On the Need to Compare the Architecture Design of Large Language Models: A Preliminary Reference Metamodel

Jesús Carreño-Bolufer[1] **and Giovanni Giachetti**[1,2]

[1]Valencian Research Institute for Artificial Intelligence (VRAIN)
Universitat Politècnica de València – Valencia – Spain

[2]Universidad Andrés Bello – Santiago – Chile

jecarbo@vrain.upv.es

Abstract. *A large number of Large Language Models (LLMs) have been developed and made available to the community. However, analyzing and comparing these models remains challenging since there is no clear, standardized way to describe their architectures precisely. While LLM developers typically provide textual model cards, configuration files, or source code, these artifacts do not follow a common standard. Consequently, researchers need to make additional efforts to reuse and compare existing LLMs. To address this issue, we propose a conceptual modeling approach to create a model-driven method for describing and analyzing LLM architectures.*

1. Introduction

The development of Large Language Models (LLMs) [Minaee et al. 2024] involves multidisciplinary teams that include machine learning engineers, mathematicians, domain experts and data scientists. However, LLMs development mainly focuses on evaluating results (i.e. executing AI models), often omitting details about the architecture design. This has a negative impact on communication and knowledge reuse when analyzing incomplete, inconsistent or heterogeneous descriptions of systems, which makes it difficult to compare and select existing LLMs solutions and may lead to manual errors during the coding phase.

The adoption of model-driven software engineering techniques [Pastor et al. 2008] can help to fill this gap in LLM development processes. Specifically, having reference conceptual models to provide a common representation to analyze and compare LLM Architecture design decisions. Since most of LLMs are based on the Transformer architecture [Vaswani et al. 2017], the goal of this research is to represent the LLM Transformer architecture as a structural model to improve analysis and understanding of specific LLM implementations through model-based methods. To achieve this, the following questions must be answered:

- To what extend can we conceptualize the constructs of the transformer architecture?
- To what extend can model-based methods improve the understandability and communication of LLM architecture?

2. Methodology

The proposed research follows the Design Science methodology considering three main stages [Wieringa 2014]: (1) Problem investigation through a literature review and empirical study considering open-source LLM projects. (2) Design of the conceptual model and

concrete syntax for the representation of a LLM architecture (treatment). (3) Validating trough a case study (treatment validation).

3. Preliminary Results

We conducted a preliminary study of the Hugging Face data repository for transformer-based models. Our findings show that existing Large Language Model (LLM) description mechanisms rely on model cards and configuration files. However, these files are generally incomplete, heterogeneous, and sometimes inconsistent with the actual implementation. Therefore, users must rely on code to find out about the model details, but this is not always available. We identified these issues in popular models, such as DeepSeek-V3.

We also conceptualized the fundamental components of the LLM architecture and created a metamodel based on the MOF standard[Object Management Group 2016]. The metamodel considers concepts such as *Component*, *Block* and *Connection*. Thus, it provides an initial Domain-Specific Modelling language to represent LLM architectures [1]. As a descriptive use case, we instantiated DeepSeek-V3.2[2].

4. Future Work

The specification of the reference LLM architecture metamodel is the starting point for the following future works:

- Analyze different LLM projects on Hugging Face to extend and validate the proposed conceptual model.
- Define and evaluate a concrete syntax supported by a specific model editor that facilitates the specification of LLMs' architectural models.
- Conduct empirical studies to evaluate the extent to which this approach reduces the effort required for designing and analyzing LLMs' architectures, as well as its feasibility for identifying issues in LLMs' specifications that have gone undetected.

5. Conclusions

This work proposes the use of conceptual models to adopt software engineering practices in the development of LLMs. In particular, the definition of a reference MOF metamodel for the LLM transformer architecture. The reference metamodel is oriented to improve the communication and understanding of the LLMs architecture and its design decisions.

References

Minaee, S. et al. (2024). Large language models: A survey.

Object Management Group (2016). *Meta-Object Facility (MOF)*. Object Management Group, 2.5.1 edition. Formal specification.

Pastor, O., España, S., Panach, J. I., and Aquino, N. (2008). Model-driven development. *Informatik-Spektrum*, 31(5):394–407.

Vaswani, A. et al. (2017). Attention is all you need.

Wieringa, R. (2014). *Design science methodology for information systems and software engineering*. Springer.

[1] https://doi.org/10.5281/zenodo.18475481

[2] https://doi.org/10.5281/zenodo.18503112