From Natural Language to Ontology Graphs in LEMP

Sedrak Grigoryan
IIAP NAS RA
Yerevan, Armenia
e-mail: sedrak.grigoryan@iiap.sci.am

Tigran Shahinyan
IIAP NAS RA
Yerevan, Armenia
e-mail: tigranshahinyan@gmail.com

***Abstract*—The work outlines our vision and implementation details for converting expert knowledge in natural language into an ontology graph (OG). The approach is used for step-by-step learning from experts by Reproducible Game (RG) Tree Solver in Learning Expert Meaning Processing (LEMP). Having an OG gives the advantage of easier knowledge integration and reasoning using tools like the SPARQL query language. Our approach is based on a combination of NLP and Transformers, i.e., spaCy and BERT; and Semantic Web, i.e., RDF, OWL and SWRL. The proposed solution consists of the following steps: (1) Preprocessing text for named entity recognition, (2) Contextual understanding and relation extraction, (3) Mapping extracted triples into new layers of knowledge in OWL/RDF, and (4) Defining and extracting rules and mapping them into SWRL.**

***Keywords*—Ontology, artificial intelligence, nlp.**

I. Introduction

- Motivation: OGs are structured representations that model knowledge as interconnected nodes and edges, clearly capturing entities, their attributes, and the relationships between them. These graphs support precise, machine-readable knowledge encoding that enables automated reasoning, consistency validation, and inferencing. In the context of the RG Solvers, OGs are particularly valuable for representing hierarchical and temporal knowledge structures. They allow an expert to pass structured knowledge step-by-step, validate logical consistency of game mechanics, and reason over evolving strategies and actor behaviors.
- Semantic Web technologies overview: Resource Description Framework (RDF), Web Ontology Language (OWL), Semantic Web Rule Language (SWRL) [1, 2, 3]. These foundational technologies provide the formal underpinnings for representing and reasoning over structured knowledge.
    - RDF is the fundamental data model for expressing facts as subject-predicate-object triples. It provides a simple yet powerful mechanism for modeling OGs by explicitly representing relationships between entities. RDF enables the integration of heterogeneous data sources, supports reasoning through its graph structure, and is easily extendable with standards like RDFS and OWL. Its triple-based structure mirrors real-world conceptual models, making it highly suitable for dynamic domains such as RGs. It enables modular, linkable representations of entities and relationships in an RG domain.
    - OWL is an ontology description language designed for creating complex ontologies. It is formally defined as an RDF vocabulary, which means that OWL ontologies are represented using RDF triples and can be processed with RDF-compatible tools. OWL extends RDF by providing additional constructs for defining classes, properties, individuals, and relationships, along with logical axioms and constraints. OWL supports richer expressions such as disjoint classes, cardinality constraints, and transitive properties, which are not directly expressible in plain RDF. While RDF provides the foundational triple-based syntax and data model, OWL supplies the vocabulary and formal semantics necessary for expressive ontological modeling. OWL ontologies are typically serialized in RDF/XML or other RDF-compatible syntaxes.
    - SWRL addresses a crucial limitation of RDF/OWL by enabling the representation of behavioral and rule-based knowledge. While RDF/OWL is powerful for defining static knowledge - such as hierarchies, types, and logical class relationships - it lacks the expressivity needed for conditional logic and dynamic rule modeling. SWRL complements OWL by allowing the formulation of rules in the form of Horn-like logical implications that operate over individuals and properties within an ontology. These rules are essential for modeling behavior, dependencies, or outcomes that depend on specific conditions. SWRL is based on Datalog, a declarative logic programming language used for deductive databases. Both languages share a similar rule structure and operate on known facts to derive new knowledge. However, SWRL is designed specifically for integration with OWL ontologies and uses the same RDF-based syntax, enabling seamless reasoning over structured semantic data. Together, OWL, RDF, and SWRL allow for a robust, machine-interpretable, and extensible semantic representation of expert knowledge.
- Challenges in translating natural language RG texts into OG representation: Translating natural language texts about RG into OG representations poses several challenges rooted in both linguistic ambiguity and knowledge formalization.

- **Ambiguity and Vagueness**: Natural language is inherently ambiguous. Terms like "advantage," "conflict," or "strategy" may lack fixed boundaries or have multiple interpretations depending on context.
- **Implicit and Contextual Knowledge**: Part of the knowledge is implicit and should be inferred from explicitly defined triples and rules. RDF/OWL reasoners allow for inferring explicit knowledge but it's still a challenging task.
- **Mapping Lexical Items to Ontological Concepts**: Building accurate and consistent mappings between natural language terms and formal ontology classes or properties is labor-intensive and error prone.
- **Scalability and Automation**: Creating OGs manually from text does not scale and may be inaccurate. Automated pipelines (e.g., NLP, ML, LLMs) must balance coverage, accuracy, and maintainability.
- **Discourse Structure and Temporal Logic**: Descriptions of RGs often involve sequences of events or actions over time. Capturing this temporal and procedural logic within a static OG requires extended formalisms or rule-based augmentation (e.g., via SWRL).

These challenges necessitate interdisciplinary solutions combining computational linguistics, formal logic, and domain-specific modeling practices.

II. Cognitive Modeling Framework.

Our cognitive modeling approach draws from Jean Piaget's developmental psychology [4], enhancing object-oriented representations of reality with English-language classifiers and relationships, and aligns with inquiries into the origins of cognition in nature [5].

We propose combinatorial games with clearly defined utilities and strategic spaces as suitable models for Human-Universe (HU) problems. Specifically, we concentrate on RG problems and their solvers, defining minimal requirements:

- Presence of interacting actors (players, competitors).
- Defined actions performed by these actors.
- Specific timing for actions.
- Clearly described situations.
- Identifiable benefits for each actor.
- Rules or regularities governing how situations change post-action.

Many problems of practical significance can be formulated within the RG class, and are reducible to one another and, ultimately, to a unified kernel problem.

For the successful study of RG expert meaning processing, we reveal the following phases to overcome:

- **First Phase** - Leveraging expert meaning processing for kernel RG problem, say chess. We consider the interaction with natural language as utilizing tool for expert knowledge. Starting with RG and the above-mentioned background we conduct meaning processing research for the RG kernel chess problem, which includes
  - Preparation of RG Expert Classifier Repository for Chess. The phase involves developing and revising the repository of expert-level classifiers for chess [5, 6]. Classifiers are organized by complexity to facilitate learning by RG Solvers.
  - Advancement in RG Expert Learning by Complexity Levels. For each specified complexity level of expert classifiers, we refine and advance the learning capabilities of the RG expert model iteratively and level by level.
  - Verification of RG Solver. Confirming workability of at the time already RG Solver learned classifiers, particularly by demonstration of abilities of learning, identification of realities, meaning to text to meaning transition.
  - Enhancement of the Solver. We further develop RG Solvers to improve their ability to acquire increasingly complex expert meanings and enhance the quality of meaning-to-text and text-to-meaning transitions.
- **Broadening Scope to the Entire RG Class.** The research expands to the whole class of RG problems, aiming for a comprehensive learning of expert meaning processing.
- **Expanding to Natural Language**. This expands the successful results from earlier phases to the whole natural language content.

III. Modeling Reproducible Games in Ontology Graphs Described in RDF/OWL/SWRL

RGs involve structured interactions between actors (e.g., players), a set of allowed actions, and rules that define outcomes or transitions. Semantic Web technologies—specifically RDF, OWL, and SWRL—enable formal modeling of this logic within OGs.

- RDF is the base data model that expresses facts using subject–predicate–object triples. In the context of RGs, RDF allows granular representation of facts such as "Knight moves LShapedMove" or "Bishop captures DiagonalCapture".
- OWL adds expressiveness, allowing classes like King, Knight, or more abstract concepts like CaptureType, LongDiagonalMove, and LandingSquareCapture to be defined with formal semantics. It supports hierarchies and constraints, e.g., the class Bishop may be defined as a subclass of Piece that moves only along LongDiagonalMove.
- SWRL allows encoding complex conditional logic that can't be captured with OWL axioms alone. For example, a rule might state: If a Knight is located on a square adjacent in an L-shaped pattern to a square occupied by an OpponentPiece, then the Knight can capture that piece using an LShapedMove. These rules integrate declarative reasoning into the OG to simulate behavioral and temporal logic.

Core Classes from the Ontology:

- Knight, Bishop, King...: specific roles/actors within an RG scenario.
- MoveType, CaptureType... : action categories.
- LShapedMove, LongDiagonalMove, AdjacentCapture: specialized action types.

Key Object Properties:

- moves: maps an actor to its legal move type.
- captures: defines how an actor can execute a capture.
- color, row, column: encodes position and state data relevant to the actor or environment.

This layered approach allows RG designers to:

- Define a rich semantic vocabulary,
- Encode static knowledge (e.g., roles and actions) via OWL,
- Extend it with dynamic behavioral rules using SWRL,
- Reason over game states and transitions with SPARQL and RDF-reasoners.

IV. Translating Text to Ontology

The NLP pipeline in LEMP is designed to convert natural language descriptions of reproducible games (RG) into structured, machine-readable semantic representations in RDF/OWL/SWRL. This pipeline integrates rule-based and supervised learning techniques to ensure both precision and adaptability in the ontology population.

Rule-based NLP pipelines are deterministic systems that apply handcrafted linguistic rules to extract structured information from unstructured text. Unlike statistical or neural models, which learn patterns from large corpora, rule-based systems rely on domain expertise and predefined patterns such as token sequences, syntactic dependencies, and lexical cues. Some of the examples of rule-based NLP pipelines are: spaCy, Stanford CoreNLP, Apache UIMA (Unstructured Information Management Architecture), GATE (General Architecture for Text Engineering) [7, 8, 9, 10].

These pipelines typically include a series of sequential components:

- Tokenization – Splitting text into words or sentences.
- Part-of-Speech (POS) Tagging – Labeling each word with its grammatical role.
- Named Entity Recognition (NER) – Identifying domain-specific concepts like Player, Action, Location.
- Dependency Parsing – Analyzing syntactic relationships between words.
- Pattern Matching – Applying linguistic rules to detect phrases and extract relations.

spaCy is an open-source Python library for advanced NLP. In our solution spaCy is used for tokenization, POS tagging, syntactic dependency parsing, which are crucial for transforming natural language into structured formats suitable for ontology construction. spaCy allows creating tailored NLP pipelines that transform narrative game descriptions into structured RDF/OWL content. When paired with triple-generation libraries like rdflib, these outputs become building blocks for semantic reasoning in RG systems.

Fine-tuning BERT model on labeled data for, sequence labeling, relation extraction and triple classification [11]. To handle broader and more variable natural language, we fine-tune a pretrained BERT model on annotated RG datasets. Two fine-tuned models are created:

- NER-classification model based on BertForTokenClassification transformer for named entities recognition
- Relation Extraction model based on BertForSequenceClassification transformer for relation extraction (e.g., capturing (Knight, Bishop))
- In the initial iteration our two BERT models are fine-tuned provided the synthetic data extracted from initial OG. After each iteration the OG is enriched with new knowledge which is then extracted and feed as labeled data to fine-tune our BERT models. For efficient learning process on each iteration the new knowledge given in natural text by the expert must contain not too many new concepts and be based on previous knowledge.

The process is repeated for successive batches of new texts, resulting in continuous enrichment of both the OG and our BERT models.

[New Natural Language Text]
↓
spaCy → Tokenization, POS, Dependency Parsing
↓
BERT (NER + Relation Extraction)
↓
Ontology Enrichment (RDF Triple Expansion)
↓
Synthetic Data Generation (Verbalization + Annotation)
↓
BERT Fine-Tuning (NER and RE models updated)
↓
Repeat

V. CONCLUSION

In this paper we presented a unified approach for translating expert-level natural language content into structured ontological knowledge, enabling step-by-step learning in the LEMP framework. By leveraging the strengths of both NLP and Semantic Web technologies, we demonstrated a scalable and semantically rich pipeline for converting unstructured game descriptions into formal RDF/OWL ontologies augmented with SWRL rules. The combined use of spaCy for linguistic preprocessing and fine-tuned BERT models for named entity recognition and relation extraction allows for accurate and context-aware ontology population. Nevertheless, the transformation from natural language into RG remains a challenging task with many obstacles.

Our multi-phase modeling strategy begins with a kernel domain such as chess and generalizes toward broader classes of RG. This process supports the progressive learning of expert meanings, validating acquired knowledge through ontology reasoning and rule execution. The resulting OG not only encodes static domain knowledge but also dynamic behavioral logic, creating a powerful foundation for interpretability, verification, and semantic enrichment in AI systems that aim to learn and reason like human experts.

Future development will focus on three key directions. First, we aim to enhance the fine-tuning of BERT models for Named Entity Recognition and Relation Extraction by expanding annotated datasets, incorporating domain-specific linguistic patterns, and optimizing transformer configurations for greater accuracy and generalization across diverse RG scenarios.

Second, we plan to deepen the integration of SWRL rules within the OG to better capture complex behavioral logic and conditional reasoning. This includes automated rule generation from textual patterns, improved rule management, and runtime execution support for dynamic inference.

Third, we will explore the transformation of OG into Object-Oriented Programming (OOP) structures. This will facilitate the generation of executable game logic directly from ontological definitions, enabling seamless transitions from formal semantic representations to practical, interpretable software components. This OG-to-OOP mapping will strengthen the bidirectional bridge between high-level expert knowledge and its operational implementation.

REFERENCES

[1] O. Lassila, and R. Swick, "Resource Description Framework (RDF) Model and Syntax Specification", *W3C Recommendation*, 1999. https://www.w3.org/TR/1999/REC-rdf-syntax-19990222/

[2] D. L. McGuinness, and F. van Harmelen, "OWL Web Ontology Language Overview", *W3C Recommendation*, 2004. https://www.w3.org/TR/owl-features/

[3] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean, "SWRL: A Semantic Web Rule Language Combining OWL and RuleML", *W3C Member Submission*, 2004. https://www.w3.org/Submission/SWRL/

[4] J. Flavell, "The developmental psychology of Jean Piaget", *D. VanNostrand Company Inc*., Princeton, New Jersey, 1963. https://doi.org/10.1037/11449-000

[5] E. Pogossian, "Constructing Models of Being by Cognizing", Academy of Sciences of Armenia, Yerevan, 2020.

[6] E. Pogossian, M. Hambartsumyan, Y. Harutunyan, "A Repository of Units of Chess Vocabulary Ordered by Complexity of their Interpretations". *National Academy of Sciences of Armenia, IIAP*, research reports (in Russian), 1974-1980.

[7] M. Honnibal, I. Montani, "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing", to appear (2017)

[8] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The Stanford CoreNLP Natural Language Processing Toolkit". *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics*: System Demonstrations, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics 2014. http://dx.doi.org/10.3115/v1/P14-5010

[9] D. Ferrucci and A. Lally. 2004. "UIMA: an architectural approach to unstructured information processing in the corporate research environment". *Natural Language Engineering 10*, 3-4, 327-348(2004) https://doi.org/10.1017/S1351324904003523

[10] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan, "GATE: an architecture for development of robust HLT applications", *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 2002. https://doi.org/10.3115/1073083.1073112

[11] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova,"BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", 2018, arXiv:1810.04805v2, https://doi.org/10.48550/arXiv.1810.04805