Mimir: Large-Scale Multilingual Concept Modeling

Anonymous ACL submission

Abstract

Current language modeling approaches are built around tokens. Text corpora are split into tokens, and models are trained by performing computations on these tokens, such as predicting the next token given the preceding ones as context. This paradigm has become the standard in modern language modeling, especially given the outstanding performance obtained by token-based architectures. However, recent works have not only begun to question how language models process and understand meaning from tokens, but also to question whether using higher levels of granularity could advance the research field. This led to the idea of Concept Modeling, that is, to directly train models for next-concept prediction rather than next-token prediction. In this work, we introduce Mimir, a 1.6B Large Concept Model trained for multilingual concept understanding and generation. We leverage a large-scale multilingual pre-training corpus (38,883,987,240 sentences) spanning 46 languages and a large-scale multi-turn and multilingual instruction-tuning dataset (66,816,428 sentences) covering a total of 35 languages. We extensively evaluate model performance against a language model with a comparable number of parameters.

1 Introduction

In recent years, Large Language Models (LLM) have completely revolutionized the Natural Language Processing (NLP) field of research (Qin et al., 2026). These models work on massive text corpora separated into tokens and learn to predict the next token autoregressively based on the ones provided as context. Despite the undeniable success of this paradigm, recent work has begun to explore alternatives to traditional token-level modeling.

One emerging research direction is **Concept Modeling**, as introduced by Large Concept Models (LCM team et al., 2024). LCMs treat sentences as concepts and perform the autoregressive prediction objective on concepts rather than tokens (Iyer et al., 2026). This approach encourages the model to reason at a higher level rather than relying solely on fine-grained lexical patterns.

One of the most remarkable advantages of the approach proposed by LCM team et al. (2024) is that the model is inherently capable of multilingual generation. This happens because concepts are represented as sentence embeddings that can be decoded into different languages through a multilingual decoder (Artetxe and Schwenk, 2019). However, as with token-based LLMs, training LCMs on data focused on a single language (e.g., English) limits their ability to support multilingual understanding. Current works on Concept Modeling have focused almost exclusively on English, without considering other non-English languages. This leaves a research gap in multilingual concept modeling. In light of this, we propose Mimir, the first Large Concept Model trained on large-scale multilingual data.

Hence, the contributions of this work are the following:

- We propose Mimir, the first Large Concept Model trained on multilingual large-scale data. We propose a 1.6B model trained on a corpus consisting of 38,883,987,240 multilingual sentences covering 46 languages;

- We perform evaluation on several languages and provide a comparison with a Large Language Model within a comparable range of parameters.

We provide all resources associated with this work to facilitate reproducibility and boost the current research trends in Concept Modeling[1].

[1]We will release all resources in case of acceptance

2 Related Works

2.1 Concepts and LLMs

Several recent works have studied the relationship between LLMs and concepts. Early studies focused on understanding whether LLMs encode and manipulate conceptual knowledge effectively. In Peng et al. (2022), the authors addressed the lack of benchmarks targeting conceptual rather than factual knowledge. To overcome this limitation, they collected 24,000 instances spanning 393 concepts and conducted extensive experiments on pre-trained language models, revealing that these models lacked conceptual knowledge. Other works have begun to question how concepts are internally represented within an LLM and whether a paradigm shift is possible. For example, Jin et al. (2025) introduced the idea of "Concept Depth". The authors suggested that LLMs learn concepts of varying difficulty across different layers, with more complex concepts learned at deeper layers of the model. Similarly, Han et al. (2025) presented a concept editing method for LLMs. Concept editing approaches focus on modifying the representations of specific concepts within LLMs to guide their outputs. They proposed a unified neuron-level paradigm for concept editing, a common framework for understanding and comparing diverse editing methods. Bhan et al. (2025) proposed the Complete Textual Concept Bottleneck Model, a method for generating concept labels that leverages a small, fine-tuned classifier language model.

Beyond analyzing conceptual representations in token-based LLMs, recent work has begun to rethink the language modeling objective itself. Iyer et al. (2026) proposed replacing the traditional next-token prediction objective with next-concept prediction, demonstrating that next-concept prediction performs better in terms of perplexity w.r.t. the traditional next-token prediction objective used in LLMs. LCM team et al. (2024) proposed the Large Concept Model, a model to perform autoregressive concept prediction in the SONAR (Duquenne et al., 2023) embedding space. They propose three variants of the LCM model: Base, One-Tower and Two-Tower.

Despite the efforts proposed to bridge the gap between concepts and LLMs, very few studies consider implementing a Large Concept Model and mostly focus on evaluating pre-trained LLMs with respect to their conceptual knowledge.

2.2 Multilingual LLMs

In recent years, there has been increasing interest in multilingual LLMs, with research focusing on scaling, evaluation, and cross-lingual learning. For example, Tanwar et al. (2023) explored in-context learning in a cross-lingual setting. They proposed a prompt construction strategy to replace the random selection of labeled training examples by combining semantic and task-based alignment. Subsequent efforts have focused on scaling the multilingual capabilities of these models to a larger number of languages. Lai et al. (2024) introduced xLLaMA-100 and xBLOOM-100, extending multilingual support to 100 languages. To do so, they proposed two datasets: one covering 100 languages for multilingual instruction tuning and one covering 30 languages for cross-lingual human preference modeling. Also focusing on scaling, Geigle et al. (2025) presented an exhaustive study on training strategies for massively multilingual multimodal LLMs. They studied best practices for multilingual and multimodal training data and proposed the Centurio model, a multimodal LLM supporting 100 languages.

Alongside model scaling, several works have addressed the evaluation and analysis of multilingual semantic capabilities. Ying et al. (2025) introduced a Dual Evaluation Framework for assessing the multilingual capabilities of LLMs. Focusing instead on internal semantic representations, Körner et al. (2026) studied the development of language-agnostic concept spaces during pre-training of EuroLLM (Martins et al., 2025) EuroLLM is a suite of open-weight multilingual LLMs capable of understanding and generating text in all official European Union languages. They showed that shared concept spaces emerge early and remain relatively stable throughout training.

Furthermore, multimodal research also expanded towards cultural understanding. Nyandwi et al. (2025) introduced CulturalGround, a dataset for evaluating multimodal LLMs' cultural understanding across 42 countries and 39 languages. They also introduced CulturalPangea, a multilingual and multimodal LLM that achieves state-of-the-art on culture-focused benchmarks.

Despite the rapid progress in multilingual language modeling, existing work remains largely intertwined with token-based architectures. To the best of our knowledge, no prior work has yet proposed a Large Concept Model trained on large-

scale multilingual data.

3 Data Collection

To train the model, we collected large-scale pre-training multilingual data. Specifically, for pre-training, we use the 350BT split of the Fineweb-edu dataset (Lozhkov et al., 2024) for the English language and the Fineweb 2 dataset (Penedo et al., 2025) for other languages. Fineweb-edu consists of educational web pages filtered from the Fineweb dataset, while Fineweb 2 is the second iteration of the Fineweb dataset, consisting of high-quality pre-training data for more than 1,000 languages. For Fineweb 2, we sample data for the languages so that the pre-train corpus is equally split into Fineweb-edu and Fineweb 2.

An overview of the language distribution in the pre-training dataset is shown in Figure 1, while detailed statistics are reported in Appendix A. We also note that Fineweb 2 unifies Chinese data under a unique language code, namely "cmn_Hani" (Mandarin Chinese). We distinguish between simplified ("zho_Hans") and traditional ("zho_Hant") Chinese using the *hanzidentifier*[2] library. We perform this additional processing step because SONAR expects either "zho_Hans" or "zho_Hant". Overall, across the entire pre-training dataset, we report 38,883,987,240 sentences covering 46 languages, of which 44.65% are in English and 55.35% in non-English languages.

For our instruction-tuning datasets, we consider the following datasets: 1) **Aya Dataset** (Singh et al., 2024), a multilingual instruction-tuning dataset curated by humans through an annotation platform. The dataset contains 204k human-annotated prompt-completion pairs across 65 diverse languages; 2) **Bactrian-X** (Li et al., 2023), a collection of 3.4M instruction-response pairs in 52 languages, that are obtained by translating 67K English instructions into 51 languages using Google Translate API. The translated instructions are then given to GPT-3.5-Turbo to obtain more natural responses; 3) **Open Assistant 2**[3], a collection of conversations collected from the open-assistant.io website. Since each conversation can have multiple paths, we reconstruct the best conversation based on the top-ranked response. The final dataset we processed consists of 19,654 instances covering 20 languages.

[2] https://github.com/tsroten/hanzidentifier

[3] https://huggingface.co/datasets/OpenAssistant/oasst2

We also add a synthetically generated multilingual dataset curated by us. Our objective was to create a multilingual multi-turn data mixture with culturally sensitive elements. We prompt the Qwen3-235B-A22B-Thinking-2507 model to provide conversational turns between a user and an AI assistant based on a given topic. The topics are sampled from the Everyday Conversations dataset (Hugging Face, 2024), which contains 2,200 multi-turn conversations generated by Llama-3.1-70B-Instruct on a given topic. The prompt used to generate this data is shown in Appendix D.

We also add a multilingual instruction-tuning dataset focused on math subjects to improve model performance on math-related tasks. In this case, we extract math problems and their solutions from the OpenMathInstruct-2 dataset (Toshniwal et al., 2024), a math instruction-tuning dataset comprising 14M problem-solution pairs generated with the Llama3.1-405B-Instruct model. The model is prompted to provide a conversation for that math problem in the target language and continue the conversation for additional turns. The prompt used to generate this data is shown in Appendix D.

For both Everyday Conversations and OpenMathInstruct, we include in the prompt instructions to randomize and diversify conversations (e.g., random number of turns, answer concisely).

Finally, we also include the train sets from MLQA (Lewis et al., 2020) and XL-Sum (Hasan et al., 2021) in the final instruction-tuning mixture. XL-Sum is a comprehensive and diverse dataset comprising 1.35 million professionally annotated article-summary pairs from BBC, extracted using a set of carefully designed heuristics, and covering 35 languages. MLQA is an extractive question-answering dataset over paragraphs covering 7 languages. The train set of MLQA is derived by machine translating the SQuAD dataset (Rajpurkar et al., 2016) into 7 languages (i.e., English, Arabic, German, Spanish, Hindi, Vietnamese, and Simplified Chinese).

To study both multilingual generalization and the impact of task-specific supervision, we consider four instruction-tuning settings: 1) multilingual with MLQA and XL-Sum in the train set; 2) multilingual without MLQA and XL-Sum in the train set; 3) English only with MLQA and XL-Sum in the train set; 4) English only without MLQA and XL-Sum in the train set. We consider variants with and without MLQA and XL-Sum in the training set to evaluate Mimir's generalization ca-

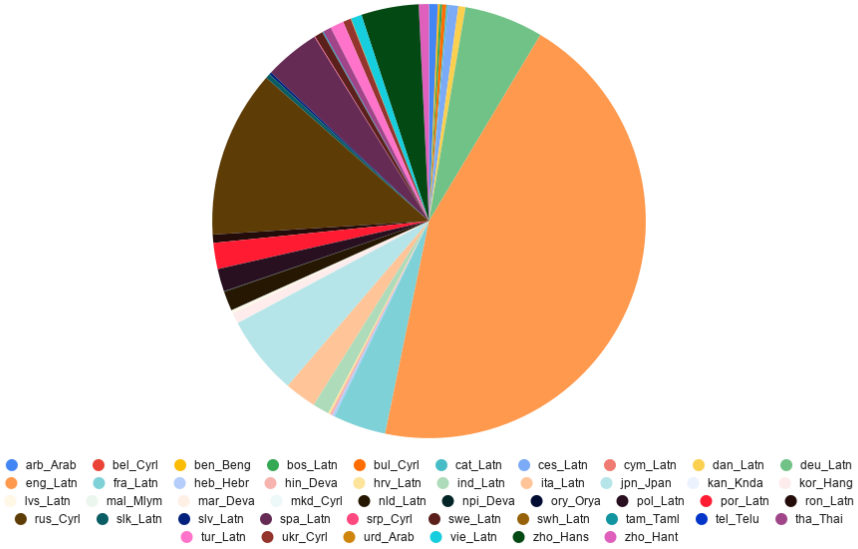

Figure 1: Pie chart of the distribution of sentences for the pre-training dataset languages

pabilities on tasks not seen during training. Similarly, the English-only variant has the aim to assess the performance discrepancy when using a model instruction-tuned only on English.

An overview of the instruction-tuning dataset cardinality per language is shown in Figure 2, while the complete statistics are reported in Appendix A. Overall, across the whole instruction-tuning dataset, we report 66,816,428 sentences covering 35 languages, of which 19.29% are in English and 80.71% in non-English languages.

In all cases, we split data into sentences using the **sat-3l** (Frohmann et al., 2024) model from *wtpsplit*[4] (Minixhofer et al., 2023). We fix a sentence threshold of 0.02 and a maximum sentence length of 256.

4 Methodology

We follow the implementation of the Two-Tower diffusion LCM model by LCM team et al. (2024). In this framework, concepts are treated as sentences, and the textual corpora are processed by splitting the input into distinct sentences. The architecture consists of two components: 1) a context encoder, where a decoder-only transformer is used to encode the contextual sequence of previous sentence embeddings; 2) a denoiser, where a stack of transformer blocks with cross-attention is used to attend over the encoded context representations. Hence, the model training objective is to generate the SONAR embedding of the next concept, given the previous sentences as context. Then, to generate text, a decoder reconstructs the original sentence from the embedding. Both the embedding encoder and the decoder are inherited from SONAR (Duquenne et al., 2023).

One of the advantages is their inherent support for multilingual language generation. Since the SONAR decoder can reconstruct text in any supported target language, even if the input embedding is encoded in English, it can be decoded into another language. However, while this solves the problem of multilingual generation, it does not address multilingual language understanding. While the SONAR decoder can reconstruct text in any language, the denoiser and context encoder are trained only on English embeddings. Moreover, SONAR embeddings are not aligned with language, implying that semantically equivalent sentences across different languages may have different embeddings. To investigate this limitation, we conduct a pilot experiment to measure the alignment between English and non-English SONAR embeddings in a parallel corpus. Specifically, we use the "parallel-sentences-wikimatrix"[5] dataset as a parallel corpus. We first extract the first 1,000 sentences for each language subset from the "dev" split, and then compute the cosine similarity between the English sentence embedding and its correspond-

[4] https://github.com/segment-any-text/wtpsplit

[5] https://huggingface.co/datasets/sentence-transformers/parallel-sentences-wikimatrix

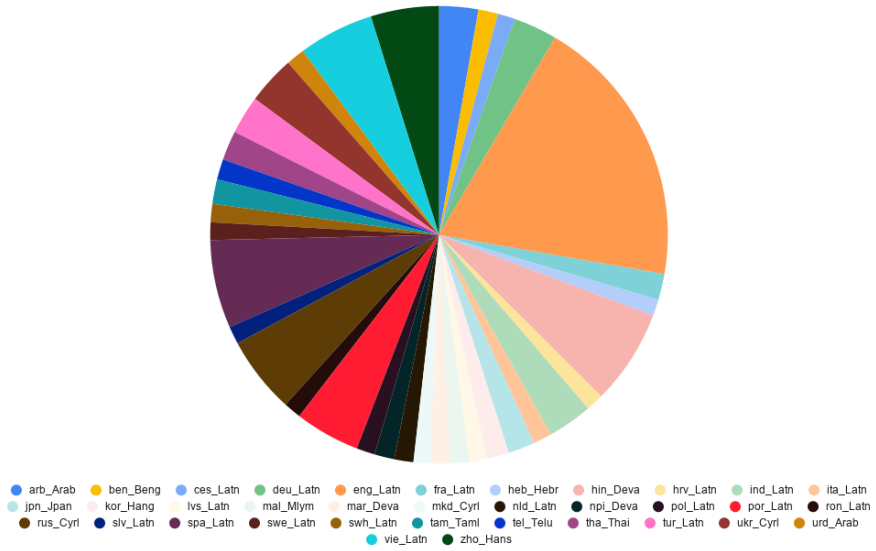

Figure 2: Pie chart of the distribution of sentences for the instruction-tuning dataset languages

ing non-English embedding. The average cosine similarity is computed for all 1,000 pairs. The results are shown in Appendix E. Overall, the results show that SONAR embeddings are not perfectly aligned across languages, as all subsets have cosine similarity scores below 0.90. This implies that models trained exclusively on English embeddings may develop a bias towards this language, making them less effective at handling prompts in other languages. A visualization of the differences between the two approaches is showcased is Appendix C. In light of this, we pre-train and fine-tune an LCM model on large-scale multilingual data.

For pre-training, the model is trained for 250,000 steps, with a 4e-4 learning rate, a cosine scheduler, 10,000 warmup steps, 0.1 weight decay, and a batch size of 229,376 sentence embeddings, following the original LCM model recipe. For instruction tuning, we train for 20,000 steps with a 1e-5 learning rate, a cosine scheduler, 0.01 weight decay, and a batch size of 512 instances. Rather than the autoregressive token-prediction objective, LCM models are trained on an autoregressive concept-prediction objective. The next-concept prediction objective is implemented using the MSE between the embedding generated by the model and the ground truth embedding of the input sentence in the SONAR space.

For pre-training, the entire input embeddings are processed autoregressively. Additionally, we append an "End of text." sentence at the end of all inputs. We use this sentence to stop generation at inference time when the model generates an embedding with a cosine similarity of 0.90 or greater to the SONAR embedding of the "End of text." sentence. Furthermore, following the original recipe by LCM team et al. (2024), we also trained a normalizer on the same data used for pre-training.

For instruction tuning, we adopt a formulation similar to completion-only training in traditional LLMs. More specifically, there is a fixed context not used for prediction (source embeddings) while the model is trained to predict only some fixed target concepts (target embeddings). In our setting, the user turn represents the source embeddings, while the assistant turn represents the target for prediction. For multi-turn data, we consider all previous user-assistant turns as context, alongside the current user turn. Hence, multi-turn data are split into multiple instances according to the number of turns (e.g., if a conversation has 4 turns, 4 instances are added to the dataset). We separate the user and assistant turns using dedicated sentences, namely "User turn." and "Assistant turn.". Finally, we append the "End of text." sentence at the end of the assistant turn, to stop generation during inference using the same strategy explained in pre-training. A formatting example is shown in Appendix B.

We perform pre-training and instruction tuning on a cluster of A100 GPUs with 64GB of VRAM, using 4 nodes with 4 GPUs each. Pre-training took approximately a month, while instruction tuning took about 10 hours. Unlike the original LCM implementation (LCM team et al., 2024), we extract

embeddings at runtime due to space limitations of the cluster where we perform the experiments.

5 Experiments

We perform inference by leveraging the same parameters used by LCM team et al. (2024). Specifically, we use 40 inference timesteps, 0.6 initial noise scale, 3.0 guidance scale, 0.7 guidance rescale, and 1.00045 epsilon scaling. Additionally, for pre-training evaluation, we limit generation to a single sentence, while for instruction-tuning evaluation, we limit generation to 16 sentences. Generated embeddings are always decoded into text using the original SONAR decoder. To assess the effectiveness of instruction tuning, we compare MIMIR with Qwen3 1.7B (Qwen Team, 2025), a modern LLM within the same parameter range. For Qwen3 1.7B, we use greedy decoding directly during inference. We do not evaluate the LCM model (LCM team et al., 2024) since the checkpoint was never publicly released. For pre-training, we evaluate on three large-scale multilingual datasets: 1) **C4**[6] (Raffel et al., 2020), a cleaned version of Common Crawl's web crawl corpus; 2) **MultiEURLEX** (Chalkidis et al., 2021), a dataset consisting of 65,000 European laws in 23 official European languages; 3) **Wiki40B** (Guo et al., 2020), a processed version of Wikipedia consisting of the full Wikipedia article after page processing that removes non-content sections and structured objects. For pre-training evaluation, we consider the **L2 distance** (L2), the **Round-trip L2 distance** (R-L2), the **Paraphrasing** (PAR) and the **Contrastive Accuracy** (CA) metrics (LCM team et al., 2024). L2 distance is the Euclidean distance between the predicted embedding and the ground truth embedding. Round-trip L2 distance is the Euclidean distance between the embedding of the decoded sentence (re-encoded in the SONAR embedding space) and the ground truth embedding. Paraphrasing is the maximum cosine similarity between the generated embedding and the context embeddings, normalized by the score of the ground truth embedding (essentially, it measures the degree to which the model relies on copying or rephrasing previous sentences w.r.t. the ground truth). Contrastive Accuracy is the ratio of times the generated embedding is closer (in terms of L2 distance) to the actual ground truth embedding than it is to other embeddings from different sentences in the same batch. We extract the first 1,000 instances with at least 9 sentences for each datasets and languages. Then, we perform inference for each sentence within each instance. That is, we perform inference by providing the model only the first sentence as context, then the first and second sentences as context, and so on.

For instruction tuning, we evaluate on two multilingual generative benchmarks: 1) XL-Sum (Hasan et al., 2021); 2) MLQA (Lewis et al., 2020). These datasets are described in Section 3, and their training set is included in our main instruction-tuning mixture. For instruction-tuning evaluation, we consider the **ROUGE-L** metric, following LCM team et al. (2024).

For pre-training, we perform evaluation on a subset of languages that are: 1) supported by both C4 and Wiki40B; 2) supported by SONAR. For instruction tuning, we perform evaluation only on the language subsets that are present in the training mixture.

5.1 Pre-Train Results

Dataset	L2	R-L2	PAR	CA
C4	0.2551	0.2411	2.4839	72.54%
MultiEURLEX	0.2063	0.1972	1.7796	56.65%
Wiki40B	0.2980	0.2793	2.3714	77.70%

Table 1: Results of pre-train evaluation

We report average pre-training evaluation results for C4, MultiEURLEX, and Wiki40B in Table 1. Per-language results for the C4 dataset on L2 and R-L2 are shown in Figure 3 while the results for Wiki40B and MultiEURLEX on L2 and R-L2, as well as the results for CA and PAR, are shown in Appendix E.

Overall, the results consistently show that R-L2 is lower w.r.t L2, implying that the decoder is capable of decoding embeddings into coherent sentences. From the average results, we find that Mimir performs best on MultiEURLEX, except for CA. This is expected since legal documents use highly constrained and repetitive vocabulary, hence the model can easily predict the exact continuation. However, this same homogeneity makes the CA evaluation unfair, because other sentences in a batch of legal texts are likely to be semantically similar to the ground truth. Furthermore, we find similar patterns for both C4 and Wiki40B, which exhibit higher PAR w.r.t. MultiEURLEX

[6] https://huggingface.co/datasets/allenai/c4

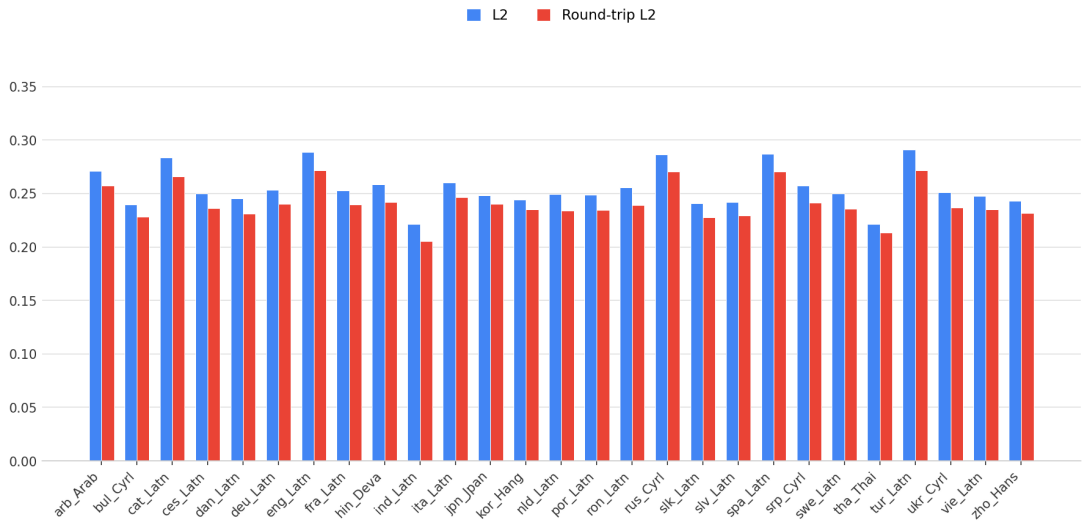

Figure 3: Bar chart showcasing pre-train evaluation results using L2 and Round-trip L2 for the C4 dataset

(since they are open-domain corpora). Interestingly, Mimir performs best on Wiki40B in terms of CA, but worse in terms of L2 and R-L2 w.r.t. C4. We attribute this to the rigorously cleaned structure of Wikipedia documents (for CA) and to the diverse vocabulary of Wikipedia text (for L2 and R-L2).

From the language split results, we find that Mimir performs worse in English than in other languages across all three datasets, particularly for L2 and R-L2. We attribute this to the nature of the pre-training data, given that we extracted all English data from Fineweb-edu, while we used Fineweb 2 for the other languages. Importantly, the results for both L2 and Round-trip L2 are comparable to those obtained by LCM team et al. (2024) on their pre-training evaluation benchmarks. This demonstrates that the model can understand context and generate embeddings that are natural continuations of the given context. Finally, manual inspection of the generated outputs further confirms that the model produces semantically consistent continuations across languages.

5.2 Instruction Tuning Results

We report average results in Table 2 while per-language results for MLQA and XL-Sum can be found in Figure 4 and in Figure 5, respectively. Configurations with English only in the train set are referred to as EN, while configurations without MLQA and XL-Sum in the train set are referred to as OOD.

Overall, Mimir performs better on XL-Sum, even when XL-Sum is not included in the train set. Remarkably, the model outperforms Qwen3 significantly on a subset of languages for XL-Sum (e.g., "ind_Latn"). We find that performance is lower on MLQA than on Qwen3, suggesting that the model is currently better suited for summarization than question-answering tasks. Finally, we find that the model trained on multilingual data performs best on MLQA, while models trained with MLQA and XL-Sum in the train set consistently outperform those trained without them. Still, models trained without MLQA and XL-Sum show a good degree of generalizability w.r.t. Qwen3 (e.g., Mimir performs better on "fra_Latn" w.r.t. Qwen3 on XL-Sum).

Model	EN	OOD	Dataset	Rouge-L
Mimir 1.6B	X	X	MLQA	27.31
Mimir 1.6B	X	X	XL-Sum	14.03
Mimir 1.6B	X	✓	MLQA	15.76
Mimir 1.6B	X	✓	XL-Sum	9.66
Mimir 1.6B	✓	X	MLQA	26.67
Mimir 1.6B	✓	X	XL-Sum	13.52
Mimir 1.6B	✓	✓	MLQA	13.77
Mimir 1.6B	✓	✓	XL-Sum	8.56
Qwen3 1.7B	-	-	MLQA	32.77
Qwen3 1.7B	-	-	XL-Sum	9.37

Table 2: Results for evaluation on MLQA and XL-Sum

After manually validating the outputs, we find that the model trained without MLQA and XL-Sum in the training set produced significantly worse-quality responses. Specifically, we found the following limitations: 1) task drift, where the model was unable to provide a response relevant to the

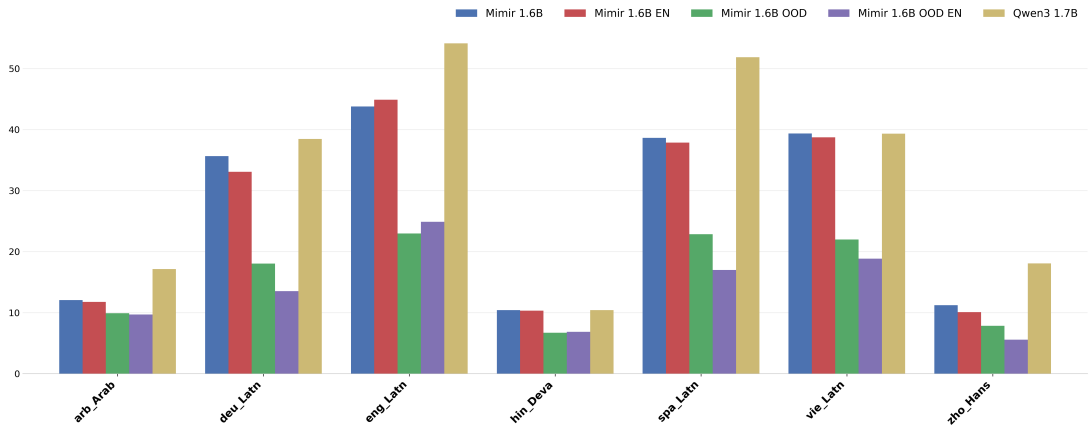

Figure 4: Bar chart showcasing instruction-tuning evaluation results using Rouge-L for the MLQA dataset

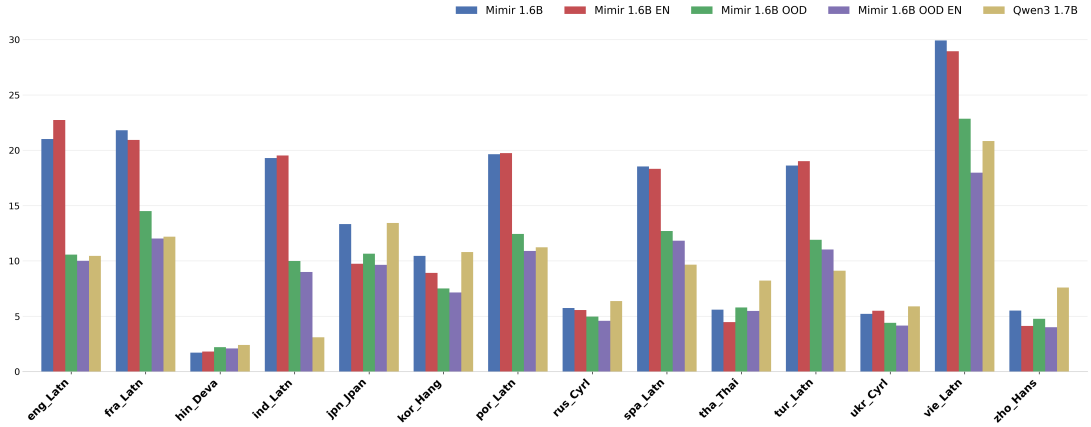

Figure 5: Bar chart showcasing instruction-tuning evaluation results using Rouge-L for the XL-Sum dataset

task (e.g., generating a new question instead of answering the input question in MLQA); 2) text repetition, where the model repeated the same terms within the same sentence (e.g., "*Walter Kasza and Walter Kasza*"). We attribute these limitations to two main factors. First of all, regarding model size: since the current model has 1.6B parameters and the next-concept prediction task is significantly more complex than next-token prediction, improved performance is expected for a model with a larger parameter count. Second, without MLQA and XL-Sum in the train set, the model was never directly provided with extractive question-answering and summarization data.

6 Conclusions and Future Works

In this work, we have introduced Mimir, the first Large Concept Model trained on large-scale multilingual data. We pre-train the model on a multilingual corpus consisting of 38,883,987,240 sentences and we perform instruction tuning on a multilingual dataset consisting of 66,816,428 sentences. Through extensive pre-training and instruction tuning, we investigated the capabilities and limitations of multilingual concept-level language modeling across four instruction-tuning mixtures. We show that Mimir performs optimally on long-context tasks (e.g., XL-Sum) and outperforms Qwen3 1.7B across most non-English languages. This result suggests that concept-level modeling is a promising research direction for multilingual semantic content generation. At the same time, our analysis revealed the model's limitations, mainly related to difficulties in task generalization, task drift, and repetitive generation patterns. These challenges indicate that concept-level language modeling remains significantly more demanding than traditional token-based language modeling, especially with smaller parameter counts. As future work, we are developing a 7B version of MIMIR and plan to extend both models to multimodal inputs. We additionally aim to extend both training and evaluation settings by including multiple-choice reasoning datasets and more challenging multilingual understanding tasks.

Limitations

Our work has the following limitations:

- **Embedding Space**: our work relies on SONAR (Duquenne et al., 2023) as embedding model following the work of LCM team et al. (2024). We do not propose a study on the applicability of other embedding models to train a Large Concept Model.

- **Evaluation Protocol**: currently, we only evaluated on MLQA and XL-Sum to consider a wide range of languages. However, Mimir should be evaluated on several additional benchmarks covering diverse skills, for example multi-turn conversations and math. We plan to further extend Mimir evaluation in the future.

- **Human Assessment**: evaluation of Mimir outputs with human evaluators over a subset of languages could have improved the strength of the conclusions and the evaluation soundness.

Ethical Considerations

In this work, the model was pre-trained using Fineweb-edu and Fineweb 2. While these datasets were subjected to rigorous quality and filtering by their original creators, we utilized them in their released state without introducing any secondary processing pipelines.

References

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.

Pierpaolo Basile, Lucia Siciliani, Elio Musacchio, and Giovanni Semeraro. 2025. Exploring the word sense disambiguation capabilities of large language models. *arXiv preprint arXiv:2503.08662*.

Milan Bhan, Yann Choho, Jean-Noël Vittaut, Nicolas Chesneau, Pierre Moreau, and Marie-Jeanne Lesot. 2025. Towards achieving concept completeness for textual concept bottleneck models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 2007–2024, Suzhou, China. Association for Computational Linguistics.

Ilias Chalkidis, Manos Fergadiotis, and Ion Androutsopoulos. 2021. Multieurlex – a multi-lingual and multi-label legal document classification dataset for zero-shot cross-lingual transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Paul-Ambroise Duquenne, Holger Schwenk, and Benoît Sagot. 2023. SONAR: sentence-level multimodal and language-agnostic representations. *CoRR*, abs/2308.11466.

Markus Frohmann, Igor Sterner, Ivan Vulić, Benjamin Minixhofer, and Markus Schedl. 2024. Segment any text: A universal approach for robust, efficient and adaptable sentence segmentation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11908–11941, Miami, Florida, USA. Association for Computational Linguistics.

Gregor Geigle, Florian Schneider, Carolin Holtermann, Chris Biemann, Radu Timofte, Anne Lauscher, and Goran Glavaš. 2025. Centurio: On drivers of multilingual ability of large vision-language model. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2831–2881, Vienna, Austria. Association for Computational Linguistics.

Mandy Guo, Zihang Dai, Denny Vrandečić, and Rami Al-Rfou. 2020. Wiki-40b: Multilingual language model dataset. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2440–2452.

Zhuowen Han, Xinwei Wu, Dan Shi, Renren Jin, and Deyi Xiong. 2025. Towards a unified paradigm of concept editing in large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 18445–18461, Suzhou, China. Association for Computational Linguistics.

Tahmid Hasan, Abhik Bhattacharjee, Md Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M Sohel Rahman, and Rifat Shahriyar. 2021. Xl-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4693–4703.

Hugging Face. 2024. Everyday conversations for llms. https://huggingface.co/datasets/HuggingFaceTB/everyday-conversations-llama3.1-2k.

Laya Iyer, Pranav Somani, Alice Guo, Dan Jurafsky, and Chen Shani. 2026. Beyond tokens: Concept-level training objectives for LLMs. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 457–474, Rabat, Morocco. Association for Computational Linguistics.

Mingyu Jin, Qinkai Yu, Jingyuan Huang, Qingcheng Zeng, Zhenting Wang, Wenyue Hua, Haiyan Zhao,

Kai Mei, Yanda Meng, Kaize Ding, and 1 others. 2025. Exploring concept depth: How large language models acquire knowledge and concept at different layers? In *Proceedings of the 31st international conference on computational linguistics*, pages 558–573.

Felicia Körner, Max Müller-Eberstein, Anna Korhonen, and Barbara Plank. 2026. When meanings meet: Investigating the emergence and quality of shared concept spaces during multilingual language model training. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3149–3169, Rabat, Morocco. Association for Computational Linguistics.

Wen Lai, Mohsen Mesgar, and Alexander Fraser. 2024. LLMs beyond English: Scaling the multilingual capability of LLMs with cross-lingual feedback. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8186–8213, Bangkok, Thailand. Association for Computational Linguistics.

LCM team, Loïc Barrault, Paul-Ambroise Duquenne, Maha Elbayad, Artyom Kozhevnikov, Belen Alastruey, Pierre Andrews, Mariano Coria, Guillaume Couairon, Marta R. Costa-jussà, David Dale, Hady Elsahar, Kevin Heffernan, João Maria Janeiro, Tuan Tran, Christophe Ropers, Eduardo Sánchez, Robin San Roman, Alexandre Mourachko, and 2 others. 2024. Large concept models: Language modeling in a sentence representation space. *CoRR*, abs/2412.08821.

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. Mlqa: Evaluating cross-lingual extractive question answering. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 7315–7330.

Haonan Li, Fajri Koto, Minghao Wu, Alham Fikri Aji, and Timothy Baldwin. 2023. Bactrian-x : A multilingual replicable instruction-following model with low-rank adaptation. *Preprint*, arXiv:2305.15011.

Anton Lozhkov, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. 2024. Fineweb-edu: the finest collection of educational content.

Pedro Henrique Martins, Patrick Fernandes, João Alves, Nuno M Guerreiro, Ricardo Rei, Duarte M Alves, José Pombal, Amin Farajian, Manuel Faysse, Mateusz Klimaszewski, and 1 others. 2025. Eurollm: Multilingual language models for europe. *Procedia Computer Science*, 255:53–62.

Benjamin Minixhofer, Jonas Pfeiffer, and Ivan Vulić. 2023. Where's the point? self-supervised multilingual punctuation-agnostic sentence segmentation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7215–7235, Toronto, Canada. Association for Computational Linguistics.

Jean De Dieu Nyandwi, Yueqi Song, Simran Khanuja, and Graham Neubig. 2025. Grounding multilingual multimodal LLMs with cultural knowledge. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 24187–24231, Suzhou, China. Association for Computational Linguistics.

Tommaso Pasini, Alessandro Raganato, and Roberto Navigli. 2021. Xl-wsd: An extra-large and cross-lingual evaluation framework for word sense disambiguation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13648–13656.

Guilherme Penedo, Hynek Kydlíček, Vinko Sabolčec, Bettina Messmer, Negar Foroutan, Amir Hossein Kargaran, Colin Raffel, Martin Jaggi, Leandro Von Werra, and Thomas Wolf. 2025. Fineweb2: One pipeline to scale them all – adapting pre-training data processing to every language. *Preprint*, arXiv:2506.20920.

Hao Peng, Xiaozhi Wang, Shengding Hu, Hailong Jin, Lei Hou, Juanzi Li, Zhiyuan Liu, and Qun Liu. 2022. Copen: Probing conceptual knowledge in pre-trained language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5015–5035.

Libo Qin, Qiguang Chen, Xiachong Feng, Yang Wu, Yongheng Zhang, Yinghui Li, Min Li, Wanxiang Che, and Philip S Yu. 2026. Large language models meet nlp: A survey. *Frontiers of Computer Science*, 20(11):2011361.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. 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.

Eshaan Tanwar, Subhabrata Dutta, Manish Borthakur, and Tanmoy Chakraborty. 2023. Multilingual LLMs

are better cross-lingual in-context learners with alignment. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6292–6307, Toronto, Canada. Association for Computational Linguistics.

Shubham Toshniwal, Wei Du, Ivan Moshkov, Branislav Kisacanin, Alexan Ayrapetyan, and Igor Gitman. 2024. Openmathinstruct-2: Accelerating ai for math with massive open-source instruction data. *arXiv preprint arXiv:2410.01560*.

Jiahao Ying, Wei Tang, Yiran Zhao, Yixin Cao, Yu Rong, and Wenxuan Zhang. 2025. Disentangling language and culture for evaluating multilingual large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22230–22251, Vienna, Austria. Association for Computational Linguistics.

A Data Details

Complete statistics are reported in Table 3 for the pre-training dataset and in Table 4 for the instruction-tuning dataset.

B Formatting

We report the formatting used for instruction tuning in Table 5.

C Methodology

We showcase a diagram representing Mimir and the methodological differences w.r.t. the Two-Tower LCM (LCM team et al., 2024) in Figure 6.

D Data Generation

We report complete prompts used for the multilingual multi-turn dataset in Table 6 and for the multilingual multi-turn math dataset in Table 7.

E Additional Evaluation Results

We report the results for the pilot study of cosine similarity for English to target language SONAR embeddings in Table 8. We report pre-training evaluation results for C4 and MultiEURLEX split by language for L2 and R-L2 in Figure 3 and in Figure 7, respectively. We report pre-training evaluation results for C4, MultiEURLEX and Wiki40B split by language for PAR in Figure 9, in Figure 10 and in Figure 11, respectively. We report pre-training evaluation results for C4, MultiEURLEX and Wiki40B split by language for CA in Figure 12, in Figure 13 and in Figure 14, respectively.

Interestingly, we report that Mimir showcased repetitive outputs for "ces_Latn" on C4 as demonstrated by the spike for that language in Figure 9.

To evaluate Mimir's ability to understand context, we test the model on the Word Sense Disambiguation (WSD) task. WSD is particularly relevant to Concept Modeling, as it requires precise contextual understanding to distinguish among different senses of the same word.

We perform the evaluation leveraging the XL-WSD (Pasini et al., 2021) benchmark extended for LLM evaluation by Basile et al. (2025). Specifically, we consider the generative split of the dataset without translations, comprising 6,757 instances in English, 1,673 in Italian, 1,248 in Spanish, 539 in French, and 263 in German. As baselines, we report the zero-shot inference results for the Llama 3.1 8B-Instruct and Llama 3.1 405B-Instruct models, as reported by Basile et al. (2025). Following Basile et al. (2025), we use ROUGE-L as the primary evaluation metric. Results show that Mimir underperforms both Llama 3.1 8B and 405B on this task. Nevertheless, it demonstrates remarkable multilingual performance, achieving non-trivial performance even on lower-resource evaluation languages such as German (11.55 Rouge-L). These findings suggest that while concept-based modeling alone is insufficient to match state-of-the-art token-based LLMs on WSD, multilingual concept representations can capture meaningful semantic information that could be further improved through targeted training for sense understanding.

F Licenses

We will release pre-train and instruction-tuning data under a license that complies with the original datasets present in the training mixture. We will release the Mimir model and the codebase under MIT license.

G Use of AI Assistants

During the preparation of this paper, we used Gemini 3.1 Pro to assist with code optimization of our scripts. Additionally, we used Grammarly to check grammar and improve the readability of the paper. We have reviewed all AI-optimized code and generated text, and we take full responsibility for the accuracy, originality, and final content of this work.

Language	Instances	Sentences
arb_Arab	6,920,272	250,934,094
bel_Cyrl	234,579	9,458,247
ben_Beng	1,695,605	53,614,461
bos_Latn	2,371,974	68,947,657
bul_Cyrl	2,902,512	114,991,120
cat_Latn	1,913,415	49,630,459
ces_Latn	7,377,000	293,734,712
cym_Latn	92,886	2,985,904
dan_Latn	5,068,336	205,778,366
deu_Latn	55,378,296	2,290,435,652
eng_Latn	339,347,842	17,363,106,068
fra_Latn	40,203,392	1,528,389,036
heb_Hebr	1,618,116	72,240,173
hin_Deva	2,406,778	74,217,167
hrv_Latn	691,812	54,837,623
ind_Latn	11,192,416	479,630,974
ita_Latn	26,684,472	919,861,585
jpn_Jpan	44,678,584	2,300,431,291
kan_Knda	266,972	8,544,099
kor_Hang	6,797,096	301,721,036
lvs_Latn	896,648	57,697,945
mal_Mlym	370,986	13,204,579
mar_Deva	436,884	14,323,222
mkd_Cyrl	463,482	11,311,454
nld_Latn	16,447,336	566,173,272
npi_Deva	545,802	12,861,045
ory_Orya	144,953	2,969,207
pol_Latn	16,968,272	657,652,847
por_Latn	22,302,304	757,671,212
ron_Latn	6,510,056	240,522,807
rus_Cyrl	78,058,128	4,847,502,965
slk_Latn	3,348,784	134,859,963
slv_Latn	1,346,495	70,469,351
spa_Latn	49,273,160	1,602,905,444
srp_Cyrl	462,948	20,646,830
swe_Latn	6,642,000	241,586,555
swh_Latn	134,693	4,211,019
tam_Taml	617,340	23,817,904
tel_Telu	219,340	11,473,008
tha_Thai	4,008,200	235,314,468
tur_Latn	10,621,912	390,860,061
ukr_Cyrl	5,929,224	227,975,182
urd_Arab	537,024	15,927,173
vie_Latn	6,818,296	316,157,489
zho_Hans	51,449,701	1,665,507,651
zho_Hant	9,049,706	296,894,863

Table 3: Complete list of all languages included in pre-training and their cardinality (both number of instances and number of sentences)

Language	Aya	Bactrian-X	Open Assistant 2	Ours	MLQA	XL-Sum
arb_Arab	66,924	794,330	120	16,210	968,702	0
ben_Beng	21,709	902,598	0	3,922	0	0
ces_Latn	0	798,389	15	20,178	0	0
deu_Latn	2,451	843,549	20,244	167,369	1,022,780	0
eng_Latn	46,309	853,400	236,677	1,510,862	1,022,762	9,218,354
fra_Latn	13,504	825,812	10,298	112,023	0	264,330
heb_Hebr	0	769,294	29	4,692	0	0
hin_Deva	12,444	860,428	0	6,964	1,041,678	2,569,314
hrv_Latn	0	831,293	0	2,306	0	0
ind_Latn	11,177	880,484	31	30,313	0	1,182,739
ita_Latn	6,176	796,651	2,643	76,201	0	0
jpn_Jpan	61,121	845,659	1,787	101,772	0	251,813
kor_Hang	3,866	864,464	25	20,903	0	165,207
lvs_Latn	0	816,503	0	1,913	0	0
mal_Mlym	17,462	893,154	0	2,495	0	0
mar_Deva	34,961	872,085	0	1,245	0	0
mkd_Cyrl	0	793,209	0	2,522	0	0
nld_Latn	20,139	835,846	144	46,460	0	0
npi_Deva	36,699	898,382	0	2,528	0	0
pol_Latn	13,080	808,567	900	50,124	0	0
por_Latn	61,850	808,277	12,656	61,529	0	2,134,442
ron_Latn	0	810,049	0	19,301	0	0
rus_Cyrl	4,034	785,054	48,847	224,329	0	2,576,458
slv_Latn	0	834,416	0	3,258	0	0
spa_Latn	32,801	824,865	106,204	138,844	1,204,589	1,863,457
swe_Latn	610,717	801,663	0	17,772	0	0
swh_Latn	3,045	860,170	0	2,382	0	0
tam_Taml	293,226	859,460	0	1,779	0	0
tel_Telu	160,897	799,914	0	1,714	0	0
tha_Thai	6,988	971,516	4,150	16,106	0	392,155
tur_Latn	33,122	850,842	99	28,978	0	893,077
ukr_Cyrl	5,655	784,703	3,061	18,235	0	1,468,201
urd_Arab	24,810	849,140	0	2,357	0	0
vie_Latn	171,693	838,376	365	17,882	1,032,551	1,503,432
zho_Hans	121,897	719,010	20,022	142,966	1,204,423	1,108,904

Table 4: Complete list of all languages included in instruction tuning, the source dataset and their cardinality (number of sentences)

CONTEXT
{USER_TURN_SENTENCE}
{PROMPT_SENTENCES}
{ASSISTANT_TURN_SENTENCE}
COMPLETION
{RESPONSE_SENTENCES}
End of text.

Table 5: Prompt used during instruction tuning. {USER_TURN_SENTENCE} is the "User turn." sentence translated to the language of the conversation. {ASSISTANT_TURN_SENTENCE} is the "Assistant turn." sentence translated to the language of the conversation. {PROMPT_SENTENCES} is the list of sentences obtained from the prompt using a sentence splitting model. {RESPONSE_SENTENCES} is the list of sentences obtained from the assistant response using a sentence splitting model.

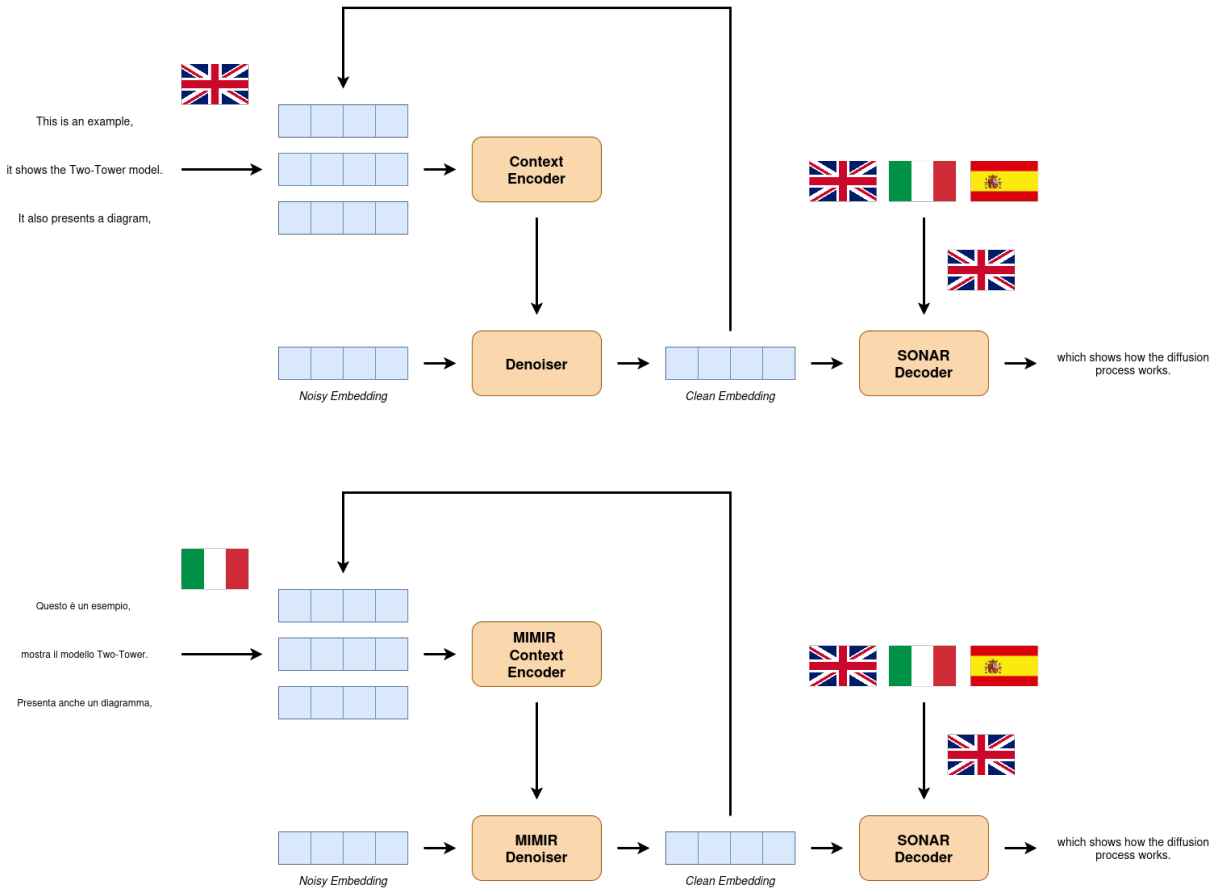

Figure 6: Overview of the diffusion process in the Two-Tower LCM architecture. The context encoder encodes SONAR sentence embeddings for the English language, which are then processed through cross-attention by the denoiser. The denoiser generates a clean embedding by attending the context. This clean embedding can then be used by the SONAR decoder to reconstruct text in any target language it supports. The cleaned embedding is also concatenated to the input embeddings to continue the generation of the next sentence. While the original LCM model only allows encoding of English sentences, Mimir allows for encoding of sentences in any language, while still providing generation in any language supported by the decoder.

You are an expert synthetic data generator. Your task is to generate a realistic, multi-turn conversation between a USER and an AI ASSISTANT based on a specific topic.

SETTINGS:
- **Topic:** {TOPIC}
- **Target Language:** {TGT_LANG}
- **Conversation Length:** Approximately {NUM_TURNS} turns (total messages between User and Assistant).

INSTRUCTIONS FOR "USER" GENERATION:
1. **Language:** Fluent, natural {TGT_LANG}.
2. **Content:** Start by asking about the Topic. Follow-up questions should dig deeper, asking for opinions, comparisons, or specific details.
3. **Cultural Relevance:** The User's perspective, idioms, and context must be culturally relevant to {TGT_LANG}. (e.g., if the language is Japanese and the topic is 'Lunch', discuss Bento or Ramen, not PB&J sandwiches).

USER CONSTRAINTS:
- **Length Constraints:** Approximately 20% of the time, the User must explicitly constrain the Assistant's output length (e.g., 'Answer with a single word', 'Give me a bulleted list', 'Keep it under 10 words').
- **Unconventional Formatting:** Occasionally, the User must request the response in an unusual format or persona (e.g., 'Return the answer as a valid JSON dictionary', 'Answer in pirate speech').

INSTRUCTIONS FOR "ASSISTANT" GENERATION:
1. **Language:** Fluent, natural {TGT_LANG}.
2. **Behavior:** Helpful, accurate, and culturally aware, unless a specific interaction scenario (below) requires otherwise.
3. **Responsiveness:** If the User sets a length or format constraint, the Assistant MUST strictly obey it.

REQUIRED INTERACTION SCENARIOS:
Select AT LEAST ONE of the following specific scenarios to naturally weave into the conversation:

NEGATIVE/ADVERSARIAL SCENARIOS:
1. **The 'Hallucination' & Correction:** The ASSISTANT provides a factually incorrect answer. The USER corrects it. The ASSISTANT apologizes and provides the correct answer.
2. **The Valid Refusal (Impossible Request):** The USER asks an impossible or out-of-bounds question. The ASSISTANT politely declines to answer.
3. **The False Correction (Assistant stands its ground):** The ASSISTANT provides a correct answer. The USER incorrectly claims it is wrong (and may provide a wrong alternative). The ASSISTANT politely but firmly asserts its correctness and explains why.

COMPLEX INTERACTION SCENARIOS:
1. **The Ambiguous Query:** The USER asks a vague follow-up using unclear pronouns (e.g., 'What about that other one?'). The ASSISTANT must politely ask for clarification before answering.
2. **The Compound Question:** The USER asks at least 3 distinct questions in a single message. The ASSISTANT must systematically answer all parts.
3. **The Pivot:** The USER abruptly changes the sub-topic mid-conversation. The ASSISTANT follows the pivot smoothly.
4. **The Goalpost Move:** The USER asks the ASSISTANT to rewrite its previous answer with a completely new constraint (e.g., 'Now make it rhyme' or 'Explain it like I am 5').

OUTPUT FORMAT:
Return strictly a valid JSON list of dictionaries. Do not include markdown formatting (like "'`json`).
Format:
[
{"turn": 1, "role": "user", "content": "..."},
{"turn": 1, "role": "assistant", "content": "..."},
...
]

Table 6: Prompt used for synthetic generation of multilingual multi-turn conversational data. {TOPIC} is a placeholder for the complete topic sampled from the Everyday Conversations dataset. {TGT_LANG} is a placeholder for the target language and script (e.g. "fra_Latn"). {NUM_TURNS} is a placeholder for a random number of turns selected randomly in [4, 6, 8, 10, 12]. Text in red is randomly removed from the prompt.

You are an expert synthetic data generator. Your task is to generate a realistic, multi-turn conversation between a USER and an AI ASSISTANT focused on solving and discussing a specific math problem.

REFERENCE MATERIAL (ENGLISH):
- **Math Problem:** {PROBLEM}
- **Reference Solution:** {SOLUTION}

SETTINGS:
- **Target Language:** {TGT_LANG}
- **Conversation Length:** Approximately {NUM_TURNS} turns (total messages between User and Assistant).

INSTRUCTIONS FOR "USER" GENERATION:
1. **Language & Natural Framing:** Fluent, natural {TGT_LANG}. Do NOT directly or stiffly translate the English reference problem. Internalize the problem, and have the User ask it naturally in {TGT_LANG} as if they just encountered it in their homework or daily life.
2. **Progression:** Start by presenting the math problem. Follow-up questions should dig deeper into the methodology, ask for alternative ways to solve it, or introduce the complex/adversarial scenarios below.

USER CONSTRAINTS:
- **Direct Answer Constraint:** Approximately 20% of the time, the User's first message must explicitly ask for the final answer ONLY, without any reasoning or step-by-step breakdown (e.g., 'Just give me the final number', 'Answer directly without explanation').
- **Unconventional Formatting:** Occasionally, the User must request the math steps in a specific format (e.g., 'Return the steps as a JSON list', 'Explain the logic using pirate speech', 'Put every mathematical operation in a separate bullet point').

INSTRUCTIONS FOR "ASSISTANT" GENERATION:
1. **Language:** Fluent, natural {TGT_LANG}. Ensure mathematical terms are correctly translated.
2. **Accuracy:** The math must be flawlessly executed and align with the logic of the Reference Solution, unless a negative scenario explicitly requires an error.
3. **Responsiveness:** If the User asks for a direct answer without reasoning in their first turn, the Assistant MUST output exactly the final number/solution and nothing else. The Assistant must strictly obey all other length or formatting constraints requested by the User.

REQUIRED INTERACTION SCENARIOS:
Select AT LEAST ONE of the following specific scenarios to naturally weave into the conversation:

NEGATIVE/ADVERSARIAL MATH SCENARIOS:
1. **The Calculation Error & Correction:** The ASSISTANT makes a subtle calculation error or uses the wrong formula in one of the steps. The USER catches the math error and corrects it. The ASSISTANT apologizes, recalculates, and provides the correct answer.
2. **The Missing Information (Valid Refusal):** The USER asks a follow-up math question that lacks the necessary variables to be solved. The ASSISTANT politely explains what information is missing and why the calculation cannot be performed.
3. **The False Correction (Assistant stands its ground):** The ASSISTANT solves a step correctly. The USER incorrectly claims it is wrong based on a common math misconception (e.g., messing up the order of operations). The ASSISTANT politely but firmly asserts its correctness and explains the mathematical rule.

COMPLEX MATH INTERACTION SCENARIOS:
1. **The Ambiguous Step Query:** The USER asks a vague follow-up about a specific number (e.g., 'Where did that 3 come from?' or 'Why did you multiply those two?'). The ASSISTANT must clarify the specific step in the reference solution.
2. **The Method Request:** The USER asks if there is an alternative mathematical way or formula to solve the exact same problem. The ASSISTANT provides a valid alternative method that yields the same result.
3. **The Variable Change (Goalpost Move):** Mid-conversation, the USER changes the numbers in the original problem (e.g., 'What if there were 10 people instead of 5?'). The ASSISTANT recalculates everything based on the new parameters.
4. **The Concept Pivot:** The USER abruptly asks for the definition of a mathematical concept related to the problem (e.g., 'By the way, what exactly is a factorial?'). The ASSISTANT explains it clearly, then ties it back to the current problem if applicable.

OUTPUT FORMAT:
Return strictly a valid JSON list of dictionaries. Do not include markdown formatting (like "'`json`).
Format:
[
{"turn": 1, "role": "user", "content": "..."},
{"turn": 1, "role": "assistant", "content": "..."},
...
]

Table 7: Prompt used for synthetic generation of multilingual multi-turn math problems. {PROBLEM} and {SOLUTION} are placeholders for the reference problem and solution from the OpenMathInstruct-2 dataset. {TGT_LANG} is a placeholder for the target language and script (e.g. "fra_Latn"). {NUM_TURNS} is a placeholder for a random number of turns selected randomly in [4, 6, 8, 10, 12]. Text in red is randomly removed from the prompt.

Target Language	Cosine Similarity
arb_Arab	0.8208
bul_Cyrl	0.8447
cat_Latn	0.8213
ces_Latn	0.8071
dan_Latn	0.8643
deu_Latn	0.7642
ell_Grek	0.8701
est_Latn	0.8359
fin_Latn	0.7954
fra_Latn	0.8652
glg_Latn	0.8447
heb_Hebr	0.8169
hin_Deva	0.7617
hrv_Latn	0.8340
hun_Latn	0.7896
ind_Latn	0.8242
ita_Latn	0.8618
jpn_Jpan	0.6030
kat_Geor	0.5483
kor_Hang	0.5586
lit_Latn	0.8081
mar_Deva	0.4885
mkd_Cyrl	0.8589
nld_Latn	0.8657
pes_Arab	0.6865
pol_Latn	0.7925
ron_Latn	0.8599
rus_Cyrl	0.7905
slk_Latn	0.8394
slv_Latn	0.8481
spa_Latn	0.8628
srp_Cyrl	0.7856
swe_Latn	0.8125
tur_Latn	0.7510
ukr_Cyrl	0.7827
vie_Latn	0.7896
zho_Hans	0.6699

Table 8: Average cosine similarity for eng_Latn to target language for SONAR embeddings

Language	Model	Rouge-L
English	Mimir 1.6B	10.36
	Llama3.1 8B-Instruct	22.60
	Llama3.1 405B-Instruct	23.93
French	Mimir 1.6B	13.23
	Llama3.1 8B-Instruct	19.01
	Llama3.1 405B-Instruct	17.51
German	Mimir 1.6B	11.55
	Llama3.1 8B-Instruct	15.86
	Llama3.1 405B-Instruct	13.43
Italian	Mimir 1.6B	8.25
	Llama3.1 8B-Instruct	13.63
	Llama3.1 405B-Instruct	15.24
Spanish	Mimir 1.6B	13.84
	Llama3.1 8B-Instruct	18.11
	Llama3.1 405B-Instruct	19.15

Table 9: Results for evaluation on XL-WSD

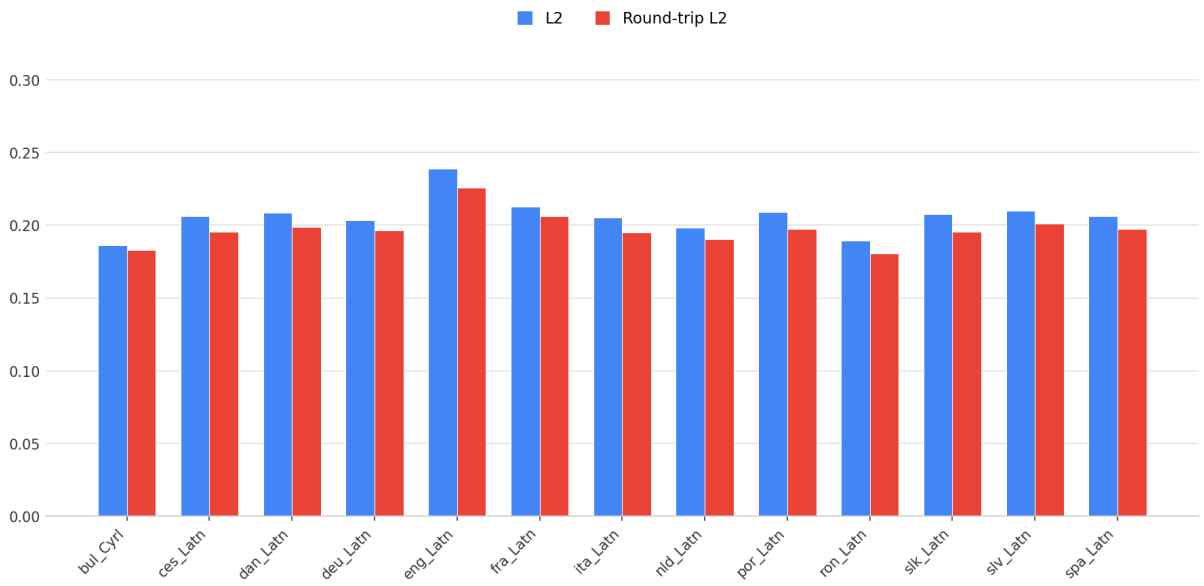

Figure 7: Bar chart showcasing pre-train evaluation results using L2 and Round-trip L2 for the MultiEURLEX dataset

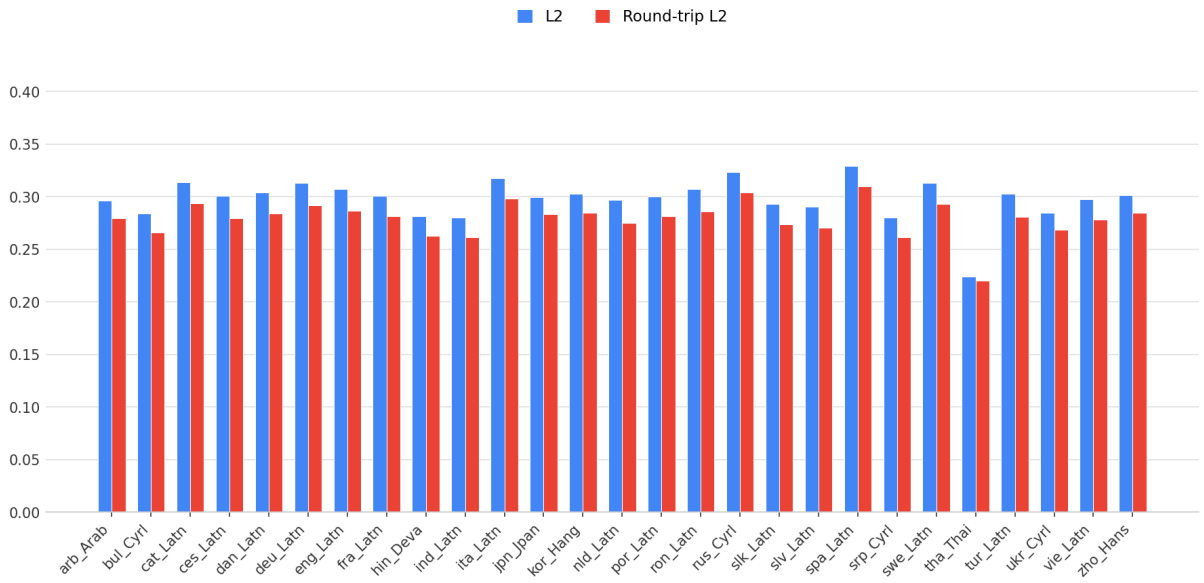

Figure 8: Bar chart showcasing pre-train evaluation results using L2 and Round-trip L2 for the Wiki40B dataset

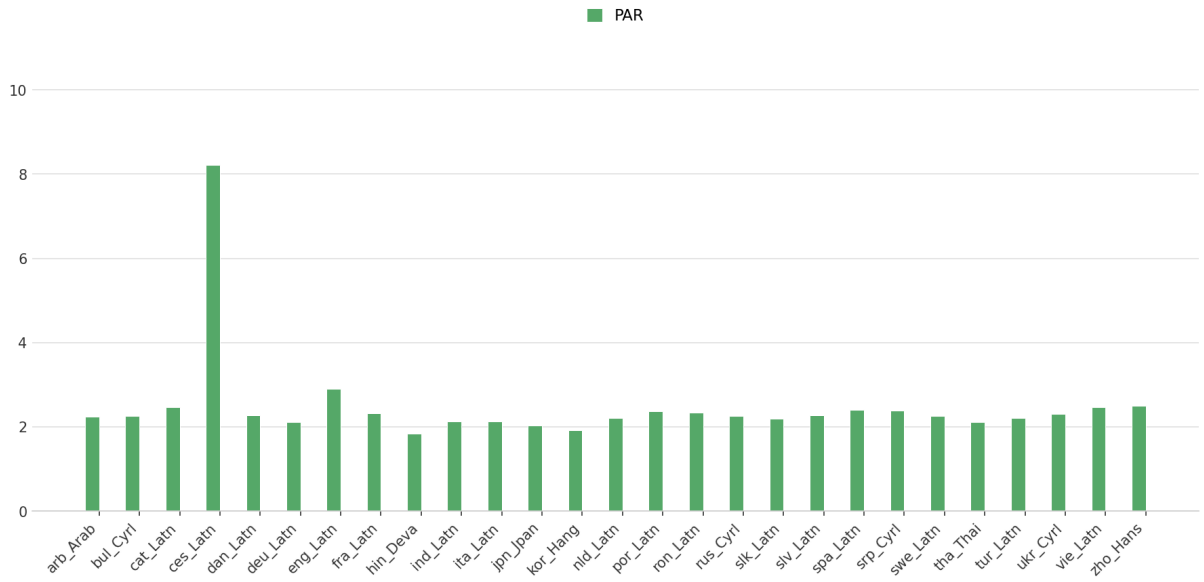

Figure 9: Bar chart showcasing pre-train evaluation results using PAR for the C4 dataset

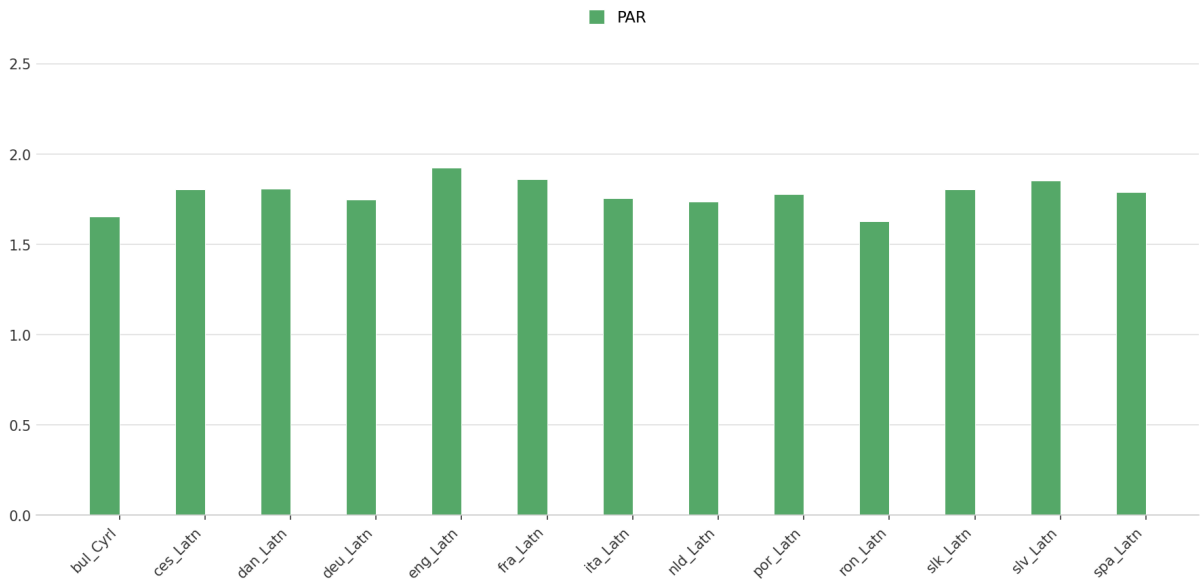

Figure 10: Bar chart showcasing pre-train evaluation results using PAR for the MultiEURLEX dataset

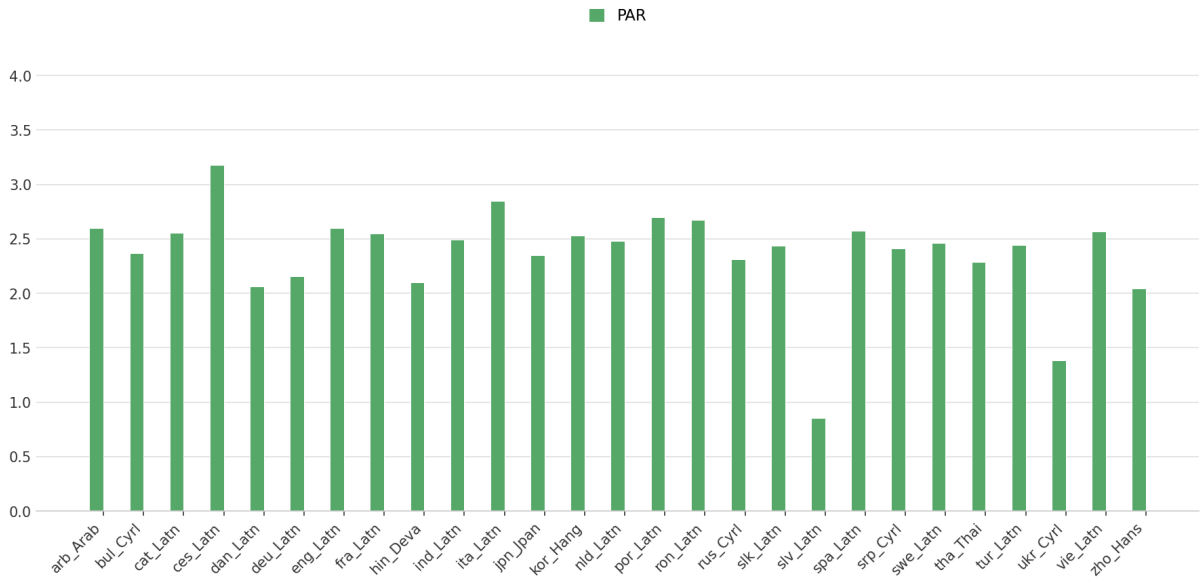

Figure 11: Bar chart showcasing pre-train evaluation results using PAR for the Wiki40B dataset

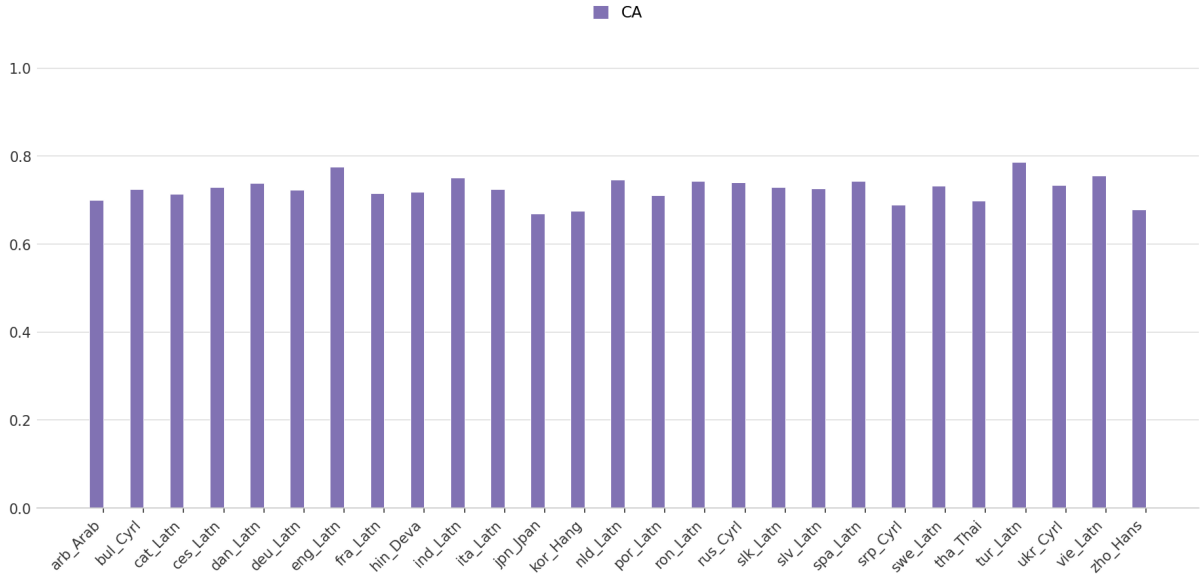

Figure 12: Bar chart showcasing pre-train evaluation results using CA for the C4 dataset

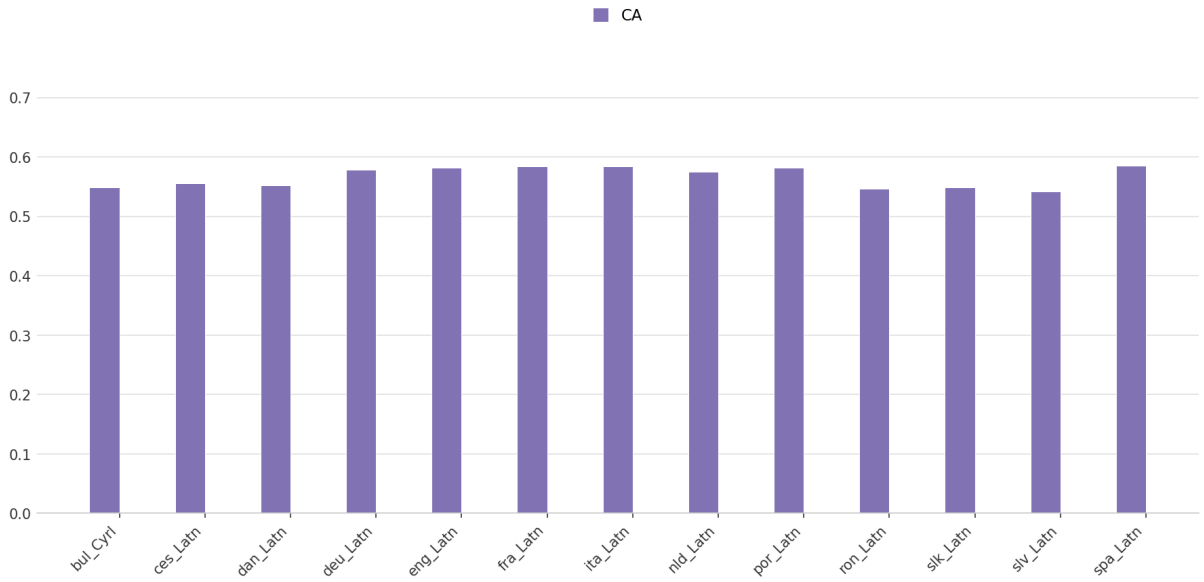

Figure 13: Bar chart showcasing pre-train evaluation results using CA for the MultiEURLEX dataset

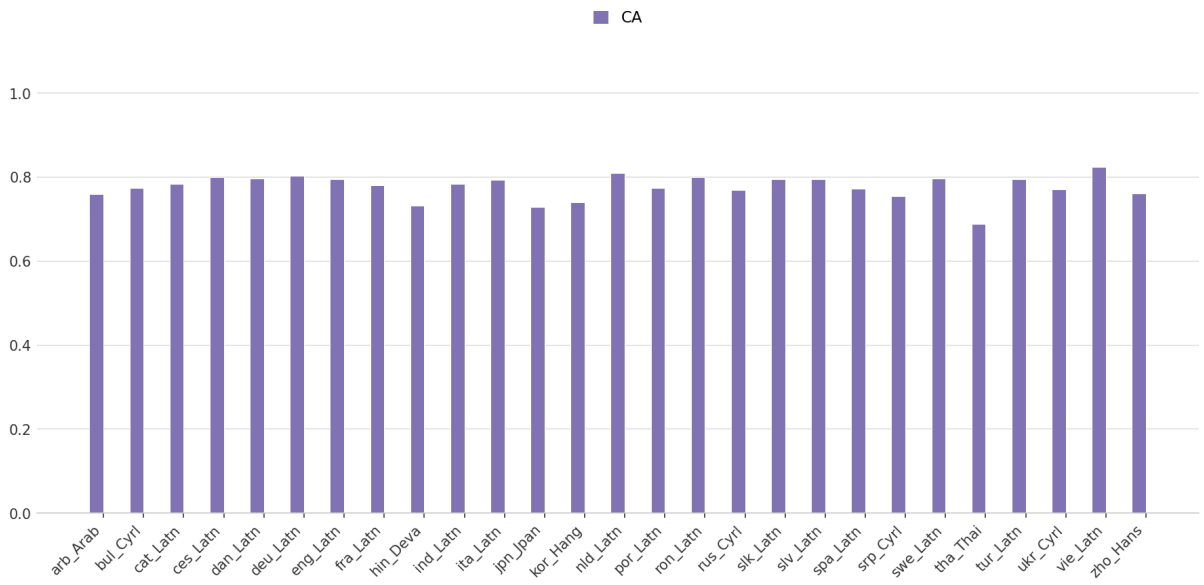

Figure 14: Bar chart showcasing pre-train evaluation results using CA for the Wiki40B dataset