Mapping Cross-Lingual Sentence Representations for Low-Resource Language Pairs Using Pre-trained Language Models

Andreea Ioana Tudor and Tsegaye Misikir Tashu
Department of Artificial Intelligence
Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence
University of Groningen, Groningen, 9747AG
andreea.14.ioana@gmail.com, t.m.tashu@rug.nl

Abstract

In this work, we explore different linear mapping techniques to learn cross-lingual document representations from pre-trained multilingual large language models for low-resource languages. Three different mapping techniques namely Linear Concept Approximation (LCA), Linear Concept Compression (LCC), and Neural Concept Approximation (NCA) and four multilingual language models such as mBERT, mT5, XLM-R, and ErnieM were used to extract embeddings. The inter-lingual representations were created mappings the monolingual representation extracted from multilingual language models. The experimental results showed that LCA and LCC significantly outperform NCA, with models like ErnieM achieving the highest alignment quality. Language pairs exhibit variable performance, influenced by linguistic similarity and data availability, with the Amharic-English pair yielding particularly high scores. The results showed the utility of LCA and LCC in enabling cross-lingual tasks for low-resource languages.

1 Introduction

“Attention is all you need.” This phrase marked a milestone in Machine Learning (ML) and Natural Language Processing (NLP) (Vaswani et al., 2023). Yet, how much attention is given to languages less common than English? Research on NLP for low-resource languages remains sparse, with studies nearly ten times fewer than those focused on English and citation rates almost twenty times lower (Poupard, 2024). This imbalance creates a gap in NLP accessibility and development for low-resource languages. While advancements in NLP have been impressive, the overwhelming focus on English limits technological inclusivity. Large Language Models (LLMs), for instance, excel in machine translation, information retrieval, question answering, and text summarization (Conneau et al., 2020; Fan et al., 2020; Tashu et al., 2023), yet most models still lack robust support for low-resource languages (Robinson et al., 2023). Some progress has been made, such as multilingual models for Indic (Dabre et al., 2021) and African languages (Ogueji et al., 2021), but challenges remain.

Training such models requires extensive data and computational resources, a significant hurdle for low-resource languages where data availability is limited. To address this, we focus on leveraging existing resources and cross-lingual learning techniques to align sentences across languages, including low-resource ones. Cross-lingual learning aligns text representations from one language to another, enabling effective knowledge transfer and facilitating robust multilingual systems without heavy reliance on machine translation (Tashu et al., 2023). Alignment can occur at different levels: word, sentence, or document. Word-level alignment brings semantically similar words close in a shared embedding space, aiding tasks like bilingual lexicon induction (Agirre, 2020). Sentence-level alignment captures full context and meaning, using techniques like LASER (Artetxe and Schwenk, 2019) to generate language-independent sentence embeddings. Document-level alignment broadens this focus, enhancing multilingual information retrieval (Tashu et al., 2023).

Our study addresses a specific gap: exploring effective methods for generating cross-lingual sentence representations from pre-trained large language models. Specifically, we ask: How effective are different mapping methods for learning cross-lingual sentence representations in low-resource language pairs? Answering this will help improve NLP inclusivity and capabilities for low-resource languages. This work builds on work by Salamon et al. (2021), Tashu et al. (2023), and Tashu et al. (2024), focusing on sentence-level representations. By emphasizing sentence-level, rather than document-level, alignment, we aim to provide a

Proceedings of the First Workshop on Language Models for Low-Resource Languages (LoResLM 2025), pages 249–257
January 20, 2025. ©2025 Association for Computational Linguistics

fine-grained understanding of multilingual semantics and bridge gaps in NLP research for underrepresented languages.

2 Methodology

Tashu et al. (2023) proposed an approach using different mapping techniques for obtaining interlingual representations, which serves as an inspiration for the current work. It involves the generation of document embeddings (representations) for the source and target languages, and then finding a mapping into an inter-lingual representation space. It further allows cross-lingual transfer learning, hence avoiding the high costs of machine translation (MT) systems and challenges with low-resource languages (Tashu et al., 2023). This study utilizes pre-trained language models to embed parallel data sets. It then employs mapping techniques to align monolingual representation spaces, creating inter-lingual document representations. This approach facilitates the effective transfer of linguistic information across different languages

2.1 Embeddings

The growing need to support a wider range of languages has led to the development of multilingual LLMs. They are pre-trained on large corpora of multilingual data, with the expectation that lower resource languages can benefit from the linguistic similarities and shared representations among language pairs (Xu et al., 2024). In this study, four multilingual language models were used to extract the unilingual representations individually for different pairs of languages: mBERT (Devlin et al., 2018), mT5 (Xue et al., 2021), XLM-RoBERTa (Conneau et al., 2020) and ErnieM (Ouyang et al., 2021).

2.2 Mapping Techniques

Given two monolingual document collections, $D_x = \{d_{x,1}, \ldots, d_{x,n}\}$ and $D_y = \{d_{y,1}, \ldots, d_{y,n}\}$, first a representation is extracted used a pretrained MLLM. However, any representation learning model which maps the document sets D_x and D_y to vectors within the $\mathbb{R}^k$ is suitable. We obtain sets of vectors, $C_x = \{\hat{d}_{x,1}, \ldots, \hat{d}_{x,n}\} \subset \mathbb{R}^k$, $C_y = \{\hat{d}_{y,1}, \ldots, \hat{d}_{y,n}\} \subset \mathbb{R}^k$. One can think of C_x, C_y as "Concept Spaces", which encode more general concepts of the language and their meaning. While the vectors in C_x, C_y might capture concepts and information, which are similar across languages, they likely encode it in different ways. Therefore, a direct comparison of $\hat{d}_{x,k}, \hat{d}_{y,k}$ is yet unlikely to reveal similarities on a content level.

2.2.1 Linear concept approximation (LCA)

LCA performs a linear transformation to map document vectors from one language's concept space to another's. This is achieved by:

1. Constructing the coefficient matrices for the projections:

$$\mathbf{A} = \mathbf{P}_{\mathbf{X}^T}\mathbf{Y}^T \in \mathbb{R}^{k_x \times k_y} \tag{1}$$

$$\mathbf{B} = \mathbf{P}_{\mathbf{Y}^T}\mathbf{X}^T \in \mathbb{R}^{k_y \times k_x}, \tag{2}$$

   where $\mathbf{P}_{\mathbf{X}^T}$ and $\mathbf{P}_{\mathbf{Y}^T}$ denote pseudo-inverses of $\mathbf{X}^T$ and $\mathbf{Y}^T$ respectively. The pseudo-inverse is used to find the best-fit linear transformation between the two spaces.

2. Calculating mappings for the document vectors $\underline{\mathbf{x}} \in \mathbb{R}^{k_x}$ in language L_x and $\underline{\mathbf{y}} \in \mathbb{R}^{k_y}$ in language L_y:

$$\underline{\hat{\mathbf{x}}} = \mathbf{A}^T \underline{x} \in \mathbb{R}^{k_y} \tag{3}$$

$$\underline{\hat{\mathbf{y}}} = \mathbf{B}^T \underline{y} \in \mathbb{R}^{k_x} \tag{4}$$

2.2.2 LCC

The LCC approach is used to align and compare document representations from different languages in a common space while preserving their information. To reiterate, the objective of LCC is to minimize the equation:

$$\min_{\mathrm{rg}(\mathbf{A})=d} \left\| \begin{bmatrix} \mathbf{C}_x & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_y \end{bmatrix} \mathbf{A} - \begin{bmatrix} \mathbf{C}_x & \mathbf{C}_x \\ \mathbf{C}_y & \mathbf{C}_y \end{bmatrix} \right\|_2^2 . \tag{5}$$

The implementation choice for LCC is described by the following steps:

1. Constructing the training matrices $\mathbf{X}$ and $\mathbf{Y}$:

$$\mathbf{X} = \begin{bmatrix} \mathbf{C}_x & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_y \end{bmatrix} \tag{6}$$

$$\mathbf{Y} = \begin{bmatrix} \mathbf{C}_x & \mathbf{C}_x \\ \mathbf{C}_y & \mathbf{C}_y \end{bmatrix} \tag{7}$$

2. Using Ridge Regression to find the transformation matrix:

The Ridge Regression helps find the best linear transformation that maps the source documents to the target documents while preventing overfitting by regularizing the size of the transformation matrix. It aims to find a matrix **W** that transforms **X** into **Y** while minimizing the regularized least squares error:

$$\hat{\mathbf{W}} = \arg\min_{\mathbf{W}} \left\{ \|\mathbf{Y} - \mathbf{X}\mathbf{W}\|_2^2 + \alpha \|\mathbf{W}\|_2^2 \right\}, \quad (8)$$

where α is the regularization parameter. The matrix $\hat{\mathbf{W}}$ serves as part of the linear mappings $\mathbf{E}_x$ and $\mathbf{E}_y$. Let $\mathbf{T}_{\mathbf{X}} = \mathbf{X}\hat{\mathbf{W}}$ be the transformed data after applying the Ridge Regression model.

3. Transforming the test data:

   For the test data, let $\mathbf{X}_{test}$ and $\mathbf{Y}_{test}$ be the concatenated test matrices corresponding to the source and target languages L_x and L_y, respectively. After applying the Ridge Regression model, we get:

$$\mathbf{T}_{\mathbf{X}_{test}} = \mathbf{X}_{test}\hat{\mathbf{W}} \quad (9)$$

$$\mathbf{T}_{\mathbf{X}_{test}} = \mathbf{Y}_{test}\hat{\mathbf{W}}. \quad (10)$$

4. Dimensionality reduction with PCA:

   After applying Ridge Regression, we employ PCA to reduce the dimensionality of the transformed test data and to map it back to the original feature space for further evaluation.

2.2.3 NCA

Two neural network models were trained to map representations between the source and target languages. The same neural network architecture was employed for both mappings: from the source to the target language and from the target to the source language. Each model consists of an input layer with dimensionality d, a hidden layer with 500 neurons using the Exponential Linear Unit (ELU) activation function, and an output layer with dimensionality d.

The ELU (Clevert et al., 2015) activation function is defined as:

$$\mathrm{ELU}(x) = \begin{cases} x & \text{if } x > 0, \\ \alpha(\exp(x) - 1) & \text{if } x \leq 0, \end{cases} \quad (11)$$

where α is a hyperparameter. This function helps mitigate the vanishing gradient problem and speeds up learning by allowing negative values, potentially improving model performance over standard activation functions like Rectified Linear Unit (ReLU).

The Huber loss function (Huber, 1964) combines the advantages of mean squared error and mean absolute error to handle outliers more robustly. It is defined as:

$$\mathrm{Huber}(a, \delta) = \begin{cases} \frac{1}{2}a^2 & \text{for } |a| \leq \delta, \\ \delta(|a| - \frac{1}{2}\delta) & \text{otherwise}, \end{cases} \quad (12)$$

where α is the residual (the difference between predicted and actual values) and δ is a threshold parameter. This loss function provides smoothness while being less sensitive to outliers than squared error.

The Adam optimizer (Kingma and Ba, 2017) integrates features from both Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp), adjusting learning rates for each parameter based on estimates of the first and second moments of the gradients. The update rule is:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \varepsilon}\hat{m}_t, \quad (13)$$

where θ denotes the model parameters, η is the learning rate, $\hat{m}_t$ and $\hat{v}_t$ are the bias-corrected estimates of the first and second moments of the gradients, and ε is a small constant to prevent division by zero.

3 Experimental Setup

3.1 Data

The NLLB dataset[1] (Fan et al., 2020; Schwenk et al., 2021) contains bitext for 1613 language pairs (148 English-centric, 1465 non-English-centric). It was created using metadata from mined bitexts made available by Meta AI, leveraging the stopes mining library[2] and LASER3 encoders (Heffernan et al., 2022). The innovation behind the NLLB project (NLLB Team et al., 2022) stands in the

[1] Available at `https://opus.nlpl.eu/NLLB/corpus/version/NLLB`

[2] stopes is a library for preparing data for MT research, part of the No Language Left Behind (NLLB) project `https://facebookresearch.github.io/stopes/`

provided solution for the automatic construction of translation pairs, done by aligning sentences from various collections of monolingual documents. This further enables the coverage of 200 languages by extending LASER's language, and the production of a substantial amount of data, including for low-resource languages.

The dataset amounts to approximately 450GB of data with over 1,500 language pairs, however for the purpose of the current project, only a few pairs were used: English-Amharic, Arabic-Somali, Bemba-Afrikaans, and Igbo-Hausa. The selection includes two Indo-European languages (English and Afrikaans), four Afro-Asiatic (Arabic, Somali, Amharic and Hausa), and two from the Niger-Congo family (Bemba and Igbo). These pairs comprise a mixture of high and low-resource languages from different language families.

3.2 Embedding the Data

To ensure a consistent and manageable dataset for embedding, we sampled $100,000$ sentences from the NLLB dataset for each language pair. Due to the smaller size of the dataset, only 58,000 sentences were sampled for the Arabic-Somali language pair. Only sentences containing a minimum of 10 words were included in the sample to ensure sufficient contextual information for accurate embeddings. This filtering step was crucial for maintaining the quality and relevance of the data used for embedding.

The embedding process was adapted for sentences rather than documents, following the methodology outlined by Tashu et al. (2024) in a similar approach used for document embeddings. Sentences were tokenized, truncated or padded to the same maximum token length of a maximum of 128 tokens, and processed through the corresponding models to compute embeddings. The attention mask ensures that only relevant tokens are considered, optimizing the representation of sentence semantics. The final hidden states from the model's encoder part are extracted to obtain embeddings for each token within the sentence. These embeddings are then aggregated using a global pooling operation to generate fixed-size vectors, ready for further analysis and mapping methods.

3.3 Evaluation metrics

After generating embeddings using the previously discussed models, in the evaluation phase, we apply the mapping methods individually to align the embedding spaces between each source language and its target counterpart, and vice versa, for each language pair. We maintain consistency by evaluating using metrics such as Mate Retrieval Rate and Mean Reciprocal Rank. This ensures direct comparison with previous studies (Tashu et al., 2024) that mainly focused on higher-resource languages, aiming to test the effectiveness of the mapping techniques in cross-lingual representation tasks, particularly in low-resource language scenarios.

Mate Retrieval Rate assesses the similarity between two documents, the query and the retrieved document. If the retrieved document matches the query document, it is termed as a mate retrieval. The mate retrieval rate is defined as:

$$MR(d) = \arg\max S_d \cdot T_d^T, \tag{14}$$

where $S(d,d')$ is given by:

$$S(d,d') = \begin{cases} 1 & \text{if } d = d' \\ 0 & \text{if } d \neq d'. \end{cases} \tag{15}$$

In this context, S represents the similarity between two documents d and d', and MR indicates the mate retrieval for a document d in the source S and target language T. Mate retrieval is deemed successful if d and d' are identical. Combining these equations, the mate retrieval rate for all documents D can be computed as:

$$\text{RetrievalRate} = \frac{1}{|D|} \sum_{d=1}^{|D|} S(d, MR(d)) \tag{16}$$

Mean Reciprocal Rank quantifies how high-ranked documents are, based on a similarity measure. Using cosine similarity, it is defined as:

$$C(d_1,d_2) = \frac{d_1 \cdot d_2}{\|d_1\| \cdot \|d_2\|}, \tag{17}$$

where the numerator is the inner product of the document vectors d_1 and d_2, and the denominator is the product of their magnitudes. The cosine similarity approaches 1 if the documents are similar and -1 if they are dissimilar. This similarity measure can be extended to a cosine similarity matrix for all documents. The rank r of a document is defined by its cosine similarity compared to other documents. If a document is most similar to itself in the target language, its rank is 1. These components are combined to calculate the mean reciprocal rank:

$$\text{ReciprocalRank} = \frac{1}{|D|} \sum_{d=1}^{|D|} \frac{1}{r_d} \quad (18)$$

3.4 Experiments

In our experiments, each language in the pairs was used once as the source and once as the target, resulting in a total of eight pairs. These pairs were embedded using the four MLLMs. Our goal was to map from the source embedding space to the target embedding space using the three different mapping methods(LCA, LCC, NCC) for each pair and each embedding model. The performance of these mappings was evaluated using both reciprocal rank and mate retrieval. We evaluated the performance across a range of dimensions from 100 up to 768, incrementing by 50.

4 Results

In this section, we present the results obtained in two parts: one focused on the pre-trained models, and another one focused on the pairs of languages used. Further, we provide the results across dimensions for a selection of the experiments run.

4.1 Results by Models

We first analyze the performance of each mapping technique based on the pre-trained models used to generate the embeddings. The highest scores for each language pair were obtained and plotted as histograms to illustrate the performance variations across different models and mapping methods. This allows us to evaluate which pre-trained models contribute most effectively to the mapping quality in the context of low-resource languages. The results showing the highest reciprocal rank scores are illustrated in Figure 1 which presents the highest values for both mate retrieval and reciprocal rank.

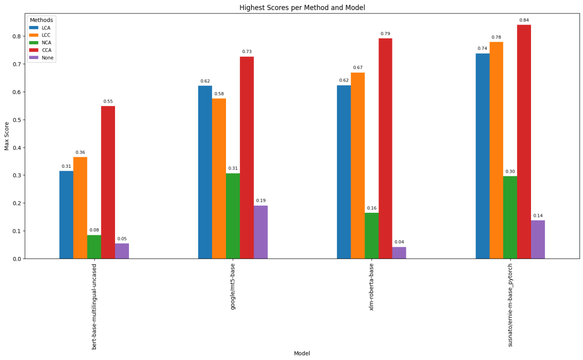


Figure 1: Highest reciprocal rank by models and mapping techniques

Among the mapping methods, LCC and LCA achieved similar scores, with the highest reciprocal ranks of 0.778 and 0.737, respectively, both reached with the ErnieM model. For NCA, scores were only marginally higher than the baseline in which no mapping approach was applied. Regarding model performance, ErnieM outperformed other models across most mapping techniques. With both LCC and LCA, ErnieM achieved strong scores, indicating its robustness across mappings. XLM-R was the second-best performing model overall, reaching a high reciprocal rank of 0.791 with LCC and maintaining high scores with LCA. This consistency underscores XLM-R's strong performance across various mapping techniques.

The mT5 model showed notable results, particularly with LCC, achieving a reciprocal rank of 0.726. It maintained respectable scores with LCA, although these were slightly lower than XLM-R and ErnieM. However, performance declined more significantly with NCA. The mBERT model consistently showed lower performance relative to the others, with its best results obtained using LCC, which yielded a reciprocal rank of 0.548—significantly lower than the top-performing models. Although LCA improved mBERT's performance slightly, the gains remained limited.

4.2 Results by Language Pairs

Next, we focus on the performance of each mapping technique based on the pairs of languages used. The highest scores from all pre-trained models were aggregated and plotted as histograms to show the effectiveness of different mappings for each language pair. This analysis helps in understanding the challenges and successes of mapping between specific low-resource language pairs and highlights the relative performance of different mapping methods. The plot can be seen in Figure 2.

For the Hausa-Igbo (ha-ig) language pair, LCC achieved the highest reciprocal rank of 0.568. For the Igbo-Hausa (ig-ha) direction, LCA achieved strong performance, with a reciprocal rank of 0.726. In both directions, scores for LCA and LCC were relatively close, ranging from 0.544 to 0.620, indicating the robustness of these mapping methods for these language pairs. For the Somali-Arabic (so-ar) pair and its reverse, LCA produced high scores of approximately 0.5, followed closely by LCC with almost identical values for the Somali-Arabic direction and slightly higher results for LCA in the Arabic-Somali (ar-so) direction. However,

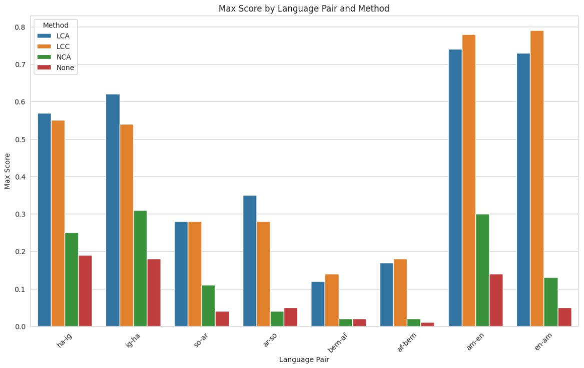


Figure 2: Highest reciprocal rank by language pairs and mapping techniques

both methods showed lower scores than for other language pairs. NCA consistently underperformed in this language pair, especially in the ar-so direction, where scores were even lower than the baseline. For the Bemba-Afrikaans (bem-af) and Afrikaans-Bemba (af-bem) language pairs, scores were comparatively lower, with reciprocal ranks not exceeding 0.184. NCA performed close to the baseline with a score near zero.

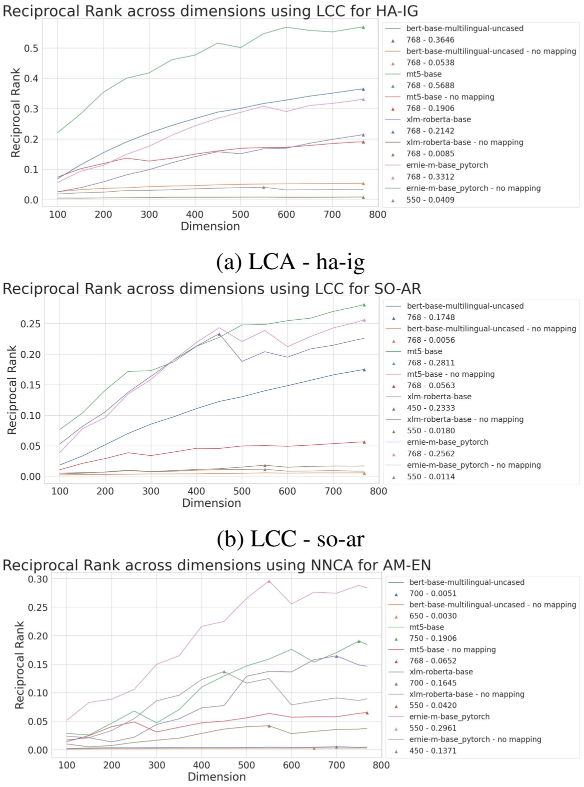
In contrast, the Amharic-English (am-en) language pair achieved excellent results with LCA, reaching the highest overall reciprocal rank of 0.840. LCC and LCA also performed well, with the highest score across all language pairs for LCC at 0.778. For the English-Amharic (en-am) direction, mapping methods yielded scores similar to the am-en direction, with the highest LCA score across all language pairs reached here at 0.737. NCA's performance was comparable to that in the ha-ig and ig-ha pairs but was considerably lower than other methods.

4.3 Results across Dimensions

To showcase the performance of the mapping techniques across dimensions, we have selected a language pair per mapping method. Given the similarity in results between source-to-target and target-to-source directions for the same language pairs, we focus on a single direction to avoid redundancy. Figure 3 presents the reciprocal rank obtained across dimensions ranging from 100 to 768, where LCA (Figure 3a), LCC (Figure 3b) and NCA (Figure 3c) were used for ha-ig, so-ar and am-en, respectively. The plots contain all embedding models, as well as the baselines, where no mapping was employed.

Across dimensions, scores generally increased for all models, while baselines (where no mapping was used) showed little to no increase. mBERT showed the highest performance for LCA and LCC in the ha-ig and so-ar pairs, while mT5 performed best with NCA in the am-en pair.

While higher dimensionalities generally correlated with better performance, there were cases where peak performance occurred at a lower dimensionality. For instance, ErnieM and XLM-R both peaked at 550 dimensions for LCA, and XLM-R peaked at 450 dimensions with LCC. For NCA, early peak scores were observed with ErnieM and XLM-R as well.



(a) LCA - ha-ig

(b) LCC - so-ar

(c) NCA - am-en

Figure 3: Reciprocal rank across dimensions using LCA, LCC and NCA for different language pairs

5 Discussion

A key outcome of this study is the effective application of LCA and LCC mapping techniques for aligning cross-lingual embeddings, as both yielded consistently higher scores than NCA across experiments. This consistency suggests that LCA and LCC can effectively capture and align semantic relationships between languages, particularly in low-resource settings. Our findings align with those of Tashu et al. (2024), who also reported strong performance for LCA and LCC, supporting their reliability in cross-lingual tasks. The relatively lower performance of NCA may be attributed to architectural limitations that limit its ability to capture the nuanced language similarities essential for effective mapping, as noted by Tashu et al. (2024).

Model performance also varied considerably, with ErnieM consistently outperforming the other models and mBERT demonstrating the lowest scores. This discrepancy may be due to the limited range of languages in mBERT's pre-training set, which included only three languages from our study: English, Afrikaans, and Arabic. Consequently, mBERT struggled with other language pairs, underscoring the importance of diverse pre-training datasets for effective multilingual representation. In contrast, XLM-R and mT5, trained on a broader range of lower-resource languages, performed well across the board, highlighting their adaptability and robustness in cross-lingual contexts.

The effectiveness of mapping techniques also varied across language pairs, indicating the unique linguistic challenges posed by different combinations. For example, the ha-ig pair achieved the highest reciprocal rank with LCC, while in the ig-ha direction, LCA performed best, demonstrating strong alignment potential between these languages, likely due to their Afro-Asiatic language roots. This performance indicates that mapping techniques may benefit from inherent structural or linguistic similarities between languages.

In the case of Somali-Arabic pairs, LCA and LCC continued to perform well but with notable score reductions compared to ha-ig. This outcome may reflect the complexity of aligning Arabic with Somali despite their shared Afro-Asiatic roots. Variations in dialect and structure may contribute to this difficulty, highlighting the need for more sophisticated mapping approaches that account for intra-family linguistic diversity.

The Bemba-Afrikaans (bem-af) and Afrikaans-Bemba (af-bem) pairs consistently achieved the lowest scores. Despite some improvement with LCA and LCC, the low performance overall suggests that the linguistic distance between Bemba, a Niger-Congo language, and Afrikaans, an Indo-European language, poses a significant challenge for mapping. The scarcity of resources and potential lack of shared linguistic structures likely contribute to the difficulty in achieving effective alignment.

The Amharic-English (am-en) pair, however, showed exceptional performance with all mapping methods, achieving the highest overall reciprocal ranks. This strong alignment suggests high compatibility, perhaps due to robust resource availability for Amharic and English. Notably, the slight score improvement in the am-en direction over en-am suggests that directionality has less impact on results than factors like model pre-training and linguistic similarity. These observations suggest that both linguistic and resource factors play crucial roles in mapping success and invite further investigation into the specific factors affecting cross-lingual performance.

Finally, exploring mapping techniques across different dimensions provided insights into the impact of embedding dimensionality on alignment quality. The results demonstrated that, generally, increasing dimensionality improves scores for LCA and LCC, though certain models achieved peak performance at lower dimensions, suggesting that optimal dimensionality may vary by model and mapping technique. Running experiments across various dimensions is valuable as it can reveal these optimal configurations, guiding resource-efficient early stopping strategies and reducing computational costs.

6 Conclusion

This study evaluated the effectiveness of multiple mapping methods in aligning cross-lingual sentence representations for pairs of low-resource languages, utilizing pre-trained multilingual LLMs. We tested LCA, LCC, and NCA mapping techniques across multiple model and language pair combinations to assess their performance in low-resource settings.

Our findings highlight the mapping techniques' success in capturing semantic relationships across languages, with LCA and LCC consistently outperforming NCA. This outcome suggests that the architectural limitations of NCA make it less effective in capturing the nuanced linguistic similarities required for cross-lingual alignment tasks. Additionally, the variability in results across models showed ErnieM's superior performance overall, with XLM-R and mT5 close behind, underscoring the importance of diverse pre-training data for robust multilingual performance. mBERT, by contrast, performed less effectively, highlighting the limitations posed by limited language exposure in pre-training.

Furthermore, our results reveal significant performance variations across language pairs, suggesting that factors like linguistic similarity and resource availability play essential roles in cross-lingual mapping. Specifically, the high compatibility and robust resource availability for Amharic-English contributed to their superior scores, illustrating how these factors can positively impact mapping performance. Overall, these findings demonstrate the utility of LCA and LCC as effective mapping methods for low-resource cross-lingual tasks and highlight the importance of training data diversity in enhancing model adaptability.

References

Eneko Agirre. 2020. Cross-Lingual Word Embeddings. *Computational Linguistics*, 46(1):245–248.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2015. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). *Under Review of ICLR2016 (1997)*.

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. *Preprint*, arXiv:1911.02116.

Raj Dabre, Himani Shrotriya, Anoop Kunchukuttan, Ratish Puduppully, Mitesh M Khapra, and Pratyush Kumar. 2021. IndicBART: A pre-trained model for indic natural language generation. *arXiv preprint arXiv:2109.02903*.

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.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, Armand Joulin, and Facebook Ai. 2020. *Beyond English-Centric Multilingual Machine Translation*.

Kevin Heffernan, Onur Çelebi, and Holger Schwenk. 2022. Bitext Mining Using Distilled Sentence Representations for Low-Resource Languages. *Preprint*, arXiv:2205.12654.

Peter J. Huber. 1964. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1):73 – 101.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. *Preprint*, arXiv:1412.6980.

NLLB Team, Marta Costa-Jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, and John Hoffman. 2022. No Language Left Behind: Scaling Human-Centered Machine Translation.

Kelechi Ogueji, Yuxin Zhu, and Jimmy Lin. 2021. Small data? no problem! exploring the viability of pretrained multilingual language models for low-resourced languages. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 116–126, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xuan Ouyang, Shuohuan Wang, Chao Pang, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2021. Ernie-m: Enhanced multilingual representation by aligning cross-lingual semantics with monolingual corpora. *Preprint*, arXiv:2012.15674.

Duncan Poupard. 2024. Attention is all low-resource languages need. *Translation Studies*, pages 1–4.

Nathaniel R. Robinson, Perez Ogayo, David R. Mortensen, and Graham Neubig. 2023. ChatGPT MT: Competitive for High- (but not Low-) Resource Languages. *Preprint*, arXiv:2309.07423.

Vilmos Tibor Salamon, Tsegaye Misikir Tashu, and Tomáš Horváth. 2021. Linear Concept Approximation for Multilingual Document Recommendation. In *Intelligent Data Engineering and Automated Learning – IDEAL 2021*, pages 147–156, Cham. Springer International Publishing.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, Armand Joulin, and Angela Fan. 2021. CCMatrix: Mining Billions of High-Quality Parallel Sentences on the Web. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6490–6500, Online. Association for Computational Linguistics.

Tsegaye Tashu, Eduard-Raul Kontos, Matthia Sabatelli, and Matias Valdenegro-Toro. 2024. Mapping Transformer Leveraged Embeddings for Cross-Lingual Document Representation (A Preprint).

Tsegaye Misikir Tashu, Marc Lenz, and Tomáš Horváth. 2023. NCC: Neural concept compression for multilingual document recommendation. *Applied Soft Computing*, 142:110348.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention Is All You Need. *Preprint*, arXiv:1706.03762.

Yuemei Xu, Ling Hu, Jiayi Zhao, Zihan Qiu, Yuqi Ye, and Hanwen Gu. 2024. A Survey on Multilingual Large Language Models: Corpora, Alignment, and Bias.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.