Bahnar–Vietnamese Parallel-Sentence Retrieval: A Low-Resource Case Study with Lexical, Neural, and Hybrid Methods

Thi Ty Nguyen[1], Long S. T. Nguyen[2,3,1], Phat T. Tran-Truong[2,3], Tan Sang Nguyen[4], Tho T. Quan[2,3*]

[1]Center for AI Research (CAIR), VinUniversity, Vietnam

[2]URA Research Group, Ho Chi Minh City University of Technology (HCMUT), Vietnam

[3]Vietnam National University Ho Chi Minh City, Vietnam

[4]National University of Singapore, Singapore

[*]**Correspondence:** qttho@hcmut.edu.vn

Abstract

Cross-lingual sentence retrieval makes existing translations accessible for reuse in low-resource language technology. For Bahnar–Vietnamese, we investigate when lexical evidence, multilingual representations, or their combination best identifies a query's translation. We compare lexical, alignment-based, neural, and hybrid methods in a fixed candidate pool, with configurations selected on development data. Their strengths differ by ranking depth: symmetric IBM1 reaches 0.4933 Recall@1 against 0.2424 for LaBSE, the strongest off-the-shelf encoder, while supervised DPR-XM leads at Recall@10 with 0.7221. IBM1 reranking of XLM-R LoRA candidates yields the highest Recall@1 point estimate, 0.5162. Paired tests support the gain over the neural first stage; the difference from IBM1 is not significant at the 0.05 level. Candidate-set analysis links reranking performance to first-stage coverage. Experiments on Khmer–Vietnamese, Lao–Vietnamese, and Zhuang–Chinese show that pair-specific adaptation improves neural retrieval, while lexical reranking gains vary. These findings make ranking depth central to method selection and motivate checking neural headroom and lexical representation quality when considering reranking. Data and code are available in our GitHub repository.

1 Introduction

Finding translations in existing texts can expand the parallel data available for low-resource *Natural Language Processing* (NLP), as demonstrated by improvements in low-resource translation from mined sentence pairs (Keung et al., 2020). Cross-lingual sentence retrieval locates such pairs by ranking sentences in one language for a query in another. This ranking step supports parallel-data mining (Artetxe and Schwenk, 2019; Schwenk et al., 2021); the resulting aligned pairs can also populate translation memories for computer-assisted translation (Semmar et al., 2006).

Bahnar–Vietnamese offers a case study in how to retrieve such pairs with limited language-specific resources. Bahnar is spoken by the Bahnar people in Vietnam and belongs to Bahnaric, a broader Austroasiatic subgroup (Gregerson et al., 1976; Nguyen et al., 2025). We use *Bahnar–Vietnamese* for the evaluated pair and Bahnaric–Vietnamese when discussing prior language-technology work. Our evaluation uses a fixed Vietnamese candidate pool with one documented counterpart per Bahnar query, as defined in Section 3. This setting isolates candidate ranking from the additional collection and filtering steps of open-web bitext mining.

Prior NLP work on Bahnaric–Vietnamese has mainly focused on translation, demonstrating the utility of lexical resources, loanwords, dictionary mappings, and rule-based components under low-resource conditions (Bui et al., 2024; Nguyen et al., 2025). These findings motivate a retrieval-specific comparison of lexical evidence and multilingual representations for ranking existing sentences. Three linked questions guide this comparison: which signal best identifies the documented counterpart, how task-specific adaptation changes neural retrieval performance, and whether lexical and neural signals complement each other. The intended use also determines the relevant ranking depth. Returning one sentence requires accurate top-rank decisions, whereas presenting candidates for reranking or human review requires coverage beyond the first rank.

We investigate these questions by comparing lexical, alignment-based, neural, and hybrid methods in the candidate-ranking setting of prior sentence retrieval and parallel sentence mining work

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 1260–1280

November 6-10, 2026 ©2026 Association for Computational Linguistics

(Trijakwanich et al., 2021; Schwenk et al., 2021). The same development-selection rule applies to off-the-shelf representations and models trained on language-pair-specific supervision. To test complementarity, we combine sentence scores from *XLM-RoBERTa* (XLM-R) (Conneau et al., 2020) trained with *Low-Rank Adaptation* (LoRA) (Hu et al., 2022) and lexical scores from *IBM Model 1* (IBM1) (Brown et al., 1993). Selecting both stage order and score combination allows us to examine which component should retrieve candidates and which should refine their ranking.

Symmetric IBM1 is the strongest single system at the top rank, while the supervised modular multilingual variant of *Dense Passage Retrieval* (DPR), termed DPR-XM (Karpukhin et al., 2020; Louis et al., 2025), provides the best coverage at deeper ranks. The XLM-R-first IBM1 hybrid gives the highest Recall@1 point estimate and improves *Mean Reciprocal Rank* (MRR) over its first stage.

To test how far these findings extend, we also train separate adaptation models on three further pairs: Khmer–Vietnamese, Lao–Vietnamese, and Zhuang–Chinese. Together, the controlled comparison and cross-pair diagnostics can inform model selection for translation lookup and candidate-review tools, including when lexical reranking is useful.

We make four empirical contributions:

- We provide a controlled comparison of lexical, alignment-based, neural, and hybrid retrieval for Bahnar–Vietnamese under a fixed candidate pool and common development-selection protocol, with released data and code.
- We show that top-rank accuracy and deeper candidate coverage favor different systems: symmetric IBM1 leads among single systems at rank 1, while DPR-XM leads at deeper cutoffs. Surface-overlap analysis further reveals differences obscured by aggregate scores.
- We demonstrate gains over the neural first stage through XLM-R-first IBM1 reranking, supported by paired tests, and examine how stage order and candidate coverage shape hybrid reranking performance.
- We train pair-specific models for Khmer–Vietnamese, Lao–Vietnamese, and Zhuang–Chinese and use post-hoc diagnostics to examine how reranking gains vary with neural headroom and lexical tokenization.

2 Related Work

Our comparison connects Bahnaric–Vietnamese language technology with lexical, neural, and hybrid retrieval. We review these strands to distinguish the evidence motivating each component from the retrieval questions evaluated here.

Bahnaric–Vietnamese NLP. Linguistic work situates Bahnar within the Bahnaric subgroup of Mon-Khmer languages in Vietnam (Gregerson et al., 1976), while the NLP studies discussed here focus on translation. Vo et al. (2024) combine dictionary lookup, entity mapping, and fine-tuned BARTPho (Tran et al., 2022) for Vietnamese–Bahnar translation. BARTBahnar combines continual pretraining, bilingual fine-tuning, lexical mapping, and post-processing (Nguyen et al., 2025). This use of lexical resources alongside neural models reflects broader low-resource NLP practice (Agić et al., 2015; Haddow et al., 2022; Wang et al., 2021a) and motivates testing their roles in sentence retrieval.

Lexical and word-alignment methods. *Term Frequency–Inverse Document Frequency* (TF-IDF) and *Best Matching 25* (BM25) use surface overlap for sparse retrieval (Salton and Buckley, 1988; Robertson and Zaragoza, 2009); Levenshtein distance measures string similarity through edit operations (Levenshtein, 1966). Loanwords and lexical overlap in Bahnaric–Vietnamese resources motivate these baselines (Nguyen et al., 2025). With parallel training data, IBM1 learns lexical translation probabilities (Brown et al., 1993), providing cross-lingual evidence beyond surface matching. GIZA++ and FastAlign offer related statistical alignment models (Och and Ney, 2003; Dyer et al., 2013), and *Cross-lingual Contextualized Phrase Retrieval* uses word-aligned phrase pairs to train a contrastive retriever (Li et al., 2024). We examine IBM1 probabilities as direct sentence scores and as reranking evidence.

Cross-lingual representations. Static embedding alignment combines subword-aware fastText representations (Bojanowski et al., 2017) with bilingual mappings learned through orthogonal Procrustes alignment (Artetxe et al., 2017). Contextual alternatives include *multilingual BERT* (mBERT), the multilingual version of *Bidirectional Encoder Representations from Transformers* (BERT), alongside XLM-R, multilingual MiniLM, and *Language-agnostic BERT Sentence Embedding* (LaBSE) (Devlin et al., 2019; Conneau et al., 2020; Wang et al., 2020; Feng et al., 2022). These models span the

static and contextual representations we compare.

Neural retrieval. Contrastive learning supports both unsupervised dense retrieval (Lei et al., 2023) and *Language Agnostic Multilingual Information Retrieval* (LAMIR), which combines semantic and language objectives (Hu et al., 2023). These retrieval models differ in how they represent and compare sentences. *Contextualized Late Interaction over BERT* (ColBERT) extends to cross-language retrieval through ColBERT-X (Nair et al., 2022); ColBERT-XM adds a modular multilingual backbone and multi-vector *Maximum Similarity* (MaxSim) interaction (Louis et al., 2025). Its DPR-XM ablation uses one pooled vector per sentence and dot-product scoring (Louis et al., 2025). Other approaches combine dense, sparse, and multi-vector retrieval (Chen et al., 2024), use contrastive context prediction (Wu et al., 2022), or distill representations through optimal transport on bitext (Huang et al., 2023).

Task-specific adaptation and retrieval scoring. LoRA enables adaptation through low-rank updates to pretrained models (Hu et al., 2022). Contrastive representation learning with *Information Noise-Contrastive Estimation* (InfoNCE) (Oord et al., 2018) also informs sentence embedding methods such as *Simple Contrastive Learning of Sentence Embeddings* (Gao et al., 2021). For retrieval, *Cross-domain Similarity Local Scaling* (CSLS) (Lample et al., 2018) and ratio margin (Artetxe and Schwenk, 2019) adjust cosine similarity using neighborhood statistics. These methods motivate our comparison of task-specific adaptation and retrieval scoring.

Hybrid lexical-dense retrieval. Hybrid retrieval combines lexical and neural signals through score interpolation and reranking (Karpukhin et al., 2020; Luan et al., 2021; Wang et al., 2021b). We use symmetric IBM1 probabilities as bilingual lexical evidence alongside XLM-R LoRA sentence similarity. The empirical question is which component should generate candidates and whether combining their scores improves the ranking.

Cross-lingual transfer. Whether transfer succeeds depends on the language pair, the quality of the data, and the available supervision (Lin et al., 2019). We therefore test the adaptation recipe on two pairs from the *Facebook Low Resource* (FLORES) benchmark (Goyal et al., 2022), Khmer–Vietnamese and Lao–Vietnamese, and on Zhuang–Chinese, training a separate model for each pair. Direct reuse of a Bahnar-trained checkpoint is evaluated separately in Appendix G.

3 Task and Data

The benchmark holds the candidate pool fixed so that differences in retrieval reflect how systems rank the available sentences. We define this setting, its supervision sources, and the metrics below.

Task definition. Following prior work on cross-lingual sentence retrieval and parallel sentence mining (Trijakwanich et al., 2021; Schwenk et al., 2021), we formulate Bahnar–Vietnamese parallel-sentence retrieval (*bitext retrieval*) as a controlled candidate-ranking task. Given a Bahnar query sentence q and a Vietnamese candidate set $\mathcal{C} = \{c_1, \ldots, c_n\}$ of size n, the system ranks the candidates by how well they correspond to the query. Each query has exactly one documented Vietnamese counterpart in the candidate pool. For final evaluation, each Bahnar sentence in the held-out test set is used as a query and all Vietnamese sentences in the same test set form the candidate pool.

Data resources and splits. We use sentence-level parallel pairs for model training, development selection, and retrieval evaluation, and a separate lexicon for fastText-Procrustes mapping and Kabsch alignment. The sentence resource contains 51,930 pairs for train-development splitting; exact duplicate bilingual pairs are grouped before a deterministic seed-42 split, and no exact bilingual pair is shared among the training, development, and test sets. A separate 2,001-pair test set is held out for final retrieval evaluation. The lexicon contains 10,358 single-word and multi-word entries; both alignment methods use only its training split. Table 1 gives the size of each split.

Table 1: Bahnar–Vietnamese sentence and lexicon split sizes. The lexicon has no development split (–).

Resource	Train	Dev	Test
Sentence pairs	46,736	5,194	2,001
Lexicon resource	8,285	–	2,073

Data construction and verification. The corpus was compiled from sources containing paired Bahnar and Vietnamese forms. Field materials include short stories, folk verses, and other texts from Bahnar-speaking communities in Vietnam, while documentary sources include dictionary entries, books, and songs. After digitization, native

Bahnar speakers checked the Bahnar text and corresponding Vietnamese forms for transcription errors, incorrect pairings, and obvious translation errors. The sentence data span heterogeneous domains and styles, including administrative text, agricultural and technical instructions, everyday and narrative sentences, and religious-style passages. This variation introduces differences in sentence length, topic, vocabulary, and cross-lingual lexical overlap; Appendix A gives examples.

Evaluation metrics. We report *Recall*@1, Recall@5, and Recall@10 (R@1, R@5, R@10) and MRR (Manning et al., 2008). Recall@10, for example, is the fraction of queries whose documented counterpart occurs among the ten highest-ranked candidates. With one documented counterpart per query, Recall@1 equals top-1 accuracy. In the primary benchmark, MRR averages the reciprocal rank of the counterpart within the ten highest-ranked candidates and assigns zero when it is ranked lower; for the hybrid, these are the ten reranked first-stage candidates. All methods therefore use the same cutoffs; Appendix B gives the formal definitions and statistical tests.

4 Methods

In the setting of Section 3, each method assigns a relevance score $s(q, c)$ to a Bahnar query q and Vietnamese candidate c, using surface similarity, learned lexical probabilities, or cross-lingual representations. We first describe the individual signals, then combine XLM-R LoRA and IBM1 in a two-stage reranker; Appendix C details the baselines.

Embedding-based retrieval scores. We compare cosine similarity, CSLS (Lample et al., 2018), and the ratio margin of Artetxe and Schwenk (2019) for fastText-Procrustes, the off-the-shelf encoders, XLM-R LoRA, and full XLM-R fine-tuning. Let $\mathcal{Q}$ be the query set, $\cos(q, c)$ the cosine similarity of the sentence embeddings of q and c, and $\mathcal{N}_k(q) \subset \mathcal{C}$ and $\mathcal{N}_k(c) \subset \mathcal{Q}$ the k candidates nearest to q and the k queries nearest to c. Equation 1 gives their mean similarities:

$$r_q = \frac{1}{k} \sum_{c' \in \mathcal{N}_k(q)} \cos(q, c'), \qquad r_c = \frac{1}{k} \sum_{q' \in \mathcal{N}_k(c)} \cos(q', c). \tag{1}$$

Equations 2 and 3 use these means to adjust cosine similarity for the local neighborhoods:

$$s_{\mathrm{CSLS}}(q, c) = 2\cos(q, c) - r_q - r_c, \tag{2}$$

$$s_{\mathrm{margin}}(q, c) = \frac{\cos(q, c)}{\max\{\frac{1}{2}(r_q + r_c),\, \epsilon_{\mathrm{num}}\}}. \tag{3}$$

Here, $\epsilon_{\mathrm{num}} > 0$ is a small numerical floor on the denominator. We use $k = 10$ for both criteria and compute the neighborhood means over the full query set and candidate pool before ranking.

4.1 Lexical and Static Embedding Methods

Lexical and edit-distance baselines. We evaluate character n-gram TF-IDF and BM25 as surface lexical baselines (Salton and Buckley, 1988; Robertson and Zaragoza, 2009). These methods test whether surface overlap, names, numbers, loanwords, and related Latin-script forms provide useful retrieval evidence. We also evaluate Levenshtein-based string similarity (Levenshtein, 1966) and token Jaccard similarity (Jaccard, 1912). Table 7 lists normalization variants: accent, punctuation, and space removal.

Static embedding alignment. We train separate Bahnar and Vietnamese fastText embeddings (Bojanowski et al., 2017) as a non-transformer baseline and learn a supervised orthogonal Procrustes mapping from the lexicon training split, following standard bilingual embedding alignment methods (Artetxe et al., 2017). At retrieval time, Bahnar query embeddings are mapped into the Vietnamese space, and candidates are ranked with cosine similarity, CSLS, or ratio margin (Lample et al., 2018; Artetxe and Schwenk, 2019). Alignment-unit, pooling, and normalization variants are listed in Appendix D.

IBM1 word-alignment baseline. We evaluate IBM1 as a classical lexical alignment baseline (Brown et al., 1993). It learns word-level translation probabilities from the Bahnar–Vietnamese training pairs and belongs to the statistical alignment tradition behind GIZA++ and the related FastAlign system (Och and Ney, 2003; Dyer et al., 2013). For a Bahnar query q and Vietnamese candidate c, Equation 4 defines the forward IBM1 retrieval score as

$$s_{\rightarrow}(q, c) = \frac{1}{|q|} \sum_{b \in q} \log \max\Big\{ \max_{v \in c} p(v \mid b),\, \epsilon \Big\}, \tag{4}$$

where b and v range over the token occurrences in q and c, $|q|$ and $|c|$ are their token counts, and $p(v \mid b)$ is the learned Bahnar-to-Vietnamese lexi-

cal probability. A small positive probability floor ϵ keeps the natural logarithm finite. The backward score $s_{\leftarrow}(q, c)$ uses the reverse model $p(b \mid v)$, maximizes over $b \in q$, and averages over the $|c|$ candidate tokens. Equation 5 averages the two directions to obtain the symmetric score:

$$s_{\text{sym}}(q, c) = \frac{1}{2}\left[s_{\rightarrow}(q, c) + s_{\leftarrow}(q, c)\right]. \quad (5)$$

The development-selected IBM1 variant additionally uses text normalization and subtracts a length penalty with weight λ:

$$s_{\text{IBM1}}(q, c) = s_{\text{sym}}(q, c) - \lambda \frac{\big||q| - |c|\big|}{\max(|q|, |c|)}. \quad (6)$$

4.2 Neural Encoders and Adaptation

Off-the-shelf multilingual encoders. We evaluate mBERT (Devlin et al., 2019; Pires et al., 2019), XLM-R (Conneau et al., 2020), multilingual MiniLM (Wang et al., 2020), and LaBSE (Feng et al., 2022) without task-specific fine-tuning, using mean-pooled token representations or the native pooling of sentence-transformer models.

Modern dense retrievers. We train four supervised dense retrievers on the same 46,736 training pairs: LAMIR, DPR-XM, ColBERT-X, and ColBERT-XM (Hu et al., 2023; Nair et al., 2022; Louis et al., 2025). DPR-XM encodes each sentence as one L2-normalized vector and scores candidates by their dot product, whereas ColBERT-X and ColBERT-XM use token-level late interaction; Table 5 in Appendix D lists the selected checkpoints.

XLM-R LoRA adaptation. We adapt XLM-R using LoRA adapters and separate source and target projection heads (Conneau et al., 2020; Hu et al., 2022). The projection heads map contextual representations into the shared 256-dimensional space of the query and candidate embeddings in Figure 1. Let $\mathbf{u}_i$ and $\mathbf{v}_i$ be the projected sentence embeddings of the i-th Bahnar and Vietnamese sentences in a minibatch of B training pairs. The model is trained with a symmetric in-batch InfoNCE objective (Oord et al., 2018), related to contrastive sentence representation learning (Reimers and Gurevych, 2019; Gao et al., 2021):

$$\mathcal{L} = -\frac{1}{2B}\sum_{i=1}^{B}\Big[\log \frac{e^{\cos(\mathbf{u}_i, \mathbf{v}_i)/\tau}}{\sum_{j=1}^{B} e^{\cos(\mathbf{u}_i, \mathbf{v}_j)/\tau}} + \log \frac{e^{\cos(\mathbf{u}_i, \mathbf{v}_i)/\tau}}{\sum_{j=1}^{B} e^{\cos(\mathbf{u}_j, \mathbf{v}_i)/\tau}}\Big]. \quad (7)$$

Here, $\tau = 0.07$ is the temperature, and j ranges over all pairs in the minibatch: the aligned pair has $j = i$, and the other pairs supply in-batch negatives. We compare sentence-mean and token-mean pooling with token pooling weighted by *Inverse Document Frequency* (IDF), together with optional lexicon-derived Kabsch alignment (Kabsch, 1976). At retrieval time, the resulting sentence representations are evaluated with cosine, CSLS, and ratio-margin scoring. Algorithm 1 in Appendix D summarizes training and retrieval.

Full-encoder XLM-R fine-tuning. To compare *Parameter-Efficient Fine-Tuning* (PEFT) with full fine-tuning, we train separate Bahnar and Vietnamese XLM-R encoders initialized from XLM-R base (Conneau et al., 2020). We compare one-way and symmetric InfoNCE-style contrastive objectives (Oord et al., 2018; Gao et al., 2021), together with alternative pooling and projection settings and all three retrieval criteria.

4.3 Hybrid Reranking

Figure 1 organizes the two-stage pipeline into a neural retriever, an IBM1 lexical scorer, and a hybrid reranker. XLM-R LoRA and IBM1 learn from the same parallel training pairs; the dashed arrow denotes InfoNCE training of XLM-R LoRA.

Neural retriever: candidate generation. Development selection compares XLM-R-first and IBM1-first pipelines and chooses XLM-R LoRA with ratio margin ($s_{\text{neural}} = s_{\text{margin}}$) to generate candidates. Ranking the Vietnamese candidate pool by this score yields the neural top-K set $\mathcal{C}_K(q) \subset \mathcal{C}$, with the selected $K = 10$ distinct from the neighborhood size k. First-stage coverage bounds reranking recall because the documented counterpart must be retained to be retrieved.

IBM1 lexical scorer: lexical evidence. The IBM1 lexical scorer combines forward and backward scores using Equation 5 and applies the length penalty in Equation 6 to obtain $s_{\text{IBM1}}(q, c)$ for each retained candidate. These scores provide word-level translation evidence to complement neural similarity within the same candidate set.

Hybrid reranker: score interpolation. For each retained candidate $c \in \mathcal{C}_K(q)$, each component score s is min–max normalized:

$$\tilde{s}(q, c) = \frac{s(q, c) - \min_{c'} s(q, c')}{\max_{c'} s(q, c') - \min_{c'} s(q, c')}.$$

Both extrema are taken over $c' \in \mathcal{C}_K(q)$. When

the score range is below a small positive threshold ϵ_{norm}, all normalized scores for that component are set to zero, including when $K = 1$. Equation 8 combines the normalized neural and IBM1 scores with an optional string-similarity term:

$$s_{\text{hybrid}}(q, c) = \alpha \tilde{s}_{\text{neural}}(q, c) + \beta \tilde{s}_{\text{IBM1}}(q, c) + \gamma \tilde{s}_{\text{edit}}(q, c), \tag{8}$$

where the weights α, β, and γ scale the neural, IBM1, and string-similarity components, respectively, and $s_{\text{edit}} \in [0, 1]$ is a character-level string-similarity ratio between q and c. Development selection gives $\alpha = 0.25$, $\beta = 0.75$, and $\gamma = 0$, with IBM1 length penalty $\lambda = 0.1$. Sorting $\mathcal{C}_K(q)$ by s_{hybrid} gives the reranked list in Figure 1, where $c_{(i)}$ and $\hat{c}_{(i)}$ mark the candidate at rank i before and after reranking, respectively.

5 Bahnar–Vietnamese Experiments

The experiments in this section compare individual retrieval signals and their combination through reranking; the next section examines transfer across language pairs. Together, they address six *Research Questions* (RQs):

RQ1. When are surface lexical, edit-distance, and static embedding methods effective?

RQ2. Can classical word alignment provide a strong sentence-level retrieval signal?

RQ3. How well do off-the-shelf encoders and in-domain modern dense retrievers perform?

RQ4. Does task-specific XLM-R adaptation improve neural retrieval?

RQ5. Do lexical and neural retrieval signals complement each other in hybrid reranking?

RQ6. Do the main findings hold for other low-resource language pairs?

Section 5.2 addresses RQ1–RQ5 on the Bahnar–Vietnamese test set, and Section 6 addresses RQ6.

5.1 Experimental Setup

Training and evaluation data. Using the splits defined in Section 3, we fit sentence-trained components on the 46,736 training pairs and select configurations on the 5,194 development pairs, reserving the 2,001-pair test set for final evaluation. IBM1 lexical probabilities and the monolingual fastText embeddings use only the sentence training pairs. Both the fastText-Procrustes mapping and Kabsch alignment use only the lexicon training split.

Selection protocol. Every model, preprocessing, training, and retrieval choice is made on development data alone. Within each method family, configurations are ranked by development Recall@1, then MRR and Recall@5 as tie-breakers, followed by configuration name if all three metrics tie. For the primary benchmark, only the development-selected configuration from each family is evaluated on the held-out test set. Later sensitivity and diagnostic analyses on the test set are post hoc and do not affect selection.

Dense and hybrid selection. For the modern-dense family, selection covers both architecture and checkpoint and yields DPR-XM. For the hybrid family, IBM1 and XLM-R LoRA are first selected independently. We then select stage order, K, score normalization, reranking formulation, interpolation weights, and length penalty on the same development set. Appendix D details the search spaces; Tables 6 and 9 report development results by family and scoring criterion, respectively.

5.2 Retrieval Results

Table 2 reports the held-out test result for one development-selected configuration per method family, using the methods described in Section 4. By Recall@1, the hybrid, symmetric IBM1, and supervised DPR-XM occupy the top three positions, followed by edit distance and BM25 ahead of XLM-R LoRA.

Top-rank accuracy and coverage favor different systems (RQ2–RQ4). Symmetric IBM1 is the strongest single system at the top rank, whereas DPR-XM leads at Recall@5 and Recall@10. The preferred retriever therefore depends on whether the task requires the documented counterpart at rank 1 or within a larger candidate set. Task-adapted XLM-R LoRA also outperforms the selected off-the-shelf encoder, reaching 0.3648 Recall@1 compared with 0.2424 for LaBSE. Full XLM-R fine-tuning has lower Recall@1 than both. Combining XLM-R LoRA with IBM1 gives the highest Recall@1 point estimate; the paired comparisons below assess this gain.

BM25 and fastText differ by overlap (RQ1). The aggregate advantage of BM25 over fastText-Procrustes comes mainly from pairs with higher surface overlap. fastText-Procrustes outperforms BM25 at Recall@1 in the two lowest character-overlap quartiles, whereas the ordering reverses in the high-overlap half. The contrast is largest in

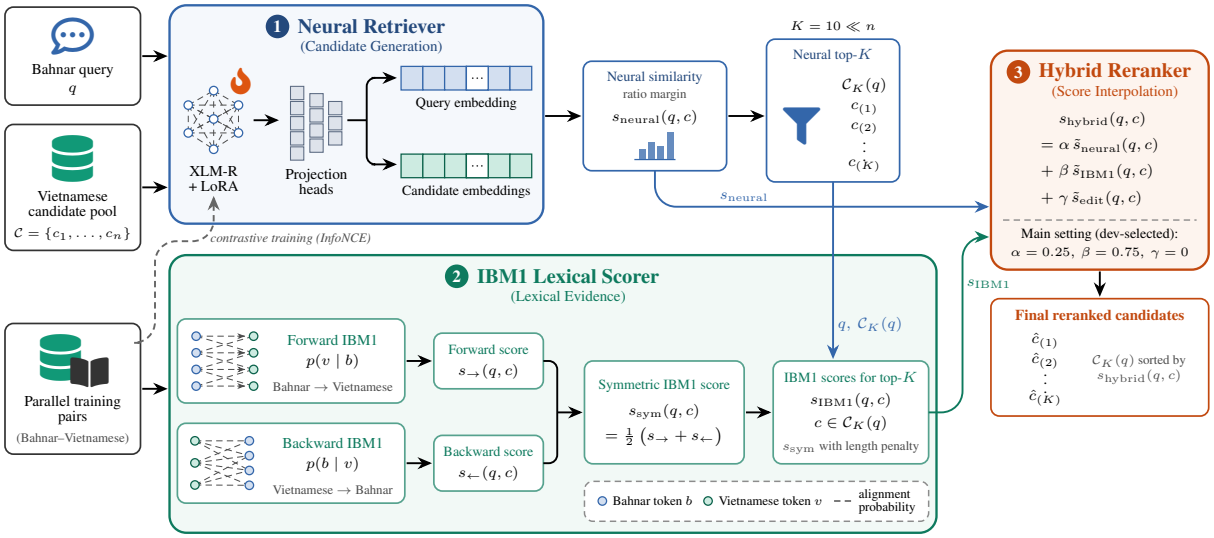


Figure 1: Development-selected XLM-R LoRA candidate generation and IBM1 reranking pipeline.

Table 2: Held-out test results for the development-selected configuration of each method family. **Bold** marks the best score in each column and underlining the second best; shaded rows are our methods.

Group	Development-selected method	R@1	MRR	R@5	R@10
Surface lexical	BM25 character n-gram, 2–5	0.3723	0.4004	0.4393	0.4548
Edit distance	Levenshtein ratio, strip accents	0.4248	0.4447	0.4703	0.4913
Static embedding	fastText-Procrustes, ratio margin	0.1654	0.2114	0.2689	0.3298
Word alignment	Symmetric IBM1, length penalty	0.4933	0.5459	0.6167	0.6662
Off-the-shelf encoder	LaBSE, ratio margin	0.2424	0.2734	0.3153	0.3518
Modern dense retrieval	DPR-XM, 20 epochs	0.4743	0.5506	**0.6522**	**0.7221**
Task-adapted neural	XLM-R LoRA, token mean, ratio margin	0.3648	0.4462	0.5517	0.6402
Full fine-tuning	Full XLM-R, token mean, CSLS	0.2224	0.2985	0.3968	0.4808
Hybrid	XLM-R-first + symmetric IBM1 reranking	**0.5162**	**0.5579**	0.6157	0.6402

the highest-overlap quartile, where BM25 reaches 0.9361 Recall@1 compared with 0.0279 for fastText. Figure 2 shows all four quartiles.

Reranking improves the neural first stage (RQ5). Reranking raises Recall@1 over the XLM-R LoRA first stage by 0.1514, from 0.3648 to 0.5162, and MRR by 0.1117. The paired-bootstrap 95% *Confidence Interval* (CI) for each gain (Efron and Tibshirani, 1994) excludes zero, as shown in Figure 3, and the exact McNemar test (McNemar, 1947) gives $p < 0.001$. Against symmetric IBM1, the hybrid has higher point estimates, but both CIs include zero and the difference is not significant at the 0.05 level. Appendix F gives the full statistics.

Development selection favors XLM-R-first ordering (RQ5). On development data, the XLM-R LoRA first stage keeps the counterpart in its top 10 for 0.7657 of queries, against 0.6213 for sym-

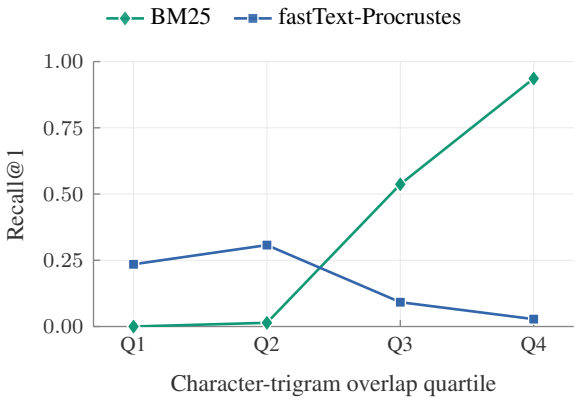


Figure 2: Recall@1 of the development-selected BM25 and fastText-Procrustes systems by character-trigram-overlap quartile of the 2,001 test queries. Q1 contains the queries with the lowest overlap, and Q4 contains those with the highest.

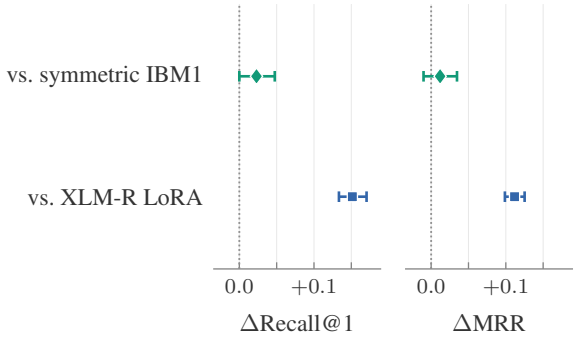


Figure 3: Hybrid-minus-comparator differences in Recall@1 and MRR on the 2,001-query test set. Whiskers show paired-bootstrap 95% CIs, and the dotted line marks zero difference.

metric IBM1. The XLM-R-first pipeline also leads after reranking on every reported metric, as shown in Figure 6 in Appendix E. These results support generating candidates with the neural model and refining them with IBM1 lexical evidence.

Reranking improves all three seed runs (RQ5). To assess variation across training runs, we repeat the selected XLM-R LoRA configuration with different random seeds while keeping the retrieval and reranking settings fixed. Across three XLM-R LoRA training seeds, reranking raises Recall@1 by 0.1369–0.1584, and every hybrid run exceeds symmetric IBM1. Figure 7 in Appendix F shows the individual runs and their variation.

Larger candidate sets improve hybrid retrieval (RQ5). The size of the candidate set determines which translations remain available to the reranker. In the post-hoc sensitivity analysis shown in Figure 8 in Appendix F, hybrid Recall@1 rises from 0.5162 at the development-selected $K = 10$ to 0.6157 at $K = 100$, where first-stage coverage reaches 0.8951. At smaller candidate sizes, Recall@5 and Recall@10 coincide with first-stage coverage for $K \leq 5$ and $K \leq 10$, respectively, since reranking only reorders the retained set.

6 Transfer to Other Language Pairs

To answer RQ6, we evaluate the adaptation recipe and lexical reranking on three additional language pairs, training a separate model for each pair. The comparison asks whether adaptation improves neural retrieval in these settings and whether lexical evidence adds further gains.

Transfer setup. Models for Khmer–Vietnamese and Lao–Vietnamese are trained on the 997 aligned FLORES development pairs and evaluated on the 1,012-pair devtest split (Goyal et al., 2022), where each source sentence is a query and all Vietnamese sentences are candidates. Zhuang–Chinese uses the corpus released with ZhuangBench (Zhang et al., 2024): 4,944 training pairs and a separate 200-pair evaluation pool in which each Zhuang query ranks all 200 Chinese candidates. Appendix G describes these setups, with hyperparameters in Table 8 and data sizes and surface overlap across all four pairs in Table 10.

Adaptation improves neural retrieval. In all three transfer settings, pair-specific XLM-R LoRA with CSLS gives the strongest neural retrieval results. Adaptation improves both top-rank accuracy and candidate coverage over off-the-shelf XLM-R, as shown in Figure 4. For example, Khmer–Vietnamese Recall@1 rises from 0.5030 to 0.8014 under the same CSLS criterion. Lao–Vietnamese and Zhuang–Chinese show the same pattern, which supports training a separate model for each pair.

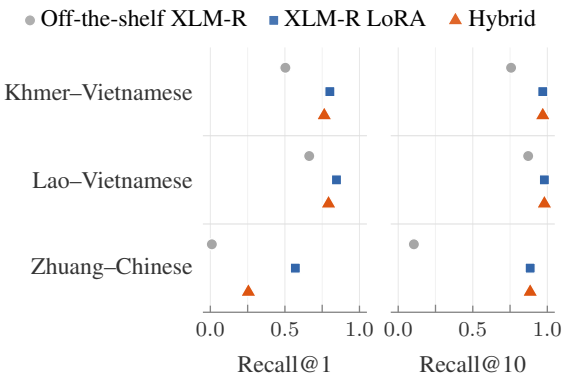


Figure 4: Recall@1 and Recall@10 on the three transfer language pairs for off-the-shelf XLM-R, pair-specific XLM-R LoRA, and the hybrid, all with CSLS.

Neural ranking leads at the top rank in the original transfer setups. Adding lexical reranking to the adapted neural retriever lowers Recall@1 and MRR in all three transfer setups while preserving Recall@10, as reported in Table 11 in Appendix G. The unchanged Recall@10 follows from reranking the same ten neural candidates. The Zhuang–Chinese result, which uses Chinese whitespace tokenization, motivates the diagnostic below.

Higher neural accuracy comes with smaller reranking gains. Across the four evaluated language pairs, reranking improves top-rank accuracy on Bahnar–Vietnamese and on Zhuang–Chinese with character tokenization, while reducing it on the two FLORES pairs, whose neural first stages are stronger. Figure 5 summarizes this monotonic

association, with Spearman correlation $\rho = -1$ across the four heterogeneous settings. This diagnostic motivates examining how much room remains for improving the neural ranking before adding lexical evidence.

Chinese tokenization changes lexical reranking performance. In the Zhuang–Chinese comparison reported in Table 12 and discussed in Appendix H, whitespace tokenization yields sparse IBM1 statistics, whereas character tokenization changes hybrid Recall@1 from 0.2550 to 0.6450. This post-hoc diagnostic uses the evaluation pool without a separate Zhuang development set. Table 11 reports the original transfer evaluation. Together, these observations suggest examining both neural headroom and whether tokenization yields reusable bilingual lexical units when applying lexical reranking.

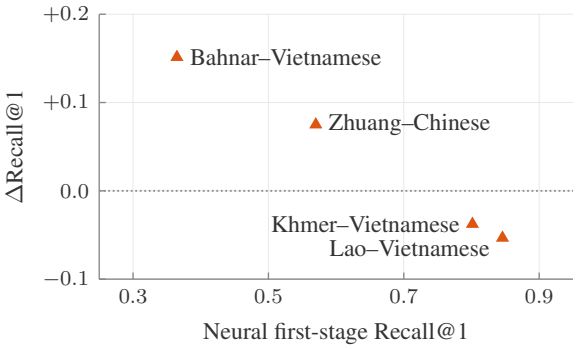


Figure 5: Neural first-stage Recall@1 and the hybrid-minus-first-stage difference in Recall@1 across the four pairs; Zhuang–Chinese uses character tokenization.

7 Conclusion

We compare lexical, alignment-based, neural, and hybrid sentence retrieval for Bahnar–Vietnamese under a common development-selection protocol. Symmetric IBM1 is the strongest single system at the top rank, whereas supervised DPR-XM gives the best coverage at deeper ranks. Reranking XLM-R LoRA candidates with IBM1 yields the highest Recall@1 point estimate. Paired tests support the gain over the neural first stage but not a significant difference from symmetric IBM1 at the 0.05 level. Across three additional language pairs, pair-specific adaptation improves neural retrieval, while lexical reranking gains vary. Transfer diagnostics suggest checking neural headroom and lexical representation quality before adding reranking. These findings can guide retrieval depth and reranking choices for translation lookup and tools that present candidates for human review.

Limitations

The evaluation in Section 3 focuses on Bahnar within the Bahnaric subgroup and uses a fixed candidate pool with one documented counterpart per query. Other Bahnaric languages, multiple valid translations, missing matches, and noisy or partially aligned collections require separate evaluation. Adaptation findings concern XLM-R, while cross-pair comparisons involve separately trained models with different training-data sizes and supervision conditions.

Impact Statement

These findings can guide translation lookup and parallel-data curation for underrepresented languages. Tools that help speakers and domain experts review candidate translations could support language documentation and education. Human validation remains appropriate in these applications and in administrative settings, where retrieval errors may propagate into downstream resources. Manual review of the representative sentence- and lexicon-level examples found no personally identifying information or offensive content.

Data and Code Availability

We make the data, code, and experimental artifacts available on GitHub. The repository includes split manifests, selected configurations, records of the development selection, and result artifacts to document the evaluation setup and model choices. The code covers the comparison methods, including the supervised modern dense retrievers, and the hybrid pipeline. Together, these resources support inspection of the reported comparisons and reuse of the data and retrieval components in future work.

AI Assistance Statement

We used GPT-6 Astra and Claude Opus 5.5 for assistance with manuscript drafting and language editing, as well as experiment implementation. They also assisted in redrawing Figure 1 from a sketch prepared by the authors. The authors formulated the research questions, designed the experiments, and conducted the analyses. All code, results, figures, and text were reviewed and verified by the authors, who take full responsibility for the manuscript and its scientific claims.

References

Željko Agić, Dirk Hovy, and Anders Søgaard. 2015. If all you have is a bit of the Bible: Learning POS taggers for truly low-resource languages. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 268–272, Beijing, China. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, Vancouver, Canada. Association for Computational Linguistics.

Mikel Artetxe and Holger Schwenk. 2019. Margin-based Parallel Corpus Mining with Multilingual Sentence Embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3197–3203, Florence, Italy. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311.

Long-Ngo-Hoang Bui, Huu-Thien-Phu Nguyen, Minh-Khoi Le, Cong-Thien Pham, and Thanh-Tho Quan. 2024. Handling imbalanced resources and loanwords in Vietnamese-Bahnaric neural machine translation. *Int. J. Intell. Inf. Database Syst.*, 16(4):451–472.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. M3-Embedding: Multi-Linguality, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2318–2335, Bangkok, Thailand. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The Hitchhiker's Guide to Testing Statistical Significance in Natural Language Processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392, Melbourne, Australia. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A Simple, Fast, and Effective Reparameterization of IBM Model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

B. Efron and R.J. Tibshirani. 1994. *An Introduction to the Bootstrap*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT Sentence Embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The Flores-101 Evaluation Benchmark for Low-Resource and Multilingual Machine Translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Kenneth Gregerson, Kenneth Smith, and David Thomas. 1976. The Place of Bahnar within Bahnaric. *Oceanic Linguistics Special Publications*, 13:371–406.

Barry Haddow, Rachel Bawden, Antonio Valerio Miceli Barone, Jindřich Helcl, and Alexandra Birch. 2022. Survey of Low-Resource Machine Translation. *Computational Linguistics*, 48(3):673–732.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew

Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, and 7 others. 2020. Array programming with NumPy. *Nature*, 585(7825):357–362.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Xiyang Hu, Xinchi Chen, Peng Qi, Deguang Kong, Kunlun Liu, William Yang Wang, and Zhiheng Huang. 2023. Language Agnostic Multilingual Information Retrieval with Contrastive Learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9133–9146, Toronto, Canada. Association for Computational Linguistics.

Zhiqi Huang, Puxuan Yu, and James Allan. 2023. Improving Cross-lingual Information Retrieval on Low-Resource Languages via Optimal Transport Distillation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, WSDM '23, page 1048–1056, New York, NY, USA. Association for Computing Machinery.

Paul Jaccard. 1912. The Distribution of the Flora in the Alpine Zone. *The New Phytologist*, 11(2):37–50.

W. Kabsch. 1976. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 32(5):922–923.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Phillip Keung, Julian Salazar, Yichao Lu, and Noah A. Smith. 2020. Unsupervised Bitext Mining and Translation via Self-Trained Contextual Embeddings. *Transactions of the Association for Computational Linguistics*, 8:828–841.

Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *International Conference on Learning Representations*.

Yibin Lei, Liang Ding, Yu Cao, Changtong Zan, Andrew Yates, and Dacheng Tao. 2023. Unsupervised Dense Retrieval with Relevance-Aware Contrastive Pre-Training. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10932–10940, Toronto, Canada. Association for Computational Linguistics.

V. I. Levenshtein. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707.

Huayang Li, Deng Cai, Zhi Qu, Qu Cui, Hidetaka Kamigaito, Lemao Liu, and Taro Watanabe. 2024. Cross-lingual Contextualized Phrase Retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6562–6576, Miami, Florida, USA. Association for Computational Linguistics.

Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, Antonios Anastasopoulos, Patrick Littell, and Graham Neubig. 2019. Choosing Transfer Languages for Cross-Lingual Learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3125–3135, Florence, Italy. Association for Computational Linguistics.

Antoine Louis, Vageesh Kumar Saxena, Gijs van Dijck, and Gerasimos Spanakis. 2025. ColBERT-XM: A Modular Multi-Vector Representation Model for Zero-Shot Multilingual Information Retrieval. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 4370–4383, Abu Dhabi, UAE. Association for Computational Linguistics.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, Dense, and Attentional Representations for Text Retrieval. *Transactions of the Association for Computational Linguistics*, 9:329–345.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, Benjamin Bossan, and Marian Tietz. 2022. PEFT: State-of-the-art Parameter-Efficient Fine-Tuning methods. `https://github.com/huggingface/peft`.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Wes McKinney. 2010. Data Structures for Statistical Computing in Python. *SciPy 2010*.

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

Suraj Nair, Eugene Yang, Dawn Lawrie, Kevin Duh, Paul McNamee, Kenton Murray, James Mayfield, and Douglas W. Oard. 2022. Transfer Learning Approaches for Building Cross-Language Dense Retrieval Models. In *Advances in Information Retrieval*, pages 382–396, Cham. Springer International Publishing.

Long S. T. Nguyen, Tran T. B. Le, Huong P. N. Nguyen, Quynh T. N. Vo, Phong H. N. Nguyen, and Tho T. Quan. 2025. Serving the Underserved: Leveraging BARTBahnar Language Model for Bahnaric-Vietnamese Translation. In *Proceedings of the 1st*

Workshop on Language Models for Underserved Communities (LM4UC 2025), pages 32–41, Albuquerque, New Mexico. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation Learning with Contrastive Predictive Coding.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, and 2 others. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA. Curran Associates Inc.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.*, 12:2825–2830.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How Multilingual is Multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA. http://is.muni.cz/publication/884893/en.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021. WikiMatrix: Mining 135M Parallel Sentences in 1620 Language Pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online. Association for Computational Linguistics.

Nasredine Semmar, Meriama Laib, and Christian Fluhr. 2006. Using Cross-language Information Retrieval for Sentence Alignment. In *Proceedings of the International Conference on the Challenge of Arabic for NLP/MT*, pages 95–104, London, UK.

Nguyen Luong Tran, Duong Minh Le, and Dat Quoc Nguyen. 2022. BARTpho: Pre-trained Sequence-to-Sequence Models for Vietnamese. In *Proceedings of the 23rd Annual Conference of the International Speech Communication Association*.

Nattapol Trijakwanich, Peerat Limkonchotiwat, Raheem Sarwar, Wannaphong Phatthiyaphaibun, Ekapol Chuangsuwanich, and Sarana Nutanong. 2021. Robust Fragment-Based Framework for Cross-lingual Sentence Retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 935–944, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hoang Nhat Khang Vo, Duc Dong Le, Tran Minh Dat Phan, Tan Sang Nguyen, Quoc Nguyen Pham, Ngoc Oanh Tran, Quang Duc Nguyen, Tran Minh Hieu Vo, and Tho Quan. 2024. Revitalizing Bahnaric Language through Neural Machine Translation: Challenges, Strategies, and Promising Outcomes. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press.

Rui Wang, Xu Tan, Renqian Luo, Tao Qin, and Tie-Yan Liu. 2021a. A Survey on Low-Resource Neural Machine Translation. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4636–4643. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Shuai Wang, Shengyao Zhuang, and Guido Zuccon. 2021b. BERT-based Dense Retrievers Require Interpolation with BM25 for Effective Passage Retrieval. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*, ICTIR '21, page 317–324, New York, NY, USA. Association for Computing Machinery.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. In *Proceedings*

of the 34th International Conference on Neural Information Processing Systems, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Ning Wu, Yaobo Liang, Houxing Ren, Linjun Shou, Nan Duan, Ming Gong, and Daxin Jiang. 2022. Unsupervised Context Aware Sentence Representation Pretraining for Multi-lingual Dense Retrieval. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 4411–4417. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Chen Zhang, Xiao Liu, Jiuheng Lin, and Yansong Feng. 2024. Teaching Large Language Models an Unseen Language on the Fly. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8783–8800, Bangkok, Thailand. Association for Computational Linguistics.

A Representative Data Examples

The examples below illustrate the lexical expressions and sentence styles represented in the resources used for training and evaluation.

Lexicon examples. Table 3 shows representative Bahnar–Vietnamese lexicon entries, which include single-word mappings and multi-word expressions.

Table 3: Representative lexicon entries.

Bahnar	Vietnamese	English gloss
'bia	*mượt mà*	smooth
'biah	*bằng phẳng*	flat
'blĕk 'blŏk	*đổi ý bất thường*	change one's mind unpredictably
'brui	*chẳng có gì cả*	nothing at all
'bap 'bap	*mấp máy môi*	move one's lips slightly

Sentence-level examples. Table 4 shows representative sentence-level Bahnar–Vietnamese examples from the dataset, with English translations provided for readability. The administrative and instructional examples contain longer clauses and specialized vocabulary, while the narrative examples illustrate other sentence styles.

B Metrics and Statistical Analysis

This section defines the metrics and statistics reported in the paper. Let $\mathcal{Q}$ be the evaluation queries and $\mathrm{rank}(q)$ the rank of the documented counterpart of q in the list returned for q, with $\mathrm{rank}(q) = \infty$ when the counterpart is not returned.

Retrieval metrics. In the primary benchmark, Recall@t at cutoff t and MRR are

$$\begin{aligned} \mathrm{Recall@}t &= \frac{1}{|\mathcal{Q}|}\sum_{q\in\mathcal{Q}} \mathbf{1}[\mathrm{rank}(q) \le t], \\ \mathrm{MRR} &= \frac{1}{|\mathcal{Q}|}\sum_{q\in\mathcal{Q}} \frac{\mathbf{1}[\mathrm{rank}(q) \le 10]}{\mathrm{rank}(q)}. \end{aligned} \tag{9}$$

Here, $\mathbf{1}[\cdot]$ is one when its condition holds and zero otherwise; a counterpart not returned contributes zero. Because each query has one documented counterpart, Recall@t is the hit rate at cutoff t, and Recall@1 is top-1 accuracy. MRR in the primary benchmark is truncated at the ten highest-ranked candidates. In the candidate-size sensitivity analysis, the MRR cutoff instead follows K: reciprocal rank is computed over all K reranked candidates. For the hybrid, $\mathrm{rank}(q)$ is the position after reranking within $\mathcal{C}_K(q)$.

First-stage coverage. First-stage Recall@K is the fraction of queries whose counterpart is in $\mathcal{C}_K(q)$. Reranking only reorders $\mathcal{C}_K(q)$, so this value bounds hybrid Recall@t for every t, as illustrated in Figure 8. For $t \ge K$, hybrid Recall@t equals first-stage Recall@K.

Paired differences. For systems M_1 and M_2, let μ be the per-query metric: $\mathbf{1}[\mathrm{rank}(q) = 1]$ for Recall@1 or the truncated reciprocal rank in Equation 9 for MRR. Equation 10 defines their mean paired difference:

$$\Delta = \frac{1}{|\mathcal{Q}|}\sum_{q\in\mathcal{Q}} d(q), \quad d(q) = \mu_{M_1}(q) - \mu_{M_2}(q). \tag{10}$$

The paired bootstrap resamples $\mathcal{Q}$ with replacement 10,000 times and reports the 2.5th and 97.5th percentiles of the resampled Δ as the 95% CI. The sign-flip test draws 10,000 vectors of independent signs $\varepsilon_q \in \{-1, +1\}$. It estimates the two-sided p-value as the fraction of draws for which $\left|\frac{1}{|\mathcal{Q}|}\sum_q \varepsilon_q d(q)\right|$ is at least $|\Delta|$.

McNemar test. For two systems M_1 and M_2, let n_{10} be the number of queries that only M_1 answers correctly at rank 1, n_{01} the number that only M_2 answers correctly, and $n_{\mathrm{d}} = n_{10} + n_{01}$. Under the

Table 4: Representative Bahnar–Vietnamese sentence pairs with English translations.

Domain	Bahnar	Vietnamese	English translation
Administrative-style	*Sở nông nghiệp păng tơ iung pơ lei đei xơ nong tổng hợp, pơ tâng uỷ ban nhân dân tỉnh, Bộ nông nghiệp păng tơ iung pơ lei adroi năr 25 rim khei (phara khei 12 pơ truh pơ tâng adroi năr 10) Păng pơ tâng đột xuất ah đei hơ 'nha.*	*Sở Nông nghiệp và Phát triển nông thôn có trách nhiệm tổng hợp, báo cáo Ủy ban nhân dân tỉnh, Bộ Nông nghiệp và Phát triển nông thôn trước ngày 25 hàng tháng (riêng tháng 12 gửi báo cáo trước ngày 10) và báo cáo đột xuất khi có yêu cầu.*	*The Department of Agriculture and Rural Development is responsible for compiling and reporting to the Provincial People's Committee and the Ministry of Agriculture and Rural Development before the 25th of each month (for December, reports must be submitted before the 10th), and for submitting ad hoc reports when requested.*
Agricultural and technical instruction	*Wei năng rim năr; rim năr a thai ling lang năng xu tơ drong hoat động dih ka vă đei tơ drong xư lý ko tă ah tình huống krup oei lech. Năng xu mă tơ pă dơng tơ drong xa truh truh tơ drong hơ blih môi trường(mă loi, PH hơ duh mê).*	*Quản lý hằng ngày: Hằng ngày phải thường xuyên theo dõi quá trình hoạt động của cá để có biện pháp xử lý kịp thời khi tình huống xấu xảy ra. Theo dõi thật kỹ từ cách cho ăn đến sự biến đổi môi trường (đặc biệt pH, khí độc).*	*Daily management: Each day, the activity of the fish must be monitored regularly so that timely measures can be taken when adverse situations occur. Monitoring should be very careful, from feeding practices to changes in the environment, especially pH and toxic gases.*
Everyday narrative	*Kcma dang ei bë bon wih ta palei bon, anhàk ba hloi bre bomà bon anoh!*	*Cho nên chúng ta sẽ về làng, dẫn theo cả hai đầy tớ chúng ta đó.*	*Therefore, we will return to the village and take both of our servants with us.*
Religious-style narrative	*Hăp tok ko jung truh dôm linh tơ plenh, ăm ko minh pah linh noh păng minh pah dôm xơ nglong a thai văr truh teh, noh hăp joa jâng tok.*	*Nó vươn cao tới tận các đạo binh trên trời, làm cho một phần đạo binh đó và một phần các tinh tú phải nhào xuống đất, rồi nó đạp chân lên.*	*It grew high, reaching the armies of heaven; it caused part of that army and some of the stars to fall to the ground, and then it trampled on them.*

null hypothesis, $n_{10} \sim \mathrm{Binomial}(n_{\mathrm{d}}, \frac{1}{2})$. Equation 11 gives the exact two-sided p-value:

$$p = \min\Bigg(1,\ 2 \sum_{j=0}^{\min(n_{10},n_{01})} \binom{n_{\mathrm{d}}}{j} 2^{-n_{\mathrm{d}}}\Bigg). \quad (11)$$

Variation across seeds. Seed variation is reported as the mean $\pm$ one sample *Standard Deviation* (SD) over the $S = 3$ seeds:

$$\mathrm{SD} = \sqrt{\frac{1}{S-1}\sum_{j=1}^{S}(a_j - \bar{a})^2}.$$

Here, a_j is the metric value for run j, and $\bar{a}$ is the mean across runs.

Cross-pair correlation. The cross-pair association uses Spearman's rank correlation:

$$\rho = 1 - \frac{6\sum_{i=1}^{L}\delta_i^2}{L(L^2-1)},$$

where $L = 4$ is the number of language pairs and δ_i is the rank difference for pair i between neural Recall@1 and ΔRecall@1 (hybrid minus neural-first-stage Recall@1). Neither variable has tied ranks in this comparison.

Surface overlap and token statistics. For a sentence pair (x, y), overlap is the Jaccard index $J(\mathcal{S}, \mathcal{T}) = |\mathcal{S} \cap \mathcal{T}|/|\mathcal{S} \cup \mathcal{T}|$, where $\mathcal{S}$ and $\mathcal{T}$ are the normalized whitespace-token sets, or the character-trigram sets, of x and y, respectively. We report the mean over evaluation pairs and use character-trigram Jaccard to divide the test pairs into the quartiles shown in Figure 2. The same-script rate is the fraction of evaluation pairs whose two sides are written in the same script. In Table 12, average tokens are counted per training target, the vocabulary is the number of distinct token types, and the hapax rate is the fraction of types that occur once.

C Baseline Systems

This section describes the comparison systems of Section 4. Table 7 lists the evaluated and selected settings of each family.

TF-IDF and BM25. Both represent each sentence as a bag of character n-grams (n from 2 to 5 or from 3 to 6). TF-IDF weights n-grams by sublinear term frequency and inverse document frequency and ranks candidates by cosine similarity (Salton and Buckley, 1988). BM25 uses $k_1 = 1.5$ for term-frequency saturation and $b = 0.75$ for length normalization (Robertson and Zaragoza, 2009).

Levenshtein distance and token Jaccard. The Levenshtein distance $\mathrm{dist}(q, c)$ counts the minimum number of character insertions, deletions, and substitutions that turn one string into the other (Levenshtein, 1966); the Levenshtein ratio rescales it to a similarity, $1 - \mathrm{dist}(q, c)/\max(|q|, |c|)$. Lengths here are measured in characters, whereas the IBM1 length penalty uses token counts. Token Jaccard ranks candidates by the Jaccard index of the two token sets (Jaccard, 1912).

fastText-Procrustes. We train 100-dimensional skip-gram fastText embeddings (Bojanowski et al., 2017) separately on the Bahnar and Vietnamese sides of the sentence training pairs (window 5, 20 epochs). A sentence is embedded by mean or IDF-weighted pooling of its word vectors. For Procrustes alignment, the source and target matrices

X and Y contain paired embeddings from the lexicon training split as rows, using either full lexicon entries or entries in which both sides contain a single token. The mapping is the orthogonal solution $W = UV^\top$ of $\min_{W^\top W=I}\|XW - Y\|_F$, where $U\Sigma V^\top$ is the singular value decomposition of $X^\top Y$, I is the identity matrix, and $\|\cdot\|_F$ is the Frobenius norm (Artetxe et al., 2017). The optional VecMap-style normalization applies unit-length scaling and mean centering before mapping.

IBM1. IBM1 estimates the lexical translation probabilities $p(v \mid b)$ and $p(b \mid v)$ by five iterations of *Expectation Maximization* (EM) on the training pairs (Brown et al., 1993). Equations 4–6 convert these word-level probabilities into sentence scores, combining the forward and reverse directions and applying the selected length penalty.

Off-the-shelf encoders. We use the mBERT (Devlin et al., 2019), XLM-R base (Conneau et al., 2020), multilingual MiniLM (Wang et al., 2020; Reimers and Gurevych, 2019), and LaBSE (Feng et al., 2022) checkpoints without further training. mBERT and XLM-R sentence embeddings are mean-pooled token states; the two sentence-transformer models use their native pooling.

LAMIR. LAMIR trains a single-vector dual encoder with a contrastive objective for language-agnostic multilingual retrieval (Hu et al., 2023). We use an XLM-R base backbone and select the 50-epoch checkpoint on development data.

DPR-XM. DPR-XM is the single-vector dense passage retriever (Karpukhin et al., 2020) built on the *Cross-lingual Modular* (X-MOD) base encoder used by ColBERT-XM (Louis et al., 2025). Single-vector sentence embeddings are scored by cosine similarity; the 20-epoch checkpoint is selected.

ColBERT-X and ColBERT-XM. These models use MaxSim late interaction to score a candidate by $\sum_i \max_j \mathbf{e}_{q,i}^\top \mathbf{e}_{c,j}$, where $\mathbf{e}_{q,i}$ and $\mathbf{e}_{c,j}$ are the vectors of query token i and candidate token j. Each query token contributes its highest dot-product similarity to any candidate token. ColBERT-X uses an XLM-R base backbone (Nair et al., 2022), and ColBERT-XM uses the X-MOD base backbone (Louis et al., 2025). Checkpoints are evaluated every 5,000 steps; the selected checkpoints are at 50,000 and 60,000 steps, respectively.

Full XLM-R fine-tuning. Separate Bahnar and Vietnamese encoders are initialized from XLM-R base and fine-tuned with all parameters. We compare sentence or token pooling and identity, linear, or *Multi-Layer Perceptron* (MLP) projections, using either the symmetric InfoNCE objective in Equation 7 or its one-way variant.

D Method and Implementation Details

Algorithm 1 summarizes the adaptation procedure in Section 4. Here, N is the number of training pairs; superscripts B and V identify the Bahnar and Vietnamese encoders f and projection heads g. The symbol θ collectively denotes the LoRA parameters, while ϕ and ψ parameterize the projection heads. Training pooling m_{train} uses sentence mean or token mean; retrieval pooling m additionally supports token IDF. The optional Kabsch rotation R and translation $\mathbf{t}$ are learned from the lexicon training split. Token-level alignment requires token mean or token IDF pooling.

D.1 Configuration Search and Selection

The searches below follow the data restrictions and selection rule in Section 5.1. Both fastText-Procrustes mapping and optional Kabsch alignment are learned from the 8,285-entry lexicon training split.

DPR-XM leads the modern dense development comparison. Table 5 summarizes the development results of the four modern dense retrievers, all trained on the same 46,736 training pairs. DPR-XM achieves the highest development Recall@1 and represents this family on the test set.

Configuration search. Table 6 summarizes the development search for each method family; the grid-searched families total 534 configurations. Each fixed base representation in an embedding family is evaluated with cosine, CSLS, and ratio margin. The fastText-Procrustes, off-the-shelf encoder, XLM-R LoRA, and full XLM-R grids contain 7, 8, 8, and 18 base representations, respectively. The modern dense search compares four architectures, selecting a checkpoint within each as reported in Table 5. The hybrid grid contains 192 configurations for each stage order (XLM-R-first and IBM1-first), for 384 configurations in total. Table 7 lists the selected settings.

D.2 Implementation Settings

Preprocessing. We apply *Normalization Form C* (canonical decomposition followed by canonical composition) to all text, standardize apostrophe-like and dash variants, and collapse repeated whitespace. Accent stripping uses *Normalization Form*

Algorithm 1: Projection-based XLM-R adaptation and retrieval. θ denotes the LoRA parameters. Pool_m projects after mean pooling (m = sentence mean) or before mean or IDF pooling (m = token mean, token IDF).

Input: training pairs $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{N}$, lexicon training split $\mathcal{D}_{\text{lex}}$, queries $\mathcal{Q}$, candidate pool $\mathcal{C}$, training and retrieval pooling modes m_{train} and m, alignment setting (none, token, sentence), retrieval score $s \in \{\cos, s_{\text{CSLS}}, s_{\text{margin}}\}$

Output: ranking of $\mathcal{C}$ for each $q \in \mathcal{Q}$

$f_\theta^B, f_\theta^V \leftarrow \mathsf{AddLoRA}(\text{XLM-R})$; initialize projection heads g_ϕ^B, g_ψ^V;

foreach minibatch $\{(x_i, y_i)\}_{i=1}^{B} \subset \mathcal{D}_{\text{train}}$ across all training epochs **do**

  $\mathbf{u}_i \leftarrow \mathsf{Pool}_{m_{\text{train}}}(g_\phi^B, f_\theta^B(x_i))$; $\mathbf{v}_i \leftarrow \mathsf{Pool}_{m_{\text{train}}}(g_\psi^V, f_\theta^V(y_i))$;

  update θ, ϕ, ψ to minimize $\mathcal{L}$ in Equation 7;

if Kabsch alignment is enabled **then**

  $(R, \mathbf{t}) \leftarrow \mathsf{Kabsch}(\mathcal{D}_{\text{lex}})$;

if token-level Kabsch is enabled **then**

  map each projected source token vector $\mathbf{z}$ to $R\mathbf{z} + \mathbf{t}$ inside Pool_m before pooling;

$\mathbf{u}_q \leftarrow \mathsf{Pool}_m(g_\phi^B, f_\theta^B(q))$ for $q \in \mathcal{Q}$; $\mathbf{v}_c \leftarrow \mathsf{Pool}_m(g_\psi^V, f_\theta^V(c))$ for $c \in \mathcal{C}$;

if sentence-level Kabsch is enabled **then**

  $\mathbf{u}_q \leftarrow R\,\mathbf{u}_q + \mathbf{t}$ for $q \in \mathcal{Q}$;

foreach $q \in \mathcal{Q}$ **do**

  rank $c \in \mathcal{C}$ by $s(q, c)$ computed from $\cos(\mathbf{u}_q, \mathbf{v}_c)$; *// Equations 2–3, $k = 10$*

Table 5: Development Recall@1/5/10 of the four in-domain modern dense retrievers, all trained on the same 46,736 training pairs, and their selected checkpoints.

Method	Checkpoint	R@1	R@5	R@10
DPR-XM	20 epochs	**0.5766**	**0.7403**	**0.7869**
LAMIR	50 epochs	0.5381	0.7147	0.7711
ColBERT-X	50,000 steps	0.4471	0.5941	0.6450
ColBERT-XM	60,000 steps	0.4224	0.5668	0.6136

Table 6: Development search size (#) and selected Recall@1 by method family. Counts refer to architectures for modern dense retrieval and configurations otherwise.

Family	#	Dev R@1
Surface lexical	16	0.0637
Edit distance	6	0.0760
fastText-Procrustes	21	0.1916
IBM1	5	0.4080
Off-the-shelf encoders	24	0.0630
Modern dense retrieval	4	0.5766
XLM-R LoRA	24	0.5094
Full XLM-R	54	0.2998
Hybrid	384	**0.6103**

D (canonical decomposition) followed by removal of combining marks. Punctuation removal replaces non-word, non-space characters with whitespace. Table 7 lists the evaluated settings and highlights selected values. We evaluate subsets of these combinations, with counts in Table 6.

Implementation and retrieval. All first-stage methods score the full development or test pool using the representations in Appendix C. For families evaluated with cosine, CSLS, and ratio margin, we precompute embeddings and the full cosine-similarity matrix. CSLS and ratio-margin neighborhood statistics ($k = 10$) are computed before ranking and top-K truncation. We use $\epsilon_{\text{num}} = 10^{-8}$ for the ratio-margin denominator, $\epsilon = 10^{-9}$ for IBM1 lexical probabilities, and $\epsilon_{\text{norm}} = 10^{-12}$ for min–max normalization. The reranker reuses the IBM1 lexical tables to score retained candidates.

Hardware and software. We implement all experiments in Python 3.11 with NumPy (Harris et al., 2020), pandas (McKinney, 2010), scikit-learn (Pedregosa et al., 2011), gensim (Řehůřek and Sojka, 2010), PyTorch (Paszke et al., 2019), Hugging Face Transformers (Wolf et al., 2020), SentenceTransformers (Reimers and Gurevych, 2019), and PEFT (Mangrulkar et al., 2022) as applicable. Neural jobs use one GPU, four CPU cores, and 16 GB of memory. Maximum sequence lengths are 256 for XLM-R LoRA and 128 for DPR-XM. Table 8 lists the XLM-R LoRA and hybrid hyperparameters.

Table 7: Evaluated settings by method family; highlighted values are development-selected. A, P, and S denote removal of accents, punctuation, and spaces.

Family	Setting	Evaluated values
Surface lexical	Scoring	TF-IDF, BM25
	Character n-grams	2–5, 3–6
	Text normalization	raw, A, A+P, A+P+S
Edit distance	Measure	Levenshtein distance, Levenshtein ratio, token Jaccard
	Text normalization	raw, A, A+P, A+P+S
fastText-Procrustes	Alignment unit	full lexicon entry, single-token entry
	Pooling	mean, IDF
	Embedding normalization	none, VecMap
	Text normalization	raw, P, A+P
	Retrieval criterion	cosine, CSLS, ratio margin
IBM1	Direction	forward, symmetric
	Text normalization	raw, A, A+P
	Length penalty λ	0, 0.1
Off-the-shelf encoders	Model	mBERT, XLM-R, multilingual MiniLM, LaBSE
	Text normalization	raw, A+P
	Retrieval criterion	cosine, CSLS, ratio margin
Modern dense retrieval	Architecture	LAMIR, DPR-XM, ColBERT-X, ColBERT-XM
	Checkpoint	selected per architecture; 20 epochs for DPR-XM
XLM-R LoRA	Pooling	sentence mean, token mean, token IDF
	Kabsch alignment	none, sentence-level, token-level
	Retrieval criterion	cosine, CSLS, ratio margin
	Epochs	50
Full XLM-R	Pooling	sentence, token
	Projection	identity, linear-256, MLP-128, MLP-256
	Objective	one-way, symmetric InfoNCE
	Temperature / learning rate	0.05, 0.07 / 5×10^{-6}, 10^{-5}
	Epochs	3, 5, 10
	Retrieval criterion	cosine, CSLS, ratio margin
Hybrid	Stage order	IBM1-first, XLM-R-first
	Candidate-set size K	5, 10
	Neural score	raw, reciprocal rank
	Score normalization	min–max, rank
	Combination	weighted sum, weighted product
	Weights (α, β), $\gamma = 0$	(1, 0), (0.75, 0.25), (0.5, 0.5), (0.25, 0.75), (0, 1)
	Weights (α, β, γ), $\gamma > 0$	(0.5, 0.4, 0.1)
	Length penalty λ	0, 0.1

E Development and Overlap Analysis

The development-set comparisons below explain the configuration choices evaluated in Table 2. A separate held-out surface-overlap diagnostic examines where the selected systems succeed.

Development and test pools. Table 6 reports the development Recall@1 of each selected configuration. The development pool contains 5,194 candidates, whereas the test pool contains 2,001. De-

Table 8: Training and reranking hyperparameters of XLM-R LoRA and the hybrid. Bahnar denotes Bahnar–Vietnamese, and Transfer the other three pairs.

Hyperparameter	Bahnar	Transfer
XLM-R LoRA		
Initialization	XLM-R base	XLM-R base
LoRA rank	8	8
LoRA scaling	16	16
LoRA dropout	0.05	0.05
Pooling	token mean	token mean
Projection dimension	256	256
Max. sequence length	256	256
Epochs	50	10
Learning rate	2×10^{-4}	2×10^{-4}
Temperature τ	0.07	0.07
Seed	42	42
Retrieval		
Criterion	ratio margin	CSLS
Neighborhood size k	10	10
Kabsch alignment	none	none
Hybrid		
Candidate-set size K	10	10
Weights (α, β, γ)	(0.25, 0.75, 0)	(0.5, 0.5, 0)
Length penalty λ	0.1	0.1
IBM1 EM iterations	5	5

velopment scores therefore compare configurations within that pool; differences from test scores also reflect the change in candidate pool.

The best retrieval criterion depends on the representation. Neighborhood correction generally improves over cosine, but neither CSLS nor ratio margin dominates across representations. Table 9 compares the three criteria with representation and preprocessing held fixed within each row. Under the selection order in Appendix D, LaBSE ties on Recall@1 under CSLS and ratio margin, with ratio margin selected by MRR. Full XLM-R ties on both metrics, with CSLS selected by Recall@5.

Hybrid stage order. Figure 6 compares the best development configuration for each stage order. The selected score weights also differ: in Equation 8, the IBM1-first configuration uses $\alpha = 0.75$ and $\beta = 0.25$, and the XLM-R-first configuration uses $\alpha = 0.25$ and $\beta = 0.75$. The stronger development results of the XLM-R-first configuration motivate generating candidates with XLM-R LoRA and reranking them with IBM1, the pipeline evaluated on the test set in Section 5.2.

Table 9: Development Recall@1 under cosine, CSLS, and ratio margin (Margin). Scores are compared within each row; tied scores share a mark.

Representation	Cosine	CSLS	Margin
fastText-Procrustes	0.1323	0.1902	**0.1916**
mBERT	0.0347	**0.0356**	0.0348
Off-the-shelf XLM-R	0.0239	0.0275	**0.0277**
Multilingual MiniLM	0.0273	**0.0306**	0.0287
LaBSE	0.0522	**0.0630**	**0.0630**
XLM-R LoRA	0.4686	0.5079	**0.5094**
Full XLM-R	0.2738	**0.2998**	**0.2998**

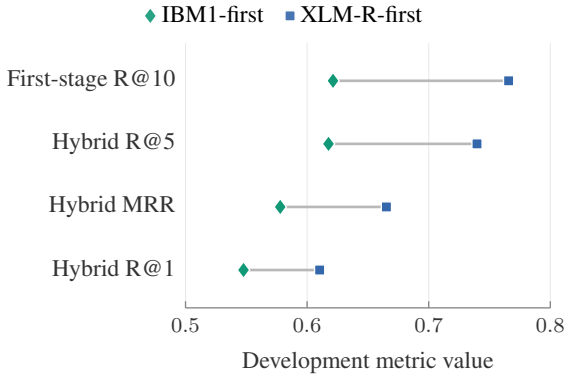


Figure 6: Development results of the best hybrid configuration for each first-stage retriever, both with $K = 10$.

BM25 and fastText reverse their ordering across overlap quartiles. We analyze the development-selected BM25 and fastText-Procrustes systems on the held-out test set to examine how their relative performance varies with surface overlap. Table 10 reports mean token and character-trigram Jaccard overlap after normalization. Figure 2 separates the pairs into character-trigram-overlap quartiles, revealing the reversal in method ordering between low- and high-overlap pairs. Q1 contains the 498 queries with the lowest overlap; Q2–Q4 contain 501 queries each.

F Uncertainty and Sensitivity to K

We examine three aspects of the hybrid results: uncertainty in paired comparisons, variation across training seeds, and sensitivity to K.

F.1 Paired Comparisons

We compare the development-selected XLM-R-first IBM1 hybrid with symmetric IBM1 and with

its XLM-R LoRA first stage using matched outcomes on the 2,001 held-out queries. Following standard practice for significance testing in NLP (Dror et al., 2018), we use exact McNemar tests (McNemar, 1947) for Recall@1, paired bootstrap (Efron and Tibshirani, 1994; Koehn, 2004) 95% CIs for differences in Recall@1 and MRR, and paired sign-flip tests. The bootstrap and sign-flip procedures each use 10,000 samples. Figure 3 shows the hybrid-minus-comparator differences with their CIs.

Paired tests support the gain over the neural first stage. Recall@1 improves by 0.1514 with a 95% CI of $[0.13343, 0.17042]$, and MRR improves by 0.1117 with a CI of $[0.09876, 0.12519]$. Both intervals exclude zero. At rank 1, 359 queries are answered correctly only by the hybrid and 56 only by the first stage. The exact McNemar test gives $p = 3.29 \times 10^{-55}$, and both sign-flip tests give $p < 0.001$ as well.

The gain over symmetric IBM1 is not statistically significant. The hybrid has higher point estimates than symmetric IBM1, but the paired differences are not significant at the 0.05 level. The Recall@1 gain is 0.0230 with a 95% CI of $[-0.00001, 0.04748]$; the MRR gain is 0.0120 with a CI of $[-0.01019, 0.03461]$. Both intervals include zero. At rank 1, 317 queries are answered correctly only by the hybrid and 271 only by IBM1. The exact McNemar test gives $p = 0.0634$, and the Recall@1 sign-flip test gives $p = 0.0653$.

F.2 Variation Across Training Seeds

Reranking improves retrieval in each of the three training runs shown in Figure 7. We repeat the development-selected XLM-R LoRA configuration with seeds 42, 13, and 87, retaining all runs and fixing the architecture, preprocessing, retrieval criterion, and $K = 10$ hybrid configuration. Mean Recall@1 increases from 0.3951 ± 0.0265 for the neural first stage to 0.5440 ± 0.0280 for the hybrid, where variation is one sample SD. The corresponding MRR values are 0.4757 ± 0.0261 and 0.5840 ± 0.0257. Recall@1 gains range from 0.1369 to 0.1584 across the runs, and all three hybrid runs exceed symmetric IBM1 (0.4933).

F.3 Sensitivity to Candidate Size

Sensitivity setup. To characterize the candidate-recall bottleneck, we rerun the selected hybrid with $K \in \{1, 3, 5, 10, 20, 50, 100\}$, keeping its stage order, retrieval criterion, normalization, interpolation weights, and IBM1 length penalty fixed. Values for $K \neq 10$ are post hoc and do not change the selected configuration, which uses $K = 10$.

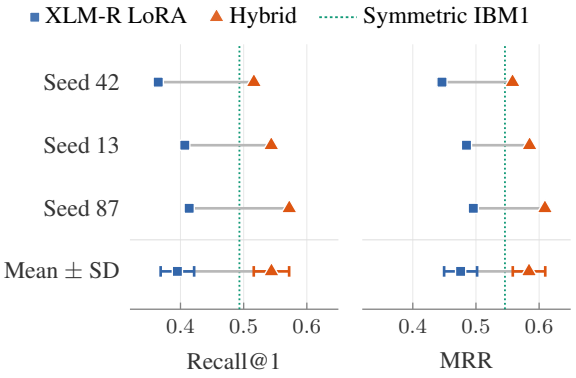


Figure 7: Recall@1 and MRR of the XLM-R LoRA first stage and the hybrid for three training seeds on the 2,001-query test set. The bottom row shows the mean ± 1 sample SD; the dotted line is symmetric IBM1.

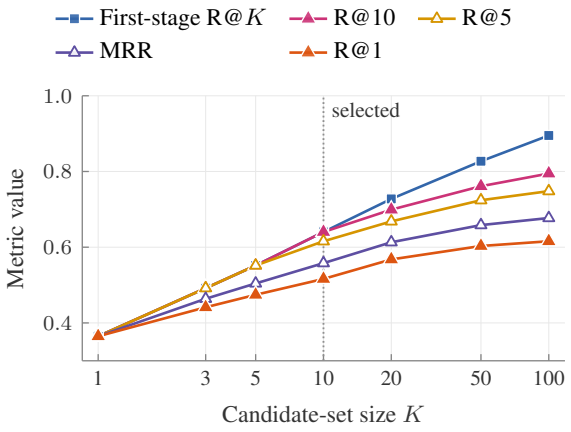


Figure 8: Hybrid retrieval and first-stage coverage (Recall@K) on the 2,001-query test set. MRR uses all K retained candidates. The dotted line marks the selected $K = 10$; other values are post hoc.

Coverage and metric cutoffs. The first-stage Recall@K curve gives the fraction of queries whose counterpart is retained for reranking and therefore provides an upper bound on recall attainable from that candidate set. Figure 8 reports how retrieval performance and first-stage candidate coverage vary with K. When an evaluation cutoff exceeds K, recall is computed over the available candidates, so hybrid Recall@5 and Recall@10 equal the first-stage ceiling for $K \leq 5$ and $K \leq 10$, respectively; MRR is computed over the K reranked candidates. At $K = 1$, reranking cannot change the first-stage decision, so every metric equals 0.3648. Section 5.2 relates the gains at larger K to the in-

Table 10: Data sizes and surface-overlap statistics of the four language pairs. Each evaluation set is also the candidate pool; for Bahnar–Vietnamese, it is the held-out test set.

	Statistic	Bahnar–Vietnamese	Khmer–Vietnamese	Lao–Vietnamese	Zhuang–Chinese
Data	Training pairs	46,736	997	997	4,944
	Development pairs	5,194	–	–	–
	Evaluation pairs	2,001	1,012	1,012	200
Overlap	Token Jaccard	0.0986	0.0186	0.0359	0.0018
	Char. 3-gram Jaccard	0.1114	0.0207	0.0246	0.0001
	Same-script rate	1.0000	0.0020	0.0069	0.0000

Table 11: Retrieval results on FLORES Khmer–Vietnamese and Lao–Vietnamese, and Zhuang–Chinese. Scores are compared within each pair.

Pair	Method	R@1	MRR	R@5	R@10
Khmer–Vietnamese	Off-the-shelf XLM-R, cosine	0.1700	0.2442	0.3468	0.4427
	Off-the-shelf XLM-R, CSLS	0.5030	0.5806	0.6828	0.7569
	XLM-R LoRA, cosine	0.7441	0.8233	0.9289	0.9585
	XLM-R LoRA, CSLS	**0.8014**	**0.8631**	**0.9407**	**0.9694**
	Hybrid	0.7638	0.8322	0.9180	**0.9694**
Lao–Vietnamese	Off-the-shelf XLM-R, cosine	0.3320	0.4274	0.5553	0.6512
	Off-the-shelf XLM-R, CSLS	0.6630	0.7368	0.8370	0.8715
	XLM-R LoRA, cosine	0.7905	0.8580	0.9506	0.9694
	XLM-R LoRA, CSLS	**0.8458**	**0.8982**	**0.9654**	**0.9802**
	Hybrid	0.7925	0.8587	0.9496	**0.9802**
Zhuang–Chinese	Off-the-shelf XLM-R, cosine	0.0000	0.0172	0.0400	0.0700
	Off-the-shelf XLM-R, CSLS	0.0100	0.0302	0.0400	0.1050
	XLM-R LoRA, cosine	0.5100	0.6249	0.7750	0.8500
	XLM-R LoRA, CSLS	**0.5700**	**0.6788**	**0.8400**	**0.8850**
	Hybrid	0.2550	0.4712	0.7350	**0.8850**

Table 12: Post-hoc Zhuang–Chinese IBM1 tokenization diagnostic. Token statistics are computed on the 4,944 normalized Chinese training targets. R@10 is identical for all tokenizations and is left unmarked.

Chinese tokenization	Avg. tokens	Vocab.	Hapax	R@1	MRR	R@5	R@10
Whitespace	2.24	9,649	92.5%	0.2550	0.4712	0.7350	0.8850
Character	18.01	2,235	19.3%	**0.6450**	**0.7568**	**0.8850**	0.8850
Chinese BERT WordPiece	17.88	2,376	20.9%	**0.6450**	0.7565	0.8800	0.8850

creased first-stage coverage.

G Transfer Setups and Results

These experiments evaluate models trained separately for each language pair under the setups below. They use the fixed hyperparameters listed in Table 8, with a batch size of 16 and gradient accumulation over 2 steps, without the development selection used for Bahnar–Vietnamese in Section 5.1. Table 10 gives the data sizes and surface overlap of each pair. A separate exploratory analysis evaluates zero-shot checkpoint transfer.

FLORES transfer setup. Khmer–Vietnamese and Lao–Vietnamese use the aligned FLORES development pairs as training data and the FLORES devtest split as the evaluation pool. Each source-

language sentence is a query, and all Vietnamese sentences in this pool are candidates. The hybrid uses fixed 0.5/0.5 neural–IBM1 weights instead of the 0.25/0.75 weights selected on the Bahnar–Vietnamese development data.

Zhuang–Chinese transfer setup. This experiment uses the parallel corpus released with Zhuang-Bench (Zhang et al., 2024), whose separate evaluation set lets each Zhuang query rank all Chinese candidates. The LoRA model is trained on the Zhuang–Chinese pairs rather than transferred from the Bahnar–Vietnamese checkpoint, and the IBM1 component uses whitespace tokenization on the Chinese side. Appendix H examines the effect of this tokenization choice. Table 11 summarizes the transfer results for Khmer–Vietnamese, Lao–Vietnamese, and Zhuang–Chinese retrieval.

Table 13: Exploratory strict Bahnar-checkpoint transfer to Zhuang–Chinese retrieval.

Scoring	R@1	MRR	R@5	R@10
Cosine	0.0450	0.1200	0.1750	0.2700
CSLS ($k = 10$)	**0.0600**	**0.1437**	**0.2100**	**0.3200**

Zero-shot checkpoint transfer. This experiment applies a Bahnar–Vietnamese checkpoint, consisting of LoRA adapters and projection heads, directly to the 200 Zhuang–Chinese evaluation pairs without Zhuang–Chinese training or selection. The checkpoint predates the train/development split used in the main comparison and is evaluated separately. Pair-specific Zhuang–Chinese training gives higher retrieval scores than the checkpoint-transfer results in Table 13.

H Cross-Pair Reranking Diagnostics

Reranking gains vary across the evaluated language pairs. The following post-hoc analyses examine neural headroom, lexical tokenization, and surface overlap to interpret these differences; they are separate from configuration selection.

Neural first-stage headroom. Figure 5 plots neural Recall@1 against the change produced by hybrid reranking to examine whether stronger first-stage rankings leave less room for improvement. For Zhuang–Chinese, the hybrid value comes from the tokenization diagnostic described below. As discussed in Section 6, the observed association is a diagnostic across the evaluated settings, whose data and supervision conditions differ.

Chinese tokenization changes hybrid performance with other settings fixed. To isolate the effect of tokenization on the result in Table 11, we vary Chinese tokenization while holding the 4,944 training pairs, neural top-10 candidates, five IBM1 iterations, preprocessing, length penalty, and 0.5/0.5 interpolation weights fixed. Table 12 reports the resulting lexical statistics and retrieval scores. Whitespace tokenization yields sparse Chinese lexical statistics: 48.9% of training targets contain at most one token, and most vocabulary types are hapaxes. Character tokenization and the WordPiece tokenizer of Chinese BERT (Devlin et al., 2019) both reach 0.6450 Recall@1 and yield identical top-1 predictions on all 200 queries. The improvement with the same training pairs points to tokenization mismatch as the main issue in this setup. These alternatives are post-hoc diagnostics without a Zhuang development set.

Surface overlap alone does not explain cross-pair reranking gains. Table 10 reports surface-overlap statistics for the four language pairs. Relative retrieval performance varies with surface overlap within Bahnar–Vietnamese, as shown in Figure 2, but the cross-pair results do not support using overlap alone as a decision threshold for lexical reranking. In particular, the Zhuang tokenization diagnostic shows a positive hybrid gain even though Zhuang–Chinese has the lowest surface overlap.