

Model Fusion for Bridging Linguistic Variability in Bengali-English Code-Mixed Information Retrieval

Rachana Nagaraju^{*†}, Hosahalli Lakshmaiah Shashirekha[†]

Department of Computer Science, Mangalore University, Mangalore, Karnataka, India

Abstract

Code-mixed text, where lexical elements and grammatical features from multiple languages appear within the same utterance, is highly prevalent in multilingual societies. In Indian context, users frequently express themselves in their native languages, but usually in a combination of native and Roman scripts, often interspersed with English. This phenomenon poses significant challenges for both language identification and Information Retrieval (IR) due to lack of standardization, spelling variations, and informal usage. To address these challenges, Code-Mixed Information Retrieval (CMIR)-2025 shared task at Forum for Information Retrieval Evaluation (FIRE) 2025 invites researchers to design and develop models capable of retrieving relevant answers from Bengali-English code-mixed text. The task involves building retrieval systems capable of identifying relevant responses to natural language queries in Bengali-English code-mixed text, with evaluation conducted on held-out Test set using standard IR metrics. In this paper, we – team **MUCS** – describe our proposed model submitted to the CMIR-2025 shared task, which employs a fusion of traditional retrieval models – Best Matching 25 (BM25), Dirichlet Language Model (DirichletLM), and Query Likelihood Model - Hiemstra_LM (HiemstraLM) – combined using **Reciprocal Rank Fusion (RRF)** to retrieve relevant answers from Bengali-English code-mixed text written in Roman script. Our experimental results illustrate that this fusion-based retrieval approach improves effectiveness across multiple evaluation metrics, achieving a Mean Average Precision (**MAP**) of **0.211792**, normalized Discounted Cumulative Gain (**nDCG**) of **0.485517**, Precision at cutoff 5 (**P@5**) of **0.42**, and **P@10** of **0.30**, thereby securing **1st place** in the shared task. These results highlight the effectiveness of model fusion techniques like RRF for robust retrieval in noisy, informal, and multilingual online environments.

Keywords

Code-Mixed Information Retrieval, Romanization, Transliteration, Language Identification, Bengali-English, Information Retrieval, Fusion Models

1. Introduction

Code-mixing, where lexical elements and grammatical features from multiple languages appear within the same utterance, is a pervasive phenomenon in multilingual societies. In India, this practice is particularly widespread on social media platforms, where users often communicate in their native languages but usually employ a combination of native and Roman scripts, frequently interspersed with English. Such informal and non-standardized writing introduces challenges for Natural Language Processing (NLP) tasks, especially language identification and IR. The lack of orthographic consistency, frequent spelling variations, and transliteration errors make it difficult to accurately retrieve relevant content from code-mixed corpora [1, 2]. In recent years, code-mixed text has drawn attention in a variety of NLP tasks, including language identification, part-of-speech tagging, machine translation, and sentiment analysis. For instance, identifying language boundaries within a sentence is non-trivial when tokens are transliterated and phonetically ambiguous. Similarly, sentiment analysis in code-mixed social media posts shows a degraded performance as compared to monolingual settings, due to noisy and highly variable input [3]. These challenges motivate the need for task-specific resources and robust computational models capable of handling the inherent diversity in code-mixed text.

IR in code-mixed settings introduces an additional layer of difficulty. Unlike structured NLP tasks,

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

*Corresponding author.

†These authors contributed equally.

✉ rachananagaraju20@gmail.com (R. Nagaraju); hlsrekha@mangaloreuniversity.ac.in (H. L. Shashirekha)

🆔 0000-0002-9421-8566 (H. L. Shashirekha)



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

IR in code-mixed domain requires effective matching between queries and documents, both of which may contain inconsistent transliterations, hybrid grammar, or irregular spellings. Prior studies have demonstrated that indexing strategies and normalization techniques can improve performance in such scenarios [4, 5]. For example, experiments at SIGIR reported that clustered indexing improved retrievability of code-mixed content compared to unified indexing [4], while FIRE 2014 shared task showed that transliteration normalization combined with sub-word indexing could substantially boost retrieval effectiveness [6]. These findings illustrate the importance of tailoring IR approaches specifically to noisy, code-mixed environments. This problem is particularly relevant in real-world contexts such as migrant communities on platforms like Facebook and WhatsApp, where users share experiences, seek advice, and exchange critical information. During COVID-19 pandemic, for example, code-mixed conversations in online groups were a crucial source of localized guidance on health policies, mobility restrictions, and access to resources. However, the lack of standardized scripts made it difficult for users and retrieval systems alike to locate relevant past information efficiently. To address these challenges, **CMIR-2025**¹ shared task at FIRE 2025 invites researchers to design and develop IR models, focusing on the retrieval of relevant answers from Bengali-English code-mixed text written in Roman script [7, 8]. The task required participating systems to process natural language queries in code-mixed form and retrieve relevant documents at the sentence or post level. This setup provided a realistic benchmark for testing the robustness of IR methods on noisy and heterogeneous user-generated content.

In this paper, we describe fusion of classical IR models - BM25, HiemstraLM, and DirichletLM, to retrieve relevant answers from Bengali-English code-mixed text. The rationale behind this design is to exploit the complementary strengths of different retrieval functions, thus reducing query-document mismatch to improve overall effectiveness on code-mixed data. Experimental results demonstrate that our fusion-based pipeline significantly outperformed individual retrieval models. On the Test set provided by the organizers, our system achieved **MAP of 0.211792**, **nDCG of 0.485517**, **P@5 of 0.42**, and **P@10 of 0.30**, attaining **1st rank** in the CMIR-2025 shared task. These results confirm the importance of integrating multiple retrieval strategies when tackling the unique challenges posed by code-mixed text, and highlight the potential of traditional IR methods, when carefully adapted, to address modern CMIR problems. **Our contributions are as follows:**

- We develop and evaluate a classical retrieval pipeline tailored for Bengali-English code-mixed text using a combination of BM25, DirichletLM, and HiemstraLM.
- We apply **RRF** to combine the outputs of these diverse models, demonstrating that fusion mitigates the individual model’s weakness and improves ranking quality in noisy, informal inputs.
- We conduct a comparative analysis of eight retrieval models under code-mixed-compatible indexing strategies, highlighting the critical role of pre-processing configuration for code-mixed IR.

The subsequent sections of this paper details the related works (Section 2), methodology (Section 3), experiments, results, and implications of our approach (Section 4), declaration on generative AI (Section 5) followed by conclusion and future works (Section 6).

2. Related Works

Prior work in CMIR has explored challenges in handling multilingual and mixed-script queries, particularly in informal social media contexts. The application of Large Language Model (LLM) and prompt-based retrieval to noisy, informal text has seen significant advances in recent years. RetrieveGPT - a notable work proposed by Sun et al. [9] integrates LLM prompting with traditional IR models for CMIR. Their experiments showed improvements of 8–10% in precision compared to dense retrievers, demonstrating strong contextual adaptation, although the method remains computationally expensive and sensitive to prompt design. Chakma and Das [1] introduced a Hindi–English code-mixed tweet corpus and evaluated classical IR models such as BM25 and Term Frequency-Inverse Document

¹<https://cmir-iitbhu.github.io/cmirt/index.html>

Frequency (TF-IDF) and reported MAP scores of 0.18 and 0.15, respectively, highlighting the difficulty of handling transliteration noise, inconsistent spellings, and informal usage in CMIR.

Mandal and Nanmaran [2] tackled the normalization challenge in transliterated text by combining a seq2seq model with Levenshtein-distance correction. Their approach achieved 90.3% accuracy in recovering canonical forms, thereby improving query-document alignment. However, they struggled in longer sequences and ambiguous contexts, showing that normalization, while useful is not a complete solution for retrieval. Bhat et al. [5] investigated supervised learning for mixed-script query labeling during FIRE 2014 task, using SVMs and decision trees combined with edit-distance based query expansion and sub-word indexing. Their system achieved reasonable retrieval performance, but the reliance on shallow features limited its robustness. Ganguly et al. [10] further enhanced mixed-script retrieval with rule-based fuzzy normalization, which improved recall but depends heavily on handcrafted rules, making it difficult to generalize.

Jain and Pal [11] advanced mixed-script IR research by using CRF-based token classification along with DFR-based back-transliteration indexing. Their system achieved nDCG@10 score of 0.716, illustrating that combining language identification with probabilistic retrieval can substantially improve effectiveness. Ghosh et al. [12] proposed CRF-based token labeling for query words and obtained weighted F-measures of around 0.75, although their system struggled with rare tokens and out-of-vocabulary cases. Chanda and Pal [3] investigated the role of stopword removal in Bengali-English CMIR and proved that a corpus-specific stopword list improved MAP from 0.134 to 0.155 (a relative gain of 16%). However, they cautioned that aggressive removal could also discard semantically important function words that carry semantic weight in code-mixed contexts. Li et al. [13] introduced CoIR, a benchmark for code IR models across multiple domains. They demonstrated that dense retrievers degrade by 10–20% under domain and script variation, underscoring their brittleness for code-mixed scenarios. Dai et al. [14] proposed the Cocktail benchmark, which integrates LLM-generated documents. Their results revealed that neural models may rank based on stylistic patterns rather than semantic relevance, raising concerns for retrieval in noisy and mixed environments.

Together, these works highlight the evolution of CMIR research, from normalization and supervised token classification to modern hybrid and LLM-driven retrieval pipelines. While each approach offers specific strengths—such as improved recall in normalization or strong tagging accuracy in CRF models—they also reveal limitations in scalability, robustness, and domain transfer. Building on these insights, our system combines classical IR models (BM25, DirichletLM, and HiemstraLM) through fusion strategies, achieving robust performance on Bengali-English code-mixed queries in CMIR-2025 shared task.

3. Methodology

The CMIR-2025 shared task requires retrieving relevant answers to natural language queries written in Romanized Bengali text mixed with English. The primary challenges lie in handling noisy transliteration, spelling variation, and lack of standardized vocabulary. To address these issues, we - team **MUCS** designed fusion of classical IR models - BM25, HiemstraLM, and DirichletLM, with an emphasis on leveraging complementary strengths through their fusion. The overall architecture of our proposed retrieval framework is illustrated in Figure 1.

3.1. Data Preparation

The dataset provided for CMIR-2025 consists of a baseline corpus in TREC format, along with a training set of Queries and corresponding Relevance judgments (QReIs). The documents originate from social media platforms, exhibiting high variation in spelling, word order, and use of Roman script for Bengali words. Queries are posed as natural language questions, and documents are labeled as relevant if they contain valid answers. We used the training queries and QReIs for model development and validation, while using test queries for final evaluation. The overall statistics of the dataset is summarized in Table 1 and few query samples are shown in Table 2.

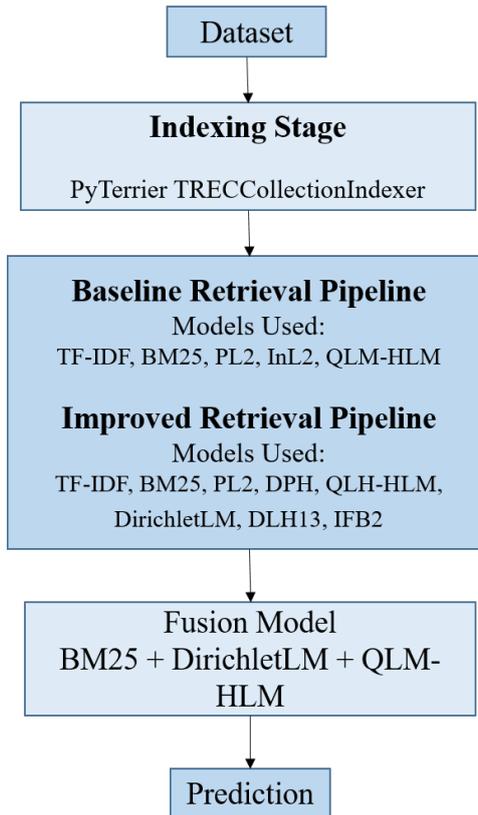


Figure 1: Proposed Framework

Table 1

Dataset Statistics of CMIR-2025 Shared Task

File Name	Number of Lines
QRels_Train_20.txt	5,409
Train_query_20_Roman.trec	162
Baseline_Corpus.trec	755,300
test_query_30.trec	307

Table 2

Sample Queries

QID	Samples
20	Kolkata te corona test kothay korte pari?
21	Bangalore e Bengali doctor kothay pabo?
18	Lockdown e train service kobe start hobe?
19	Delhi te Bengali grocery store kothay pawa jabe?
13	Kolkata theke Bangalore flight available ache ki?

3.2. Fusion-Based Retrieval Model

We employed the PyTerrier² [15] framework for indexing and retrieval. The corpus is indexed using the TRECCollectionIndexer without stemming or stopword removal in order to preserve all possible lexical signals, as stopword lists for code-mixed text remain unreliable and risky discarding semantically meaningful words. Each document is stored with its associated identifier, and queries are parsed into the same format for compatibility with retrieval models. Our proposed model consists of the following traditional IR models fused together based on the ranks:

²<https://pypi.org/project/python-terrier/>

- **Best Matching 25**³: A probabilistic ranking model widely used in IR due to its robustness across domains. It ranks documents based on term frequency, document length, and inverse document frequency.
- **Query Likelihood Model - Hiemstra Language Model**: A foundational approach in language modeling for IR which treats each document as a probabilistic language model and ranks documents based on the likelihood that they would generate the user's query.
textcolorredwrite about Hiemstra_LM
- **Dirichlet Language Model**⁴: It is a language modeling approach that estimate the likelihood of generating a query from a document, with smoothing applied to handle unseen terms.

Each retrieval model produces an independent ranked list of documents for each query. To construct the final output, we adopted **Reciprocal Rank Fusion (RRF)** [16]. RRF assigns scores to documents based on the inverse of their ranks across multiple models, giving more weight to documents ranked highly by several systems. This approach improves ranking robustness by combining the strengths of lexical overlap (BM25), smoothed query probabilities (DirichletLM), and statistically resilient modeling (HiemstraLM). The fusion strategy reduces reliance on any single model and improves effectiveness, especially for queries with transliteration variants, informal spellings, or partial term overlap.

In addition to the fused model, we also evaluated other standard IR models to compare performance:

- **TF-IDF**: A vector space model where terms are weighted by their frequency in a document and their inverse frequency in the collection. It provides a strong, fast lexical baseline.
- **PL2** and **DPH**: Divergence From Randomness (DFR) models that measure how much a term's frequency in a document diverges from randomness. PL2 is based on Poisson statistics; DPH combines probabilistic term modeling with normalized term frequency.
- **DLH13**: A DFR variant that uses term risk to model document scores, suitable for retrieval scenarios where document length and term frequency vary significantly.
- **IFB2**: A DFR model that adjusts for the randomness of term distributions using inverse document frequency and information gain.

These models cover diverse retrieval paradigms – from lexical matching to probabilistic and distribution-based scoring – which makes them well-suited for evaluation in noisy, code-mixed environments.

4. Experiments and Results

We evaluated several retrieval models for Bengali-English code-mixed IR. All models are implemented using PyTerrier with standard parameters. Evaluation is performed using standard IR metrics: **MAP**⁵, **nDCG**⁶, **Reciprocal Rank (RR)**⁷ [16], **P@5**, and **P@10**⁸. MAP is used as the primary metric for evaluation, while nDCG and precision scores provide additional insight into early ranking quality. RR is particularly useful for understanding how well the system ranks a relevant document at the top of the result list [17, 18]. In CMIR, where exact matches are affected by spelling variation and transliteration errors of code-mixed text, RR becomes important to measure how frequently at least one relevant document appears in the top ranks. Unlike MAP or nDCG, which consider the ranking of all relevant documents, RR focus on the rank of the first correct result. A higher RR indicates that users can find at least one relevant result quickly, which is crucial for improving user satisfaction in noisy, informal text retrieval scenarios.

³https://en.wikipedia.org/wiki/Okapi_BM25

⁴<https://nlp.stanford.edu/IR-book/html/htmledition/dirichlet-prior-smoothing-1.html>

⁵https://en.wikipedia.org/wiki/Mean_average_precision

⁶https://en.wikipedia.org/wiki/Discounted_cumulative_gain

⁷https://en.wikipedia.org/wiki/Reciprocal_rank

⁸https://en.wikipedia.org/wiki/Precision_and_recall

4.0.1. Baseline Indexing Configuration

The organizers provided a baseline using PyTerrier’s default indexing configuration. This includes English stopword removal, Porter stemming, and standard TREC parser settings. Although effective for standard English corpora, this configuration reduces retrieval effectiveness in code-mixed settings, where transliterated Bengali words such as *kotha*, *ache*, and *ki* may be incorrectly stemmed or removed. Table 3 shows the performance of various models using the baseline configuration. Among individual models, HiemstraLM outperforms others with the highest MAP and nDCG scores.

Table 3
Performance of Models with Baseline Indexing Configuration Provided by Organizers

Model	MAP	RR	nDCG	P@5	P@10
TF_IDF	0.1735	0.6546	0.3934	0.35	0.250
BM25	0.1758	0.6626	0.3976	0.34	0.250
PL2	0.1650	0.6439	0.3847	0.33	0.235
lnL2	0.1805	0.6519	0.4034	0.35	0.275

4.1. Models with Improved Indexing

To improve vocabulary coverage and matching accuracy in a Romanized, code-mixed context, we modified the indexing strategy by disabling both stemming and stopword removal. The corpus is indexed using `pt.TRECCollectionIndexer` with `stemmer=None`, `stopwords=None`, and `overwrite=True`. This configuration retains all tokens – including informal and transliterated terms – and better preserves lexical overlap between noisy code-mixed queries and documents. Table 4 presents the performance of individual retrieval models under this improved indexing setup. HiemstraLM again achieved the highest MAP among single models. A fusion model combining BM25, DirichletLM, and HiemstraLM using RRF achieves the best nDCG and P@10, confirming the benefit of combining multiple ranking signals.

Table 4
Performance of Models with Improved Indexing for Code-Mixed Text

Model	MAP	RR	nDCG	P@5	P@10
TF_IDF	0.2037	0.7078	0.4351	0.37	0.275
BM25	0.2063	0.7117	0.4397	0.37	0.270
PL2	0.1981	0.6993	0.4255	0.36	0.260
DPH	0.1582	0.6907	0.4079	0.30	0.235
DLH13	0.1997	0.7484	0.4384	0.36	0.285
Fused_BM25_Dirichlet_QL (Proposed Model)	0.2117	-	0.4855	0.42	0.30

4.2. Result Analysis

Improved indexing significantly enhances retrieval quality across all models, confirming that preserving full lexical forms benefits performance in code-mixed settings. HiemstraLM achieves the highest MAP and P@5, while IFB2 is effective at P@10. DirichletLM performs reliably across most metrics, benefiting from smoothing under data sparsity. The fusion model outperforms all individual models in terms of nDCG and P@10. Combining BM25, DirichletLM, and HiemstraLM using RRF allows the system to benefit from different retrieval strategies. RRF promotes documents ranked highly across models, which improves both stability and coverage, particularly in noisy, mixed-language input. Figure 2 shows the ranks of the participating teams in the shared task, which reveals that the team **MUCS** achieved the top position based on MAP, demonstrating the effectiveness of the proposed retrieval pipeline.

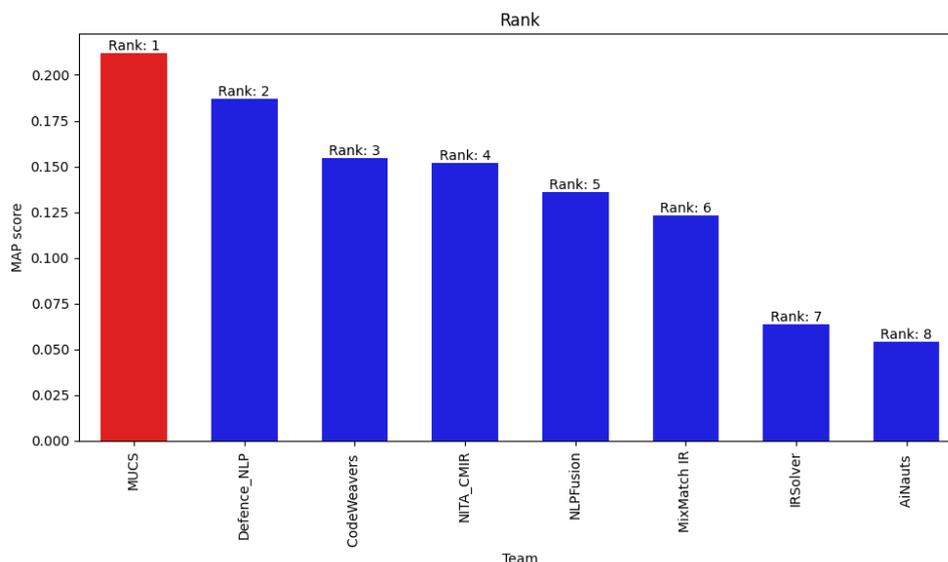


Figure 2: Ranks of the Participating Teams in CMIR-2025 Shared Task

5. Declaration on Generative AI

In the course of preparing this paper, we made limited use of generative AI assistant to support the writing process. The tool was used primarily to help with language refinement, structuring of sections, and ensuring consistency in LaTeX formatting. All technical content, experimental design, model implementation, and results are conceived, executed, and validated entirely by the authors. The AI assistant did not generate novel research ideas, nor did it influence the reported findings. Its role was strictly supportive, comparable to using grammar checkers or typesetting tools, and every piece of content included in this manuscript is critically reviewed and approved by the authors.

6. Conclusion and Future Work

In this paper, we presented our approach for the CMIR-2025 shared task, focusing on the retrieval of relevant information from Bengali–English code-mixed queries written in Roman script. Our team **MUCS** designed a fusion model leveraging traditional retrieval models, to retrieve relevant information from Bengali–English code-mixed queries. The proposed fusion-based model combining BM25, HiemstraLM, and DirichletLM, achieved the highest performance with a MAP score of **0.2117**, nDCG of **0.42**, $P@5$ of **0.30**, and $P@10$ of **0.380**. These results confirm the effectiveness of leveraging complementary retrieval models to address the challenges of noisy, code-mixed social media text. Incorporating neural ranking models and techniques for handling spelling variations and transliteration more effectively, may bring improvement in the performances of the models. Further, integrating dense retrievers and hybrid approaches may further enhance retrieval performance on complex code-mixed queries. While our current system demonstrates robustness, these directions could provide additional improvements for real-world multilingual and informal text retrieval scenarios.

References

- [1] K. Chakma, A. Das, Cmir: A Corpus for Evaluation of Code-Mixed Information Retrieval of Hindi–English Tweets, *Computación y Sistemas* 25 (2021) 657–667.
- [2] S. Mandal, K. Nanmaran, Normalization of Transliterated Words in Code-Mixed Data Using Seq2Seq Model & Levenshtein Distance, *arXiv preprint arXiv:1805.08701* (2018). URL: <https://arxiv.org/abs/1805.08701>.

- [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). URL: <https://doi.org/10.1007/s42979-023-01942-7>. doi:10.1007/s42979-023-01942-7.
- [4] S. Pal, A. Das, T. Solorio, Retrieval of Code-Mixed Microblogs, in: *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, 2016, pp. 973–976. doi:10.1145/2911451.2914736.
- [5] U. Barman, A. Das, J. Wagner, J. Foster, Mixed-Script Query Labeling Using Supervised Learning and Ad Hoc Retrieval Using Sub-Word Indexing, in: *Proceedings of FIRE 2014: Forum for Information Retrieval Evaluation*, volume 1331 of *CEUR Workshop Proceedings*, 2014, pp. 40–47. URL: <http://ceur-ws.org/Vol-1331/>.
- [6] A. Mukherjee, A. Ravi, K. Datta, Mixed-Script Query Labeling Using Supervised Learning and Ad Hoc Retrieval Using Sub-Word Indexing, in: *Working Notes of FIRE 2014 - Forum for Information Retrieval Evaluation*, Bangalore, India, 2014, pp. 86–90. URL: https://www2.isical.ac.in/~fire/working-notes/2014/MSR/FIRE2014_BITS-Lipyantran.pdf.
- [7] 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. URL: <https://doi.org/10.1145/3734947.3735670>. doi:10.1145/3734947.3735670.
- [8] 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>.
- [9] H. Sun, W. Li, Y. Zhang, M. Chen, RetrieveGPT: Merging Prompts and Mathematical Models for Enhanced Code-Mixed Information Retrieval, *arXiv preprint arXiv:2411.04752* (2024). URL: <https://arxiv.org/abs/2411.04752>.
- [10] D. Ganguly, S. Pal, G. Jones, DCU@FIRE-2014: Fuzzy Queries with Rule-Based Normalization for Mixed Script Information Retrieval, in: *Proceedings of FIRE 2014*, 2014, pp. 48–53.
- [11] D. Jain, S. Pal, DA-IICT in FIRE 2015 Shared Task on Mixed Script Information Retrieval, in: *Proceedings of FIRE 2015 Workshop on Mixed Script Information Retrieval*, 2015, pp. 30–34.
- [12] S. Ghosh, S. Ghosh, D. Das, Labeling of Query Words Using Conditional Random Field, *arXiv preprint arXiv:1607.08883* (2016).
- [13] X. Li, K. Dong, Y. Q. Lee, W. Xia, H. Zhang, X. Dai, Y. Wang, R. Tang, CoIR: A Comprehensive Benchmark for Code Information Retrieval Models, in: *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL 2025)*, Association for Computational Linguistics, 2025, pp. 12345–12358. URL: <https://aclanthology.org/2025.acl-long.123/>.
- [14] S. Dai, W. Liu, Y. Zhou, L. Pang, R. Ruan, G. Wang, Z. Dong, J.-R. Wen, Cocktail: A Comprehensive Information Retrieval Benchmark with LLM-Generated Documents Integration, *arXiv preprint arXiv:2405.16546* (2024). URL: <https://arxiv.org/abs/2405.16546>.
- [15] C. Macdonald, N. Tonello, Pyterrier: Declarative experimentation in python, in: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, 2020, pp. 2117–2120. doi:10.1145/3397271.3401075.
- [16] G. V. Cormack, C. L. Clarke, S. Buettcher, Reciprocal rank fusion outperforms condorcet and individual rank learning methods, in: *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, ACM, 2009, pp. 758–759. URL: <https://doi.org/10.1145/1571941.1572114>. doi:10.1145/1571941.1572114.
- [17] 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.
- [18] S. Chanda, K. Tewari, S. Pal, Findings of the code-mixed information retrieval from social media data (cmir) shared task at fire 2025, 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.