Cross-lingual Matryoshka Representation Learning across Speech and Text

Anonymous ACL submission

Abstract

Speakers of under-represented languages face both a **language barrier**, as most online knowledge is in a few dominant languages, and a **modality barrier**, since information is largely text-based while many languages are primarily oral. We address this for French-Wolof by training the first bilingual speech-text Matryoshka embedding model, enabling efficient retrieval of French text from Wolof speech queries without relying on a costly ASR-translation pipelines. We introduce large-scale data curation pipelines and new benchmarks, compare modeling strategies, and show that modality fusion within a frozen text Matryoshka model performs best. Although trained only for retrieval, the model generalizes well to other tasks, such as speech intent detection, indicating the learning of general semantic representations. Finally, we analyze cost-accuracy trade-offs across Matryoshka dimensions and ranks, showing that information is concentrated only in a few components, suggesting potential for efficiency improvements.

1 Introduction

1.1 Motivation

Access to information is limited by two major barriers. First, a *language barrier*: most content is written in a few high-resource languages. Second, a *modality barrier*: information retrieval systems assume text-based queries, while many under-represented languages are primarily oral. Traditional cascaded ASR-translation pipelines are costly and suffer from error propagation. We address this by training cross-lingual speech-text matryoshka representation models, enabling direct retrieval of text documents from speech queries with flexible accuracy-cost trade-offs.

We focus on Wolof-French as a representative and socially grounded case study. Wolof is primarily spoken in Senegal and is mainly oral. Due to colonial history, administrative, educational, and informational content relevant to Wolof speakers is accessible in French, yet many Wolof speakers have limited French literacy. This creates a critical mismatch: information for Wolof speakers exists in French text, while their natural query modality is Wolof speech, making cross-lingual speech-based retrieval a practical solution.

1.2 Related Works

Speech Language Models extend LLMs with speech understanding by integrating a speech encoder (Wu et al., 2023; Pareras et al., 2025; Lam et al., 2025). We similarly add speech capabilities to embedding LLMs, which remains underexplored compared to generative LLMs.

Matryoshka Representation Learning (MRL) (Kusupati et al., 2022) reduces deployment costs by learning representations at multiple dimensions simultaneously, enabling flexible dimension choice at inference. This has been generalized to vision-language and audio-visual LLMs (Hu et al., 2024; Cai et al., 2024; Cappellazzo et al., 2025) for sequence compression at different granularities. We extend this to cross-lingual speech-text retrieval and analyze performance-cost trade-offs.

Multilingual Representation Models (Conneau et al., 2019; Devlin et al., 2018; Tang et al., 2020) demonstrate strong cross-lingual transfer for low-resource languages but are generally not designed for retrieval. Recent work (Schmidt et al., 2024) integrates a massively multilingual machine translation encoder such as NLLB into an LLM to produce cross-lingual representations for over 200 languages. We show that a model 10x smaller, augmented with speech capability and trained mostly on supervised synthetic bilingual data, not only supports retrieval but also learns robust representations that transfer to non-retrieval tasks.

CLIP-style Architectures (Radford et al., 2021) use modality-specific encoders with contrastive ob-

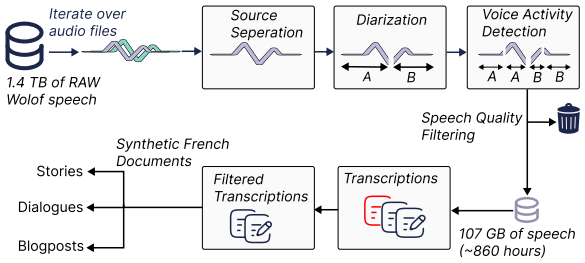

Figure 1: **Speech data pipeline**. Raw data is filtered via *Source Separation*, *Diarization*, *VAD*, and *Quality Filtering*. The resulting speech is transcribed, bad transcriptions (in red) are filtered, then used to generate French *story, dialog* and *blogpost* documents.

jectives. CLAP (Elizalde et al., 2022) extends this to audio-text. We show these dual-encoder models work for simple tasks like transcription retrieval but struggle with semantically demanding cross-lingual speech-to-document retrieval.

2 Datasets

To address the scarcity of Wolof speech-French text retrieval data, we primarily rely on synthetic documents. We first describe the training data collection pipelines for both text-only and speech-text data, then introduce the evaluation datasets. In this work, *query* is any speech or text used as a query for retrieval, whether it is a question or not.

2.1 Training Datasets

Text-only Training Dataset. We curate data from three sources: French mMARCO (Bonifacio et al., 2021) with queries translated to Wolof, Senegalese French webpages, and synthetically generated French queries then translated to Wolof. We augment the training dataset with French QA datasets, where the questions are translated to Wolof. We add Wolof-French translation pairs for cross-lingual transfer. The total text-only training dataset yields 1,176,908 query (Wolof) and document (French) pairs, corresponding to 593,284,495 document tokens.

Collecting Wolof Speech Queries We collect speech data by manually sourcing natural and spontaneous speech (podcasts, radio programs, etc.) by web browsing and listening to audio. We exclude read content like audiobooks. The scraped data over these sources yields 1.4TB of speech, but requires pre-processing for training. We evaluate multiple filtering pipelines and select one that effectively removes long silences, excessive backchannels, and unintelligible overlap. The filtering pipeline is presented in Figure 1. We iterate through the raw 1.4TB of audio and apply, in order: *Source Separation*[1] to isolate speech from other sounds, *Diarization* from `pyannote` (Plaquet and Bredin, 2023; Bredin, 2023) to split audio by speaker, and *Voice Activity Detection* from `Silero-VAD` (Team, 2024) to retain segments with speech. Finally, we keep only utterances between 3-30 seconds with a DNSMOS (Reddy et al., 2021) quality score above 3.2, resulting in 860 hours of high-quality Wolof speech queries.

Synthetic French Documents. Starting from Wolof speech queries, we synthesize French documents. We first transcribe the speech using a Wolof Speech Language Model (Sy et al., 2025), then filter transcriptions based on perplexity and lexical diversity (unique-to-total word ratio), retaining approximately one quarter of the original 860 hours of Wolof speech-text pairs. The filtered Wolof transcriptions are translated into French and used to generate three types of synthetic documents with Gemini-2.5-Flash: stories, dialogues, and web blogposts.

Instruction-Following Dataset. In real-world scenarios, databases are heterogeneous: the same input may correspond to multiple target documents. For example, for the same Wolof speech query, it is possible to retrieve in the database either the corresponding Wolof transcription or the French document. Instruction-following embedding models (Peng et al., 2024) address this by appending a task description to the query. Accordingly, we train a multitask embedding model that can be prompted at inference time, rather than optimizing solely for French document retrieval. In addition to document retrieval, we include *speech translation retrieval* (retrieving French translations from Wolof speech) and *transcription retrieval* (retrieving Wolof transcriptions from speech), which is useful for applications such as keyword spotting. Appendix A.1 details the dataset for these additional tasks.

2.2 Test Datasets

To evalute the models, we introduce two benchmarks for Wolof to French document retrieval. The first is derived from SIB-Fleurs, a benchmark based

[1]we used `https://huggingface.co/seanghay/uvr_models/blob/main/UVR-MDX-NET-Inst_HQ_3.onnx`

on Fleurs test-split. The second benchmark is based on the Kallaama test-split (Gauthier et al., 2024). We detail next both benchmarks.

Artefact 1: Evaluation Datasets for French document retrieval from Wolof queries.

We introduce *Kallaama-Retrieval-Eval* and *Fleurs-Retrieval-Eval*, two datasets for evaluating French document retrieval from Wolof text/speech queries.

Fleurs-Retrieval-Eval Because most of our training data is synthetic, we evaluate on a fully natural benchmark. We start from SIB-Fleurs (Schmidt et al., 2025), a multilingual speech topic classification dataset based on the Fleurs test split (Conneau et al., 2022). Using the French translations of the Wolof speech, we retrieve the most relevant French documents from the web, and manually filter out cases where the scraper fails or returns irrelevant results. The retrieved documents are used as-is, without cleaning or shortening, to reflect real-world usage. The resulting *Fleurs-Retrieval-Eval* dataset contains 166 pairs of Wolof speech and corresponding French text queries.documents.

Kallaama-Retrieval-Eval After manual verification and listening to the Fleurs audio, we found that the Wolof speech is unnatural, as speakers are hesitant and speak quietly. We will empirically support and explain this later in the analysis Section 5. We propose another evaluation dataset that relies on the test split of Kallaama (Gauthier et al., 2024), a dataset where native Wolof speakers converse naturally, and the speech is transcribed by professionals. From the test split, we select the 150 longest speech queries that do not exceed 30 seconds, ensuring information-rich queries while respecting the input length constraints of modern speech models. We translate the transcriptions into French and use Gemini-2.5-Pro to synthetically generate three document types: *dialogues*, *blogposts*, and *stories*. The resulting evaluation set contains 150 Wolof speech and text queries, each paired with three corresponding French documents.

3 Modeling

To enable flexible choice of the dimensions at inference-time, Matryoshka Representation Learning (Kusupati et al., 2022) optimizes a joint retrieval loss across different dimensions. Given

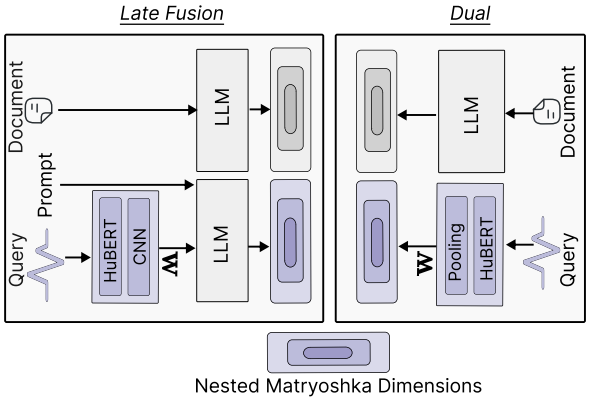

Figure 2: **Late-Fusion vs. Dual architectures**. In the *Late-Fusion* approach, speech is encoded with HuBERT, then the sequence is downsampled by a CNN (x2), projected to the LLM embedding with a matrix **W**, concatenated with the prompt token embeddings, and the whole is forwarded to the Matryoshka embedding LLM. In the *Dual* architecture, HuBERT features are pooled at the sequence level using an attention-based pooler, then projected with dimension-specific **W** matrices to obtain Matryoshka embeddings. In both architectures, text documents are embedded using the text-only Matryoshka embedding LLM.

the training loss $\mathcal{L}$ and the matryoshka dimensions $\mathcal{M} = \{d_1, d_2, \cdots, d_m\}$ where $d_1 < d_2 < \cdots < d_m$, the model optimizes this joint loss:

$$\mathcal{L}_{MRL} = \sum_{m \in \mathcal{M}} \mathcal{L}(\mathbf{Q}_{:m}, \mathbf{D}_{:m})$$

where **Q** is the query embedding matrix, **D** is the document embedding matrix, and $:m$ refers to the PyTorch slicing operator. We use InfoNCE as retrieval loss, with in-batch negative, meaning for each query-document pair, the negatives are all other documents in the batch. Next, we detail the text-only and multimodal models: **late-fusion** and **dual** encoders.

3.1 Text-only Embedding

We use Qwen3-0.6-Embedding, an MRL Embedding LLM that can represent text in dimensions 32, 64, 128, 256, 512, and 1024. Since the training of Matryoshka is costly when there are too many dimensions, due to the loss summation, we only use dimensions 128, 256, 512, and 1024. We first finetune Qwen3-0.6-Embedding on the text-only data presented previously, so the model learns cross-lingual representations between Wolof and French. This model is trained with in-batch InfoNCE loss. We next introduce and study two approaches for in-

tegrating the speech modality within this text-only model.

Artefact 2: Cross-lingual speech and text representation models for Wolof and French

We introduce a series of cross-lingual matryoshka retrieval models for Wolof and French. This includes both text-only and speech-text representation models.

3.2 Late Modality Fusion

In *late-fusion* (Liu et al., 2023; Wu et al., 2023) is a simple and efficient approach for integrating vision or speech capabilities into pre-trained LLMs. In this approach, the speech features from the speech encoder and text token embeddings from the embedding layer are projected separately and then concatenated before being fed into the pretrained language model. The *fusion* is late because the two modalities only interact within the language model, as opposed to *early-fusion*, where the modalities are combined before. For the speech encoder, we use a continued-pretrained Wolof HuBERT model (Sy et al., 2025). Figure 2 illustrates this approach. Wolof query speech features from all HuBERT layers are concatenated, then passed through a CNN to reduce the sequence length and projected to the LLM embedding space using $\mathbf{W}$. The resulting speech embeddings are concatenated at the sequence level with the prompt token embedding and forwarded to the LLM. The document is embedded separately by the LLM. The model is trained end-to-end on the Instruction-Following Speech-Text dataset using a joint InfoNCE loss over the different Matryoshka dimensions. Only the CNN and $\mathbf{W}$ are trained, while HuBERT and the LLM remain frozen.

3.3 Dual Modality

An alternative is a dual-encoder architecture, similar to CLIP (Radford et al., 2021) or CLAP (Elizalde et al., 2022), illustrated on the right of Figure 2. The pipeline is similar to *late-fusion*, except that speech features are not forwarded to the LLM. In this approach, a pooling function is required to map the speech sequence vectors to a single vector. This is unnecessary in late-fusion, which naturally uses the pooling mechanism of Qwen3-0.6B-Embedding, namely the final token representation. For *dual* architectures, we use *attention-based* pooling with a learnable query. We define a pooling query parameter $\mathbf{q} \in \mathbb{R}^{1\times d}$, and given the HuBERT features $\mathbf{X} \in \mathbb{R}^{s\times d}$, the pooling is defined as $\text{softmax}(\frac{\mathbf{q}\mathbf{X}^T}{\sqrt{d}})\mathbf{X}$

Compared to the late-fusion approach, this dual-encoder architecture has notable limitations. First, it is inherently less expressive, as speech queries are represented without the depth of contextualization provided by the pre-trained language model's transformer layers. Second, the dual approach is incompatible with *task prompting*, since the speech query is encoded solely by the speech modules, which cannot process text prompts. So, to provide more expressivity during training, we unfreeze the HuBERT, and in place of the simple slicing, we introduce different trainable linear projections for each matryoshka dimension. We study two different training objectives for the dual architectures.

Dual - *Retrieval*. This first approach trains the dual architecture with in-batch InfoNCE retrieval loss. During training, the text LLM is frozen, contrary to the speech modules which are unfrozen.

Dual - *Query Alignment*. We found that dual approaches trained on document retrieval fail to converge well, likely because the 90M-parameter HuBERT model is too small. We study a simpler alternative: aligning Wolof speech query representations with their transcriptions via distillation. Since French document retrieval from Wolof text queries works well, aligning the representation speech queries with their transcriptions should enable Wolof speech to French document retrieval without direct training of speech to document retrieval. We use a distillation loss consisting of a joint loss of cosine similarity loss (`CosineSimilarity` in PyTorch) and l_1 loss (`L1Loss` in PyTorch).

4 Experiments

Using the Sentence-Transformers library, we implement the four models: *text-only*, *late-fusion*, *dual-retrieval*, and *dual-query alignment*. All models are trained for 1 epoch with batch size 16, maximum length 2048, and learning rate $3 \cdot 10^{-4}$. The *text-only* and *late-fusion* models are trained on their respective instruction datasets. The dual-encoder models use the same data as *late-fusion* but without prompts, since the query is only processed by the speech encoder that cannot process texts.

Approach	dim=4096		dim=1024		dim=512		dim=256		dim=128	
	nDCG@5	nDCG@10	nDCG@5	nDCG@10	nDCG@5	nDCG@10	nDCG@5	nDCG@10	nDCG@5	nDCG@10
NLLB-LLM2Vec	57.98	61.53	–	–	–	–	–	–	–	–
Late-fusion	–	–	**69.85**	**74.49**	66.86	71.04	61.58	67.05	56.13	62.30
Pipelined	–	–	57.09	62.82	54.07	59.14	45.60	51.56	41.21	47.34
Dual – *Retrieval*	–	–	46.96	53.70	45.27	52.48	41.97	49.78	38.47	44.40
Dual – *Query Alignment*	–	–	41.56	47.42	41.02	46.76	38.24	44.18	35.47	40.47

Table 1: nDCG@5 and nDCG@10 document retrieval results on **Kallaama-Retrieval-Eval** for the different approaches across dimensions. NLLB-LLM2Vec is not Matryoshka and uses the full model dimension (4096), while other approaches use matryoshka dimensions. The most performant speech-based approach is colorized and the best scores are in bold.

Approach	dim=4096		dim=1024		dim=512		dim=256		dim=128	
	nDCG@5	nDCG@10	nDCG@5	nDCG@10	nDCG@5	nDCG@10	nDCG@5	nDCG@10	nDCG@5	nDCG@10
NLLB-LLM2Vec	55.98	59.43	–	–	–	–	–	–	–	–
Late-Fusion	–	–	**57.89**	**61.19**	55.03	58.90	50.87	54.43	41.56	46.60
Dual – *Retrieval*	–	–	41.28	45.54	40.18	45.06	39.84	45.17	39.24	44.34
Dual – *Query Alignment*	–	–	38.07	41.82	37.33	41.69	38.04	41.29	34.46	38.53

Table 2: nDCG@5 and nDCG@10 document retrieval results on **Fleurs-Retrieval-Eval** for the different approaches across dimensions. NLLB-LLM2Vec uses the full embedding dimension (4096), while other approaches use matryoshka dimensions. The most performant speech-based approach is colorized and the best scores are in bold.

4.1 Experiment 1: Retrieval Tasks

Evaluation. We evaluate the trained models on Kallaama-Retrieval-Eval and Fleurs-Retrieval-Eval using nDCG, a standard metric to evaluate the ranking of recommender systems. nDCG@k measures the ranking quality of the top k retrieved documents by comparing their graded relevance to an ideal ranking, with higher scores indicating better alignment with the ground truth. We implement the evaluation script using `InformationRetrievalEvaluator` class from Sentence-Transformers.

Baseline. We compare our speech–text retrieval models with NLLB-LLM2Vec (Schmidt et al., 2024)[2] , a massively multilingual text-only encoder built on the NLLB machine translation model and LLaMA3-8B, trained via self-distillation and supporting over 200 languages, including Wolof. NLLB-LLM2Vec has over 8B parameters—about 10x more than our models, and produces fixed 4096-dimensional embeddings, whereas our approach outputs Matryoshka embeddings ranging from 128 to 1024 dimensions. We also include a *pipelined* baseline, where speech is first transcribed and the transcription is then used as a text query with our text-only model.

Results. Table 1 presents retrieval results on the *Kallaama-Retrieval-Eval* dataset. The *late-fusion* model outperforms the text-only NLLB-LLM2Vec baseline despite being 10× smaller and using 4× fewer embedding dimensions. It also surpasses the pipelined approach, which is affected by transcription error propagation. These results indicate that directly leveraging speech features through late fusion reduces error accumulation and yields more effective retrieval. The dual architecture performs poorly overall, although training with a *retrieval* objective yields better results than *Query Alignment*. Table 2 also reports results on the *fleurs-Retrieval-Eval* dataset. While late fusion still performs best, the performance gap across Matryoshka dimensions is larger. As we show later, this is due to the lower speech quality in FLEURS, indicating that higher Matryoshka dimensions are preferable in low-quality speech settings.

Finding 1: *Late-fusion* overcomes *dual* architectures for fast cross-modal adaptation

Compared to *dual* architectures, *late-fusion* shows better cross-modal generalization while having fewer trainable parameters.

4.2 Experiment 2: Speech Keyword Spotting

Evaluation. We evaluate the models on speech keyword spotting using the test split of Urban Bus (DIOP, 2021), a dataset of common places in Dakar city. The task is to detect the station or common place pronounced by the user. This is exactly tran-

[2] https://huggingface.co/fdschmidt93/NLLB-LLM2Vec-Meta-Llama-31-8B-Instruct-mntp-unsup-simcse

scription retrieval, and our models have learned such tasks during training.

Results. Table 3 reports Keyword Spotting Recall and F1 scores across Matryoshka dimensions. The *late-fusion* approach outperforms the *dual* architecture at all dimensions. Compared to the retrieval results in Tables 1 and 2, the performance of dual architectures is higher overall, suggesting that for less semantically demanding tasks such as transcription retrieval, the dual architecture can be a viable solution. Training the dual architecture with the *Query Alignment* objective yields better performance than the *retrieval* objective.

Finding 2: *Dual* models perform well on tasks that are not semantically demanding.

Dual architectures perform poorly on speech-to-document retrieval, but show good results on transcription retrieval tasks, which don't require deep semantic understanding.

4.3 Experiment 3: Unseen Task

So far, we have evaluated performance on tasks seen during training. We now assess the models' generalization to an unseen task using a speech intent detection task in a few-shot setting.

4.3.1 Data

We use WolBanking77 (Kandji et al., 2025), an intent detection dataset with both speech and text queries, where text queries are transcriptions of the speech. The task is to classify each speech request into one of 10 intents.

4.3.2 Method

We evaluate the generalization ability of the *late-fusion* model in zero-shot and few-shot settings. In **the zero-shot** setting, the model encodes the intent labels and the user's speech request separately, and classifies the intent of the speech as the label with the highest similarity score. In the **n-Shot** setting, the model is finetuned on n examples per class in the training set. Following SetFit (Tunstall et al., 2022), the few-shot learning is performed in two stages.

Stage 1. In this stage, the model learns better representations of the instances of each label. The model is finetuned on pairs of positive and negative examples drawn from the few-shot training dataset. This is achieved by pairing each example of a class (positive) with all other examples of different classes in the dataset (negatives). We then perform contrastive training of the *late-fusion* model on the positive-negative pairs, varying the n-Shot setting in $\{1, 2, 4, 8, 16\}$.

Stage 2. After Stage 1 training, where the model has been trained to represent and discriminate speech instances across labels, the second stage finetunes a classifier head on top of this frozen model using standard multiclass logistic regression.

4.4 Results

Performance of Speech Intent Detection at 1024 dimension. Table 3 shows the results for a Matryoshka Dimension of 1024. The 0-Shot F1-Score is already better than random, which is approximatively 10%. Adding a few more examples greatly improves overall performance, achieving an F1 Score of 96.11 with 16 Shot training examples.

Performance of Speech Intent Detection across dimensions. Figure 6 shows the evolution of Speech Intent Detection performance for different matryoshka dimensions as the number of shots increases. While higher dimensions perform better with fewer examples, lower dimensions catch up when increasing the number of examples. Since, as we will show it, lower dimensions are cheaper at inference, these results suggest it is worthwhile to pay higher training costs in exchange for reduced inference costs.

Finding 3: Big dimensions adapt quickly to new tasks, smaller ones need more data.

When only a few examples are available, big dimensions perform better. However, with more training examples, small dimensions catch up.

5 Analysis

We provide analysis, interpretations, and justifications of the results. First, we analyze the speech quality of Fleurs and Kallaama, showing that Fleurs' lower quality explains the weaker performance on the Fleurs-Retrieval-Eval. Second, we analyze how matryoshka dimensions represent information by examining their rank. Third, we compare the deployment costs across different matryoshka dimensions. And finally, we analyze the instruction-following capabilities of the model in Appendix A.2. All the analyses are done on the late-fusion model.

Approach		dim=1024	dim=512	dim=256	dim=128
Late-Fusion	**F1 Score**	88.79	84.85	79.80	76.86
	Recall	89.79	86.49	81.98	79.88
Dual - *Retrieval*	**F1 Score**	68.64	58.33	50.86	44.59
	Recall	71.17	62.76	56.76	52.25
Dual - *Query Alignment*	**F1 Score**	76.07	67.67	67.67	65.22
	Recall	77.78	70.87	70.87	68.47

Figure 3: **Keyword Spotting** (Urban Bus) Performance Comparison across different embedding dimensions.

n-Shot	F1 Score	Recall
0	44.79	50.64
1	56.29	58.50
2	60.40	61.93
4	87.18	88.46
8	92.54	92.51
16	96.11	96.10

Table 3: *n*-Shot Performance for **Speech Intent Detection** at dimension 1024.

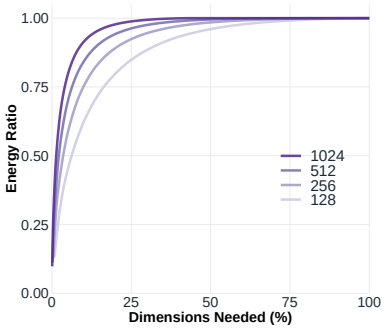

Figure 4: Percentage of dimensions needed to represent a given energy ratio.

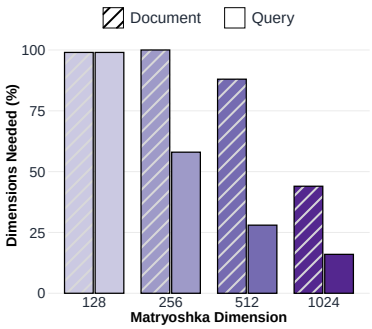

Figure 5: Percentage of dimensions needed to represent the full energy

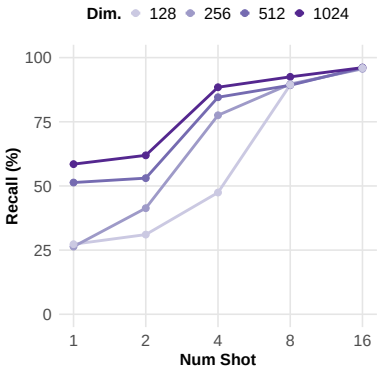

Figure 6: How the recall improves across matryoshka dimension as a function of the number of **few-shot examples**.

5.1 On the speech quality of Fleurs

We analyze the speech quality of both datasets and show that Fleurs exhibits lower quality due to its data collection design. Wolof speakers were asked to read translated texts on diverse topics such as history, geography, and astronomy. However, Wolof is primarily an oral language, and its standardized script is not taught in education systems, so most speakers are not used reading it. This is reflected in the Fleurs dataset, where speech contains frequent hesitations and speakers appear less confident. We support this by comparing Fleurs to Kallaama, which consists of transcribed spontaneous speech rather than read text.

Fleurs contains more hesitations. We measure the Characters per second as a proxy for hesitation. We found that Fleurs' speech contains 7.51 characters per second, over than 2 times fewer than Kallaama where speakers produce 16.88 characters per second.

Fleurs is of lower volume. We also observed that speakers in Fleurs are less confident. This is reflected in the speech being less loud compared to Kallaama. The average volume in dB is -49.01 for Fleurs, while it's -23.81 Kallaama

These observations explain the speech-to-document retrieval performance on Fleurs (see Table 2). The results also suggest that, for lower speech quality, higher matryoshka dimensions are preferable.

Finding 4: Wolof speech in Fleurs is unnatural.
Wolof is mainly an oral language. The speakers are not used to reading it, leading to more *hesitations* and *lower volume* in Fleurs read speech.

5.2 The rank of matryoshka dimensions

We have seen in Section 4.3 that smaller dimensions need more training time to catch up on the higher dimensions. However, for retrieval, all the dimensions are trained on the same amount of data, and the results in Table 1 and 2 show that small dimensions perform worst. This suggest that small dimensions fail to represent the full information. We study next the rank of the matryoshka dimensions to better provide more substance to interpretate these results.

Measuring the rank There are many ways

to study the rank. We analyze the cumulative energy ratio $R(k)$, which is the proportion of total variance explained by the top-k eigenvalues of the covariance matrix: $R(k) = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{j=1}^{d} \lambda_j}$ for $k \in \{1, 2, \ldots, d\}$. Where λ_i are the eigenvalues of the covariance matrix, sorted in descending order: $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d$. For each matryoshka dimension, we compute the covariance matrix on the full test dataset of Kallaama. This ratio tells us how the information is spread within the matryoshka vector. If the vector is low rank, then R reaches 1.0 (100%) more quickly, which means it needs only few dimensions to represent the full information.

Result 1: Ranks of matryoshka dimensions. Figure 4 shows the energy distribution for different matryoshka dimensions. Higher dimensions reach 1.0 more quickly, indicating they are lower rank and require only a small fraction of the full dimension to represent information. However, as shown in Tables 1 and 2, lower dimensions fail to match their performance. If higher dimensions are low-rank, why can't lower dimensions achieve similar results? This suggests that matryoshka representation learning fails to retain critical information during compression. It has been suggested (Zhang et al., 2025; Wen et al., 2025) that MRL's joint training and rigid compression rule, which is just a slicing, increases gradient variance during training and discards critical information.

Result 2: Rank of queries and documents. We also compare the rank of speech queries and documents. Figure 5 shows the fraction of dimensions needed to represent the full energy. Documents have higher ranks than queries since they contain more information. At dimensions 128 and 256, the vectors are full-rank, which may explain their lower performance, as all document information might not be fully represented.

Finding 5: Information is inequally distributed in matryoshka dimensions.

The rank analysis of the matryoshka dimensions shows that most information is concentrated in a few dimensions, suggesting that MRL does not achieve optimal compression and leaves room for further compression.

5.3 Costs

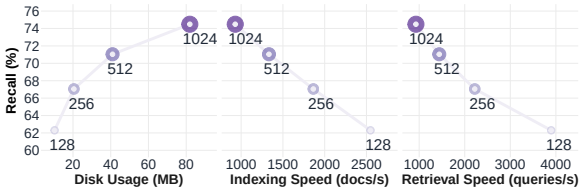

Figure 7: **Costs** across matryoshka dimensions.

Method. We analyze the matryoshka dimensions costs using `chromadb`, which is an open-source application for search and retrieval. We embed in float16 precision the documents in the different Matryoshka dimension. We measure the time in second for indexing documents (docs/s) and the final disk usage in MB. We then measure the time to query the documents store for one result.

Results. Figure 7 presents the relationship between performance and costs. For disk usage, we observe diminishing returns: increasing the matryoshka dimension yields marginal accuracy gains while incurring higher memory costs. Smaller embedding dimensions enable faster retrieval, particularly at lower dimensions due to reduced FLOPs. However, indexing speed improves more slowly. The trade-offs depend on the user's compute and the downstream task. As shown in Section 4.3, smaller dimensions can match the performance of bigger dimensions with more training data.

6 Conclusion and Future Work

We presented the first cross-lingual speech-text Matryoshka embedding models for Wolof and French, enabling direct retrieval of French documents from Wolof speech without ASR-translation pipelines. We collect large-scale speech-text training data and compare modeling comparison, showing that late modality fusion within a frozen text Matryoshka model achieves the best trade-off between expressivity, generalization, and efficiency. Late-fusion consistently outperforms pipelined and dual-encoder approaches on semantically demanding tasks and generalizes to unseen tasks like speech intent detection. Our rank analysis reveals that information concentrates in few dimensions, suggesting suboptimal compression in current Matryoshka training methods.

7 Limitations

Generalization to other languages. Our study focuses on a single language pair, Wolof-French. While this setting is socially and practically motivated, it remains unclear how well the proposed approach generalizes to other under-represented languages with different phonological properties, writing systems, or sociolinguistic contexts. Extending the approach to additional language pairs is an important direction for future research.

Synthetic Dataset. The majority of the training data relies on synthetic documents generated from transcribed and translated speech. While this enables scaling in low-resource settings, synthetic data may not fully capture the diversity and noise of real-world documents, potentially limiting robustness at deployment time. Morever, this approach supposes to have a functional ASR and translation systems, which is not always available for many low-resource and under-represented languages.

Beyond Matryoshkas. Our analysis of Matryoshka Representation Learning reveals that only few dimensions represent the information, suggesting suboptimal compression. There are many new works exploring alternatives to Matryoshka Representation Learning (Wen et al., 2025). We plan to explore dynamic structured sparsity, a more granular and flexible compression approach.

References

Luiz Henrique Bonifacio, Israel Campiotti, Roberto Lotufo, and Rodrigo Nogueira. 2021. mMARCO: A multilingual version of MS MARCO passage ranking dataset. *arXiv:2108.13897*.

Hervé Bredin. 2023. pyannote.audio 2.1 speaker diarization pipeline: principle, benchmark, and recipe. In *Proc. INTERSPEECH 2023*.

Mu Cai, Jianwei Yang, Jianfeng Gao, and Yong Jae Lee. 2024. Matryoshka multimodal models. *Preprint*, arXiv:2405.17430.

Umberto Cappellazzo, Minsu Kim, and Stavros Petridis. 2025. Adaptive audio-visual speech recognition via matryoshka-based multimodal llms. *Preprint*, arXiv:2503.06362.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Alexis Conneau, Min Ma, Simran Khanuja, Yu Zhang, Vera Axelrod, Siddharth Dalmia, Jason Riesa, Clara Rivera, and Ankur Bapna. 2022. Fleurs: Few-shot learning evaluation of universal representations of speech. *arXiv preprint arXiv:2205.12446*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Thierno Ibrahima DIOP. 2021. Wolof asr data on urban transport.

Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. 2022. Clap: Learning audio concepts from natural language supervision. *Preprint*, arXiv:2206.04769.

Elodie Gauthier, Laurent Besacier, Sylvie Voisin, Michael Melese, and Uriel Pascal Elingui. 2016. Collecting resources in sub-saharan african languages for automatic speech recognition: a case study of wolof. *LREC*.

Elodie Gauthier, Aminata Ndiaye, and Abdoulaye Guissé. 2024. Kallaama: A transcribed speech dataset about agriculture in the three most widely spoken languages in Senegal. In *Proceedings of the Fifth Workshop on Resources for African Indigenous Languages @ LREC-COLING 2024*, pages 10–19, Torino, Italia. ELRA and ICCL.

Wenbo Hu, Zi-Yi Dou, Liunian Harold Li, Amita Kamath, Nanyun Peng, and Kai-Wei Chang. 2024. Matryoshka query transformer for large vision-language models. *Preprint*, arXiv:2405.19315.

Abdou Karim Kandji, Frédéric Precioso, Cheikh Ba, Samba Ndiaye, and Augustin Ndione. 2025. Wolbanking77: Wolof banking speech intent classification dataset. *Preprint*, arXiv:2509.19271.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and Ali Farhadi. 2022. Matryoshka representation learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 30233–30249. Curran Associates, Inc.

Tsz Kin Lam, Marco Gaido, Sara Papi, Luisa Bentivogli, and Barry Haddow. 2025. Prepending or cross-attention for speech-to-text? an empirical comparison. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2994–3006, Albuquerque, New Mexico. Association for Computational Linguistics.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, pages 34892–34916. Curran Associates, Inc.

Oriol Pareras, Gerard I. Gállego, Federico Costa, Cristina España-Bonet, and Javier Hernando. 2025. Revisiting direct speech-to-text translation with speech llms: Better scaling than cot prompting? *Preprint*, arXiv:2510.03093.

Letian Peng, Yuwei Zhang, Zilong Wang, Jayanth Srinivasa, Gaowen Liu, Zihan Wang, and Jingbo Shang. 2024. Answer is all you need: Instruction-following text embedding via answering the question. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 459–477, Bangkok, Thailand. Association for Computational Linguistics.

Alexis Plaquet and Hervé Bredin. 2023. Powerset multi-class cross entropy loss for neural speaker diarization. In *Proc. INTERSPEECH 2023*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. *Preprint*, arXiv:2103.00020.

Chandan K A Reddy, Vishak Gopal, and Ross Cutler. 2021. Dnsmos: A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors. *Preprint*, arXiv:2010.15258.

Fabian David Schmidt, Philipp Borchert, Ivan Vulić, and Goran Glavaš. 2024. Self-distillation for model stacking unlocks cross-lingual NLU in 200+ languages. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6724–6743, Miami, Florida, USA. Association for Computational Linguistics.

Fabian David Schmidt, Ivan Vulić, Goran Glavaš, and David Ifeoluwa Adelani. 2025. Fleurs-slu: A massively multilingual benchmark for spoken language understanding. *Preprint*, arXiv:2501.06117.

Yaya Sy, Dioula Doucouré, Christophe Cerisara, and Irina Illina. 2025. Speech language models for under-represented languages: Insights from wolof. *Preprint*, arXiv:2509.15362.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning.

Silero Team. 2024. Silero vad: pre-trained enterprise-grade voice activity detector (vad), number detector and language classifier. `https://github.com/snakers4/silero-vad`.

Lewis Tunstall, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. 2022. Efficient few-shot learning without prompts. *arXiv preprint*.

Tiansheng Wen, Yifei Wang, Zequn Zeng, Zhong Peng, Yudi Su, Xinyang Liu, Bo Chen, Hongwei Liu, Stefanie Jegelka, and Chenyu You. 2025. Beyond matryoshka: Revisiting sparse coding for adaptive representation. *Preprint*, arXiv:2503.01776.

Jian Wu, Yashesh Gaur, Zhuo Chen, Long Zhou, Yimeng Zhu, Tianrui Wang, Jinyu Li, Shujie Liu, Bo Ren, Linquan Liu, and Yu Wu. 2023. On decoder-only architecture for speech-to-text and large language model integration. *Preprint*, arXiv:2307.03917.

Biao Zhang, Lixin Chen, Tong Liu, and Bo Zheng. 2025. SMEC:rethinking matryoshka representation learning for retrieval embedding compression. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 26220–26233, Suzhou, China. Association for Computational Linguistics.

A Multitask Capabilities

A.1 Beyond speech-to-document retrieval

We extend the synthetic speech-to-document dataset with natural ASR data from multiple sources: **FLEURS**, **ALFA** (Gauthier et al., 2016), **Common Voice** (**CV**), **Kallaama** (Gauthier et al., 2024), and **Urban-Bus** (**UB**) (DIOP, 2021). We retain the official train/test splits for FLEURS and create random splits for the other datasets. During training, speech serves as the query and transcriptions or their translations as retrieval targets. Table 4 summarizes the dataset distribution.

A.2 Instruction-Following Capabilities

The late-fusion model is multilingual, multimodal, and multitask, as the retrieval target can be specified via the prompt at inference time. We analyze retrieval and keyword-spotting performance under different prompts. Table 5 shows that performance is highest when the task description and task objective are aligned.

Split	Fleurs	Alfa	CV	Kallaama	UB	Total
Train	8.72	16.13	34.97	33.60	4.52	**97.94**
Test	1.75	2.84	6.21	5.91	1.12	**17.83**

Table 4: The non-synthetic ASR dataset.

	Tasks			
	Document Retrieval		**Keyword Spotting**	
Prompt	**nDCG@5**	**nDCG@10**	**F1**	**Recall**
Document Retrieval	68.85	74.49	85.35	87.39
Transcription Retrieval	66.08	71.38	88.79	89.79

Table 5: Instruction-following performance of *late-fusion* model (d=1024) on document retrieval and keyword spotting with different prompts.