*Computational Models of Argument*
*C. Reed et al. (Eds.)*


*doi:10.3233/FAIA240329*

# Translating Natural Language Arguments to Computational Arguments Using LLMs

Guilherme TRAJANO [a], Débora C. ENGELMANN [b], Rafel H. BORDINI [c], Stefan SARKADI [d,1], Jack MUMFORD [e], and Alison R. PANISSON [a,2]
[a] *Department of Computing, Federal University of Santa Catarina, Brazil*
[b] *Institute of Artificial Intelligence in Health, Brazil*
[c] *School of Technology, Pontifical Catholic University of Rio Grande do Sul, Brazil*
[d] *Department of Informatics, King's College London, United Kingdom*
[e] *Department of Computer Science, University of Liverpool, United Kingdom*

**Abstract.** Large Language Models (LLMs) have become a significant milestone in the history of artificial intelligence, representing a powerful technology that drives advancements in natural language understanding and generation. In this paper, we propose an approach in which LLMs are utilized to support the task of translating natural language arguments into computational representations. Our approach is grounded in using argumentation schemes to classify arguments, providing context to LLMs for performing the proposed task. Our results demonstrate that LLMs, even with a short context, can handle simple argument structures. Moreover, our findings suggest that a larger context would likely enhance the performance, particularly when dealing with more complex argument structures.

**Keywords.** Argumentation, Large Language Models, Human-Agent Interaction

## 1. Introduction

In recent years, the intersection of Artificial Intelligence (AI) and Natural Language Processing (NLP) has led to groundbreaking advancements in various domains. One such area of intense research and development is the use of Large Language Models (LLMs), which can be applied to diverse tasks in different application domains. LLMs have been used in medical advice consultation, mental health analysis, e-government, serving as a writing or reading assistant in education, legal document analysis, financial sentiment analysis and even as an assistant on scientific research tasks [1,2]. One of the most outstanding uses for LLMs is their integration into human-computer interaction interfaces, representing a pivotal advancement in AI technology. LLMs have revolutionized how to interact with machines, enabling natural and intuitive communication where AI systems can understand, interpret, and respond to human interactions with context sensitivity.

In contrast, certain powerful AI paradigms, such as multi-agent systems, operate on symbolic reasoning principles. However, there is a pressing need to explore how LLMs

[1]Was supported by the Royal Academy of Engineering and the Office of the Chief Science Adviser for National Security under the UK ICRF fellowship.
[2]Corresponding Author: Alison R. Panisson, e-mail: alison.panisson@ufsc.br.

can effectively bridge the gap between human interaction and symbolic representation within these paradigms. This investigation is crucial for unlocking the full potential of LLMs in facilitating seamless communication and interaction between humans and AI systems operating on symbolic reasoning. In this paper, we propose an approach in which LLMs are used to support the task of translating natural language arguments into computational arguments. This task is essential to developing AI applications with sophisticated interaction interfaces, which also contextualize recent advances in Hybrid Intelligence (HI) [3]. It moves beyond simple question/command-answer systems to systems that can understand, reason, and articulate complex arguments.

Our approach is structured around a workflow where natural language arguments are initially classified based on the reasoning pattern (argumentation scheme) used in each argument. We then employ a Retrieval Augmented Generation (RAG) methodology to contextualize LLMs, guiding them in the task of translating natural language arguments into computational representations. Our approach is modular, allowing flexibility in the choice of LLM and context size. To evaluate the LLMs' capability in performing the proposed task, we sought feedback from experts in knowledge representation and argumentation on the computational arguments generated.

## 2. Background

### *2.1. Large Language Models*

Language Models (LMs) are computational systems engineered to comprehend and generate natural language text [4]. In a study by [1], the authors observed that scaling the model size (number of parameters) and training data size typically improves model performance on downstream tasks. Large Language Models (LLMs), often containing more than 10 billion parameters[3], and trained on massive text data, rely primarily on the Transformer architecture. This architecture features the stacking of multi-head attention layers in an extensively deep neural network [1]. According to [1], while language modelling is not a new concept, it has evolved with advancements in AI. Early language models primarily focused on generating text data, while recent models, such as GPT-4 [5], are capable of solving complex tasks.

LLMs have been widely used in the research community, solving classic NLP tasks, acting as information retrieval models and recommender systems, processing and integrating information from various modalities, such as multimodal LLMs, and serving as LLM-based agents. They are also utilized in specific domain applications, including healthcare, education, law, finance, and scientific research [1]. One task with significant potential for LLMs is translating natural language text. NLP tasks such as text generation have been extensively studied, and LLMs exhibit strong language generation skills due to their pre-training being text prediction [1]. In this paper, we propose to evaluate the capacity of LLMs to generate computational arguments based on arguments in natural language and their associated argumentation schemes.

[3]There is no formal consensus on the minimum parameter scale for a LM to be considered an LLM [1].

### 2.2. Retrieval Augmented Generation

Despite the good performance that LLMs exhibit on NLP tasks, they still suffer from limitations such as the inability to expand their memory, reliance on outdated knowledge, and a tendency to 'confabulate' [6]. Retrieval Augmented Generation (RAG) is considered a possible solution to these problems, allowing the LLM to access external information beyond its training data [7]. RAG operates by retrieving documents represented as vector embeddings that closely match the user's prompt. When presented with a specific prompt, RAG retrieves the top-$k$ documents most relevant to that prompt [8]. This mechanism enables the addition, modification, or removal of knowledge bases for large language models.

In this paper, we used RAG to retrieve documents containing relevant information for the task of translating natural language arguments into computational form, considering the associated argumentation scheme of the natural language argument. This was facilitated by RAG's capability to flexibly create and manipulate an external knowledge base. The implementation details of the RAG pipeline will be presented in Section 3.

### 2.3. Argumentation

Argumentation, particularly computational models of argument, is emerging as a central component in many aspects of AI research because it provides a robust approach to handling incomplete and inconsistent information, similar to how humans approach this task [2]. Furthermore, argumentation offers a sophisticated form of communication that meets current needs in AI applications, such as interpretability and explainability [3,9,10]. In summary, argumentation provides sophisticated reasoning and communication components. An agent can build and evaluate arguments and counterarguments supporting its conclusions and decision-making, as well as engage in discussions or debates where arguments are exchanged with humans or other AI agents [2,11].

Argumentation encompasses a multifaceted phenomenon that extends beyond mere awareness of arguments; this complexity must also be reflected in computational models of arguments. It necessitates a deep comprehension of the argument structure, delving into the implicit information present within it. To achieve such understanding, an agent needs to associate the reasoning pattern employed in constructing the argument. This association provides the necessary reference points for a more intricate understanding. Argumentation schemes are a recognized form of providing such reasoning patterns. They are considered patterns for arguments (or inferences) representing the structure of common types of arguments used in everyday discourse as well as in special contexts such as legal and scientific argumentation [12,13]. These schemes have been extensively catalogued by several authors across multiple application domains [12,13,14,15,16,17]. For example, consider the argumentation schemes below, based on the *Argument from Position to Know* scheme from [12].

> "Agent $Ag$ is in a position $P$ that implies knowing things in a certain subject domain $S$ containing proposition $A$ **(Major Premise)**. $Ag$ asserts that $A$ **(Minor Premise)**. Then we should conclude that $A$ **(Conclusion)**".

This argumentation scheme provides a reasoning pattern that can be used to instantiate arguments. Instantiating this scheme involves replacing the variables $Ag$, $P$, $A$, and $S$ with specific contextual elements from the application domain. For instance:

> "*Peter* is in a position *doctor* that implies knowing things in a certain subject domain *medicine* containing proposition *smoking causes cancer*. *John* asserts that *smoking causes cancer*. Then we should conclude that *smoking causes cancer*".

Understanding the argumentation scheme associated with the instantiated argument provides a deep understanding of it. This is because argumentation schemes offer references to implicit information related to arguments of that kind, as highlighted by critical questions [13]. Such understanding enables agents to conduct a thorough analysis of argument acceptance while exploring sophisticated links of information during dialogues.

Some approaches in the argumentation literature have suggested that argumentation schemes [12,13] could be translated into computational structures using defeasible inferences [18,19,20]. The acceptability of arguments instantiated using these rules could then be used to instantiate frameworks for computational argumentation, such as ASPIC+ [21], DeLP [22], and others [23]. Our approach aligns with this line of work. However, in this paper, we specifically focus on translating natural language arguments into a computational representation suitable for these computational argumentation frameworks. We maintain the link between the argument and the argumentation scheme used to instantiate it, thereby providing agents with a deep understanding of the argument.

## 3. LLMs Supporting Argument Translation

In this paper, we propose an approach in which LLMs are used to support the task of translating natural language arguments into computational arguments. This task is essential in the context of developing natural language communication interfaces between intelligent agents (software) and humans, with a focus on sophisticated communication phenomena such as argumentation.

There are many argumentation frameworks in the literature, some of which have been implemented and integrated with reasoning mechanisms for intelligent agents [11, 23,24]. This powerful integration allows intelligent agents to reason and communicate using arguments. This means that agents can engage in sophisticated reasoning processes based on the construction of arguments for and against their conclusions and decision-making, as well as communicate in a more informed manner, justifying their positions in dialogues using arguments [11]. While this represents an innovative approach in the area of multi-agent systems, there is still a missing approach that allows agents and humans to communicate with the same sophistication, which, for example, underpins the development of hybrid intelligence [3].

One of the initial steps toward achieving such human-agent communication interfaces is devising mechanisms for translating the complex structures found in natural language arguments into computational ones (similar to those argument structures already utilized in the aforementioned computational argumentation frameworks). To advance in this direction, we propose leveraging LLMs to assist in the translation of natural language arguments into computational ones. For example, our goal is to develop an approach that receives as input an argument in natural language, such as:

> *"Peter is a doctor and says that smoking causes cancer. Therefore, we can conclude that smoking causes cancer."*

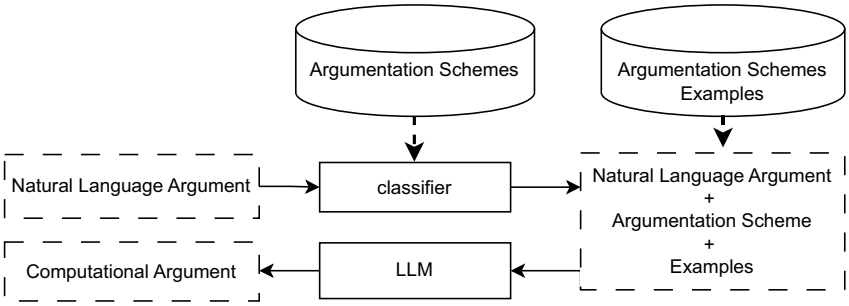


**Figure 1.** Overview for the Proposed Approach.

then understands the reasoning pattern (argumentation scheme) used to instantiate that argument and then translates it to a computational representation of that argument following the computational structure of the argumentation scheme used to instantiate it. Consider a simple computational structure for arguments given by $\langle$`[premises],conclusion`$\rangle$. Using this simple structure, it is possible to represent argumentation schemes and then instantiate them. For example, the argumentation scheme position to know can be represented as $\langle$ [ `position_to_know(Ag,S)`, `asserts(Ag,A)`, `contain(S,A)` ], `A` $\rangle$, and then it can be instantiated, according to the natural language argument, as follow:

$\langle$ [ `position_to_know(peter,medicine)`, `asserts(peter,causes(smoking,cancer))`, `contain(medicine,causes(smoking,cancer))` ], `causes(smoking,cancer)` $\rangle$

In this process of translating natural language to computational arguments, we desire that the reasoning pattern, argumentation scheme, available to agents be respected. That is, all predicates present in the argumentation schemes will also be present in the computational argument, and the variables correctly instantiated. There are different levels of complexity to instantiate variables, according to the argumentation scheme used and the application domain. In the example above, variables are instantiated with terms, for example, $\{\texttt{S} \mapsto \texttt{medicine}\}$ and with predicates $\{\texttt{A} \mapsto \texttt{causes(smoking,cancer)}\}$. The second case seems more challenging.

### *3.1. Proposed Approach*

Our approach is grounded on the use of argumentation schemes to classify arguments and provide argument structure, also following those works that define computational arguments based on argumentation schemes [13]. In our approach argumentation schemes are used to provide references to the LLMs in the task of translating natural language to computational arguments.

A high-level overview of the proposed approach is depicted in Figure 1. In this diagram, our approach takes a natural language argument as input and generates a computational representation of that argument based on the computational argumentation framework being used. The process involves several steps. First, the natural language argument is classified according to the argumentation schemes available to that specific application domain [25]. Second, the approach retrieves a computational representation of the corresponding argumentation scheme, along with examples showcasing the mapping between natural language arguments and computational representations based on that argumenta-

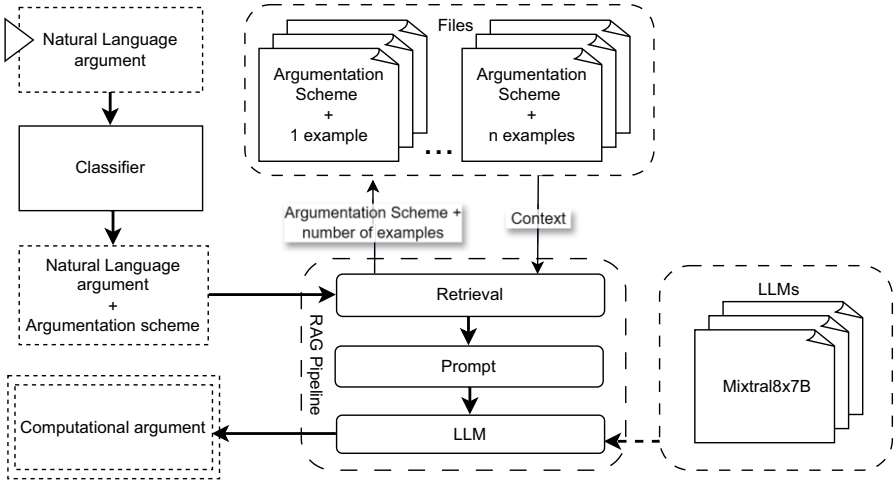


**Figure 2.** Proposed Approach with RAG Pipeline.

tion scheme. These elements are then provided to a LLM, which uses the argumentation scheme, examples, and the original natural language argument to perform the translation.

At the time of this work, two main methodologies are commonly applied when using LLMs: *Retrieval-Augmented Generation* (RAG), which was discussed in Section 2, and *fine-tuning*. *Fine-tuning* the model consists of retraining the model on new data, which can be computationally expensive, especially when working with models with a large number of parameters [26]. Due to the significant computational cost of the *fine-tuning* method, we opted to utilize *Retrieval Augmented Generation* in this work. In future work, we intend to develop and compare both approaches.

To utilize RAG, we established a pipeline. The initial step involved defining a function that would extract features of the information to be shown to the LLM, known as the embedding function. Following this, we created a vector database. We selected the open-source vector database Chroma[4], specifically designed for storing vector embeddings, a crucial task in Natural Language Processing. After creating it, files containing the context for the LLM are added to the vector database. One important note here is that if all the files were added to the database simultaneously, the RAG method could present the LLM with the context of the wrong argumentation scheme since there could be potential semantic similarities between the user's prompt and an example of a different argumentation scheme in the vector database. That is why we integrated a classifier [25] into our pipeline. First, it classifies the natural language argument according to the available argumentation schemes. Then, our approach retrieves examples related to the argumentation scheme used to instantiate the input argument, providing the necessary context for the translation task in our approach.

Currently, many LLMs could be incorporated into our approach. They have different restrictions regarding the context size, which reflects the length of the context that can be provided to them. To make our approach generic regarding the model used and the context length, we modelled these choices into two separate components. In our approach, we set which LLMs, from those available, will be used, as well as the number of examples that will be provided as context to the model. Context examples were or-

[4] https://www.trychroma.com/

ganized into different files based on the number of examples. Thus, after classifying the argumentation scheme used to instantiate the input argument, those files are retrieved according to the argumentation scheme and the number of examples. Then, this context is used to build the prompt, which is executed over the selected LLM; it then performs the translation and returns the computational representation of that argument.

We conducted several experiments using different context sizes and various LLM models, evaluating their ability to perform the proposed task. In the next section, we describe part of the experiments and our findings.

### *3.2. Evaluation*

In order to evaluate whether LLMs can efficiently translate natural language into computational arguments, we first created a small knowledge base containing examples that match natural language and computational arguments. Each example references the argumentation scheme used to instantiate it. This small knowledge base was organized in different files. In each file, we provide an argumentation scheme along with different matching examples for that scheme. Based on the number of examples specified in our approach, the corresponding file was retrieved to provide context to the LLMs.

Furthermore, we created a second knowledge base containing only natural language arguments, distinct from those used in the first knowledge base that provides context to the LLMs. These natural language arguments from the second knowledge base are later used as input in our approach to evaluate the LLM's capability to translate natural language arguments into computational arguments. In summary, the LLM receives a natural language argument and its corresponding context (the argumentation scheme used to instantiate that argument and a number of examples of the task it should perform). It then generates a computational representation of that argument based on the computational representation of the argumentation scheme also provided in the context.

During the process of creating the knowledge bases necessary to evaluate our approach, we randomly selected a total of 10 argumentation schemes from existing knowledge bases[5]. A total of 8 argumentation schemes from the general domain and 2 from the hospital bed allocation domain were selected. The argumentation schemes selected from the general domain were: *Classification*, *Analogy Based on Classification*, *Necessary Condition*, *Need for Help*, *Position to Know*, *Argument from Opposities*, *Moral Justification Ad Populum* and *Cause to Effect*. These schemes offer a diverse array of structures and complexities, allowing us to comprehensively evaluate the LLMs' capacity to translate arguments into computational form. Shorter argumentation schemes such as *Position to Know* and *Classification* enable assessment of the LLMs' performance in handling simpler arguments, providing insights into basic translation capabilities. Conversely, longer schemes like *Analogy Based on Classification* and *Effect to Cause* present more intricate contexts and information, challenging the LLMs' ability to maintain context and accurately instantiate variables within complex arguments. Further, argumentation schemes from specific application domains, such as bed allocation [28], provide an overview of the LLM's capability of performing this task in specific domains.

For each argumentation scheme, we created 5 different files containing the computational representation of that scheme and examples of the proposed task (matching ar-

[5]For example, the knowledge base from [27], available at https://carnelian-brow-e71.notion.site/Base-de-Conhecimento-7ff0ef55a217411f84ff1aca7d652488

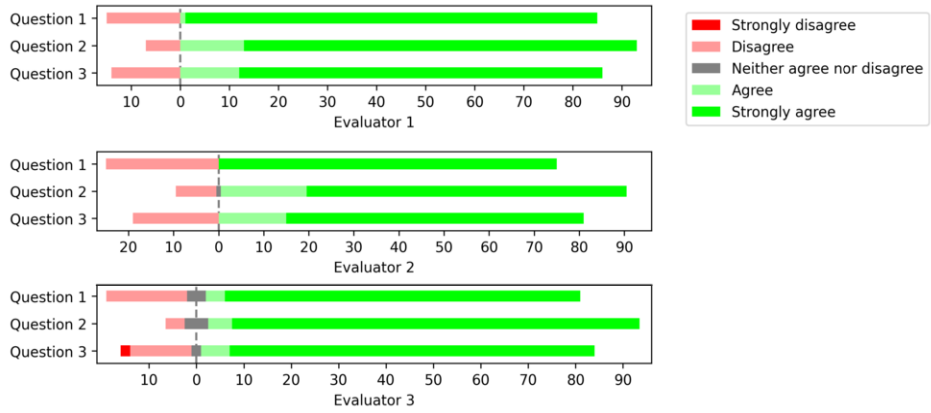


**Figure 3.** Distribution of the evaluators' answers based on the 5-point Likert Scale.

guments in natural language and computational language). Each file contains a varying number of examples, ranging from 1 to 5 examples. The structure of these files includes one line with the argumentation scheme using generic variables, followed by one line for each example presenting the natural language argument and its corresponding translation to the computational argument. Next, to evaluate the proposed approach, we create the second knowledge base containing 10 natural language arguments for each considered argumentation scheme (i.e., 100 natural language arguments in total). We ensure that these natural language arguments were different from those used to build the first knowledge base, which is used as context to the LLM.

Finally, we evaluated our approach by providing the arguments from the second knowledge base as input, varying the number of examples and the LLMs model[6]. One of our goals was to determine the minimum number of examples required for LLMs to accurately instantiate and organize the variables and predicates of argumentation schemes in computational arguments. We observed that this requirement depends on the LLM model used. One LLM model that was able to execute the task (i.e., instantiate all argumentation schemes according to the input argument) using only one example was the Mixtral8x7B model [29]. Therefore, we took the resulting computational arguments, translated by the Mixtral8x7B model using only one example, to be further evaluated by experts in the field of knowledge representation and argumentation. We provided them with the computational representation of the argumentation scheme, the natural language argument, and the resulting computational argument, then asked them to answer the following three questions: **Question 1** - Did the LLM follow the provided argumentation scheme? **Question 2** - Are the variables correctly instantiated in the computational representation of the argument? **Question 3** - Does it generate good semantics for instantiated variables?

We provided 5 choices of answer based on the Likert scale [30,31]. The results of this evaluation are provided by Figure 3. The average percentage of "strongly agree" responses from all evaluators for questions 1, 2, and 3 were 74.4%, 78.7%, and 72%, respectively. When grouping classes into "agree" (combining "strongly agree" and "agree" answers) and "disagree" (combining "disagree" and "strongly disagree"), the average percentages of "agree" were 79.3%, 91%, and 83%, respectively. The percent agreement [32] among the evaluators was 0.88, 0.62, and 0.62 for exact matches on answers

[6]Each choice of parameter (n examples and/or LLM) generates 100 computational arguments.

by the three evaluators. When considering the super classes "agree" and "disagree", the percent agreements were 0.88, 0.77, and 0.75, respectively. These percent agreements are considered substantial[7] according to [33]. When considering only argumentation schemes with simple structures, where variables are instantiated with simple terms, the average percentage of "strongly agree" or "agree" responses from all evaluators for questions 1, 2, and 3 were 79%, 97%, and 100%, respectively. The percent agreement for those argumentation schemes were: 0.98, 0.90, and 0.94 considering the exact match, and 0.98, 0.92, and 1.00 considering grouping classes in "agree" and "disagree".

Although we are able to observe some degree of subjectivity among the evaluators, we consider that the Mixtral8x7B model has performed well on this task, bearing in mind that for this evaluation, we provided a short context containing only one example of matching natural language and computational argument. Additionally, it can be observed that the model's performance decreases when translating complex arguments that require instantiating a variable with a predicate instead of simple terms. In our tests, we observed that the LLM requires more context to execute this more complex task and improves when provided with more examples in the context. Preliminary results on increasing the context to 2 and 3 examples demonstrate that a larger context resolves 38 to 46% of the issues highlighted by the reviewers. We reached these results asking reviewers to indicate, for each argument they identified a problem with, if the larger context had resolved the issue. Finally, we observed that LLMs are able to deal with *enthymemes*, which we believe is a significant finding highly relevant to AI literature. For example, the argument described in Section 3, in which it adds to the computational representation of that argument the premise `contain(medicine, causes(smoking, cancer))`, which was not present in the natural language argument.

### *3.3. Challenges and Limitations*

Given the constraints on the context size of LLM models, we used simple and straightforward prompts. Additionally, even when the model was asked to return only the computational argument, it would still attempt to explain why it chose a certain object from the text to instantiate a certain variable in the argumentation scheme. As a result, it was necessary to create a filter for the LLM's response. The filter returns only the argument and can also be used to solve some problems pointed out by the evaluators, for example, ensuring that terms have a lowercase representation. Sometimes, the LLM insisted on instantiating variables with uppercase letters for proper names.

Another challenge in building the proposed approach was creating a mechanism in which only the context related to the argumentation scheme used was provided to the LLM. This was necessary because similar arguments, even based on different schemes, usually resulted in mistakes due to semantic similarity. For instance, if the user's prompt is: *"All roses are flowers. Red roses are roses. Therefore, red roses are flowers"*, this would correspond to an argument from *Classification*. However, if there is a similar example in the context provided to the LLM, such as *"James is a botanist and says that roses are plants. Therefore, we can conclude that roses are plants"*, which is an argument from *Position to Know*, the RAG pipeline might retrieve the second example from the database due to the semantic similarity (involving roses) between both sentences. That

[7] [33] propose the following scale: poor (<0.0), slight (0.0-0.2), fair (0.21-0.4), moderate (0.41-0.6), substantial (0.61-0.8), and almost perfect (0.81-1).

could lead to a potential error in the model's analysis, as it would associate the user's argument with the wrong argumentation scheme. For that reason, we used a classifier [25], adding only the relevant context for that entry.

Finally, it is a challenge to find experts in the field of knowledge representation and argumentation available to evaluate hundreds of matching natural language and computational arguments. In this work, we consulted a considerable number of academics, but very few were available to perform the evaluation. As a result, only three evaluators who had not participated in the development of the work were involved. In our future work, we intend to provide a detailed guide for this particular evaluation task, allowing regular individuals to participate. Additionally, once we have created a validated knowledge base, the evaluation process may be conducted using matching algorithms.

## 4. Related Work

LLMs have been used in the literature to support different tasks in the research field of argumentation, as demonstrated by studies such as [34] and [35]. In the former, the authors investigate the effectiveness of LLMs for argument mining via prompting. The latter analyzes whether including context in the classification process of argumentation components may improve the accuracy of contextual language models, thereby enhancing the argumentation mining process.

Our work is also inspired by the idea of classifying arguments according to their reasoning patterns, a topic widely discussed in the literature. For example, [36] explains the importance of classifying argumentation schemes and provides a survey of the literature on scheme classification. An alternative approach to classifying arguments is presented in [37] through the creation of a *Periodic Table of Arguments*. Furthermore, [38] demonstrates that the structure of argumentation schemes can provide useful information for automatically identifying complex argumentative structures in natural language text.

Few works in the literature have utilized LLMs to translate natural language into computational representations. For instance, [39] investigates whether LLMs can translate natural language goals into a structured planning language. Additionally, [40] utilizes LLMs to translate a natural language problem into a symbolic formulation, enabling the use of symbolic solvers to perform inference on the formulated problem. However, to the best of our knowledge, our work is the first to propose an approach in which LLMs support the task of translating natural language arguments into computational arguments.

## 5. Conclusion

In this work, we proposed an approach in which LLMs are used to support the task of translating natural language arguments into computational arguments. The approach is based on classifying natural language arguments according to the argumentation schemes used to create them, providing argumentation schemes and examples as context for LLMs. Our approach is flexible and can incorporate various LLMs as well as a range of context sizes, i.e., different numbers of examples.

The proposed approach aims to create sophisticated communication interfaces between intelligent autonomous agents operating with symbolic representations and hu-

mans communicating in natural language. While our focus was on the specific task of translating natural language arguments into computational representations, the proposed approach also bridges existing argumentation-based frameworks implemented as core components of agents and natural language argumentation. Consequently, it directly moves towards the development of communication interfaces where humans and agents can engage in argumentation-based dialogues.

To evaluate our approach, we solicited feedback from experts in the fields of knowledge representation and argumentation. They reviewed 100 arguments generated by our approach, which were based on 10 different argumentation schemes: 5 classified as complex and 5 as simple schemes. In this first evaluation, we provided only one example as context to the Mixtral8x7B model specifically. Our findings show that this particular model is able to perform the task efficiently, particularly in translating simple argument structures. The model's performance might improve with a larger context (a greater number of examples), and this investigation is part of our ongoing work. In this direction, we intend to propose an interactive evaluation where we increase the context provided to various LLMs. This evaluation would determine the optimal context size (number of examples) required for each LLM currently available. While we observed good results using the Mixtral8x7B model with only one example of the task, our preliminary results also show that increasing the number of examples improves the model's performance in translating more complex argument structures.

## References


[1] Zhao WX, Zhou K, Li J, Tang T, Wang X, Hou Y, et al. A survey of large language models. arXiv preprint arXiv:230318223. 2023.

[2] Atkinson K, Baroni P, Giacomin M, Hunter A, Prakken H, Reed C, et al. Towards artificial argumentation. AI magazine. 2017;38(3):25-36.

[3] Akata Z, Balliet D, de Rijke M, Dignum F, Dignum V, Eiben G, et al. A Research Agenda for Hybrid Intelligence: Augmenting Human Intellect With Collaborative, Adaptive, Responsible, and Explainable Artificial Intelligence. Computer. 2020;53(8):18-28.

[4] Chang Y, Wang X, Wang J, Wu Y, Yang L, Zhu K, et al. A survey on evaluation of large language models. ACM Transactions on Intelligent Systems and Technology. 2023.

[5] Achiam J, Adler S, Agarwal S, Ahmad L, Akkaya I, Aleman FL, et al. GPT-4 technical report. arXiv preprint arXiv:230308774. 2023.

[6] Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in Neural Information Processing Systems. 2020;33:9459-74.

[7] Chen J, Lin H, Han X, Sun L. Benchmarking large language models in retrieval-augmented generation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38; 2024. p. 17754-62.

[8] Feng Z, Feng X, Zhao D, Yang M, Qin B. Retrieval-generation synergy augmented large language models. In: IEEE International Conference on Acoustics, Speech and Signal Processing; 2024.

[9] Gunning D. Explainable Artificial Intelligence (XAI). Defense Advanced Research Projects Agency (DARPA), nd Web. 2017.

[10] Panisson AR, Engelmann DC, Bordini RH. Engineering explainable agents: An argumentation-based approach. In: International Workshop on Engineering Multi-Agent Systems. Springer; 2021. p. 273-91.

[11] Maudet N, Parsons S, Rahwan I. Argumentation in Multi-Agent Systems: Context and Recent Developments. In: Maudet N, Parsons S, Rahwan I, editors. ArgMAS. vol. 4766 of Lecture Notes in Computer Science. Springer; 2006. p. 1-16.

[12] Walton D. Argumentation schemes for presumptive reasoning. Routledge; 1996.

[13] Walton D, Reed C, Macagno F. Argumentation Schemes. Cambridge University Press; 2008.

[14] Toniolo A, Cerutti F, Oren N, Norman TJ, Sycara K. Making Informed Decisions with Provenance and Argumentation Schemes. Proceedings of the Eleventh International Workshop on Argumentation in Multi-Agent Systems, 2014.; 2014. .

[15] Parsons S, Atkinson K, Haigh K, Levitt K, Rowe PMJ, Singh MP, et al. Argument schemes for reasoning about trust. Computational Models of Argument: Proceedings of COMMA 2012. 2012;245:430.
[16] Tolchinsky P, Atkinson K, McBurney P, Modgil S, Cortés U. Agents deliberating over action proposals using the ProCLAIM model. In: International Central and Eastern European Conference on Multi-Agent Systems. Springer; 2007. p. 32-41.
[17] Panisson AR, Ali A, McBurney P, Bordini RH. Argumentation Schemes for Data Access Control. In: Computational Models of Argument (COMMA); 2018. p. 361-8.
[18] Prakken H. An Abstract Framework for Argumentation with Structured Arguments. Argument and Computation. 2011;1(2):93-124.
[19] Panisson AR, Bordini RH. Uttering only what is needed: Enthymemes in multi-agent systems. In: International Conference on Autonomous Agents and MultiAgent Systems; 2017. p. 1670-2.
[20] Panisson AR, Bordini RH. Argumentation Schemes in Multi-agent Systems: A Social Perspective. In: International Workshop on Engineering Multi-Agent Systems; 2017. p. 92-108.
[21] Modgil S, Prakken H. The ASPIC+ framework for structured argumentation: a tutorial. Argument & Computation. 2014;5(1):31-62.
[22] García AJ, Simari GR. Defeasible logic programming: Delp-servers, contextual queries, and explanations for answers. Argument & Computation. 2014;5(1):63-88.
[23] Panisson AR, McBurney P, Bordini RH. A computational model of argumentation schemes for multi-agent systems. Argument & Computation. 2021;12(3):357-95.
[24] Carrera Á, Iglesias CA. A systematic review of argumentation techniques for multi-agent systems research. Artificial Intelligence Review. 2015;44:509-35.
[25] de Sousa LHH, Trajano G, Morales AS, Sarkadi S, Panisson AR. Using Chatbot Technologies to Support Argumentation. In: International Conference on Agents and Artificial Intelligence. SciTePress. 2024.
[26] Kim J, Lee JH, Kim S, Park J, Yoo KM, Kwon SJ, et al. Memory-efficient fine-tuning of compressed large language models via sub-4-bit integer quantization. Advances in Neural Information Processing Systems. 2024;36.
[27] Ferreira CEA, Engelmann DC, Bordini RH, Carbonera JL, Panisson AR. A Knowledge Base of Argumentation Schemes for Multi-Agent Systems. In: International Conference on Enterprise Information Systems (ICEIS); 2024. p. 1-15.
[28] Engelmann DC, Cezar LD, Panisson AR, Bordini RH. A conversational agent to support hospital bed allocation. In: Brazilian Conference on Intelligent Systems. Springer; 2021. p. 3-17.
[29] Jiang AQ, Sablayrolles A, Roux A, Mensch A, Savary B, Bamford C, et al. Mixtral of experts. arXiv preprint arXiv:240104088. 2024.
[30] Likert R. A technique for the measurement of attitudes. Archives of psychology. 1932.
[31] Krosnick JA. Maximizing questionnaire quality. Measures of political attitudes. 1999;2:37-58.
[32] Gwet KL. Handbook of inter-rater reliability: The definitive guide to measuring the extent of agreement among raters. Advanced Analytics, LLC; 2014.
[33] Landis JR, Koch GG. The measurement of observer agreement for categorical data. Biometrics. 1977:159-74.
[34] Al Zubaer A, Granitzer M, Mitrović J. Performance analysis of large language models in the domain of legal argument mining. Frontiers in Artificial Intelligence. 2023;6.
[35] Hidayaturrahman, Dave E, Suhartono D, Arymurthy AM. Enhancing argumentation component classification using contextual language model. Journal of Big Data. 2021;8:1-17.
[36] Walton D, Macagno F. A Classification System for Argumentation Schemes. Argument and Computation. 2015;6(3):219-45.
[37] Wagemans J. Constructing a periodic table of arguments. In: Argumentation, objectivity, and bias: International Conference of the Ontario Society for the Study of Argumentation, Windsor; 2016.
[38] Lawrence J, Reed C. Argument Mining Using Argumentation Scheme Structures. In: COMMA; 2016. p. 379-90.
[39] Xie Y, Yu C, Zhu T, Bai J, Gong Z, Soh H. Translating natural language to planning goals with large-language models. arXiv preprint arXiv:230205128. 2023.
[40] Pan L, Albalak A, Wang X, Wang WY. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. arXiv preprint arXiv:230512295. 2023.