CoLIR: Late Interaction based Code-Mixed Information Retrieval for English-Bengali language pair

Devansh Shroff[†]

Abstract

There has been limited work towards developing information retrieval models for Indic languages, more so for code-mixed information retrieval. Code-mixing is an issue for Indic languages, as it is usually accompanied by transliteration and the absence of a standard CMIR model for the same. Based on the aforementioned premise, the paper attempts to develop a model specifically for the English-Bengali language pair. This work uses ColBERT on a test dataset containing code-mixed and transliterated queries developed from the content extracted from social media in English and Bengali. Query expansion using a phonetic algorithm is performed on the dataset to allow the parsing of texts to be more effective. A few experiments are performed by combining classical ranking models as initial rankers of the documents along with a neural re-ranker, in this case ColBERT. The experiments are carried out using PyTerrier. The results place *Team IRSolver 7th* with a *mAP score* of *0.063702*, and indicate that neural IR on the type of documents in our dataset seems to be the way forward for effective CMIR on Indic language texts.

Keywords

code-mixed, transliteration, ColBERT, Query Expansion, PyTerrier, CMIR, neural IR, information retrieval

1. Introduction

Indic languages have always been approached meticulously when it comes to various NLP tasks due to the diversity and uniqueness of the languages in terms of morphology and scripts. This however poses certain problems, especially for the models to parse a document written in those languages effectively. The problem is aggravated when the text is code-mixed and more so when it is transliterated as well. For example, a word like **আছে** is often transliterated as *aache* while on social media, it sometimes turns up as *a6e*. Consider another word like **ভালো** which is transliterated as *bhaalo* and a lot of times as *vaalo, valo, bhalo* etc. online. In information retrieval, this is a major hurdle, as our model must be robust enough to handle the linguistic dynamics in play in order to be able to effectively retrieve meaningful answers. In that case, how can we retrieve answers from documents that cannot even be parsed properly? How can we ensure that our model captures the linguistic nuances of Indian languages, even if we can parse them?

The paper aims to develop a model that can compute query relevance for a particular document, both of which are code-mixed and transliterated, specifically for English, Roman-transliterated Bengali and Brahmi script Bengali. Given a query and a document, the goal is to determine the relevance score of the query to the document and rank them accordingly. This involves handling the complexities of code-mixing, where elements from both languages are used within the same text, as well as dealing with the informal and non-standardized nature of the language. The system must accurately capture the semantic relationship between the query and the document despite these linguistic challenges. It is important because, first of all, lesser than usual amount of work has been documented for this specific use case. Presenting a significant outcome in this direction would give way to the development of highly robust search engines that truly return meaningful answers to queries in the vernacular language with a fundamental understanding of modern social contexts. Moreover, it would enable wider access to groups previously disadvantaged due to the absence of search systems that process vernacular low-resource languages, especially on social media platforms and online community forums.

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

[†]Work done during an internship at IIT-BHU

devansh.shroff@gmail.com (D. Shroff)

The importance of why one must consider the encoding of transliterated text is best shown in [1] and [2], which show a significant difference in performance when phonetic encoding is brought into play. Phonetic encoding has generally not been considered as a supplementary tool with neural rankers. This work explores the difference in performance by incorporating phonetic encoding in the form of query expansion and using it with a neural ranker and re-ranker to observe increased accuracy in document retrieval.

2. Dataset

This shared task consists of a single dataset [3, 4] for code-mixed information retrieval. The corpus consists of 107900 documents and 50 queries in total. The dataset is in Roman transliterated Bengali mixed with English language.

3. Related Work

The entire state-of-the-art was eventually narrowed down to 3 major techniques in order to develop effective Information Retrieval models for the purposes of this paper, which seem pertinent to the topic: **Prompting, Late-Interaction and Sparse Retrieval**.

3.1. Prompting

Prompting of LLMs has been a common occurrence across papers for Information Retrieval in the years following the development of the transformer [5]. It essentially involves zero-shot/few-shot prompting LLMs to make them rank the documents based on their relevance to the given query. One of the initial methods of ranking documents was done using Pointwise Ranking Prompting [6][7] wherein a relevance generation method is used [6] that involves the calculation of a relevance score based on a probabilistic function. Additionally, a query generation method [7] has also been used where a LLM was prompted to generate a query based on the document and the probability of generating the actual query would be checked. A more effective approach was devised by [8] called Pairwise Ranking Prompting in which relevance is checked pairwise owing to LLM's inherent ability to understand pairwise relevance order of documents. Experiments were carried out on test sets of TREC-DL2019 and TREC-DL2020 along with the BEIR dataset, giving higher NDCG@10 score than RankGPT and almost at par performance with GPT-4 and GPT-3.5 Turbo. An improvement over this is the Setwise approach [9] which increased the performance of [8] by comparing multiple candidates at once. Experiments for [9] were performed using BEIR and TREC-DL2019 as well, giving slightly higher NDCG@10 score than [8].

3.2. Sparse Retrieval

Sparse Retrieval is a technique for information retrieval wherein query and documents are converted into a high-dimensional sparse vector over a vocabulary and using inverted index to look up exact or weighted term matches, followed by ranking. One of the significant works in this sphere is [10] which introduced DeepCT (Deep Contextualized Term Weighting) framework that maps BERT's contextualized embeddings to context-aware term wights for sentences and passages [11], for first-stage retrieval. Another very important development in this area is [12] that introduced SPLADE (Sparse Lexical and Expansion Model) with the primary focus on first-stage ranking, giving much higher NDCG@10 and MRR@10 scores as compared to DeepCT [10] and other sparse retrieval models, upon evaluation on MS-MARCO dataset.

3.3. Late-Interaction

Late-interaction methods, introduced recently, have proven to be a game-changer in retrieval tasks due to their unique mechanism of individual parsing of queries and documents by a cross-encoder followed

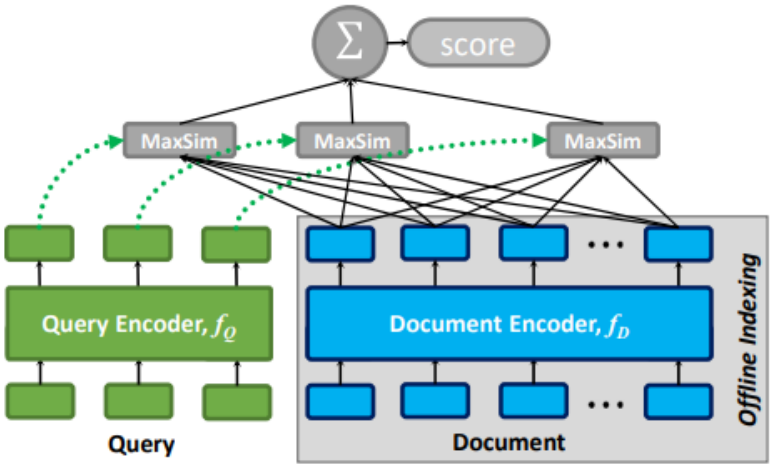

Figure 1: Base architecture of the ColBERT model

by similarity estimation using the *MaxSim*[1] function ensuring the preservation of important context in both, unlike coalescing them into a high-dimension vector. It was introduced by [13] through the ColBERT model (**Figure 1**). ColBERT gives a better MRR@10 and Recall score than existing methods, on MS MARCO. Its variant ColBERTv2 [14] performs even better when compared head to head with SPLADEv2 on the BEIR and LoTTE benchmark. Moreover, ColBERTv2 improves upon the latency of ColBERT, in addition to enabling higher retrieval quality and a much smaller index, thus saving space.

3.4. Approaches specifically for CMIR

Although all of the models mentioned above were not specifically made keeping in mind code-mixing, they are important cornerstones to be based off of, during the development of a robust model than can tackle code-mixing. When specifically considering code-mixed IR work on Indic languages, one of the benchmark works in this use case are [15] and [1] that have experimented on corpus-based stopwords removal and phonetic encoding for query expansion, respectively, proving to be quite effective and have given a completely new way to think about Indic texts for retrieval. An addition to the aforementioned papers, instrumental in devising a new method to effectively parse Indic texts, is [2] that introduces a new phonetic algorithm called Hindex. It defines a set of rules to encode transliterated Hindi texts based on the phonemes of the consonants and vowels in Hindi. [15] achieved a 16% increase in mAP score over the removal of non-corpus-based stopwords. [1] achieved over 15% increase in mAP for Hindex-encoded Named Entity and English tags, showcasing the efficacy of Hindex phonetic algorithm. [16] and [17] have explored this topic from the point of view of Sentence-BERT in addition to the Graph Neural Network (GNN), and a mathematical model based on the prompting GPT 3.5 Turbo and the sequential nature of the documents, respectively. An important contribution towards Indic IR in terms of resources has been IndicIRSuite [18], which introduces open-source ColBERT models called IndicColBERT, finetuned in 11 Indian languages using the Indic-MS MARCO dataset.

4. Methodology

The aim is to perform empirical research on selected Information Retrieval techniques that may prove effective for code-mixed transliterated social media data to achieve a significant mAP score and obtain meaningful search results when applied to queries with the same constraints, for English and Bengali. Moreover, a comparison is made between the traditional IR methods and these experimental methods

[1]MaxSim operator computes the maximum similarity between query embeddings and document embeddings, the scalar outputs of multiple such operators are summed across query terms [13]

to understand the difference in performance. The main research questions to be asked based on the above premise are as follows.

RQ1. Can we solely rely on a neural model for CMIR, without a first stage retrieval?

RQ2. How important is accurate phoneme-based parsing in terms of improving performance on code-mixed transliterated Indic language texts?

The experiments presented in this paper are carried out using PyTerrier, a Python interface for Terrier IR Engine developed by [19], that enables the creation of flexible retrieval pipelines. The models and indexer used here are taken from the PyTerrier libraries , like TRECCollectionIndexer for indexing documents as per the indexes used by algorithms like BM25 and ColBERTIndexer for indexing documents according to ColBERT's index structure. The models used for this experiment[20] are BM25 and ColBERT, wherein BM25 is loaded using the *pt.terrier.Retriever*()[2] function and ColBERT is intialised using the *ranking_factory*()[3] function from the PyTerrier extension of ColBERT. A small comparitive study is drawn up between BM25, ColBERT and a pipeline containing BM25 as the first stage ranker with ColBERT as a neural re-ranker. This is done using the *then* operator (represented by »), which combines two transformers [19] (retrieval transformer in this case). The experiments were carried out on training data provided by [3] wherein the first case included non-expanded queries and the other case included Hindex-expanded queries. Hindex is applied on Named-Entity tags and English tags only since this setup yielded the best results in [1]. Additionally, the *BM25»ColBERT* model is applied on the provided test queries [3]. This paper has used a pre-trained ColBERT model for the purposes of experimentation, no explicit fine-tuning was carried out on ColBERT.

The metrics used here to measure the performance of each model are mAP, (Mean Average Precision), NDCG (Normalised Discounted Cumulative Gain), P@5 (Precision@5) and P@10 (Precision@10), with greater emphasis on the mAP score.

5. Results

It is evident from **Table 1** and **Table 2** that query expansion improves the mAP, NDCG, P@5 and P@10 scores of our chosen models thus answering *RQ2* that phoneme-based parsing of transliterated code-mixed text for English and Bengali introduces variations of NE and English tags into the expanded query that aid in retrieval. It is also evident that solely using a neural ranker does not really yield a significant mAP score, rather a more sophisticated approach is required if an effective single-stage neural ranker is to be obtained, thus answering *RQ1*. The scores obtained on the test set were quite low and hence underscore the need to adopt a stronger approach.

Model	MAP Score	ndcg Score	P@5 Score	P@10 Score
BM25	0.175802	0.397553	0.34	0.25
ColBERT	0.074576	0.347530	0.22	0.145
BM25»ColBERT	0.119620	0.225578	0.37	0.250

Table 1
Results on training data non-expanded queries

Model	MAP Score	ndcg Score	P@5 Score	P@10 Score
BM25	0.213307	0.447165	0.39	0.27
ColBERT	0.093105	0.313139	0.16	0.14
BM25» ColBERT	0.161327	0.270455	0.37	0.27

Table 2
Results on training data Hindex-expanded queries

[2]https://pyterrier.readthedocs.io/en/latest/_modules/pyterrier/terrier/retriever.htmlRetriever
[3]https://github.com/terrierteam/pyterrier_colbert/blob/main/pyterrier_colbert/indexing.py

Rank	Team Name	Submission File	MAP Score	ndcg Score	P@5 Score	P@10 Score
1	IRSolver	Run 1	0.063702	0.147071	0.226667	0.156667
2	IRSolver	Run 2	0.05711	0.139316	0.206667	0.156667

Table 3
Submission results for IRSolver Team

Query	Document	Score	Rank
hi hyderabad e rapid antigen test kothay kora hochhe keu janate parben its urgent jate test korar 1 2 ghontar modhhe result pete pari	acha tmr chele thikache to covid test korar por test korar por kono problm hochhe na toh	17.857134	1
hi hyderabad e rapid antigen test kothay kora hochhe keu janate parben its urgent jate test korar 1 2 ghontar modhhe result pete pari	test er no e o onek kichu edik odik ache like delhi rt oct half kore half er besi rapid test korche	17.432081	2
hi hyderabad e rapid antigen test kothay kora hochhe keu janate parben its urgent jate test korar 1 2 ghontar modhhe result pete pari	rapid test authentic na apni normal ta karun citizen hospital e retcr ta korate parben	17.305046	3
hi hyderabad e rapid antigen test kothay kora hochhe keu janate parben its urgent jate test korar 1 2 ghontar modhhe result pete pari	pdf search korar valo kichhu website janate parben	16.580173	4
hi hyderabad e rapid antigen test kothay kora hochhe keu janate parben its urgent jate test korar 1 2 ghontar modhhe result pete pari	amader bareti 15 ta rapid antigen test kit a6e didi health worker bole	15.604569	5

Table 4
Sample test query with a list of relevant documents along with their relevance score obtained by the BM25 » ColBERT model

The scores obtained on the test set after successful submission of the obtained order of relevant documents for individual test queries, are presented in **Table 3** along with a list of sample query and documents along with their relevance scores from the test set presented in **Table 4**. Thus, team "IRSolver"[21, 22] secured 7th rank amongst all the submissions with a mAP score of *0.063702* when evaluated on test data by the organisers.

6. Conclusion

Code-mixing and transliteration complicate retrieval tasks on Indic language texts due to the sheer diversity of the scripts and the lingustic dynamics, which needs a more specialised approach. Consequently, traditional information retrieval methods cannot be employed to circumvent this problem.

This paper explores the efficacy of ColBERT as a single stage neural ranker for retrieval, and highlights the superiority of two-stage retrieval engine with BM25 as the first stage ranker and ColBERT as the neural re-ranker compared to the prior, in code-mixed information retrieval task. Moreover, the paper also highlights the usefulness of the phoneme-based approach Hindex for encoding texts in queries to retrieve the most relevant documents , and the boost in performance brought about by the combination of these two approaches.

Declaration on Generative AI

During the preparation of this work, the author used Grammarly to grammar and spelling check. The author reviewed and edited the content as needed and took full responsibility for the publication's content.

References

[1] S. Chanda, Text Processing on Code Mixed Social Media Data (2025).

[2] D. K. Prabhakar, S. Pal, C. Kumar, Query Expansion for Transliterated Text Retrieval, Transactions on Asian and Low-Resource Language Information Processing 20 (2021) 1–34.

[3] 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, FIRE '24, Association for Computing Machinery, New York, NY, USA, 2025, p. 29–31. URL: https://doi.org/10.1145/3734947.3735670. doi:10.1145/3734947.3735670.

[4] S. Chanda, S. Pal, Overview of the shared task on code-mixed information retrieval from social media data, in: FIRE 2024 Working Notesl, CEUR Workshop Proceedings, 2024, p. 124–128. URL: https://ceur-ws.org/Vol-4054/T2-1.pdf.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is All You Need, Advances in Neural Information Processing Systems 30 (2017).

[6] P. Liang, R. Bommasani, T. Lee, D. Tsipras, D. Soylu, M. Yasunaga, Y. Zhang, D. Narayanan, Y. Wu, A. Kumar, et al., Holistic Evaluation of Language Models, arXiv preprint arXiv:2211.09110 (2022).

[7] D. S. Sachan, M. Lewis, M. Joshi, A. Aghajanyan, W.-t. Yih, J. Pineau, L. Zettlemoyer, Improving Passage Retrieval with Zero-Shot Question Generation, arXiv preprint arXiv:2204.07496 (2022).

[8] Z. Qin, R. Jagerman, K. Hui, H. Zhuang, J. Wu, L. Yan, J. Shen, T. Liu, J. Liu, D. Metzler, et al., Large Language Models are Effective Text Rankers with Pairwise Ranking Prompting, arXiv preprint arXiv:2306.17563 (2023).

[9] S. Zhuang, H. Zhuang, B. Koopman, G. Zuccon, A Setwise Approach for Effective and Highly Efficient Zero-Shot Ranking with Large Language Models, in: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2024, pp. 38–47.

[10] Z. Dai, J. Callan, Context-Aware Sentence/Passage Term Importance Estimation for First Stage Retrieval, arXiv preprint arXiv:1910.10687 (2019).

[11] Z. Dai, J. Callan, Context-Aware Term Weighting for First Stage Passage Retrieval, in: Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, 2020, pp. 1533–1536.

[12] T. Formal, B. Piwowarski, S. Clinchant, SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2021, pp. 2288–2292.

[13] O. Khattab, M. Zaharia, ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT, in: Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, 2020, pp. 39–48.

[14] K. Santhanam, O. Khattab, J. Saad-Falcon, C. Potts, M. Zaharia, ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction, arXiv preprint arXiv:2112.01488 (2021).

[15] 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). URL: https://doi.org/10.1007/s42979-023-01942-7. doi:10.1007/s42979-023-01942-7.

[16] A. Sharma, InfoTextCM: Addressing Code-Mixed Data Retrieval Challenges via Text Classification, in: Forum of Information Retrieval and Evaluation (FIRE-2024), 2024.

[17] A. Deroy, S. Maity, RetrieveGPT: Merging Prompts and Mathematical Models for Enhanced Code-Mixed Information Retrieval, 2025. URL: https://arxiv.org/abs/2411.04752. arXiv:2411.04752.

[18] S. Haq, A. Sharma, P. Bhattacharyya, IndicIRSuite: Multilingual Dataset and Neural Information Models for Indian Languages, 2023. URL: https://arxiv.org/abs/2312.09508. arXiv:2312.09508.
[19] C. Macdonald, N. Tonellotto, Declarative Experimentation in Information Retrieval using PyTerrier, in: Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval, 2020, pp. 161–168.
[20] C. Macdonald, N. Tonellotto, S. MacAvaney, I. Ounis, PyTerrier: Declarative experimentation in Python from BM25 to dense retrieval, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, 2021, pp. 4526–4533.
[21] S. Chanda, K. Tewari, S. Pal, Findings of the code-mixed information retrieval from social media data (cmir) shared task at fire 2025, in: Forum for Information Retrieval Evaluation (Working Notes) (FIRE 2025) December 17-20, Varanasi , India, CEUR-WS.org, 2025.
[22] 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.