the model based on errors and sensitivity are conducted on the training set. Table 4 shows the performance metrics of the model on training queries.

**Table 4**

Performance of CodeWeavers runs on train queries across different metrics.

Run	MAP score	NDCG score	P@5 Score	P@10 Score
Run 1	0.4162	0.4711	0.39	0.285
<b>Run 2 (LexiSemIR)</b>	<b>0.644196</b>	<b>0.411855</b>	<b>0.469999</b>	<b>0.335</b>

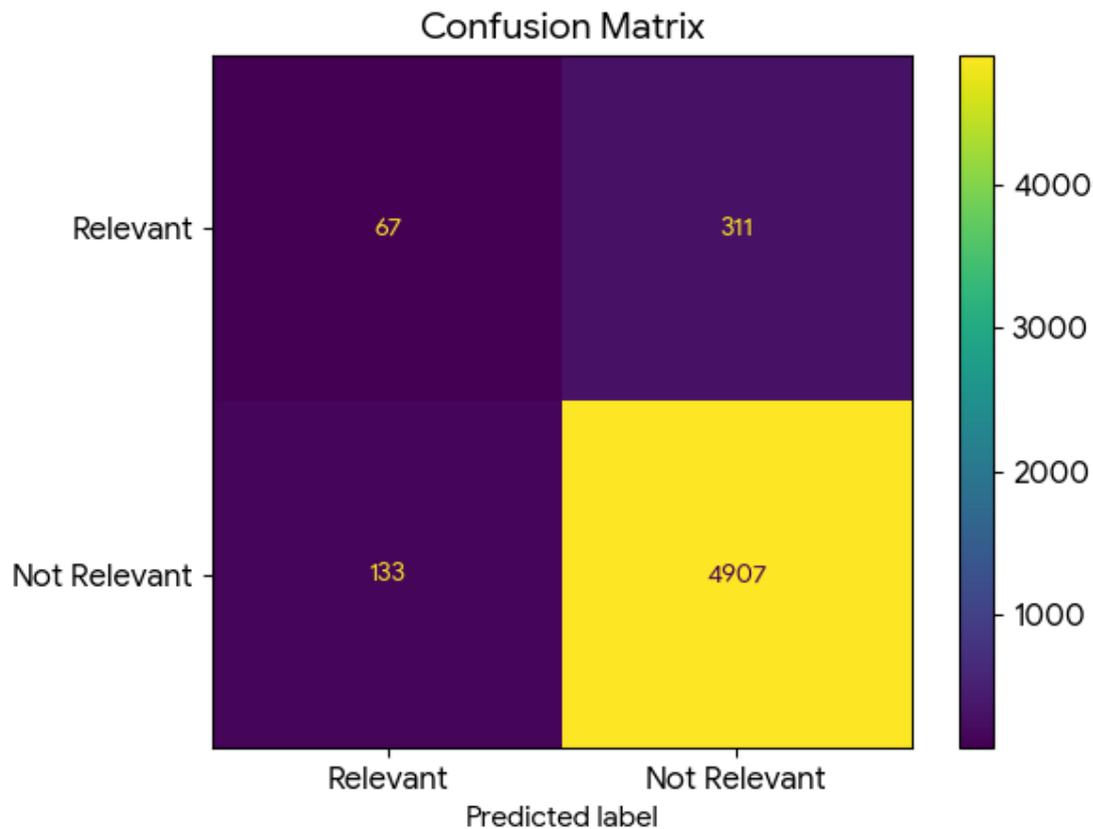
### 4.1. Error Analysis

This section talks about the error analysis of the proposed model in the training phase. Figure 2 shows the confusion matrix corresponding to the proposed LexiSemIR model. From the figure, it is evident that a total of 67 out of 378 relevant documents were retrieved by the model. It can also be observed that the false-negative (FN) rate is higher than the false-positive (FP) rate. The FN rate shows that a considerable portion of relevant documents were not ranked in the top ten, and allows for an area of significant improvement. The FP rate is also quite high, signalling that the model is also confident about placing irrelevant documents in the top ten.

Figure 3 shows the rank distribution of the 67 retrieved documents. A positive trend can be observed in it, where the overall number of retrieved documents decreases as the rank goes down, showing that the model successfully places the most relevant documents in the top ranks.

For a more detailed error analysis, an evaluation of the retrieved documents based on each query is presented in Table 5. It is observed from the table that queries 19 and 1 retrieved the highest number of relevant documents, followed by queries 13 and 14. The worst performing queries are 2 and 12, with 0 retrieved documents. Queries 21 and 25 have the highest number of false negatives, while queries 2 and 12 have the highest number of false positives.

A closer inspection reveals that the poor performance for queries 2 and 12 may be attributed to their predominant use of Bengali tokens with minimal English code-mixing. Since all-mpnet-base-v2



**Figure 2:** Confusion matrix of the LexiSemIR model on training data

is primarily trained on English text, it lacks robust multilingual alignment capabilities. Consequently, Bengali-dominant inputs fall outside its pretraining distribution, leading to weaker semantic representations. The absence of sufficient English context limits the model’s ability to generate meaningful embeddings, thereby reducing retrieval effectiveness for such queries.

Overall, this analysis shows that the model’s performance variation is highly query-specific. Addressing this issue by improving the handling of queries with limited code-mixing could significantly enhance the model’s ability to mitigate false negatives and improve generalization in future iterations.

## 5. Discussion

The proposed model secured third place in the overall CMIR-2025 competition. This achievement indicates the effectiveness of the model. Although the model performed well on the dataset, it is important to note that the query set sizes for the train and the test phases were relatively small (20 and 30, respectively). This design makes the dataset highly scalable, making space for better results when applied to larger datasets. While the model showed comparable performance with the hyperparameter settings mentioned in Section 3.3, an additional analysis based on the *top\_k* hyperparameter has been performed to assess the model’s sensitivity towards it. The other two hyperparameters ( $\{k_1\}$  and  $\{b\}$ ) are kept at their default standard values. Similar to the error analysis, the sensitivity analysis is also performed on the train queries due to the unavailability of the relevance judgements for test queries.

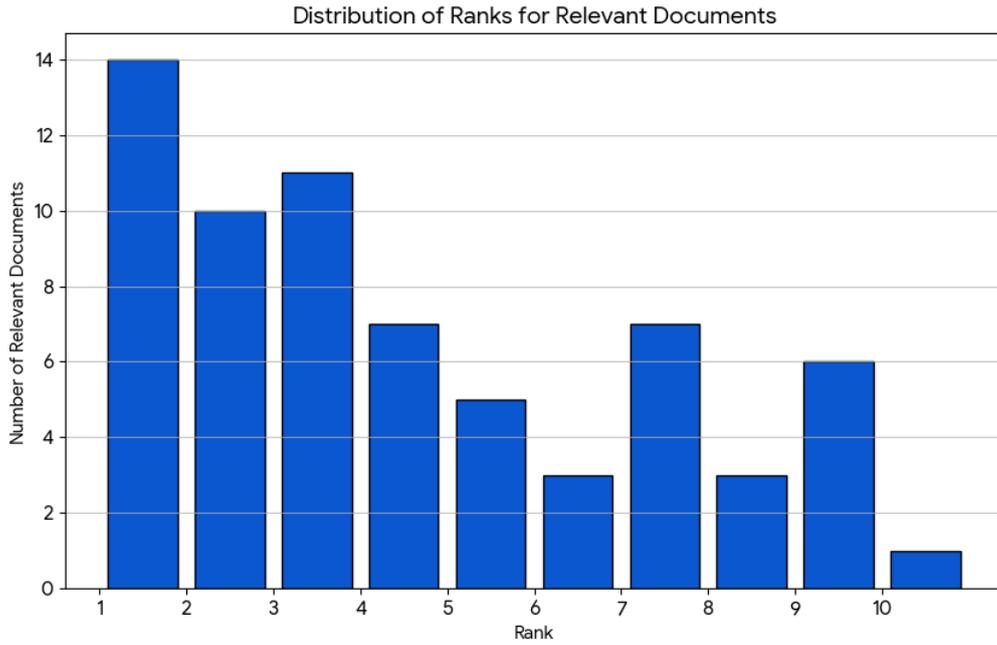


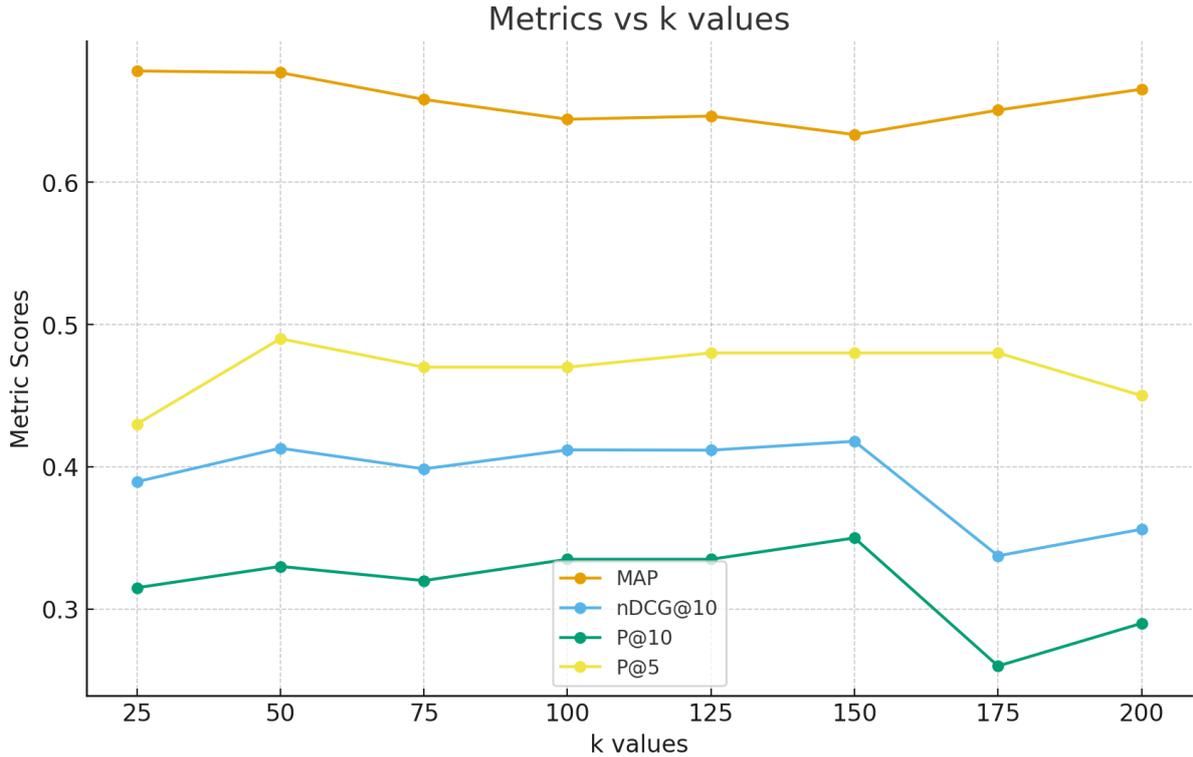
Figure 3: Rank distribution of the retrieved documents

Table 5  
Per-Query Performance Analysis on Training Data

qid	Total Relevant Docs	Relevant Docs Found	False Positives	False Negatives
19	21	7	3	14
1	37	7	3	30
13	24	5	5	19
14	20	5	4	15
23	31	4	5	27
25	35	4	2	31
15	11	4	6	7
24	15	4	6	11
11	11	3	7	8
3	20	3	7	17
17	20	3	7	17
20	19	3	7	16
7	10	3	6	7
22	14	3	7	11
4	11	3	7	8
5	11	3	7	8
18	8	2	6	6
21	33	1	9	32
12	12	0	10	12
2	15	0	10	15

### 5.1. Sensitivity Analysis

This section talks about the sensitivity of the model on changing the  $top_k$  value in the first stage of retrieval. After initially submitting the model by tuning it with  $top_k = 100$ , it was later evaluated with seven other values of  $top_k$  to analyse its sensitivity, and the corresponding performance metrics on the train data were observed. The graph corresponding to the observed values of performance metrics



**Figure 4:** Training evaluation metrics for different values of  $k$  in the first stage of retrieval

with respect to different values of  $top\_k$  is shown in Figure 4.

From the figure, it can be deciphered that the overall best value of  $top\_k$  is 150, with the highest value in both NDCG@10 and P@10.  $top\_k = 25$  had the highest MAP score, while  $top\_k = 50$  had the highest value for P@5.

It can also be observed that the MAP score is the least sensitive to the change in  $top\_k$  value. On the other hand, P@5 shows drastic sensitivity to changing the  $top\_k$  value from 25 to 50, where it rises from being the minimum to the maximum. After that, it remains fairly stable before dipping again at 200. For both NDCG@10 and P@10, a gradual but uneven upward trend is observed as the value of  $top\_k$  increases from 25 to 150. However, at 175, the values of these two metrics drop drastically.

## 6. Conclusion

This study proposed a two-stage re-ranking-based framework as a part of the CMIR-2025 shared task in code-mixed information retrieval [25], [26]. The model uses BM25 as the first-stage ranker to retrieve the top 100 documents. These documents are then passed to the second stage, where a zero-shot bi-encoder computes the query and document embeddings. A cosine similarity score between them provides the relevance measure, which in turn is used to retrieve the top 10 final documents.

The proposed LexiSemIR model, which represents the second run submitted for the task, secured the third rank in the competition, demonstrating competitive performance. Although Run 1 achieved slightly higher NDCG and MAP scores, Run 2 (LexiSemIR) showed improvements in P@5 and P@10, indicating enhanced precision at higher ranks through some comparable trade-offs.

This study also provides a detailed analysis of the model on the training data. A comprehensive error analysis was conducted on the training queries, offering valuable insights into the model's performance, biases, and potential areas for improvement. In addition, a sensitivity analysis on the training queries highlighted the performance trend with varying values of the

Future work could focus on more advanced preprocessing of code-mixed data and the incorporation

of language-specific encoders to achieve a more accurate semantic understanding.

Future scope of this study could involve more advanced preprocessing of code-mixed data and incorporating language-specific encoders to achieve more accurate semantic understanding.

## Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

## References

- [1] H. Schütze, C. D. Manning, P. Raghavan, Introduction to information retrieval, volume 39, Cambridge University Press Cambridge, 2008.
- [2] K. A. Hambarde, H. Proenca, Information retrieval: recent advances and beyond, *IEEE Access* 11 (2023) 76581–76604.
- [3] S. Chanda, S. Pal, The effect of stopword removal on information retrieval for code-mixed data obtained via social media, *SN Comput. Sci.* 4 (2023).
- [4] S. Chanda, S. Pal, Overview of the shared task on code-mixed information retrieval from social media data, in: *FIRE 2024 Working Notes*, CEUR Workshop Proceedings, 2024, p. 124–128. URL: <https://ceur-ws.org/Vol-4054/T2-1.pdf>.
- [5] S. Chanda, S. Pal, Overview of the shared task on code-mixed information retrieval from social media data, in: *Proceedings of the 16th Annual Meeting of the Forum for Information Retrieval Evaluation*, Association for Computing Machinery, 2025, pp. 29–31.
- [6] J. Do, J. Lee, S.-w. Hwang, Contrastivemix: overcoming code-mixing dilemma in cross-lingual transfer for information retrieval, in: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, 2024, pp. 197–204.
- [7] W. B. Croft, D. Metzler, T. Strohman, et al., *Search engines: Information retrieval in practice*, volume 520, Addison-Wesley Reading, 2010.
- [8] A. Kaushish, A. Vijayvargiya, K. Rawat, M. Kumar, V. Jain, Optimized spoken query cross-lingual document retrieval using bm25 and neural re-ranking with adamw, in: *2025 4th OPJU International Technology Conference (OTCON) on Smart Computing for Innovation and Advancement in Industry 5.0*, IEEE, 2025, pp. 1–7.
- [9] R. Nogueira, W. Yang, K. Cho, J. Lin, Multi-stage document ranking with bert, *arXiv preprint arXiv:1910.14424* (2019).
- [10] R. Nogueira, K. Cho, Passage re-ranking with bert, *arXiv preprint arXiv:1901.04085* (2019).
- [11] M. Anand, J. Zhang, S. Ding, J. Xin, J. Lin, Serverless bm25 search and bert reranking., in: *DESIRES*, 2021, pp. 3–9.
- [12] S. Sannigrahi, J. Van Genabith, C. España-Bonet, Are the best multilingual document embeddings simply based on sentence embeddings?, *arXiv preprint arXiv:2304.14796* (2023).
- [13] R. Litschko, E. Artemova, B. Plank, Boosting zero-shot cross-lingual retrieval by training on artificially code-switched data, *arXiv preprint arXiv:2305.05295* (2023).
- [14] W. A. G. Kodri, M. Haris, R. Fitriadi, Fine-hybrid: Integration of bm25 and finetuned sbert to enhance search relevance, *Teknika* 14 (2025) 213–222.
- [15] K. Chakma, A. Das, Cmir: A corpus for evaluation of code mixed information retrieval of hindi-english tweets, *Computación y Sistemas* 20 (2016) 425–434.
- [16] S. Banerjee, K. Chakma, S. K. Naskar, A. Das, P. Rosso, S. Bandyopadhyay, M. Choudhury, Overview of the mixed script information retrieval (msir) at fire-2016, in: *Forum for information retrieval evaluation*, Springer, 2016, pp. 39–49.
- [17] M. Kowsher, I. Hossen, S. Ahmed, Bengali information retrieval system (birs), *International Journal on Natural Language Computing (IJNLC)* 8 (2019).

- [18] A. Deroy, S. Maity, Retrievegpt: Merging prompts and mathematical models for enhanced code-mixed information retrieval, arXiv preprint arXiv:2411.04752 (2024).
- [19] J. Savoy, L. Dolamic, M. Akasereh, Information retrieval with hindi, bengali, and marathi languages: Evaluation and analysis, in: Multilingual Information Access in South Asian Languages: Second International Workshop, FIRE 2010, Gandhinagar, India, February 19-21, 2010 and Third International Workshop, FIRE 2011, Bombay, India, December 2-4, 2011, Revised Selected Papers, Springer, 2013, pp. 334–352.
- [20] J. Leveling, G. J. Jones, Sub-word indexing and blind relevance feedback for english, bengali, hindi, and marathi ir, ACM Transactions on Asian Language Information Processing (TALIP) 9 (2010) 1–30.
- [21] D. Mandal, S. Dandapat, M. Gupta, P. Banerjee, S. Sarkar, Bengali and hindi to english cross-language text retrieval under limited resources., in: CLEF (Working Notes), 2007.
- [22] S. Bandyopadhyay, T. Mondal, S. K. Naskar, A. Ekbal, R. Haque, S. R. Godhavarthy, Bengali, hindi and telugu to english ad-hoc bilingual task at clef 2007, in: Workshop of the Cross-Language Evaluation Forum for European Languages, Springer, 2007, pp. 88–94.
- [23] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, M. Gatford, et al., Okapi at TREC-3, British Library Research and Development Department, 1995.
- [24] K. Song, X. Tan, T. Qin, J. Lu, T.-Y. Liu, Mpnet: Masked and permuted pre-training for language understanding, Advances in neural information processing systems 33 (2020) 16857–16867.
- [25] S. Chanda, K. Tewari, S. Pal, Overview of the cmir track at fire 2025: Code-mixed information retrieval from social media data, in: FIRE '25: Proceedings of the 17th Annual Meeting of the Forum for Information Retrieval Evaluation. December 17-20, Varanasi , India, Association for Computing Machinery (ACM), New York, NY, USA, 2025.
- [26] S. Chanda, K. Tewari, S. Pal, Overview of the cmir track at fire 2025: Code-mixed information retrieval from social media data, in: K. Ghosh, T. Mandl, S. Pal, S. Majumdar, A. Chakraborty (Eds.), Forum for Information Retrieval Evaluation (Working Notes) (FIRE 2025) December 17-20, Varanasi , India, CEUR-WS.org, 2025.