# Bootstrapping Embeddings for Low Resource Languages

**Merve Basoz**[a] **and Andrew Horne**[b] **and Mattia Opper**[a]
[a]School of Informatics, University of Edinburgh, UK
[b] Edina, University of Edinburgh, UK
{s2451985, ahorne2, m.opper}@ed.ac.uk

## Abstract

Embedding models are crucial to modern NLP. However, the creation of the most effective models relies on carefully constructed supervised finetuning data. For high resource languages, such as English, such datasets are readily available. However, for hundreds of other languages, they are simply non-existent. We investigate whether the advent of large language models can help to bridge this gap. We test three different strategies for generating synthetic triplet data used to optimise embedding models. These include in-context learning as well as two novel approaches, leveraging adapter composition and cross lingual finetuning of the LLM generator (XL-LoRA) respectively. We find that while in-context learning still falls short of strong non-synthetic baselines, adapter composition and XL-LoRA yield strong performance gains across a wide array of tasks and languages, offering a clear, scalable pathway to producing performant embedding models for a wide variety of languages.

## 1 Introduction

In natural language processing (NLP), embeddings play an important role, powering crucial applications such as retrieval augmented generation, semantic matching and classification, among others (Cer et al., 2018; Lewis et al., 2020; Thakur et al., 2021; Muennighoff et al., 2022). Creating the most effective embedding models relies on human annotated triplet data, which allows the model to learn complex semantic relationships during finetuning (Reimers and Gurevych, 2019; Ni et al., 2022). However, such datasets are costly to produce and although they are abundant in high resource languages such as English, they are simply unavailable for many languages, despite these languages often having many millions of speakers and high demand for NLP technologies. While unsupervised approaches have shown promise as a potential solution, their performance still lags substantially behind what can be achieved using curated supervision (Gao et al., 2021; Wu et al., 2022).

The rise of large language models holds the promise of bridging this gap. Given their impressive broad spectrum capabilities (Bubeck et al., 2023), can we use them to generate high-quality synthetic data, cheaply and effectively bridging the gap for languages and domains where human annotated resources simply do not exist? Recent work, investigating synthetic data generation (Zhang et al., 2023b), has shown promising results on standard English benchmarks. Here we examine whether it can be applied to low resource languages where the impact is arguably far greater, as alternatives often simply do not exist.

We examine the efficacy of three approaches to synthetic data generation: one uses in-context learning following prior work (Zhang et al., 2023b) and two novel methods; one in which we optimise the LLM data generator using adapter composition and the other in which we use a specialised cross lingual adaptation regime we dub XL-LoRA[1]. We find that these latter two approaches show clear consistent gains against strong non-synthetic baselines across a wide array of tasks and languages. Moreover, XL-LoRA, our best performing method, requires no data in the target language to optimise the generator and produces highly competitive results from just a small scale finetuning dataset, though it can in principle be scaled further. We believe this offers promising pathway for producing performant models for the myriad languages where the requisite resources would otherwise be unavailable.

## 2 Background and Related Work

**Transformer Embeddings:** Early encoder-only transformers such as BERT (Devlin et al., 2019) produced poor embedding models, underperforming simple word embeddings despite being much

[1]Code available at: github.com/mbasoz/xllora-embedding



*Proceedings of the Second Workshop on Language Models for Low-Resource Languages (LoResLM 2026)*, pages 408–425

more powerful in theory (Reimers and Gurevych, 2019). This was in large part due to a tendency to produce anisotropic embedding spaces (Su et al., 2021; Machina and Mercer, 2024), which is provably harmful for learning useful representations (Wang and Isola, 2020a). However, these limitations were short-lived. Seminal work by Reimers and Gurevych (2019) demonstrated that by harnessing supervised NLI datasets and an appropriate representation level objective, transformer embeddings could achieve a new state of the art. This discovery was quickly followed by the SimCSE framework of Gao et al. (2021). In SimCSE representations are optimised using a contrastive objective inspired by advances in computer vision (Chen et al., 2020). The objective requires maximising the cosine similarity between an anchor sentence and a positive example, while simultaneously minimising similarity to a set of negative examples. It is defined as follows:

$$-\log \frac{e^{\text{sim}(h_i,h_i^+)/\tau}}{\sum_{j=1}^{M}\left(e^{\text{sim}(h_i,h_j^+)/\tau}+e^{\text{sim}(h_i,h_i^-)/\tau}\right)}. \quad (1)$$

Where $h_i$ denotes the anchor, $h_i^+$ denotes the positive target and $h_i^-$ denotes the hard negative. The objective can be applied in an unsupervised manner, in which case the term including $h_i^-$ is omitted, and the positive simply corresponds to the anchor with a different dropout mask applied. However, performance improves significantly when the objective is applied in a supervised setting, where for each example sentence a human annotated positive and hard negative is provided forming a triplet. The SimCSE framework set a new state of the art and has remained dominant since. However, its success is crucially dependent on the existence of high-quality triplet data, with the best reported performance coming from triplets obtained via the NLI dataset (Bowman et al., 2015).

**Embeddings for Low Resource Languages:** A core challenge of the transformer-based approach is its reliance on supervision and scale, making it difficult to apply to low resource languages. One approach to this issue is to use lightweight models that can be trained fully unsupervised (Mao and Nakagawa, 2023; Bestgen, 2024; Opper and Siddharth, 2024, 2025). However, these models have weaker capacity than large scale transformers, which places a ceiling on their efficacy. A second approach is to attempt to train multilingual models that have aligned embedding spaces, so that knowledge of a high resource language can at least partially transfer to a low resource one. This can be achieved implicitly through the inclusion of multiple languages in the pre-training data and careful control over sampling from each (Conneau et al., 2018; Marone et al., 2025). Or explicitly through the use of large-scale parallel corpora (Lample et al., 2017; Feng et al., 2020; Wieting et al., 2021; Heffernan et al., 2022). Aligned embedding spaces enable cross lingual transfer (Reimers and Gurevych, 2020; Nair et al., 2022), whereby such models can be finetuned on high quality supervised data in one language and then partially transfer the same capability to the target language. This approach can produce very strong results. However, it remains suboptimal compared with the ideal situation where high-quality supervised data is available in the target language.

**Data Synthesis Using LLMs:** As the capabilities of LLMs grow, they have begun to gain ever-increasing traction as tools for data synthesis. Arising from the hope that the sheer amount of knowledge they have internalised can be used to address the challenge of obtaining high-quality data for the multitudes of tasks and domains that cover NLP (Hartvigsen et al., 2022; Sahu et al., 2022; Wang et al., 2023; Honovich et al., 2023). Most relevant for our work, Zhang et al. (2023b) introduce SynCSE: a technique for synthetically generating triplet data required to train effective embedding models. To do so they use five-shot in-context learning, with alternating prompts and examples and either generate the positive and negative examples conditioned on a provided anchor sentence or generate all three components of the triplet simultaneously. Zhang et al. (2023b) demonstrated strong results on standard English benchmarks, leaving open the promise that the same technique could be applied to generate data for specialised domains or language adaptation.

**Summary:** When coupled with high-quality triplet data, transformer encoders can produce powerful, effective embeddings. For high resource languages like English, where such datasets abound, there exist a vast array of effective embedding models. However, for low resource languages, where such data is lacking, producing embeddings remains a

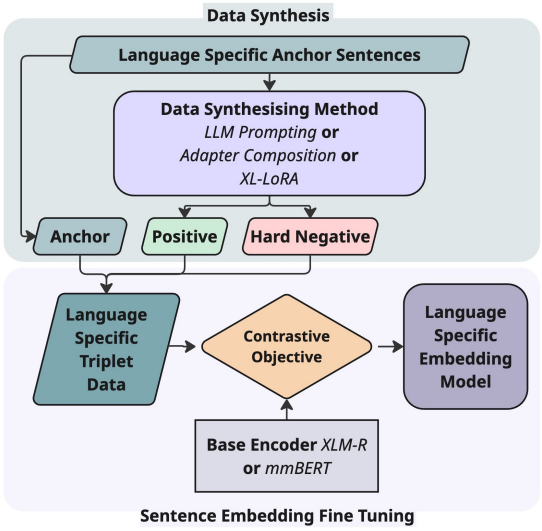


Figure 1: Pipeline overview: data is synthesised using an LLM, the result of which is then used to finetune an encoder, resulting in the final embedding model.

challenge. We explore whether the advent of powerful decoder only language models can allow us to bridge this gap. Can we synthesise high-quality examples for low resource languages, and what are the best approaches to do so?

## 3 Pipeline and Methodology

In this section, we outline a) the overall data generation pipeline and subsequent training of the embedding model and b) the different techniques used to optimise the LLM data generator.

### 3.1 Pipeline

Our data synthesis pipeline consists of three core stages. First, we collect anchor sentences for the target language from widely available web corpora. Second, we pass these sentences to the LLM generator which outputs the corresponding positive and hard negative pairs for the anchor, forming the triplet. Finally, once we have created the entire dataset, we then use it to optimise our transformer encoder backbone using the supervised SimCSE contrastive objective (Gao et al., 2021). The resulting embedding model is then evaluated zeroshot on downstream tasks. Figure 1 shows an overview of the pipeline. For all experiments, we used Gemma 3 27b (Team, 2025) as the generator, which we found to be the most effective multilingual model given our compute budget.

**In-Context Learning with Prompting:** Our first approach follows that of SynCSE (Zhang et al., 2023b), which demonstrated that with fewshot prompting applied to English, language models can generate effective synthetic data for training embedding models, outperforming unsupervised baselines and approaching parity with human annotations. Motivated by these findings, we tested whether SynCSE can be applied to low resource languages, where the need for synthetic data is arguably far higher. An overview of the exact pipeline for generating prompts can be found in Appendix A.3 while the prompts themselves can be found in A.4. Our approach closely mirrors the original SynCSE, though it differs in two regards. First, we used Gemma 3 27b rather than GPT 3.5 for parity with our other experiments. Second, we instruct the model to provide the output in the target language while the examples are in English. We also experimented with translating the prompt and examples, but found this to have a negligible effect compared to using English.

**Adapter Composition:** Beyond simple prompting, we also explored whether optimising the LLM generator itself could prove beneficial. After all, producing quality triplets requires an excellent grasp of semantics, and on top of that, the model must also be able to apply this knowledge to a low resource language. To this end, we turned to LoRA (Hu et al., 2021) finetuning, as it has demonstrated high efficacy in low-data training regimes (Whitehouse et al., 2024) and provides minimal computational overhead. A further desirable characteristic of LoRA is that adapters can be composed, allowing for flexible combination of different capabilities (Zhang et al., 2023a; Zhao et al., 2024). This is particularly desirable in our case because we are looking to optimise for two characteristics: a) the model must be able to understand what constitutes a useful positive or negative example for a given anchor, and b) it must have competency in the target language to generate grammatical outputs. As a result, we turn to AdamergeX. A technique recently introduced by Zhao et al. (2024), which showed strong results in cross lingual transfer in key tasks, including natural language understanding, summarisation, and reasoning. The core equation underlying AdamergeX composition is given as follows:

$$\underbrace{\mathbf{TA}_{tgt}}_{\text{target task}} = \underbrace{\mathbf{TA}_{src}}_{\text{source triplet tuning}} \overbrace{+}^{\text{elem-wise}} \underbrace{\lambda\left(\mathbf{LA}_{tgt} - \mathbf{LA}_{src}\right)}_{\text{Generic Instruction Tuning}} \tag{2}$$

Shown above, AdamergeX requires training three separate adapters: an adapter for the task in the source language and two generic language adapters, one in the source language and one in the target. The source language adapter is subtracted from that of the target to remove any artifacts outside of pure linguistic competency and is then added to the task adapter, with a hyperparameter lambda controlling the impact of the language adapter on the composition. In our case, the task is triplet generation and we train the corresponding adapter using a subset of English NLI (Williams et al., 2017; Bowman et al., 2015). For training the task adapter we use an instruction tuning format, inputting premise and outputting contradiction or entailment sentences. The prompts to convert the task to the requisite format can be found in Appendix A.8. Language adapters are trained using causal language modeling on the Aya dataset. We used either the core dataset or the machine translated collections split based on availability (Singh et al., 2024). Following ablations, which can be found in A.5, we observed that training two separate Task Adapters, for generating the positive and hard negative respectively, improved performance compared with using a single adapter for both. Further ablations over hyperparameters can be found in Appendix A.8. Finally, following Zhao et al. (2024) and due to the limited size of Aya, we used 10k examples to train each adapter.

**XL-LoRA:** For our final approach, we introduce a technique we term XL-LoRA (cross lingual LoRA). Here we do not require the generator to produce outputs in the target language. Rather we generate English positives and negatives while only the anchor remains in the target. This approach is motivated by two key observations. First, LLMs display internal cross lingual alignment (Wendler et al., 2024). Whereby in the middle layers tokens frequently occupy an abstract concept space, shared across languages. Second, LLMs have a strong bias towards outputting English. This means that even if they understand a task completely, the very fact of having to output non-English tokens can be highly detrimental to performance, masking true capabilities and perhaps partially being responsible for the capability gap observed in low resource languages (Pomerenke et al., 2025; Ahuja et al., 2023). Consequently, we constructed an additional finetuning set where anchors are in the target language, but the positives and negatives (i.e. generation targets for the LLM) are in English. We used the following steps to create this dataset as illustrated in Figure 2.

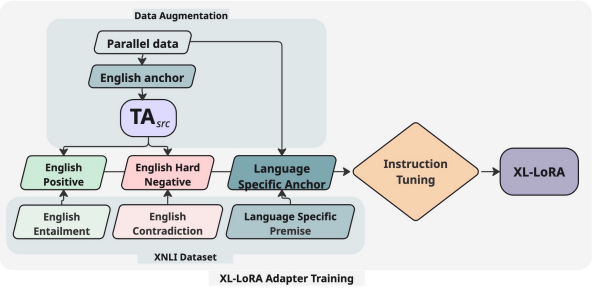


Figure 2: XL-LoRA training data construction. Starting with high quality translations we generate positives and negatives based on English, before swapping back the anchor to the original non-English language. Resulting examples are used to finetune the XL-LoRA generator.

**Step 1:** First, we sourced high-quality translations between English and non-English languages[2]. One of these is the XNLI test set (Conneau et al., 2018), which unlike the XNLI training set, contains translations by human experts. In this case, we already had positives and negatives available for each example so we simply swapped the English anchor for multilingual ones. As a result, the languages used to train the adapter are restricted to those present in XNLI and are largely differfrom the target languages used as the anchor sentences later, with Hindi as the sole exception. This avoids the need to include target language supervision, which is often difficult to obtain for low-resource languages.

**Step 2:** To further increase the number of examples to 10k, matching the SFT set sizes used for AdaMergeX, we could not rely on the XNLI test set alone, as it is very limited in scale. For further data, we found high-quality human translation data (Tiedemann and de Gibert, 2023) in the same languages covered by XNLI. We then used our triplet generation model, finetuned on English, to generate the corresponding positives

[2]Training of the XL-LoRA adapter includes the following languages: Arabic, Bulgarian, German, Greek, Spanish, French, Hindi, Russian, Swahili, Thai, Turkish, Urdu, Vietnamese and Chinese.

and negatives before swapping out the English anchor with its multilingual equivalent. We found that high-quality translations in the XL-LoRA training were absolutely crucial to ensuring success (ablations can be found later in the paper), and using machine translated examples was highly detrimental. However, we note that this is only required for the small scale finetuning of the generator. Additionally, we emphasize that, unlike in the adapter composition approach, no data in the target language is required for training the XL-LoRA adapter, and the vast majority of the languages we later evaluate on are not present in the training data.

**Step 3:** We train the XL-LoRA adapter using instruction tuning, where the model is prompted in a zero shot manner using the positive and negative pair generation instructions described in Appendix A.8. Each prompt takes a non-English anchor sentence as a premise and instructs the model to generate either a contradictory sentence as a hard negative or an entailment sentence as a positive in English. During training, the reference positives and hard negatives are provided only as target outputs for supervision and are not included in the input prompt. Once this is completed, the generator can then be deployed at scale for data synthesis.

**Summary:** We present three methods for optimising our synthetic data generator: one based on prompting, the second on adapter composition and the final uses cross lingual examples. Next, we test them experimentally.

## 4 Experiments

**Setup:** We want to evaluate the efficacy of synthetic data for training embedding models suitable for low resource languages. To do so we select two popular and capable multilingual models, XLM-R (Conneau et al., 2020) and mmBERT (Marone et al., 2025), as backbones that we finetune into embedding models. We compare against the following baselines not reliant on synthetic data:

1. **Base Encoder:** The first baselines are simply the pretrained backbones without any finetuning. These are unlikely to be good embedding models out of the box (Reimers and Gurevych, 2019), but provide a useful lower bound.

2. **Unsupervised:** Here we finetune backbones using the unsupervised SimCSE objective (Gao et al., 2021) on unlabeled data in the target language. This approach mitigates the lack of human annotated data, and has been shown to produce strong results on English STS.

3. **Cross Lingual:** While unsupervised SimCSE is a strong method for training embedding models its performance still lags in comparison to human annotated data. An alternative approach is to take advantage of the fact that multilingual encoders learn partially shared representations across languages (Conneau et al., 2020), which enables a good degree of zeroshot cross lingual transfer. Therefore, a highly performant alternative approach is to finetune using human annotated English data, and then evaluate zeroshot transfer to the target language (Nair et al., 2022).

We compare these baselines against the three approaches for synthetic data generation described in Section 3. Namely, prompting using in-context examples, adapter composition and finally XL-LoRA. The cross lingual baseline is trained using supervised SimCSE on the English NLI dataset from Williams et al. (2017); Bowman et al. (2015). The unsupervised baseline for each target language is trained using sentences sourced from the Leipzig Corpora Collection (Goldhahn et al., 2012) except for the Hausa data sourced from Opus (Nygaard and Tiedemann, 2003), while the synthetic approaches use these same sentences as anchors and then generate the corresponding hard positives and negatives - totalling 275k training examples, in line with the English NLI sample size. We evaluate performance using both STS/STR tasks from (Ousidhoum et al., 2024) and a subset of retrieval tasks from MTEB (Muennighoff et al., 2022) specifically focusing on low resource languages. Further training and hyperparameter details can be found in Appendix A.2.1.

### 4.1 Results

STS Results can be found in Table 1. Overall, we see that all three synthetic data approaches outperform both the base encoder and unsupervised baselines by a substantial margin. The cross lingual baseline nevertheless proves a tough competitor, and outperforms the prompt based synthetic approach across the board. However, the more sophisticated synthetic approaches, adapter composition

| Method | XLM-R Afr | Hin | Mar | Tel | Ind | Hau | Kor | Score | MM-BERT Afr | Hin | Mar | Tel | Ind | Hau | Kor | Score |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Base Encoder | 56.2 | 52.7 | 55.7 | 46.3 | 46.7 | 4.1 | 60.8 | 46.1 | 72.0 | 63.8 | 70.7 | 66.8 | **52.2** | 21.7 | 62.0 | 58.5 |
| Unsupervised | 74.8 ±0.3 | 69.7 ±0.2 | 77.1 ±0.2 | 76.2 ±0.3 | 39.2 ±0.3 | 45.4 ±0.7 | 70.9 ±0.3 | 64.7 ±0.2 | 75.5 ±1.2 | 58.0 ±2.6 | 68.6 ±1.3 | 62.1 ±3.5 | 44.7 ±1.7 | 25.5 ±1.2 | 58.0 ±2.9 | 56.1 ±0.8 |
| Cross Lingual | 78.0 ±0.3 | 77.4 ±0.1 | 81.6 ±0.2 | 80.9 ±0.7 | 47.1 ±0.6 | 48.2 ±1.0 | **79.9** ±0.1 | 70.4 ±0.2 | 78.0 ±0.4 | 77.7 ±0.1 | 79.8 ±0.4 | 73.6 ±0.6 | 49.3 ±0.6 | 29.4 ±1.7 | **79.6** ±0.1 | 66.8 ±0.4 |
| Synth - Prompting | **81.4** ±0.5 | 77.9 ±0.3 | 82.3 ±0.3 | 81.4 ±0.5 | 38.3 ±0.9 | 47.4 ±1.2 | 69.9 ±0.5 | 68.4 ±0.2 | 80.6 ±0.3 | 76.8 ±0.4 | 76.3 ±0.6 | 76.6 ±0.3 | 41.3 ±1.0 | 45.2 ±1.4 | 68.0 ±0.4 | 66.4 ±0.4 |
| Synth - Adapter Composition | 80.4 ±0.3 | 77.6 ±0.2 | **84.8** ±0.2 | 83.0 ±0.2 | 46.4 ±0.3 | 49.5 ±0.5 | 74.7 ±0.6 | 70.9 ±0.2 | 79.9 ±0.3 | 77.6 ±0.2 | 82.2 ±0.4 | 81.1 ±0.4 | 46.2 ±0.6 | 42.1 ±1.4 | 72.6 ±0.9 | 68.8 ±0.2 |
| Synth - XL-LoRA | 81.0 ±0.2 | **78.0** ±0.2 | 84.3 ±0.2 | **84.0** ±0.2 | **47.8** ±0.3 | **58.4** ±0.3 | 72.8 ±0.3 | **72.3** ±0.1 | **80.4** ±0.3 | **79.5** ±0.4 | **84.6** ±0.3 | **83.5** ±0.3 | 49.1 ±0.6 | **56.1** ±1.5 | 72.6 ±0.3 | **72.3** ±0.3 |

Table 1: Embedding performance on STS tasks (Spearman's correlation). We report results for two backbones, with mean ± standard deviation over four random seeds where applicable.

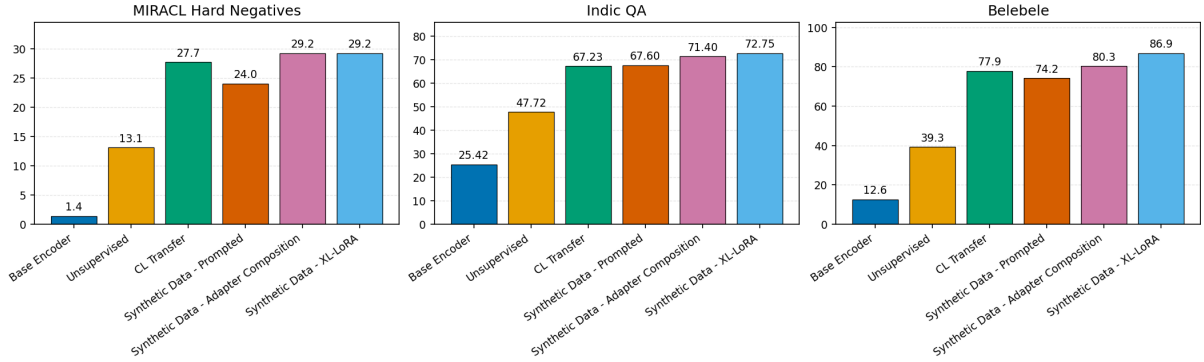


Figure 3: Retrieval results across multiple benchmarks. Results are averaged across backbones (XLM-R and mmBERT) and across languages. Metric is recall@10. Full results for all tasks and backbones can be found in Appendix A.7, but reflect the same clear trend depicted here.

and XL-LoRA, show consistent improvements over the cross lingual baseline, with XL-LoRA performing best of all. The same pattern can be seen in the retrieval results shown in Figure 3, where adapter composition and XL-LoRA outperform all baselines by a considerable margin, while the prompt based approach lags behind the cross lingual baseline. This trend is observable across backbones and across tasks, indicating that - provided the right considerations are taken in training the generator - synthetic data can be a powerful tool to bootstrap training where resources are otherwise unavailable. However, its success requires careful consideration and can be subject to multiple challenges and potential failure modes. In the following section, we will explore the behaviour of the various data generation methods and attempt to shed light on their strengths and weaknesses, together with their subsequent effects on the representation space.

## 5 Analysis and Ablations

Successfully generating high quality synthetic data can be challenging, particularly in somewhat OOD settings such as low resource languages. Here we highlight some of the key failure modes and challenges we encountered in the hope they will prove useful for future development.

The first approach we tried was prompt based generation using in-context examples. This is the SynCSE approach advocated by Zhang et al. (2023b), which showed very strong performance on standard English benchmarks, and the promise of being easily extensible to other languages and domains. However, in our experiments we found this approach suffers from three key difficulties:

1. Lack of competency in the target language: The model often struggled to generate fluent and coherent sentences in the target language. Frequently mixing in words from other languages, and producing ungrammatical sen-

| | Data type | Anchor | Positive | Hard Negative |
|---|---|---|---|---|
| **Example 1** | Annotated | yeah it's really sad i don't know i just think um | That's a sad thing. | Its an uplifting topic to talk about. |
| | Synthetic - Prompted | | The person appears to be expressing emotional distress and uncertainty. | yeah it's great I'm sure I just feel like yeah |
| | Synthetic - LoRA | | Yes, I agree that is really sad. | Yeah I know, it's actually very happy. |
| **Example 2** | Annotated | that one person has total control and i always figured at least in a day care center there are other people around and if you get one bad apple there's are at least other people that can see it they can watch and i just kind of always felt that the chances of something happening were less | It seems like even though there are some bad kids in daycare, there's always someone around to supervise. | I've never experienced any kids with bad behaviour in daycare. |
| | Synthetic - Prompted | | I figured a daycare center with multiple staff would be safer because others can monitor and prevent bad behavior. | I figured one bad apple could cause harm with no others to supervise. |
| | Synthetic - LoRA | | I always thought that there were less chances of something happening in a day care because there are other people watching. | I thought the chances of something happening were high. |

Table 2: Synthetic triplet examples vs annotations (English).

| Synthesised Sentence | Type |
|---|---|
| Depremde hasar gören köprülerin restorasyonuna **bereits** başladı. | Hard negative |
| Niyeti **сегодня** muhtemelen açık bir şekilde belli etti. | Hard negative |
| Tatil beldesinde meydana gelen gelişmeleri理解 etti. | Positive |
| Son **xuâtlanan** roman gerçekten mükemmel. | Positive |

Figure 4: Synthetic Turkish data produced via ICL-Prompting. Unintended code switching and poor intelligibility are apparent throughout.

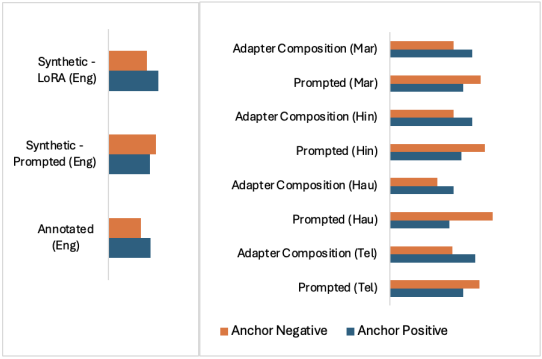


Figure 5: Lexical overlap between the anchor and positive/negative pairs computed via Dice coefficient. English results, including gold human annotations, are shown on the left. Right compares adapter composition with prompted data in low resource languages. XL-LoRA is excluded as languages differ within the triplet.

tences. These issues compound the more low resource the language is as demonstrated in Figure 4.

2. Contextual and pragmatic mismatch: The generated hard negatives often failed to be contextually or pragmatically aligned with the anchor sentence. For example, defaulting to a generic style whereas the anchor may be e.g. more conversational/informal as observed in the first example of Table 2. Human annotations, meanwhile, mirror the tone.

3. High lexical overlap between anchor and hard negative compared to anchor and positive (Figure 5): While it may not be initially obvious why this is an issue, it differs strongly from the human annotations, and is indicative of the model using lazy strategies such as simply including a negation (e.g., that was *not* good) which offer limited semantic diversity and scope for learning. This pattern is also observed in the qualitative analysis in Appendix A.9.

Overall, the adapter composition approach yields stronger results than the prompting approach and shows consistent, though moderate, improvements over the cross-lingual baseline. However, while we found that task adaptation worked extremely well as seen in Appendix A.1, the performance of the composed adapters was more variable. Alignment and uniformity analysis (Wang and Isola, 2020b) in Figure 7 reveals that embeddings produced by adapter composition exhibit weaker alignment than those from cross-lingual and prompting methods. This behavior is intuitive when considering that adapter composition relies on anchor–positive pairs with higher lexical overlap than the prompting synthesised method (see Figure 5) reducing their lexical diversity. As a result, the model may struggle to group semantically similar sentences that differ lexically. While adapter composition demonstrates a strong potential, its weaker alignment highlights clear opportunities for further refinement to enhance its efficacy in matching semantically similar sentences.

| Data type | Anchor | Positive | Hard Negative |
|---|---|---|---|
| **MT** | Arabalar ucuz benzin satılan istasyona yönelmediler. | Cheap gasoline is the station they are going to . | Cars headed to the station because gas was expensive there . |
| **HT + Syn** | (The cars did not head toward the station where cheap gasoline was sold.) | Cars didn't go to the station that sold cheaper petrol. | All of the cars were driving to the gas station that had the most expensive gas. |
| **MT** | Kimya laboratuvarındaki hava ölümcül. | The air in the laboratory was lethally silent . | The chemistry lab is empty . |
| **HT + Syn** | (The air in the chemistry laboratory is deadly.) | The air in the chemistry lab will kill you. | The chemistry lab is a very safe place. |
| **MT** | Zehra elinde kalan son filmleri izlemeye başlamadı. | She has watched all of her movies but a few . | Zehra 's favorite movie is playing . |
| **HT + Syn** | (Zehra didn't start watching the remaining films.) | Zehra had not commenced watching the remaining films. | Zehra watched her slides shows |

Figure 6: Examples of data synthetsised by the XL-LoRA generator when trained on machine-translated (MT) and human-translated plus synthesised (HT+Syn) data. MT data shows leads to lower quality.

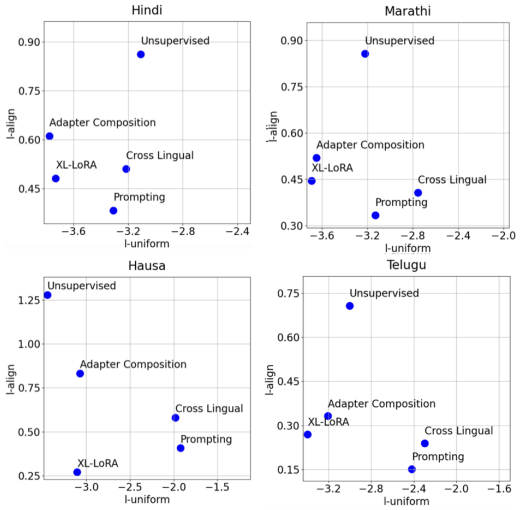


Figure 7: Alignment and Uniformity values of finetuned embedding models, based on the corresponding STR language embeddings[3]. Lower $\ell_{\text{align}}$ values indicate that similar sentences are clustered closely together in the embedding space, while lower $\ell_{\text{uniform}}$ values indicate a more evenly distributed embedding space, reducing anisotropy. Here mmBERT is used as the base encoder, but trends are consistent across backbones (see A.10)

The final model we evaluated was XL-LoRA, which leverages the stronger proficiency of models in high resource languages by generating hard positives and negatives in English while maintaining the language-specific anchors. Beyond achieving strong performance across all evaluation settings, XL-LoRA generally exhibits improved uniformity across languages relative to other synthetic approaches and cross lingual method. Furthermore, it maintains strong alignment compared to both adapter composition and unsupervised methods, as illustrated in Figure 7. This results in a more balanced embedding space that effectively clusters semantically similar pairs while preserving separation between distinct examples.

[3]Alignment and uniformity are computed per language from the STS test set, using sentence pairs with normalized similarity $\geq 0.8$ as positives and all test sentences to estimate the data distribution.

| | 10k | 20k |
|---|---|---|
| hin | $79.5 \pm 0.4$ | $\mathbf{80.0 \pm 0.1}$ |
| mar | $\mathbf{84.6 \pm 0.3}$ | $83.9 \pm 0.1$ |
| tel | $83.5 \pm 0.3$ | $\mathbf{84.9 \pm 0.6}$ |
| hau | $56.1 \pm 1.5$ | $\mathbf{59.3 \pm 0.9}$ |
| avg | $75.9 \pm 0.5$ | $\mathbf{77.0 \pm 0.2}$ |

Table 3: STS results (mean $\pm$ std over 4 seeds, Spearman's correlation) for XL-LoRA method (mmBERT as the base encoder) trained with 10k vs. 20k data.

We further observe that XL-LoRA is highly sensitive to training data quality: when trained on XNLI machine-translated data, the LLM generator produces positives which exhibit notable semantic mismatches with their anchors, as shown in the qualitative analysis in Figure 6 in Appendix A.6. Replacing machine-translated data with XNLI human-translated data augmented by synthesised examples resolves this issue, highlighting the importance of high-quality training data for effective sentence generation.

Finally, we conducted a limited experiment to study whether scaling the amount of training data improves the performance of the XL-LoRA adapter. In this setup, the training data for both the XL-LoRA adapter and the LoRA based task adapter used for the data augmentation step is increased from 10k to 20k. As shown in Table 3, even this basic increase leads to improved STS performance across most languages. We note that this is only one axis of scaling, and there are many options including increasing source dataset diversity and LoRA rank.

Additional ablation studies are included in the Appendix section, detailing hyperparameter tuning for non-English sentence embedding models (Appendix A.2), adapter composition ablations for triplet data synthesis strategies (Appendix A.5) and XL-LoRA triplet data synthesis strategies (Appendix A.6). Full retrieval evaluation tables are also provided in Appendix A.7.

## 6 Conclusions and Future Work

We investigate whether synthetic data can be used in order to train capable embedding models for low resource languages. We find that while in-context learning through prompting is unlikely to be sufficient, more sophisticated approaches which use resource efficient LoRA adaptation of the LLM data synthesiser can prove highly effective. Notably, our best performing method, XL-LoRA, achieves considerable gains without requiring parallel data in the target language. Thouhg we limited the number of examples for finetuning in order to maintain parity with the adapter composition. *Many more* examples can be sourced both to optimise the English triplet generator, and as quality substitutions for the English anchor. With greater scale, performance should improve even further, as hinted by the results in 3. Compounded with the unrelenting advances in LLM capabilities, we believe this offers a bright future for low resource NLP.

## 7 Limitations

While we aim for broad experimental coverage, extending evaluation to additional language families and analysing performance with respect to typological features such as morphology, word order, and script would be valuable future work. Our experiments are also constrained by compute, limiting evaluation to LLMs below 100B parameters; scaling to larger or more recent models may further improve synthetic data quality, particularly for lower-resource and typologically distant languages. Finally, our pipeline focuses on encoder-based embedding models, and we do not explore alternative embedding paradigms such as decoder-based embeddings from instruction-tuned LLMs, which may prove far more capable backbones.

## 8 Acknowledgments

We thank Mirella Lapata, Edoardo Ponti, Sahil Verma and Vivek Iyer for their insightful feedback and discussions over the course of this project. We also give particular thanks to Su Kara for her valuable comments and suggestions on draft versions of this paper. Finally, MO was funded by a PhD studentship through Huawei-Edinburgh Research Lab Project 10410153 which helped to enable this work.

## References


Kabir Ahuja, Harshita Diddee, Rishav Hada, Millicent Ochieng, Krithika Ramesh, Prachi Jain, Akshay Nambi, Tanuja Ganu, Sameer Segal, Maxamed Axmed, Kalika Bali, and Sunayana Sitaram. 2023. Mega: Multilingual evaluation of generative ai. *Preprint*, arXiv:2303.12528.

Yves Bestgen. 2024. SATLab at SemEval-2024 task 1: A fully instance-specific approach for semantic textual relatedness prediction. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 95–100, Mexico City, Mexico. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *Preprint*, arXiv:2303.12712.

Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder. *Preprint*, arXiv:1803.11175.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. *Preprint*, arXiv:2002.05709.

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.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. 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.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic BERT sentence embedding. *CoRR*, abs/2007.01852.

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.

Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. 2012. Building large monolingual dictionaries at the Leipzig corpora collection: From 100 to 200 languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 759–765, Istanbul, Turkey. European Language Resources Association (ELRA).

Mustafa Halat and Ümit Atlamaz. 2024. ImplicaTR: A granular dataset for natural language inference and pragmatic reasoning in Turkish. In *Proceedings of the First Workshop on Natural Language Processing for Turkic Languages (SIGTURK 2024)*, pages 29–41, Bangkok, Thailand and Online. Association for Computational Linguistics.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022. ToxiGen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3309–3326, Dublin, Ireland. Association for Computational Linguistics.

Kevin Heffernan, Onur Çelebi, and Holger Schwenk. 2022. Bitext mining using distilled sentence representations for low-resource languages. *Preprint*, arXiv:2205.12654.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2023. Unnatural instructions: Tuning language models with (almost) no human labor. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14409–14428, Toronto, Canada. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Haoyang Huang, Tianyi Tang, Dongdong Zhang, Xin Zhao, Ting Song, Yan Xia, and Furu Wei. 2023. Not all languages are created equal in LLMs: Improving multilingual capability by cross-lingual-thought

prompting. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12365–12394, Singapore. Association for Computational Linguistics.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. *CoRR*, abs/2005.11401.

Anemily Machina and Robert Mercer. 2024. Anisotropy is not inherent to transformers. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4892–4907, Mexico City, Mexico. Association for Computational Linguistics.

Zhuoyuan Mao and Tetsuji Nakagawa. 2023. Lealla: Learning lightweight language-agnostic sentence embeddings with knowledge distillation. *Preprint*, arXiv:2302.08387.

Marc Marone, Orion Weller, William Fleshman, Eugene Yang, Dawn Lawrie, and Benjamin Van Durme. 2025. mmbert: A modern multilingual encoder with annealed language learning. *Preprint*, arXiv:2509.06888.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

Suraj Nair, Eugene Yang, Dawn J. Lawrie, Kevin Duh, Paul McNamee, Kenton Murray, James Mayfield, and Douglas W. Oard. 2022. Transfer learning approaches for building cross-language dense retrieval models. *CoRR*, abs/2201.08471.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-t5: Scalable sentence encoders from pretrained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874, Dublin, Ireland. Association for Computational Linguistics.

Lars Nygaard and Jörg Tiedemann. 2003. Opus—an open source parallel corpus. In *Proceedings of the 13th Nordic Conference on Computational Linguistics*.

Mattia Opper and N. Siddharth. 2024. Self-strae at semeval-2024 task 1: Making self-structuring autoencoders learn more with less. *Preprint*, arXiv:2404.01860.

Mattia Opper and N. Siddharth. 2025. Banyan: Improved representation learning with explicit structure. *Preprint*, arXiv:2407.17771.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024. SemEval-2024 task 1: Semantic textual relatedness for african and asian languages. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

David Pomerenke, Jonas Nothnagel, and Simon Ostermann. 2025. The ai language proficiency monitor – tracking the progress of llms on multilingual benchmarks. *Preprint*, arXiv:2507.08538.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. pages 3982–3992.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.

Gaurav Sahu, Pau Rodriguez, Issam Laradji, Parmida Atighehchian, David Vazquez, and Dzmitry Bahdanau. 2022. Data augmentation for intent classification with off-the-shelf large language models. In *Proceedings of the 4th Workshop on NLP for Conversational AI*, pages 47–57, Dublin, Ireland. Association for Computational Linguistics.

Shivalika Singh, Freddie Vargus, Daniel Dsouza, Börje F. Karlsson, Abinaya Mahendiran, Wei-Yin Ko, Herumb Shandilya, Jay Patel, Deividas Mataciunas, Laura OMahony, Mike Zhang, Ramith Hettiarachchi, Joseph Wilson, Marina Machado, Luisa Souza Moura, Dominik Krzemiński, Hakimeh Fadaei, Irem Ergün, Ifeoma Okoh, and 14 others. 2024. Aya dataset: An open-access collection for multilingual instruction tuning. *Preprint*, arXiv:2402.06619.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval. *CoRR*, abs/2103.15316.

Gemma Team. 2025. Gemma 3.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *Preprint*, arXiv:2104.08663.

Jörg Tiedemann and Ona de Gibert. 2023. The OPUS-MT dashboard – a toolkit for a systematic evaluation of open machine translation models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System*

*Demonstrations)*, pages 315–327, Toronto, Canada. Association for Computational Linguistics.

Tongzhou Wang and Phillip Isola. 2020a. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. *CoRR*, abs/2005.10242.

Tongzhou Wang and Phillip Isola. 2020b. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning (ICML)*, pages 9929–9939. PMLR.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. 2024. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. *Preprint*, arXiv:2412.13663.

Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. 2024. Do llamas work in English? on the latent language of multilingual transformers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15366–15394, Bangkok, Thailand. Association for Computational Linguistics.

Chenxi Whitehouse, Fantine Huot, Jasmijn Bastings, Mostafa Dehghani, Chu-Cheng Lin, and Mirella Lapata. 2024. Low-rank adaptation for multilingual summarization: An empirical study. *Preprint*, arXiv:2311.08572.

John Wieting, Kevin Gimpel, Graham Neubig, and Taylor Berg-Kirkpatrick. 2021. Paraphrastic representations at scale. *CoRR*, abs/2104.15114.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *CoRR*, abs/1704.05426.

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.

Qiyu Wu, Chongyang Tao, Tao Shen, Can Xu, Xiubo Geng, and Daxin Jiang. 2022. PCL: Peer-contrastive learning with diverse augmentations for unsupervised sentence embeddings. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 12052–12066, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jinghan Zhang, Shiqi Chen, Junteng Liu, and Junxian He. 2023a. Composing parameter-efficient modules with arithmetic operations. *Preprint*, arXiv:2306.14870.

Junlei Zhang, Zhenzhong Lan, and Junxian He. 2023b. Contrastive learning of sentence embeddings from scratch. *Preprint*, arXiv:2305.15077.

Yiran Zhao, Wenxuan Zhang, Huiming Wang, Kenji Kawaguchi, and Lidong Bing. 2024. Adamergex: Cross-lingual transfer with large language models via adaptive adapter merging. *Preprint*, arXiv:2402.18913.

# A Appendix

## A.1 English Experiments Setup

| Base Encoder | Base Model Description | Synthesised Data Generation Method | STS-B |
|---|---|---|---|
| RoBERTa Base | unsupervised SimCSE | – | 80.0 |
| | supervised-SimCSE on annotated English triplets | – | **85.4** |
| | supervised SimCSE | SynCSE-partial (GPT3.5) | 83.3 |
| | | SynCSE-partial (Gemma3-27b)* | 83.1 |
| | | LoRA TA (Gemma3-1b)* | 83.2 |
| | | LoRA TA (Gemma3-27b)* | 84.3 |

Table 4: STS-B performance (Spearman's correlation) of SimCSE-based English models. Results are reproduced via SimCSE fine-tuning methodologies, using their hyperparameter settings. * denotes synthesised data. Original scores for unsupervised SimCSE, supervised SimCSE, and SynCSE-partial (GPT-3.5) are 80.2, 85.8, and 83.9 (Gao et al., 2021; Zhang et al., 2023b).

## A.2 Non-English Setup Ablations

### A.2.1 Sentence Embedding Models Training and Hyperparameter Settings

To train our models, we utilise the SimCSE package, which is built on top of the Transformers library (Wolf et al., 2020), modifying the package to be able to extend the base encoder use to ModernBERT (Warner et al., 2024) architecture. Our approach builds upon pre-trained language models, leveraging the XLM-RoBERTa Base model (Conneau et al., 2020) and the mmBERT Base model (Marone et al., 2025) as the foundations for our low resource language models. We drew upon the hyperparameters established in the original SynCSE and SimCSE studies except the pooling method. Our ablation study indicates that using the average first last pooling method during training and inference, a batch size of 512, a learning rate of 5e-5 and a maximum sequence length of 32 consistently results in high scores across two different languages. For training the unsupervised SimCSE models, we use the same hyperparameters as the original unsupervised SimCSE, including a batch size of 512, a maximum sequence length of 32, and a learning rate of 1e-5. The ablation studies were conducted on the development sets of the evaluation data. All results report the average first–last pooling method, except for the Base Encoder models in the STS experiments, which use average pooling, as this pooler yields higher baseline scores for this specific task.

| Pooler Type | STR | |
|---|---|---|
| | Afr | Hin |
| cls + mlp | 77.9 | 81.8 |
| avg | 77.6 | 81.3 |
| afl | **78.4** | **82.0** |
| at2 | 78.0 | 81.6 |
| cbp | 77.7 | 81.8 |

Table 5: Spearman correlations for STR performance fine-tuned with Prompting synthesis method in Afrikaans and Hindi, using different pooler types

| Batch Size | MLM Obj | STR | |
|---|---|---|---|
| | | Afr | Hin |
| 128 | FALSE | 78.4 | 82.0 |
| 128 | TRUE | 78.5 | 82.1 |
| 512 | FALSE | **79.0** | **82.8** |

Table 6: Spearman correlations for STR performance fine-tuned with Prompting synthesis method in Afrikaans and Hindi, using different batch sizes and MLM objective.

## A.3 Localised SynCSE-partial pipeline

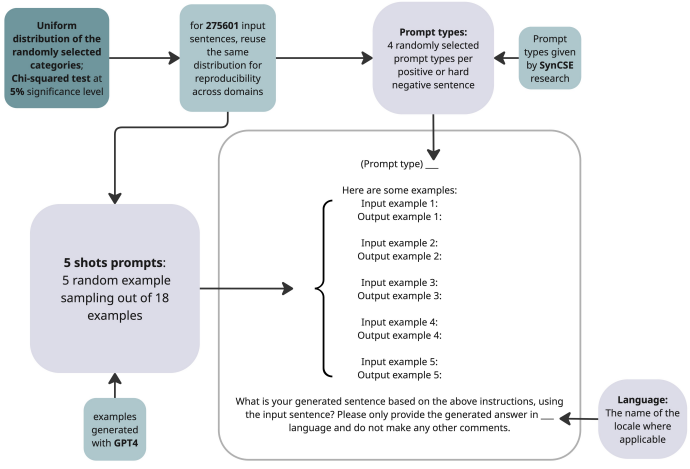


Figure 8: The prompting strategy for localised SynCSE-partial. As in the original method, each prompt includes five randomly selected examples from a pool of 18, together with a randomly chosen opening sentence from one of the four prompt types shown in Figure 9. Language specific details are added in the closing prompt.

## A.4 SynCSE-partial Positive and Hard Negative Prompts

| Positive prompts pools | Hard negative prompts pools |
|---|---|
| **Prompt1**: Please paraphrase the input sentence or phrase, providing an alternative expression with the same meaning. | **Prompt1**: Revise the provided sentence by swapping, changing, or contradicting some details in order to express a different meaning, while maintaining the general context and structure. |
| **Prompt2:** Rewrite the following sentence or phrase using different words and sentence structure while preserving its original meaning. | **Prompt2:** Generate a slightly modified version of the provided sentence to express an opposing or alternate meaning by changing one or two specific elements, while maintaining the overall context and sentence structure. |
| **Prompt3**: Create a sentence or phrase that is also true, assuming the provided input sentence or phrase is true. | **Prompt3**: Transform the input sentence by adjusting, altering, or contradicting its original meaning to create a logical and sensible output sentence with a different meaning from the input sentence. |
| **Prompt4**: Please provide a concise paraphrase of the input sentence or phrase, maintaining the core meaning while altering the words and sentence structure. Feel free to omit some of the non-essential details like adjectives or adverbs. | **Prompt4:** Generate a sentence that conveys a altering, contrasting or opposite idea to the given input sentence, while ensuring the new sentence is logical, realistic, and grounded in common sense. |

Figure 9: Positive and hard negative prompts used in the SynCSE-partial methodology (Zhang et al., 2023b) to promote diversity in data synthesis.

## A.5 Adapter Composition Triplet Data Synthesis Ablations

### A.5.1 Ablation Design

Ablation studies of the adapter composition method for triplet data synthesis include reproducing the pipeline of (Zhao et al., 2024) for the XNLI task using Gemma3 1B model and implementing the approach to the STR task for Gemma3 1B and 27B models. For ensuring the effectiveness of the implementation, we tested the performance of different task adapter training methods (Table 9) and different sources of training data for the language adapters (Table 10), rank values (Table 11) and lambda hyperparameters (Figure 10). The development set is used for the STR evaluation throughout the ablation experiments. The initial experiments synthesise a smaller set (50k) of data due to compute limits.

### A.5.2 LoRA Hyperparameters

For training all the LoRA adapters, the following hyperparameters are used; lora_alpha: 16, target_modules: all-linear following the AdaMergeX parameters. Rank is set to 8 based on our ablation results.

| Task Adapter | Adapter Composition Method | XNLI Accuracy Score | |
|---|---|---|---|
| | | XLM-R | Gemma3-1b |
| Base model | – | 33.3 | 35.3 |
| Hindi XNLI | – | 57.2 | 65.7 |
| English XNLI | – | 56.8 | 65.3 |
| English XNLI | AdaMergeX with Hindi LA & English LA | **60.5** | **65.8** |

Table 7: XNLI accuracy on the Hindi test set for Base models, LoRA task adapters (TAs) in Hindi and English, and the AdaMergeX cross-lingual TA.

| Data Synthesis Method | STR (Hindi) | |
|---|---|---|
| | Gemma3-1b | Gemma3-27b |
| Base | 79.1 | 81.0 |
| LoRA TA trained on English data | 80.2 | 82.5 |
| Crosslingual TA (Hindi) via Adamergex | **80.5** | **82.7** |

Table 8: Spearman correlations for STR performance in Hindi, using different data synthesis methods with 50,000 synthesized samples.

| TA Adapter Specs | LA Adapter Specs | STR (Hindi) |
|---|---|---|
| Trained a combined model with both negative and positive prompt examples | Aya annotated dataset | 81.1 |
| | Combined sentence corpora | 82.0 |
| | Individual sentences | 81.8 |
| | Aya annotated + collections dataset | 82.4 |
| Trained two individual models with negative and positive prompt examples | Aya annotated dataset | **82.5** |
| | Combined sentence corpora | **83.0** |
| | Individual sentences | 81.8 |
| | Aya annotated + collections dataset | **82.7** |

Table 9: Spearman correlations for STR performance in Hindi, using different language adapter (LA) and task adapter (TA) specifications with 50,000 synthesized data samples. Aya annotated dataset contains the human annotated samples (Singh et al., 2024), collections dataset denotes the Aya collections of machine translated samples. Combined sentence corpora and individual sentences experiments source the multilingual corpora from (Opper and Siddharth, 2025)

| LA Adapter Specs | STR | | |
|---|---|---|---|
| | Hindi | Telugu | Afrikaans |
| Aya annotated dataset | 82.2 | 81.0 | – |
| Combined sentence corpora | 82.7 | **81.4** | 77.4 |
| Aya annotated + collections dataset | **83.0** | 81.3 | **77.5** |

Table 10: Spearman correlations for STR performance across Hindi, Telugu, and Afrikaans for different language adapter (LA) and task adapter (TA) specifications synthesising the full dataset. Data for Afrikaans in the Aya annotated category was unavailable.

| | STR | | |
|---|---|---|---|
| | **Hindi** | **Telugu** | **Afrikaans** |
| Rank 4 | 82.1 | 81.7 | 77.0 |
| Rank 8 | **83.0** | 81.3 | **77.5** |
| Rank 16 | 82.4 | **81.9** | 77.4 |

Table 11: Spearman correlations for STR performance across Hindi, Telugu, and Afrikaans for different ranks synthesising the full dataset.

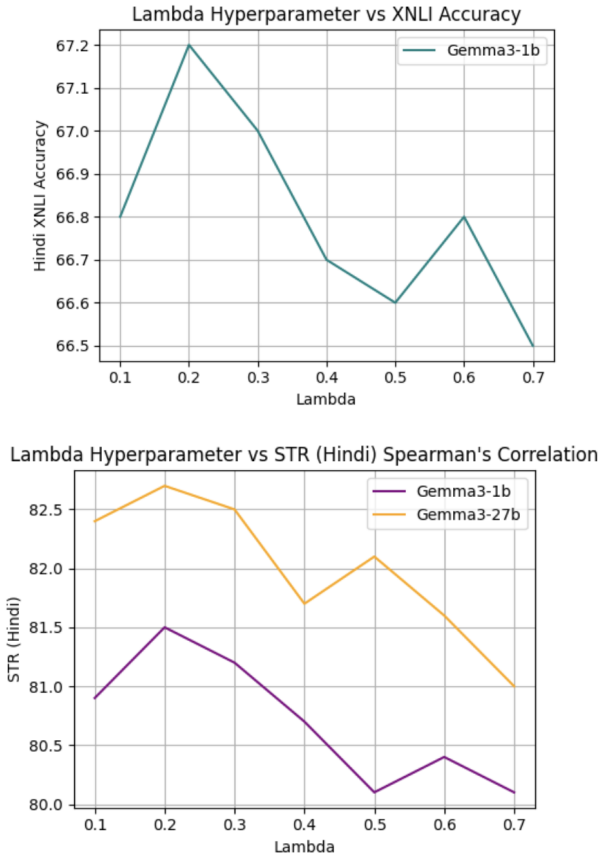


Figure 10: Accuracy for XNLI (dev) and STR (dev) performances across different lambdas.

## A.6 XL-LoRA Triplet Data Synthesis Ablations

| | XLM-R | | | | | | | mmBERT | | | | | | | Score |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | **Afr** | **Hin** | **Mar** | **Tel** | **Ind** | **Hau** | **Kor** | **Afr** | **Hin** | **Mar** | **Tel** | **Ind** | **Hau** | **Kor** | |
| MT | 80.0 | 78.2 | 83.6 | **84.7** | 48.3 | **61.0** | **73.9** | 80.5 | **79.8** | 83.9 | 83.1 | 50.7 | **59.4** | **74.1** | 72.9 |
| HT + MT | 80.2 | **78.4** | 83.5 | 84.1 | **49.3** | 60.7 | 72.9 | **80.7** | **79.8** | 84.5 | **84.6** | **51.7** | 59.1 | 72.7 | **73.0** |
| HT + Syn | **81.2** | 77.8 | **84.0** | 84.0 | 47.8 | 58.0 | 72.9 | 80.6 | 79.5 | **85.0** | 83.7 | 48.9 | 57.7 | 72.9 | 72.4 |

Table 12: STR results on synthesised data methods using XL-LoRA models trained on machine-translated (MT), human-translated plus machine-translated (HT+MT) and human-translated plus synthesised (HT+Syn) data.

| | nDCG@10 | | | | | | | | | Recall@10 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | XLM-R | | | | mmBERT | | | | Score | XLM-R | | | | mmBERT | | | | Score |
| **Method** | **Hin** | **Tel** | **Ind** | **Kor** | **Hin** | **Tel** | **Ind** | **Kor** | | **Hin** | **Tel** | **Ind** | **Kor** | **Hin** | **Tel** | **Ind** | **Kor** | |
| MT | 17.2 | 13.4 | 14.3 | 25.4 | 16.2 | 8.1 | 18.7 | 26.3 | 17.5 | 26.9 | 22.4 | 21.4 | 35.9 | 23.4 | 14.3 | 25.8 | 35.8 | 25.7 |
| HT + MT | 18.2 | 16.6 | 17.1 | 25.2 | 18.3 | **11.9** | 19.0 | **28.9** | 19.4 | **29.3** | 26.4 | 24.2 | 34.6 | 27.1 | **20.1** | 25.7 | **37.7** | 28.1 |
| HT + Syn | **18.8** | **16.7** | **18.1** | **25.9** | **20.8** | 11.6 | **22.5** | 26.9 | **20.2** | 28.4 | **28.0** | **25.2** | **37.1** | **29.3** | 19.4 | **29.8** | 36.2 | **29.2** |

Table 13: MIRACL Retrieval Hard Negative results on synthesised data methods (XL-LoRA; MT vs. HT+MT vs. HT+Syn)

| | nDCG@10 | | | | | | | Recall@10 | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | XLM-R | | | mmBERT | | | Score | XLM-R | | | mmBERT | | | Score |
| **Method** | **Hin** | **Mar** | **Tel** | **Hin** | **Mar** | **Tel** | | **Hin** | **Mar** | **Tel** | **Hin** | **Mar** | **Tel** | |
| MT | 49.9 | 56.8 | 54.2 | 46.6 | 55.7 | 49.5 | 52.1 | 69.4 | 75.0 | 74.6 | 65.2 | 74.3 | 67.8 | 71.1 |
| HT + MT | 50.2 | 57.5 | 54.9 | 48.2 | 56.2 | 51.1 | 53.0 | 68.7 | 75.3 | 74.7 | 67.2 | 75.0 | 68.9 | 71.6 |
| HT + Syn | **51.2** | **57.6** | **55.1** | **48.8** | **58.6** | **51.6** | **53.8** | **70.6** | **75.8** | **75.4** | **68.2** | **76.8** | **69.7** | **72.8** |

Table 14: Indic QA Retrieval results on synthesised data (XL-LoRA; MT vs. HT+MT vs. HT+Syn).

| **nDCG@10** | XLM-R | | | | | | | mmBERT | | | | | | | Score |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | **Afr** | **Hin** | **Mar** | **Tel** | **Ind** | **Hau** | **Kor** | **Afr** | **Hin** | **Mar** | **Tel** | **Ind** | **Hau** | **Kor** | |
| MT | 76.8 | 68.2 | 69.6 | 63.6 | 77.7 | 65.4 | 75.4 | 78.7 | 69.4 | 70.5 | 65.7 | 82.4 | **66.3** | 79.1 | 72.1 |
| HT + MT | 78.0 | **70.1** | **71.8** | 64.4 | 78.9 | 64.6 | 76.3 | 81.1 | **70.8** | 72.6 | **65.6** | 82.0 | 62.3 | 80.4 | 72.8 |
| HT + Syn | **79.2** | **70.1** | 71.6 | **64.6** | **79.1** | **65.8** | **78.4** | **81.9** | **70.8** | **73.9** | 65.5 | **83.7** | 58.9 | **81.0** | **73.2** |

Table 15: Belebele Retrieval nDCG@10 results on synthesised data (XL-LoRA; MT vs. HT+MT vs. HT+Syn)

| **Recall@10** | XLM-R | | | | | | | mmBERT | | | | | | | Score |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | **Afr** | **Hin** | **Mar** | **Tel** | **Ind** | **Hau** | **Kor** | **Afr** | **Hin** | **Mar** | **Tel** | **Ind** | **Hau** | **Kor** | |
| MT | 88.9 | 84.1 | 84.6 | 80.1 | 90.1 | 79.0 | 89.4 | 90.4 | 86.1 | 85.3 | 80.6 | 93.4 | **79.8** | 91.8 | 86.0 |
| HT + MT | 90.1 | 84.7 | 86.8 | **81.1** | 91.0 | 79.1 | 89.7 | 92.6 | 85.2 | 86.4 | 80.2 | 93.0 | 79.2 | 92.1 | 86.5 |
| HT + Syn | **91.1** | **84.8** | **87.0** | 80.2 | **91.9** | **80.1** | **91.6** | **92.7** | **86.3** | **87.8** | **81.2** | **94.0** | 74.2 | **93.3** | **86.9** |

Table 16: Belebele Retrieval Recall@10 results on synthesised data (XL-LoRA; MT vs. HT+MT vs. HT+Syn)

## A.7 Retrieval evaluation

| | nDCG@10 | | | | | | | | | Recall@10 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | XLM-R | | | | mmBERT | | | | Score | XLM-R | | | | mmBERT | | | | Score |
| **Method** | **Hin** | **Tel** | **Ind** | **Kor** | **Hin** | **Tel** | **Ind** | **Kor** | | **Hin** | **Tel** | **Ind** | **Kor** | **Hin** | **Tel** | **Ind** | **Kor** | |
| Base Encoder | 1.4 | 0.6 | 0.3 | 4.9 | 0.3 | 0.0 | 0.0 | 0.7 | 1.0 | 1.7 | 0.7 | 0.6 | 6.6 | 0.4 | 0.0 | 0.1 | 0.7 | 1.4 |
| Unsupervised | 16.1 | 11.0 | 8.6 | 22.7 | 2.2 | 0.7 | 3.3 | 7.7 | 9.0 | 23.5 | 17.6 | 12.1 | 29.1 | 3.8 | 1.0 | 5.2 | 12.1 | 13.1 |
| Cross Lingual | **19.8** | 13.3 | 16.5 | **27.4** | **22.0** | 8.2 | 20.0 | 28.3 | 19.4 | **29.8** | 22.0 | 23.9 | 36.3 | **30.6** | 13.6 | 27.7 | 37.5 | 27.7 |
| Synth - Prompting | 16.7 | 16.3 | 15.1 | 21.6 | 9.5 | 11.4 | 9.7 | 24.5 | 15.6 | 24.5 | **28.8** | 21.6 | 32.5 | 13.6 | **20.2** | 14.5 | 36.6 | 24.0 |
| Synth - Adapter Composition | 16.5 | **17.0** | **18.8** | 25.5 | 16.6 | 11.5 | 21.7 | **29.1** | 19.6 | 26.5 | 28.4 | **26.4** | 37.4 | 25.4 | 19.8 | 29.4 | **40.5** | **29.2** |
| Synth - XL-LoRA | 18.8 | 16.7 | 18.1 | 25.9 | 20.8 | **11.6** | **22.5** | 26.9 | **20.2** | 28.4 | 28.0 | 25.2 | **37.1** | 29.3 | 19.4 | **29.8** | 36.2 | **29.2** |

Table 17: MIRACL Retrieval Hard Negative results across languages.

| | nDCG@10 | | | | | | | Recall@10 | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | XLM-R | | | mmBERT | | | Score | XLM-R | | | mmBERT | | | Score |
| **Method** | **Hin** | **Mar** | **Tel** | **Hin** | **Mar** | **Tel** | | **Hin** | **Mar** | **Tel** | **Hin** | **Mar** | **Tel** | |
| Base Encoder | 5.9 | 21.0 | 21.7 | 1.9 | 2.0 | 39.6 | 15.4 | 12.2 | 31.1 | 43.1 | 4.3 | 4.6 | 57.2 | 25.4 |
| Unsupervised | 43.2 | 44.6 | 48.2 | 17.1 | 23.1 | 13.9 | 31.7 | 61.9 | 62.8 | 67.2 | 31.5 | 38.7 | 24.2 | 47.7 |
| Cross Lingual | 50.7 | 55.0 | 50.8 | 46.2 | 50.1 | 39.6 | 48.7 | 70.0 | 73.4 | 69.6 | 64.9 | 68.3 | 57.2 | 67.2 |
| Synth - Prompting | 52.3 | 52.3 | 51.8 | 44.4 | 47.8 | 42.5 | 48.5 | **72.0** | 71.7 | 72.1 | 62.9 | 67.0 | 59.9 | 67.6 |
| Synth - Adapter Composition | **52.5** | 57.5 | **57.7** | 48.5 | 52.7 | 47.0 | 52.7 | 71.5 | **75.8** | **77.5** | 66.2 | 72.3 | 65.1 | 71.4 |
| Synth - XL-LoRA | 51.2 | **57.6** | 55.1 | **48.8** | **58.6** | **51.6** | **53.8** | 70.6 | **75.8** | 75.4 | **68.2** | **76.8** | **69.7** | **72.8** |

Table 18: Indic QA Retrieval results across languages

| **nDCG@10** | XLM-R | | | | | | | mmBERT | | | | | | | Score |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | **Afr** | **Hin** | **Mar** | **Tel** | **Ind** | **Hau** | **Kor** | **Afr** | **Hin** | **Mar** | **Tel** | **Ind** | **Hau** | **Kor** | |
| Base Encoder | 14.7 | 11.5 | 13.8 | 14.8 | 27.9 | 5.9 | 23.0 | 1.1 | 0.8 | 0.9 | 1.0 | 1.1 | 1.2 | 1.2 | 8.5 |
| Unsupervised | 7.4 | 8.0 | 5.7 | 49.7 | 66.5 | 18.6 | 62.4 | 36.1 | 20.1 | 16.8 | 12.5 | 54.4 | 5.0 | 32.1 | 28.2 |
| Cross Lingual | 74.9 | 65.6 | 66.8 | 60.2 | **79.1** | 41.9 | 74.3 | 78.5 | 64.7 | 63.0 | 48.0 | 82.9 | 30.0 | 77.4 | 64.8 |
| Synth - Prompting | 70.5 | 58.5 | 61.0 | 56.6 | 71.0 | 48.3 | 72.7 | 65.0 | 53.6 | 53.5 | 48.7 | 73.2 | 28.0 | 70.9 | 59.4 |
| Synth - Adapter Composition | 77.7 | 64.0 | 68.6 | 61.3 | 76.9 | 56.7 | 73.1 | 76.6 | 58.8 | 61.0 | 57.6 | 79.9 | 43.6 | 72.7 | 66.3 |
| Synth - XL-LoRA | **79.2** | **70.1** | **71.6** | **64.6** | **79.1** | **65.8** | **78.4** | **81.9** | **70.8** | **73.9** | **65.5** | **83.7** | **58.9** | **81.0** | **73.2** |

Table 19: Belebele Retrieval nDCG@10 results across languages

| **Recall@10** | XLM-R | | | | | | | mmBERT | | | | | | | Score |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | **Afr** | **Hin** | **Mar** | **Tel** | **Ind** | **Hau** | **Kor** | **Afr** | **Hin** | **Mar** | **Tel** | **Ind** | **Hau** | **Kor** | |
| Base Encoder | 20.8 | 16.8 | 20.4 | 20.9 | 39.8 | 10.6 | 31.9 | 2.2 | 1.7 | 2.0 | 2.1 | 2.1 | 2.4 | 2.4 | 12.6 |
| Unsupervised | 12.8 | 13.7 | 11.2 | 64.1 | 81.2 | 30.0 | 78.3 | 52.2 | 32.0 | 27.0 | 22.0 | 69.2 | 8.9 | 47.0 | 39.3 |
| Cross Lingual | 87.4 | 80.0 | 83.1 | 77.1 | 90.3 | 54.2 | 86.9 | 90.6 | 78.7 | 78.1 | 62.8 | 92.7 | 37.3 | 90.7 | 77.9 |
| Synth - Prompting | 84.1 | 73.4 | 77.3 | 71.7 | 84.6 | 63.8 | 85.9 | 81.1 | 69.6 | 70.8 | 64.9 | 86.6 | 40.9 | 84.4 | 74.2 |
| Synth - Adapter Composition | 89.1 | 78.2 | 83.8 | 75.7 | 89.3 | 72.4 | 85.9 | 90.4 | 75.6 | 76.6 | 72.4 | 90.7 | 58.8 | 85.1 | 80.3 |
| Synth - XL-LoRA | **91.1** | **84.8** | **87.0** | **80.2** | **91.9** | **80.1** | **91.6** | **92.7** | **86.3** | **87.8** | **81.2** | **94.0** | **74.2** | **93.3** | **86.9** |

Table 20: Belebele Retrieval Recall@10 results across languages

## A.8 Zero-shot Prompting Strategies for Task Adapters

| Prompting Strategy | Zero-Shot Prompt |
|---|---|
| Huang et al. (2023) Prompting – XNLI Classification | I want you to act as a natural language inference expert for {language}. Premise: {premise}. Hypothesis: {hypothesis}. You should retell the premise and hypothesis in English. You should judge whether the hypothesis is true (entailment), false (contradiction), or undetermined (neutral) given the premise. The relationship can be chosen from entailment, contradiction and neutral. You should step-by-step answer the request. You should tell me the relationship in this format 'Relationship:'. |
| Modified Prompting – XNLI Classification (This Work) | You are an expert at natural language inference. Given a premise and hypothesis (in {language}), you should return an integer classification. The options are as follows: 0 for entailment, 1 for neutral, and 2 for contradiction. Return only the integer without any preamble or explanation. Premise: {premise}. Hypothesis: {hypothesis}. |
| Prompting – Positive Pair Generation (This Work) | You are an expert at natural language inference. Given a premise, you should return an entailment sentence example (in {language}) to the premise. Return only the entailment sentence example without any preamble or explanation. Premise: {premise}. |
| Prompting – Negative Pair Generation (This Work) | You are an expert at natural language inference. Given a premise, you should return a contradiction sentence example (in {language}) to the premise. Return only the contradiction sentence example without any preamble or explanation. Premise: {premise}. |

Table 21: Zero-shot prompting strategies used in this study.

## A.9 Qualitative Analysis - Turkish data

| | Data Type | Anchor | Positive | Hard Negative | Notes |
|---|---|---|---|---|---|
| **Example 1** | **Annotated** | Yemeğe giden arkadaşlarına bağırdı. | Yemeğe giden arkadaşlarına seslendi. | Yemeğe giden arkadaşlarına seslenmedi. | Grammar mistake on positive example of the prompted method. CLLoRA-Mix anchor and positive sentences are very similar. |
| | **Synthetic - Prompted** | | Yemeğe giden **arkadaşlarını bağırdı.** | Yemeğe giden arkadaşlarına yumuşak bir sesle teşekkür etti. | |
| | **Synthetic - Adapter Composition** | | Yemeğe giden arkadaşlarına seslendi. | Suskun bir şekilde yemeğe giden arkadaşlarını izledi. | |
| | **Synthetic - XL-LoRA** | | She shouted to her friends who were eating. | She quietly ate her meal alone. | |
| **Example 2** | **Annotated** | Kafasındaki uçağı herhalde derste tasarlamamış. | Kafasındaki uçağı derste tasarladığı yüzde yüz değilmiş. | Kafasındaki uçağı yüzde yüz derste tasarlamış. | Hard negative of the prompted method looks like a mirrored version of the anchor, flipping words into their opposites in a shallow way. |
| | **Synthetic - Prompted** | | Kafasındaki uçağı ders sırasında tasarlamamıştır. | Kafasındaki uçağı herhalde derste tasarlamış. | |
| | **Synthetic - Adapter Composition** | | Ders dışındaki bir zamanda tasarlamış. | Uçağı tasarladığına eminim. | |
| | **Synthetic - XL-LoRA** | | He didn't design that plane in class, I think. | He didn't think about planes at all during the lesson. | |

Figure 11: Qualitative review comparing the Turkish semi annotated data (Halat and Atlamaz, 2024), synthetic data via Prompted, Adapter Composition and XL-LoRA methods. We constructed the annotated triplet dataset from the ImplicaTR dataset (Halat and Atlamaz, 2024) by treating entailment pairs as positive examples and contradiction pairs as hard negatives.

## A.10 Alignment and Uniformity

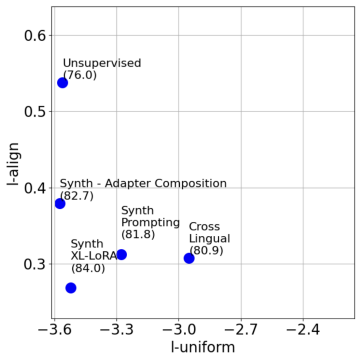


Figure 12: Alignment and Uniformity analysis of XLM-R based fine tuned sentence embedding models, based on Telugu STR language embeddings.