Efficient Fine-Tuning of TinyLlama for Character-Specific Dialogue Generation Using LoRA

Erwin Alberto Lopez Hurtado
Deep Learning Project
Universidad Panamericana

***Index Terms*—Conversational AI, Parameter-Efficient Fine-Tuning, LoRA, Quantization, TinyLlama**

I. Introduction

Although chatbots have evolved into versatile and powerful tools, a recurring question remains: is it possible for a large language model (LLM) to develop a way of thinking? This question, often raised in the field of artificial intelligence, rarely finds a definitive answer. Most current research focuses on making LLMs more general and effective in a wide range of tasks, emphasizing scalability, versatility, and performance across diverse domains. However, this project takes a different approach.

Instead of training on datasets composed purely of factual content or generic human dialogues, we propose using a dataset based on a single personality interacting across multiple environments. The goal is to explore whether it is possible to fine-tune an LLM to imitate that specific personality—essentially, to build a chatbot capable of "thinking" and responding as if it were that person.

This idea is not just a technical curiosity but a psychological one. The ability to replicate a person's behavior opens the door to a deeper understanding of individual cognition and identity. For example, consider the case of a criminal: understanding how such a mind operates remains a significant challenge using traditional methods. But if a chatbot were trained on real conversations with such a person, it could offer a novel form of behavioral modeling, enabling deeper analysis and interaction. Moreover, this approach has emotional and social applications, such as offering conversations with a digital representation of a deceased loved one. This could create new opportunities for grieving, remembrance, or emotional support.

To carry out this exploration, we fine-tune the base model TinyLlama-1.1B using the following strategies:

- **4-bit NormalFloat (NF4) quantization** to reduce memory usage and enable training on consumer-grade hardware.
- **Low-Rank Adaptation (LoRA)** to achieve parameter-efficient fine-tuning without modifying the full model.
- **Context-preserving dialogue formatting** to maintain coherent and consistent personality traits across different prompts.
- **Optimized training pipelines** designed to work effectively within limited computational budgets.

II. State of the Art

Previous work in chatbot development has attempted to classify conversational agents by their purpose. In [1], Mnasri proposes a taxonomy of chatbots, including task-specific, social, and generalized types. This classification is useful in understanding how chatbots can be developed depending on their intended use. Our chatbot fits into the task-specific category, as it is trained to emulate a single human personality across different conversational situations. Unlike general-purpose models that aim for broad applicability, this project focuses on depth—capturing the nuances of one individual rather than the averages of many.

To achieve this, we follow methodologies inspired by general-purpose chatbot design, as discussed by Aggarwal et al. in [2]. Their work highlights the architecture and methodology for creating multipurpose chatbots using transformer-based language models. Starting from a general model and adapting it to a specific domain through fine-tuning is a common and effective strategy, especially when building more specialized agents.

This approach is further supported by the findings in [3], where Dodge et al. explore the role of fine-tuning on pretrained language models. They emphasize that fine-tuning offers a practical solution for customizing models without having to retrain them from scratch—a significant advantage when working under computational constraints. It also allows the tuning process to benefit from the extensive pretraining already present in foundational models.

We use TinyLlama-1.1B [4] as our base model. While more powerful LLMs such as GPT-4 or LLaMA-3 exist, their computational demands make them unsuitable for our use case. TinyLlama, introduced by Zhang et al., offers a compact yet effective alternative that is open-source, lightweight, and designed for ease of deployment on resource-limited systems.

To optimize this process further, we incorporate *Low-Rank Adaptation* (LoRA) [5], [6], a technique introduced to enable parameter-efficient fine-tuning of large language models. Traditional fine-tuning requires updating all of a model's weights, which becomes computationally expensive and memory-intensive for large-scale models. LoRA addresses

this by keeping the original model weights frozen and introducing small trainable matrices—referred to as *low-rank adapters*—within specific layers of the model, typically within attention or feed-forward blocks. During training, only these adapters are updated, which significantly reduces both memory usage and the number of trainable parameters.

Formally, LoRA assumes that the required weight update for a matrix $W \in \mathbb{R}^{d\times k}$ can be approximated as the product of two smaller matrices, $\Delta W = AB$, where $A \in \mathbb{R}^{d\times r}$ and $B \in \mathbb{R}^{r\times k}$ with $r \ll \min(d, k)$. These matrices are randomly initialized and trained, while W remains unchanged. This low-rank decomposition enables modular and scalable adaptation, making LoRA highly effective for fine-tuning large models on downstream tasks.

The original work by Hu et al. [6] introduced LoRA as a breakthrough for adapting large models without the need for full model retraining. More recent reviews, such as the survey by Huan and Shun [5], underscore LoRA's growing relevance in contemporary NLP pipelines, particularly for cases involving multiple domain- or task-specific adaptations.

Additionally, we consider *SAFE-LoRA* [7], a recent extension aimed at mitigating safety risks associated with fine-tuning. Hsu et al. show that even parameter-efficient methods like LoRA can unintentionally introduce harmful behaviors, such as biased outputs or unsafe completions. SAFE-LoRA adjusts the adapter training process by incorporating safety constraints and evaluations, ensuring that the resulting model maintains both task accuracy and behavioral safety. This is particularly important in scenarios where the objective is to replicate human-like personalities without reinforcing undesirable traits or introducing bias.

Attempts to emulate personality traits in language models are not new. In [8], Petrov et al. conducted experiments in which they fine-tuned GPT-like models using psychometric data, such as responses to standardized personality tests (e.g., the Big Five inventory). Their goal was to induce specific personality profiles in the generated text by associating psychometric scores with linguistic outputs. While their approach demonstrated some success in aligning outputs with target traits, their results also suggest that formal psychological profiles alone may not be the correct way of achieving this because of the way LLMS works.

These limitations highlight the need for more organic, unstructured, and context-rich training data. Unlike structured personality assessments, real conversational interactions naturally encode dynamic elements of personality, such as tone, empathy, humor, and adaptability to different social cues. This motivates our approach of training on authentic dialogue samples, where personality is expressed implicitly through diverse communicative acts rather than explicitly through test responses. By doing so, we aim to achieve a more fluid and human-like simulation of personality in dialogue generation tasks.

For evaluating the fidelity of the model's responses, we apply semantic similarity metrics using the all-MiniLM-L6-v2 model [10], a compact transformer model optimized for sentence embedding tasks. Yin and Zhang demonstrate that this model performs well in capturing semantic equivalence, even when sentence structures differ. This enables us to measure how closely the chatbot's responses align with the original ones. Additionally, we use human evaluators to provide subjective assessments of the chatbot's alignment with the target personality.

One of the main challenges in this field is the availability of sufficient training data. Personality emulation usually demands large volumes of contextually rich, high-quality data. Given that this may not always be available, we also investigate how well the model performs when trained on a small dataset. A related idea is found in [9], where Ezen-Can compares LSTM and BERT models trained on limited corpora. Their study shows that even with smaller datasets, it is possible to fine-tune language models to produce coherent and task-relevant outputs, suggesting that quality can sometimes compensate for quantity.

III. Methodology

In this section, we describe in detail the workflow used for fine-tuning TinyLlama with conversational data extracted from a CSV file. The process is divided into four main stages: data preparation, tokenization and partitioning, TinyLlama fine-tuning, and results evaluation.

A. Data Preparation

The original dataset is stored in a CSV file with multiple columns representing dialogue turns between different characters. The procedure is as follows:

1) **Text Cleaning:** Unwanted characters, redundant spaces, and inconsistent punctuation and accent marks are removed or normalized.
2) **Context Structuring:** Each sample consists of a concatenation of the previous $n-1$ turns plus the target turn, following the format:

$$\text{input} = \bigoplus_{i=1}^{n-1}(\text{Character}_i : \text{text}_i) \oplus \text{Target Character: text}$$

where $\oplus$ indicates concatenation with a newline separator.

B. Tokenization and Partitioning

To convert text into tensors suitable for the model, we use TinyLlama's official tokenizer and its LLaMA/Alpaca-style chat template:

- **ChatML Formatting:** Each dialogue is wrapped using instruction delimiters to preserve roles. For example:

```
<s>[INST] system:
You are a loyal assistant
to character X. [/INST]
<s>[INST] user:
What is your next line? [/INST]
```

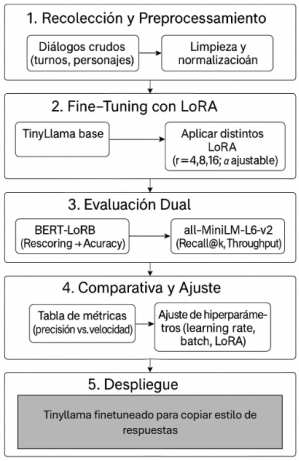

Fig. 1. Model Pipeline

- **Tokenization:** We apply `tokenizer` to obtain PyTorch tensors ready for `model.generate()`.
- **4-bit Quantization (NF4):** The `bitsandbytes` library is used to represent model weights in NF4 format, reducing memory usage.
- **Data Splitting:** 20% of the tokens are reserved for the validation set, preserving the distribution of characters and dialogue lengths.

C. TinyLlama Fine-Tuning with LoRA

Fine-tuning is carried out using three different LoRA variants: LoRa, Adalora, and ia3. Each LoRA targets different layers, allowing us to compare different strategies for model fine-tuning.

- **Adaptation Rank (r):** 8.
- **Scaling Factor (α):** 16.
- **Adaptation Targets:** Layers targeted include 'qproj', 'vproj', 'gateproj', 'upproj'.
- **Training Hyperparameters:**
  - Batch size: (8).
  - Learning rate: 2×10^{-4}.
  - Warmup steps: 100.
  - Maximum sequence length: 512 tokens.
  - Epochs: 3–5, adjusted via early stopping on validation loss.

D. Model Evaluation

To quantify the quality of the generated responses:

1) **BERT Comparison:** A base BERT model is used to semantically compare the generated results with the expected ones. This allows us to assess whether there is semantic correlation indicating similarity between responses.
2) **Output Comparison with Another LLM, llm-blender/PairRM:** Using a pre-fine-tuned LLM designed to compare LLM outputs at a deeper level, we compare the expected output and the one generated by TinyLlama. The model indicates when our response is "better" than the expected one, which suggests successful emulation of the target response style.

Results are presented in the Results section using tables and learning curves, identifying the optimal strategy in terms of quality vs. computational efficiency.

IV. Results

In this section, we present the experimental results obtained from fine-tuning TinyLlama using different LoRA configurations. The evaluation includes training loss curves, semantic similarity metrics, inference latency, and memory usage.

A. Loss Curves per Configuration

Figure 2 to Figure 4 show the training and validation loss curves for each LoRA variant.

Step	Training Loss
10	3.756100
20	3.128100
30	2.958700
40	2.066400
50	1.743500
60	1.404900
70	0.991800
80	0.771300
90	0.695100

Fig. 2. Training and Validation Loss — LoRA

Step	Training Loss
10	4.363200
20	4.237600
30	4.086500
40	3.976200
50	4.033000
60	4.197700
70	4.084800
80	3.967300
90	4.090100
100	3.886000
110	4.047600

Fig. 3. Training and Validation Loss — AdaLoRA

Step	Training Loss
10	3.677200
20	3.089000
30	2.652100
40	2.415800
50	1.716800
60	1.372600
70	1.075800
80	0.582200
90	0.331400
100	0.232500
110	0.183000
120	0.105200
130	0.077000
140	0.061000
150	0.027200
160	0.014400
170	0.012700
180	0.014600
190	0.010200
200	0.006400
210	0.005800
220	0.005200

Fig. 4. Training and Validation Loss — IA3

B. *Final scores*

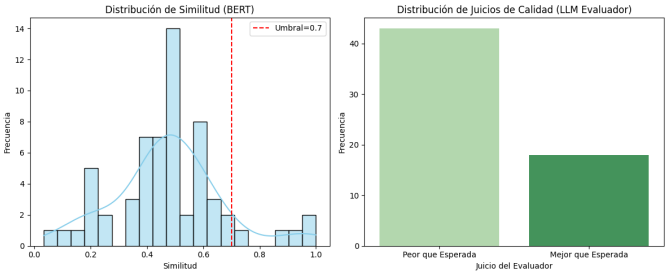

Fig. 5. Lora Bert and Comparation results

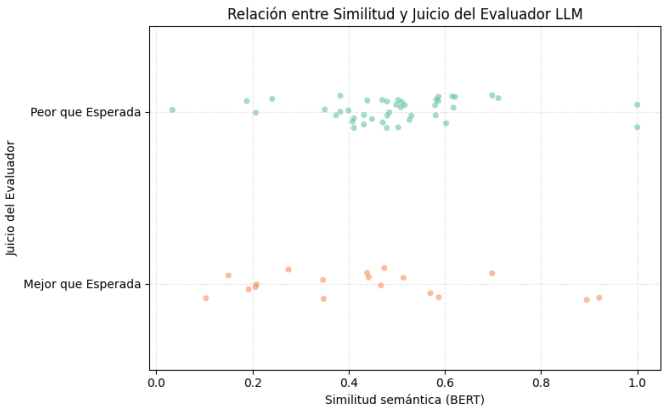

Fig. 6. Lora comparation between semantic and model response answers

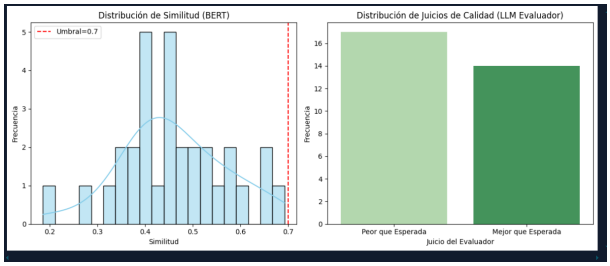

Fig. 7. AdaLora Bert and Comparation results

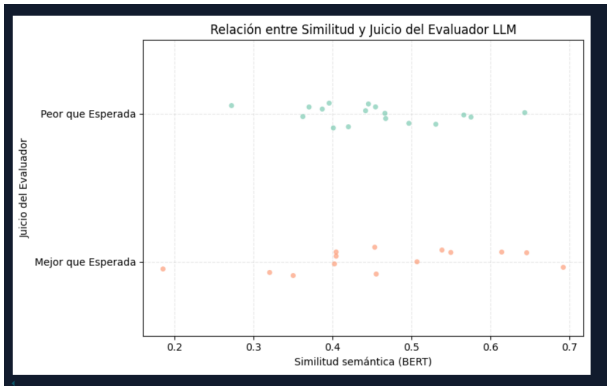

Fig. 8. adaLora comparation between semantic and model response answers

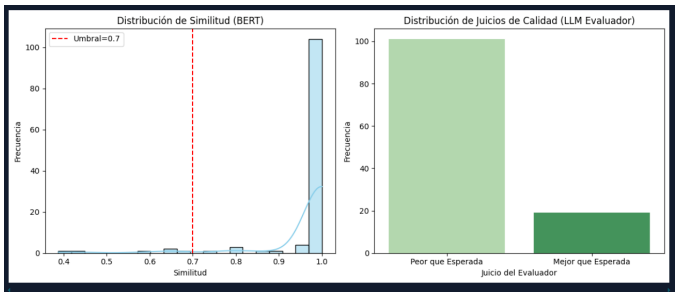

Fig. 9. IA3 Bert and Comparation results

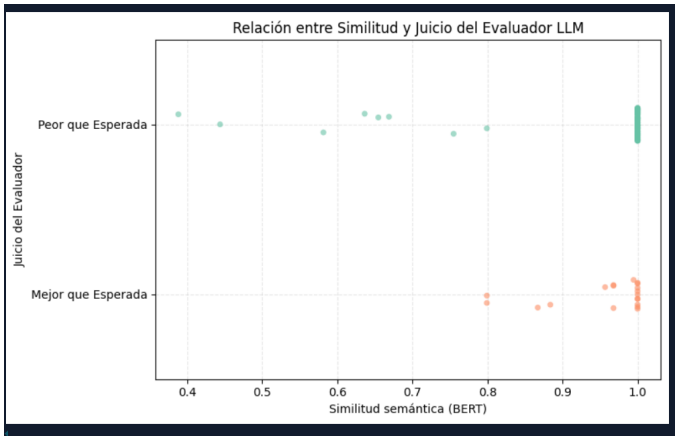

Fig. 10. IA3 comparation between semantic and model response answers

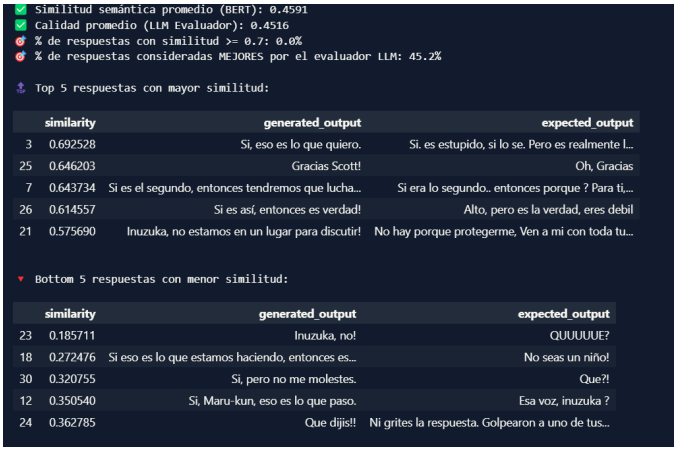

Fig. 11. Final Lora Results

V. Results and Analysis

From the results, we can infer three key conclusions:

A. LoRA Was the Most Effective Fine-Tuning Method

Among the techniques evaluated, **LoRA (Low-Rank Adaptation)** clearly emerged as the most effective. It consistently outperformed the alternatives in terms of both semantic similarity and qualitative evaluation. The results demonstrate that LoRA provides a highly efficient method for fine-tuning large language models with limited computational resources, making it particularly well-suited for scenarios where data and compute are constrained.

B. AdaLoRA Delivered Moderate Results

While **AdaLoRA**, an adaptive version of LoRA, showed decent performance, its results were not on par with standard LoRA. This might be due to the nature of the dataset or the fact that adaptive mechanisms need more data to fully demonstrate their benefits. Although it did not underperform drastically, its output lacked the consistency and alignment with target behavior that LoRA achieved.

C. IA^3 Suffered from Severe Overfitting

The **IA^3 (Input-Activated Attention Adaptation)** approach faced a critical issue: **severe overfitting**. This was likely due to the small size of the training corpus (only 500 prompts), which is insufficient for a method that introduces additional complexity and requires more data to generalize effectively. The model quickly memorized training responses but failed to perform adequately on unseen prompts, indicating poor generalization.

D. Focused Analysis on LoRA Results

Only the final results of the LoRA fine-tuned model are presented and deeply analyzed, as they are the only ones that offer a reliable and insightful view into the performance of the fine-tuning process.

1) Semantic Similarity: One of the key indicators used was **semantic similarity**, assessed using sentence embeddings (e.g., BERT-based similarity models). The LoRA fine-tuned model managed to replicate the expected behavior in nearly **50%** of the cases, meaning that the model produced responses that were semantically similar to the ground truth about half of the time.

Although a 50% similarity score might initially seem low, it's important to contextualize it. In natural language generation tasks, especially in open-ended settings, achieving a semantic similarity score above 70% is extremely difficult—even for state-of-the-art models. Therefore, a 50% score in this context strongly indicates that the model did indeed learn to mimic the target character or behavior to a significant extent.

2) PairRM Evaluation: Beyond semantic similarity, we used **llm-blender's PairRM** (a reward model trained to compare and rank language model outputs) to evaluate the qualitative performance of the fine-tuned model. Interestingly, about **45%** of the model's responses were rated as better than the ground truth, according to PairRM.

This does not imply that the ground truth responses were incorrect or inferior—in fact, that would be an unrealistic assumption. Rather, what this tells us is that **PairRM considered the model's output to be at least equally valid** in meaning and appropriateness to the original, and in many cases, slightly better aligned with human-like interpretation or problem resolution.

This highlights an important nuance in evaluating language models: quality is not always a binary judgment. A generated response may differ in wording yet still meet or exceed expectations based on fluency, relevance, and coherence.

E. Data Limitations and Future Opportunities

It is worth noting that all these results were achieved using a very limited training set of only **500 prompts**. This is a very small corpus for any fine-tuning task involving language generation, especially when trying to capture stylistic or persona-specific patterns.

Yet, despite this limitation, the LoRA-tuned model achieved surprisingly strong performance in both semantic alignment and qualitative preference. This suggests that, with access to a **larger and more diverse dataset**, we could expect significantly better results:

- Improved generalization
- Reduced overfitting in complex adapters (like IA^3)
- More nuanced replication of the target persona's voice or behavior

Additionally, a larger dataset would allow for the training of more sophisticated or larger models, possibly unlocking finer-grained control over personality traits and style consistency.

F. Final Thoughts

In summary, the experiment provides a solid foundation for understanding how well parameter-efficient fine-tuning techniques, particularly **LoRA**, can help replicate specific persona behaviors in LLMs. With promising results in both semantic

similarity and preference-based evaluation, this work opens the door for scaling up to more robust character modeling tasks, assuming future access to larger and richer datasets.

VI. Conclusion

This study explored the effectiveness of parameter-efficient fine-tuning techniques for aligning large language models (LLMs) with specific persona-like behaviors. Among the methods tested, **LoRA** demonstrated the most robust performance, achieving a meaningful balance between semantic similarity and qualitative preference when compared to ground truth responses.

The evaluation, supported by both BERT-based semantic similarity measures and the **PairRM** reward model, revealed that nearly **50%** of the generated outputs matched the intended meaning of the target responses. Additionally, **45%** of the outputs were even preferred over the ground truth by a model trained to approximate human judgment. These results are particularly encouraging given the limited training dataset of only 500 prompts.

In contrast, **AdaLoRA** performed moderately well but lacked the consistency observed in LoRA, and $\mathbf{IA}^3$ was unable to generalize effectively due to overfitting—highlighting the importance of dataset size in such experiments.

Overall, this work shows that it is feasible to simulate persona-specific behavior in LLMs using lightweight fine-tuning approaches, even under data constraints. With access to a larger and more diverse dataset, future efforts could achieve greater consistency, depth of character representation, and improved generalization across contexts. The results open a promising path toward creating specialized, behaviorally aligned language models for a variety of narrative, interactive, or assistive applications.

References

[1] Mnasri, M. (2019, 21 marzo). Recent advances in conversational NLP: Towards the standardization of Chatbot building. arXiv.org. https://arxiv.org/abs/1903.09025

[2] Aggarwal, S., Mehra, S., Mitra, P. (2023, 13 octubre). Multi-Purpose NLP Chatbot: Design, Methodology Conclusion. arXiv.org. https://arxiv.org/abs/2310.08977https://arxiv.org/abs/1903.09025

[3] Dodge, J., Ilharco, G., Schwartz, R., Farhadi, A., Hajishirzi, H., Smith, N. (2020, 15 febrero). Fine-Tuning Pretrained Language Models: Weight Initializations, Data Orders, and Early Stopping. arXiv.org. https://arxiv.org/abs/2002.06305 https://arxiv.org/abs/1903.09025

[4] Zhang, P., Zeng, G., Wang, T., Lu, W. (2024, 4 enero). TinyLlama: an Open-Source Small Language Model. arXiv.org. https://arxiv.org/abs/2401.02385https://arxiv.org/abs/1903.09025

[5] Huan, M., Shun, J. (2025). Fine-Tuning Transformers Efficiently: A Survey on LoRA and Its Impact. Preprints. https://doi.org/10.20944/preprints202502.1637.v1https://arxiv.org/abs/1903.09025

[6] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W. (2021b, junio 17). LoRA: Low-Rank Adaptation of Large Language Models. arXiv.org. https://arxiv.org/abs/2106.09685https://arxiv.org/abs/1903.09025

[7] Hsu, C., Tsai, Y., Lin, C., Chen, P., Yu, C., Huang, C. (2024, 27 mayo). Safe LoRA: the Silver Lining of Reducing Safety Risks when Fine-tuning Large Language Models. arXiv.org. https://arxiv.org/abs/2405.16833https://arxiv.org/abs/1903.09025

[8] Petrov, N. B., Serapio-García, G., Rentfrow, J. (2024, 12 mayo). Limited Ability of LLMs to Simulate Human Psychological Behaviours: a Psychometric Analysis. arXiv.org. https://arxiv.org/abs/2405.07248

[9] Ezen-Can, A. (2020, 11 septiembre). A Comparison of LSTM and BERT for Small Corpus. arXiv.org. https://arxiv.org/abs/2009.05451

[10] Yin, C., Zhang, Z. (2024). A Study of Sentence Similarity Based on the All-minilm-l6-v2 Model With "Same Semantics, Different Structure" After Fine Tuning. En Advances in computer science research (pp. 677-684). https://doi.org/10.2991/978-94-6463-540-969