Synergy between KG and LLM

Giovanni Magrone, 777341

Department of Computer Science, University of Bari Aldo Moro
g.magrone7@studenti.uniba.it

Abstract. The integration of Knowledge Graphs (KGs) and Large Language Models (LLMs) has emerged as a transformative approach to enhance AI systems' factual accuracy, reasoning capabilities, and contextual grounding. While LLMs excel in generating fluent and contextually rich text, they often suffer from biases, factual inconsistencies, and limitations in complex reasoning. Conversely, KGs provide structured, verifiable knowledge representations that can anchor LLM outputs to reliable facts.
This paper explores two synergistic integration paradigms:

KG-enhanced LLMs: Augmenting LLMs with structured knowledge through techniques like Retrieval-Augmented Generation (RAG) and graph-augmented training, improving factual grounding and reducing hallucinations.

LLM-enhanced KGs: Leveraging LLMs to populate, refine, and correct KGs by generating missing triples, suggesting relationships, and resolving incompleteness.

We further propose a hybrid framework for automated KG construction from unstructured text, combining LLM-based extraction (Meta-Llama-3-8B) with iterative graph refinement. Evaluated on Italian Public Administration documents, our approach demonstrates:

1)Domain adaptability to specialized language (e.g., legal texts).
2)Iterative self-correction to minimize errors and hallucinations.
3)Scalability through parallel processing and robust error recovery.

Results show a good results in generating relevant triples from the unstructured text. The synergy of KGs and LLMs unlocks new potential for question answering, fact-checking, and multi-hop reasoning, while future work targets real-time KG updates and distributed computation. This work bridges the gap between unstructured language generation and structured knowledge, giving the way for more reliable, knowledge-aware AI systems.

1 Introduction and Motivations

The integration of Knowledge Graphs (KGs) with Large Language Models (LLMs) represents a new powerful approach to solve some of the most famous limitations with artificial intelligence systems. While LLMs, such as GPT-4, excel in generating coherent and contextually rich text given a prompt, they often have to face challenges related to factual accuracy, reasoning capability, and the ability to manage domain-specific or less common knowledge. This comes from their needs to rely on vast corpora of unstructured data, which, even if it's a various text, those are prone to biases, inaccuracies, and full of outdated information.

On the other side, Knowledge Graphs offer structured, semantic representations of real-world entities and their relationships, providing a reliable and a more verifiable source of factual knowledge. By integrating KGs with LLMs, it can be possible to merge the generative and language processing capabilities of LLMs with the grounded, structured knowledge offered by the differents kinds of KGs.

This mixed synergy not only increases the factual accuracy of AI outputs but also makes possibile to introduce an advanced reasoning capabilities for the AI, enabling in this way, a more affordable and context-sensitive applications. This is motivated by the potential of this integration, this work explores the techniques, challenges, and opportunities of combining KGs and LLMs, highlighting their potential in different domains such as question answering, fact-checking, and complex reasoning tasks.

2 State of the Art

The combination of Knowledge Graphs (KGs) and Large Language Models (LLMs) represents a significant branch in improving the accuracy, reasoning, and factual grounding of language generation systems. While LLMs like GPT-4 and similar models excel in generating a lot of fluent and contextualized text, they sometimes produce information that is imprecise, outdated, or even incorrect. This limitation arises because LLMs are first trained on large corpora of unstructured text and do not have direct access to structured, verifiable knowledge sources like KGs and others structures.

In contrast, KGs offer a well-organized, semantic representation of real-world entities, relationships, and facts, which can serve as an essential grounding layer for LLMs. By integrating KGs with LLMs, we would like to bridge the gap between unstructured language generation and structured, factual knowledge. This paper explores two main key directions in this technique's integration: KG-enhanced LLMs and LLM-enhanced KGs are the two solutions that we are going to investigate, focusing on the main architectural challenges and the main proposed solutions in making these technologies working together.

2.1 KG-Enhanced LLMs

Overview of KG-Enhanced LLMs KGs are essentials to improve the factual accuracy and reasoning abilities of LLMs by giving structured and verifiable knowledge to the model. LLMs, even if they are well able to generate text, are limited in the ability to produce accurate and consistent text, moreover when dealing with specialized or less common knowledge.

Knowledge Graphs, such as Wikidata, gives a formal representation of entities and their relationships, which can improve the model's fact-checking and reasoning capabilities [1].

This allows for more accurate answers and more correct text generation. This integration is particularly useful for tasks such as question answering and fact verification, where accessing structured knowledge is essential.

Techniques for KG-Enhanced LLMs

- **Graph-Augmented Training:** One approach to integrating KGs with LLMs is through graph-augmented training. This method involves enhancing the training data with structured knowledge from the KG. The LLM can process this data alongside text to create more factually grounded outputs. Techniques such as entity linking (mapping text mentions to KG nodes) and knowledge embedding (embedding KG entities and relationships in vector spaces) are widely used in this process. By infusing KG information directly into the model during training, LLMs can learn to utilize the structure and relationships inherent in the KG [12] [10].
- **Retrieval-Augmented Generation (RAG):** A more dynamic approach to increase LLMs' performance is Retrieval-Augmented Generation (RAG). In this method, the LLM queries an external KG at inference time to get relevant facts, which are then used to guide text generation. This approach ensures that LLMs have access to accurate, up-to-date information from the KG during generation, instead of relying only on their pre-trained knowledge.
  RAG frameworks improve the model's factual accuracy by dynamically embedding external knowledge while generating text. The Retrieval-augmented Generation (RAG) method allows LLMs to focus on the most relevant information from the KG, rather than relying on static knowledge embedded in their training [7] [6]. By using external KGs, RAG improves the overall contextual grounding of LLM outputs and reduces the likelihood of hallucinations in the generated text [7].
- **Fine-Tuning with KG Data:** Another effective method for integrating KGs with LLMs is fine-tuning. Here, LLMs are fine-tuned on task-specific datasets that include both natural language text and KG triples (subject-predicate-object). Fine-tuning allows the model to learn how to incorporate structured knowledge from the KG for specific tasks, such as answering questions about entities or generating more factually accurate summaries [13] [15]. This process also enables the model to refine its ability to generate contextually relevant responses based on the structure of the KG, ensuring that the information output aligns with verified knowledge and improves the model's accuracy in specialized domains [13].

Challenges and Solutions

- **Semantic Alignment:** One of the first challenges in integrating KGs with LLMs is the semantic alignment between the continuous vector representations used by LLMs and the discrete symbolic data found in KGs.
  LLMs are typically trained on large, unstructured text corpora, which results in continuous embeddings. In contrast, KGs use a more discrete and structured format, where entities and relationships are explicitly defined. Bridging this gap needs sophisticated methods to align these different types of representations, such as using Graph Neural Networks (GNNs) to map the structure of the KG to the continuous vector space used by the LLM .
- **Real-Time Knowledge Integration:** We know that KGs are usually updated with new facts or entities, and integrating this information into LLMs in real-time can be challenging. Training the entire model again every time the KG is updated would be computationally expensive and inefficient. Instead, the integration occurs dynamically, with the model querying the KG for the most updated information during inference. This ensures that LLMs can work with the most current knowledge without requiring a full retraining process.
- **Resource Demands:** Integrating Knowledge Graphs (KGs) with Large Language Models (LLMs) requires substantial computational power, particularly for tasks like pre-training and fine-tuning. This process demands high-performance hardware like GPUs or TPUs, along with large memory capacities. The combination of training the LLM on vast text corpora and embedding intricate graph structures further intensifies the computational needs, making it challenging to perform this integration in resource-limited or time-sensitive environments .
- **Privacy Risk:** The inclusion of sensitive data within Knowledge Graphs—such as personal or domain-specific information—raises privacy concerns when integrated with LLMs. There's a risk that confidential details may be inadvertently revealed through model-generated outputs, especially if privacy-preserving methods are not in place. To mitigate this, it's essential that systems comply with data protection regulations and adopt techniques like differential privacy to protect sensitive information during the integration process .
- **Difficulty in Graph-Based Reasoning:** While KGs provide a well-structured knowledge framework, using this structure for reasoning and inference in LLMs presents challenges. KGs contain nodes and edges representing complex relationships, which are harder to integrate into LLMs compared to text data. The integration of graph structures into language models requires advanced encoding methods that preserve both local and global graph properties, enabling deep reasoning over these interconnected relationships .

By addressing these key challenges, future research can develop more efficient, secure, and reliable systems that combine the strengths of Knowledge Graphs and Large Language Models, allowing us to build more advanced AI applications.

2.2 LLM-Enhanced KGs

Overview of LLM-Enhanced KGs In addition to increase the performance using both LLMs with KGs, there is a significant potential for LLMs to enhance Knowledge Graphs. Specifically, LLMs can be used to populate KGs, correct errors, and expand knowledge in cases where the KG is incomplete or outdated. One of the major challenges with KGs is that they are often incomplete or lack sufficient details for certain entities or relationships. LLMs, with a huge language knowledge, can help generate new triples or expand existing ones to improve the coverage of the KG.

Techniques for LLM-Enhanced KGs

- **Entity Linking and Alignment:** One of the foundational tasks in integrating LLMs with KGs is entity linking. This process involves identifying entities in textual data and aligning them with corresponding nodes in the KG. By doing so, LLMs:

  - Increase their understanding of entities and relationships within the KG.
  - Enable more accurate text generation by retrieving additional entity-related facts from the KG.
  - Facilitate error correction in KGs by cross-referencing the linked entities with more reliable information sources.

  For instance, LLMs like GPT-4 can use Named Entity Recognition (NER) to identify entities in text, followed by linking them to the KG's existing nodes, thereby enriching the graph with new insights or relationships .
  This process allows the LLM to increase its understanding of entities and their relationships within the KG. This alignment also enables the LLM to retrieve additional relationships about those entities from the KG to improve the accuracy of generated text or responses [5].

- **Generate-on-Graph (GoG):** The Generate-on-Graph (GoG) framework is an innovative method for leveraging LLMs in incomplete KGs. In traditional Knowledge Graph Question Answering (KGQA) tasks, an incomplete KG can severely limit the model's performance. GoG addresses this by using LLMs for a dual role:
  - **As search agents:** LLMs extract relevant triples from the KG based on queries.
  - **As knowledge generators:** LLMs create new factual triples dynamically when gaps are identified in the KG [5].

This hybrid approach significantly improves KGQA performance, especially when dealing with incomplete knowledge graphs, by allowing the model to generate missing knowledge dynamically [5].

2.3 Cross-Domain Synergies: Combining KG-Enhanced LLMs and LLM-Enhanced KGs

- **Hybrid Architectures:** The true potential of KGs and LLMs lies in hybrid architectures that merge their strengths. These systems process unstructured text with LLMs while leveraging the structured reasoning capabilities of KGs. Advanced methods include:
  - **Multi-Stream Models:** Architectures that simultaneously handle structured and unstructured data for tasks such as reasoning, question answering, and text generation.
  - **KG Neural Networks (KGNNs):** Neural models designed to operate on KG structures, improving reasoning and fact retrieval.
- **Key Innovations:**
  - **Knowledge Distillation:** LLMs extract essential knowledge from unstructured text and translate it into structured formats suitable for KG integration.
  - **Efficient Fact Retrieval:** LLMs query large-scale KGs while maintaining computational scalability, ensuring fast and reliable access to relevant facts.
  - **Complex Question Answering (CQA):** CQA addresses questions requiring multi-step reasoning and knowledge integration. Combining LLMs and KGs enhances this process by linking natural language processing (via LLMs) with structured information from KGs. A hybrid model uses KGs to provide structured contexts for logical reasoning while LLMs handle language-based subtleties. For example, recent frameworks such as "Retrieve-Rewrite-Answer" focus on retrieving relevant KG subgraphs and integrating these into LLM workflows for enhanced reasoning [14].
  - **Multi-Hop Question Generation (MHQG):** MHQG involves creating questions that necessitate reasoning across multiple entities or relations, often relying on KGs for structured guidance. Techniques like relation path prediction in multi-hop retrieval use KGs to maintain logical consistency while LLMs ensure natural language fluency. These approaches improve AI training, such as generating test questions for models in science and engineering contexts [14].
  - **Knowledge Graph Chatbots:** KG chatbots enhance conversational AI by embedding KG reasoning capabilities, ensuring answers are context-aware and accurate. KGs provide structured grounding for answers, while LLMs refine responses in user-friendly language. Advanced methods integrate multi-hop reasoning for questions requiring deep exploration of interconnected knowledge [14].
- **Dynamic Growth and Enhancement:** These hybrid systems enable dynamic growth and enhance both KG accuracy and LLM-generated outputs. For example, chatbots built on KG-enhanced LLMs can provide real-time, domain-specific insights while updating KGs based on user interactions.

2.4 Evaluation Metrics

LLMs with KGs metrics For LLMs integrated with KGs, several metrics are used to evaluate how effectively the LLM utilizes the knowledge from the KG. Metrics like accuracy, ROUGE, and BLEU are essential for assessing performance in tasks such as question answering, summarization, or text generation, where the model needs to generate factually accurate and contextually relevant output. Fidelity and comprehension are particularly important in evaluating how well the model understands and integrates the graph-based knowledge during reasoning. Additionally, precision, recall, and F1-score are critical for tasks like entity recognition or classification, where the model needs to correctly identify and classify entities based on the information provided by the KG [5].

KG with LLMs metrics On the other hand, for KGs integrated with LLMs, the focus is more on the efficiency and effectiveness of using the KG during inference or training. Metrics like time cost, training time, and GPU use are important to assess the model's computational efficiency, especially in real-time or large applications where the KG needs to be queried dynamically. Latency-volume trade-off and mismatch rate are used to evaluate how the system manages the balance between processing time and data volume, ensuring smooth operation without sacrificing the quality of the output. Hits@k and exact match (EM) are key in retrieval-based tasks, where the model's ability to rank and retrieve the most relevant knowledge from the KG is evaluated [5].

2.5 Conclusion

Future Works Future research should focus on improving the integration of LLMs with graph databases and complex data structures. Key areas include refining encoding techniques to capture relationship details in graphs, enhancing LLM accuracy and contextual relevance. Domain-specific graph databases can be better integrated with LLMs through tailored adaptation algorithms[5].

Additionally, scalable, real-time learning models are needed to allow LLMs to quickly adjust to new data. Addressing scalability and performance challenges through methods like model pruning will help maintain efficiency as data volumes grow.

Finally, interdisciplinary approaches combining AI, NLP, and database technologies can support real-time learning, efficient data management, and seamless knowledge transfer, unlocking further potential for integrated systems.

Conclusion Integrating Knowledge Graphs with Large Language Models offers a interesting path forward for creating more fact- aware and reliable AI systems. Whether through KG-enhanced LLMs or LLM-enhanced KGs, the combination of these two technologies allows for more accurate text generation, improved reasoning, and the ability to work with up-to-date, structured knowledge. As the integration techniques improve and new challenges raise and we need to find new way to deal with them, this hybrid approach will become a milestone for many natural languages processing applications, including question answering, fact verification, and knowledge-driven reasoning.

3 Modeling and Implementation: Frameworks for KG Extraction from Unstructured Text

Following a preliminary investigation, we resolve to examine frameworks capable of automating the construction of Knowledge Graphs (KGs) from unstructured text. We conduct an analysis of various available tools, including **KGGEN, GraphRAG, OpenIE**, emphasizing their architectural features and culminating with a **hybrid custom approach**.

3.1 Frameworks Overview

Recent studies emphasize the complementary relationship between LLMs and KGs for extracting structured knowledge [11]. LLMs are adapt at understanding context, while KGs offer a reliable factual basis [9]. Methods like **OpenIE** utilize rule-based parsing, whereas **GraphRAG** and **KGGEN** employ LLMs to enhance semantic richness [3]. Selecting a framework involves balancing the trade-offs between precision and scalability (e.g., OpenIE).[8].

3.2 KG-GEN

Architecture: KGGEN leverages large language models (LLMs) to create triples through prompt engineering, effectively transforming unstructured text into a network of nodes and edges [11]. Its primary strength is in probabilistic relation extraction. The framework's main novelty is its two-stage knowledge extraction approach:

Neural Extraction Phase:

- Employs transformer-based language models (such as GPT-4 or architectures detailed in the documentation) for extracting information across various domains.
- Uses prompt engineering to implement constrained decoding, ensuring output in a JSON format.
- Utilizes the model's pre-existing knowledge to address coreferences and deduce implicit connections.

Symbolic Normalization Phase:

- Applies **graph clustering** algorithms to merge semantically equivalent entities (e.g., "Apple Inc." and "Apple")
- Uses **relation embedding spaces** to cluster similar predicates (e.g., "founded" $\rightarrow$ "was founder of")
- Implements **hierarchical agglomerative clustering** on entity-relation subgraphs(e.g "Cupertino" $\rightarrow$ "California" $\rightarrow$ "USA")[8].

Clustering Phase Methodology:
The framework's clustering module operates with three principal mechanisms:

1) Entity Resolution:

- Computes pairwise similarity scores
- Implements Markov Chain Clustering (MCL) for community detection

2) Relation Harmonization:
Projects extracted predicates into a learned embedding space, applies density-based clustering (DBSCAN) to identify relation clusters,maintains an evolving ontology of standardized predicates

3) Subgraph Consolidation:
Identifies isomorphic subgraphs between document boundaries. Uses Weisfeiler-Lehman graph kernels for structural similarity assessment Implements minimum description length (MDL) principles for graph compression

Technical Limitations

While KGGEN's approach shows promise, several challenges persist:

- **Computational Complexity:** Requires GPU acceleration for large-scale knowledge graphs (>100k nodes)
- **Semantic Drift:** Cross-document clustering can lead to over-aggressive merging of distinct entities, requires careful tuning of similarity thresholds
- **Temporal Dynamics:** Static clustering approaches struggle with evolving entity representations, lacks incremental update mechanisms for streaming data

3.3 GraphRAG

GraphRAG introduces a novel two-level hierarchical architecture for knowledge extraction that fundamentally differs from sequential approaches like KGGEN:

Document-Level Graph Construction: Processes individual documents through embedding-based clustering
Generates local graph structures capturing intra-document relationships.

Corpus-Level Graph Integration:
Implements cross-document node alignment using approximate nearest neighbor search

Builds global relationships through graph neural network propagation The system's key innovation lies in its topological sorting of knowledge, where lower-level entity clusters form the foundation for higher-order reasoning.

Advantage: Scales to large corpora but requires vector DBs [9].

3.4 OpenIE

OpenIE extracts simple facts (subject, predicate, object) from text using grammatical rules instead of machine learning. It analyzes sentence structure to find relationships but does not link them to a knowledge base or normalize different phrases (e.g., "created" vs. "invented" are treated as separate relations). While fast and flexible, it struggles with complex sentences and lacks deeper semantic understanding.

Key Features:

- Rule-based (no training data needed)
-
- Works on any text domain
-
- Outputs raw triples without filtering

Limitations:

- No meaning-based grouping of similar relations
-
- Cannot handle implied or indirect relationships well
-
- Best for simple, clear sentences

[2] [4]

3.5 Hybrid Approach: LLM-KG Integration Framework

Our hybrid approach combines the strengths of LLM-based knowledge extraction with iterative graph refinement, addressing key limitations of existing frameworks. The architecture implements three novel components:

We opted to run the LLM infrastructure locally with the Ollama application. Than, we downloaded the Meta-Llama-3.1-8B from HuggingFace. This iteration employs 128k tokens for context, roughly equivalent to 300-500 pages of text, depending on tokenization. It encompasses over 8 billion parameters. Additionally, it's **instruct**, meaning it is fine-tuned for instruction-following tasks. [1]

This is useful for long documents, extended conversations, or complex tasks requiring large inputs (e.g., legal contracts, research papers).

1) Adaptive Chunk Processing

- Implements paragraph-aware text segmentation respecting semantic boundaries
- Dynamically adjusts chunk sizes based on:
- Entity density (NER tag frequency)
- Lexical cohesion scores
- Document structure markers

2) Two-Phase Knowledge Extraction

- Initial Extraction: LLM generates structured JSON with nodes/edges
- Triple Normalization: Converts to canonical (subject, predicate, object) form

Implements robust JSON parsing with:

- Markdown stripping
- Error-tolerant bracket matching
- Schema Validation

3) Generate-on-Graph Refinement
Iterative knowledge completion through:

- Entity relation suggestion
- Plausibility Filtering
- Graph augmentation

Uses constrained prompting to:

- Maintain existing entity sets
- Suggest only novel relations
- Enforce consistent JSON output

Technical Advantages

- API rate limits (exponential backoff)
- Malformed LLM outputs
- Partial processing recovery

Semantic Coherence:

- Preserves discourse continuity through paragraph-bound chunking
- Maintains entity consistency across documents
- Supports multi-hop reasoning through iterative refinement

Scalability:

- Parallel file processing
- Memory-efficient graph aggregation
- Configurable refinement depth

Implementation Highlights
The system demonstrates several key improvements over baseline approaches:
Precision Enhancement

- Existing entity constraints
- Multi-iteration verification
- Confidence-based filtering

Recall Optimization

- Context-aware chunking
- Cross-document aggregation
- Implicit relationship suggestion

Practical Deployment
Implements production-ready features:

- Progress tracking
- Error isolation
- Partial result persistence

Supports:

- Batch processing
- Incremental updates
- Result visualization

Table 1. Comparative Benefits of Knowledge Graph Construction Approaches

Feature	Our Approach	KGGEN	GraphRAG	OpenIE
Iterative Refinement	✓	×	Partial	×
Error Recovery	✓	×	×	×
Cross-Document Links	✓	×	✓	×
Semantic Normalization	Partial	×	✓	×
No Training Data	✓	✓	×	✓

3.6 Experiment and results

The hybrid model was rigorously tested on Italian Public Administration datasets given by the university, this dataset was already a preprocessed one, organized in subfolders.

The experimental framework included:

- **Small-scale validation**: A selected subset (13 documents, 1 folder) was used for rapid prototyping and initial verification of the knowledge extraction pipeline, to check how it works and the given results.
- **Full-scale evaluation**: A more complete dataset package (40 documents, 3 folder) tested the system's scalability and robustness in a more realistic conditions to see how much is robust as a system

As demonstrated in Figures 1 and 4, the system achieved:

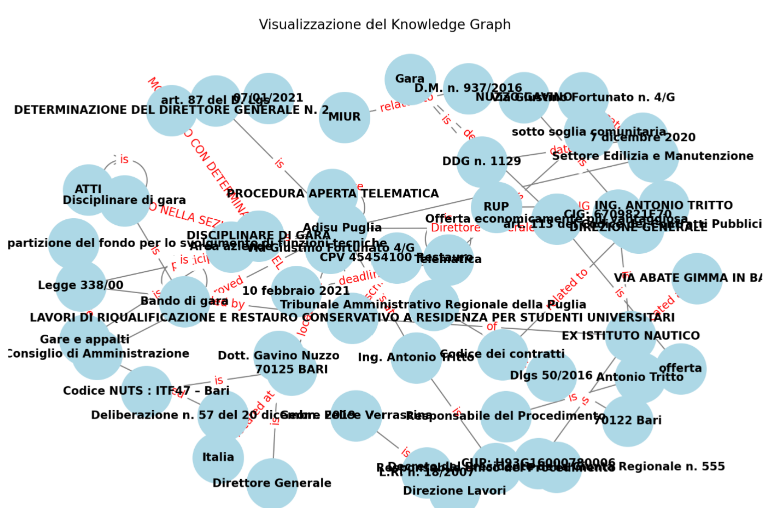

Fig. 1. Analysis over the first dataset

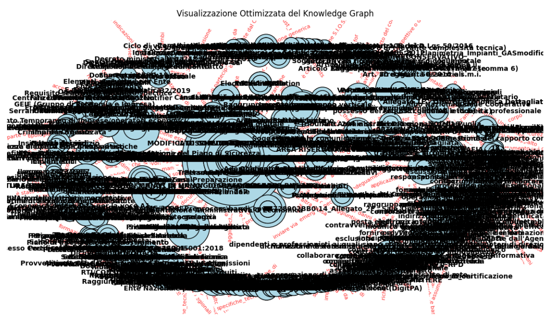

Fig. 2. Anlysis over the second dataset

Subsequently, the knowledge graph generated in the first step was compared against a reference dataset of correct tuples, referred to as the golden standard. This comparison enabled an evaluation of the system's effectiveness in accurately extracting semantic relationships.

To perform this evaluation, the **Jaccard index** was calculated. This similarity metric measures the intersection between the relations identified by the system and those present in the golden standard, relative to their union. The calculation also incorporated a syntactic similarity component, in order to account for potential lexical variations in the detected relations.

Table 2. Showing results of evaluations of the model.

Relation to analyze	16
Covered relation by our KG	4
Jaccard Index	0,25

Fig. 3. Golden standard triple

```
],
[
    "distributori automatici",
    "dispenses",
    "Bevande Calde"
],
```

Fig. 4. KG graph triple

The results show that the knowledge graph successfully identified many of the relationships expected in the golden standard and, in some cases, even generated new potentially relevant relations. However, the comparison also highlights areas for improvement, suggesting the need for additional testing, experimenting with different code configurations, and trying alternative models to assess variations in performance.

As shown in the fig.3, and fig.4, in the dataset there are several entities that from the semantics point of view should give a match, for this other approaches like bertSCORE or using a LLM to find the match can further improve the score of the Jaccard Index showing that the KG has been able to catch a better number of relations form the graph

Overall, considering the complexity of the task and the level of automation achieved, we are satisfied with the outcome: the system works and provides a solid foundation for future development.

3.7 Conclusions

The proposed methodology gives a new potential path for automated knowledge extraction from unstructured documents like these legal documents, with three core advantages:

- **Domain adaptability**: The architecture successfully handles the specialized language of PA tenders while remaining transferable to other regulatory domains
- **Self-correcting refinement**: The iterative knowledge augmentation mechanism reduces hallucination risks
- **Integration-ready output**: Generated metadata complies with standard RDF/OWL formats for immediate use in existing systems

Future research directions:

1. **Complete dataset processing**: Execution on the entire dataset, currently hindered by:
   - Hardware limitations (requires 128GB RAM)
   - Need for distributed computing architecture
2. **Standardized evaluation** using established metrics:
   - **Precision@K** (P@K): Top-K accuracy for relationship extraction
   - **Mean Reciprocal Rank** (MRR): Ranking quality of correct entities

  - **Hits@10**: Completeness of knowledge coverage
  - **Normalized DCG**: Semantic relevance scoring
3. **Incremental updating**: Development of dynamic KG maintenance protocols for:
  - Handling tender amendments
  - Temporal relationship tracking
  - Version-controlled knowledge updates

This work establishes foundations for:

- Automated analysis of multi-document datasets packages
- Real-time funding opportunity monitoring systems
- New knowledge platform for the Public Administrations

References

1. AI, M.: Llama 3: Open foundation and fine-tuned chat models. https://ai.meta.com/blog/meta-llama-3/ (2024), model cards and technical details for the 8B, 70B, and 128k-context variants.
2. Angeli, G., et al.: Leveraging linguistic structure for open domain information extraction. In: ACL (2015)
3. Borgo, R., et al.: Graphrag: Hierarchical kg indexing for llm reasoning. Microsoft Research (2023)
4. Etzioni, O., et al.: Open information extraction: The second generation. IJCAI (2011)
5. Ibrahim, N., Aboulela, S., Ibrahim, A., Kashef, R.: A survey on augmenting knowledge graphs (kgs) with large language models (llms): Models, evaluation metrics, benchmarks, and challenges. Discover Artificial Intelligence Review (2024)
6. Izacard, G., Lewis, P., Lomeli, M., Hosseini, L., Petroni, F., Schick, T., Dwivedi-Yu, J., Joulin, A., Riedel, S., Grave, E.: Few-shot learning with retrieval augmented language models. In: ACL (2022), https://aclanthology.org/2022.acl-long.225/
7. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.t., Rocktäschel, T., et al.: Retrieval-augmented generation for knowledge-intensive nlp tasks. NeurIPS (2020), https://arxiv.org/abs/2005.11401
8. Lockard, C., et al.: Deepke: Knowledge extraction with llms. AAAI (2023)
9. Pan, J., et al.: Unifying llms and kgs for enhanced reasoning. arXiv:2305.12302 (2023)
10. Peters, M., Neumann, M., Logan, R., Schwartz, R., Joshi, S., Singh, S., Smith, N.A.: Knowledge enhanced contextual word representations. EMNLP (2019), https://aclanthology.org/D19-1005/
11. Rebello, P., et al.: Rebel: Relation extraction by end-to-end language generation. EMNLP (2023)
12. Wang, X., Zhang, M., Chen, P., Wang, Y.: Kglm: Integrating knowledge graph structure in language models for link prediction. In: EMNLP (2021), https://aclanthology.org/2021.emnlp-main.787/
13. Yasunaga, M., Ren, H., Bosselut, A., Liang, P., Leskovec, J.: Deep bidirectional language-knowledge graph pretraining. NeurIPS (2022), https://arxiv.org/abs/2210.09338
14. Yu, D., Zhu, C., Yang, Y., Zeng, M.: Reasoning with language model and knowledge graph for question answering. AAAI (2023), https://arxiv.org/abs/2212.00093
15. Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., Liu, Q.: Ernie: Enhanced language representation with informative entities. In: ACL (2019), https://aclanthology.org/P19-1139/

A Appendix