Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 246 (2024) 2235–2243

Procedia Computer Science

www.elsevier.com/locate/procedia

28th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2024)

A Zero-Shot Strategy for Knowledge Graph Engineering Using GPT-3.5

Salvatore Carta[a], Alessandro Giuliani[a], Marco Manolo Manca[a], Leonardo Piano[a], Alessandro Sebastian Podda[a], Livio Pompianu[a], Sandro Gabriele Tiddia[a,*]

[a]*Department of Mathematics and Computer Science, University of Cagliari*
Palazzo delle Scienze, Via Ospedale 72, 09124, Cagliari, Italy

Abstract

In the recent digitization era, capturing, representing, and understanding knowledge is essential in countless real-world scenarios. Knowledge graphs emerged as a powerful tool for representing information through an adequately interconnected and interpretable structure in such a context. Nevertheless, generating proper knowledge graphs usually requires significant manual effort and domain expertise, resulting in graphs often affected by human subjectivity, limited scalability, or inability to capture implicit knowledge or handle heterogeneity. This paper proposes an innovative zero-shot strategy tailored to uncover reliable knowledge from text leveraging the recent highly effective generative large language models, with a particular focus on the GPT-3.5 model. Our proposal aims to create a suitable knowledge graph or improve existing ones by discovering missing qualitative triples. To assess the effectiveness of our methodology, we performed experiments on domain-specific datasets, confirming its potential for scalable and versatile knowledge discovery.

Keywords: Knowledge Engineering; Knowledge Graphs; Large Language Models.

1. Introduction

Nowadays, due to the evolution of technology, many real-world domains present an exponential increase in the volume of data. Despite the benefits such a scenario may convey, many challenges must be addressed. A central issue is the so-called *information overload*, i.e., when an excessive amount of information makes processing, analysis, and inference of meaningful insights extremely difficult. In such a context, proper knowledge representation is a crucial factor that may effectively support information extraction, organization, processing, and exploitation. Knowledge Graphs (KGs) represent a powerful structure to manage, access, understand, and utilize information effectively [9].

* Corresponding author. Tel.: +39-070-675-8705
E-mail address: sandrog.tiddia@unica.it

1877-0509
10.1016/j.procs.2024.09.573

They organize information into a graph format, wherein nodes represent entities, and edges denote relations between entities. The structured knowledge of KG offers several benefits in many Web-based services, e.g., in recommender systems [7], semantic search [21], healthcare and clinical tools [1], or finance [12].

Building a KG presents various hindrances. On the one hand, manual construction is extremely laborious and prone to human error. On the other hand, while capable of achieving excellent performance in some contexts, automated approaches are even dependent on extensive human annotations. They are usually limited to classify relationships only at the *sentence level*, losing information in broader contexts. Moreover, to ensure entity disambiguation, many approaches exploit external knowledge bases that are often not exhaustive in identifying all extracted entities. Nevertheless, manually curated and automatically generated KGs often lack completeness, containing only observed information and leaving numerous undiscovered missing connections.

To overcome the aforementioned limitations, we exploit the effectiveness of the latest generative Large Language Models (LLMs) in analyzing, processing, and understanding a noteworthy amount of natural language. Thus, our insight is not merely relying on relation extraction but to allow for better *semantification* by providing a precise description of entities and their predicates, facilitating future disambiguation.

LLMs can predict natural language elements in response to a given query, referred to as a *prompt*, that could be expressed as a question, instruction, or statement. Providing well-formulated prompts to LLMs can be highly useful in supporting the development of powerful NLP applications in many scenarios [17, 4, 22, 24]. In particular, we focus on *zero-shot* prompting, i.e., without the need for examples and fine-tuning, instead of relying on the common methods based on *few-shot* prompting, i.e., providing LLMs with a few examples of desired outputs. Indeed, on the one hand, providing a few examples may overfit the training data and perform much worse on unknown inputs [23]. On the other hand, zero-shot strategies permit building models that are more easily adaptable to new scenarios and computationally efficient. Recent works demonstrated several advantages of zero-shot methods against few-shot prompting in information extraction [20].

The main contributions of the paper can be summarized as follows :

- We propose a unified LLM *chain-of-prompts* method for Document-Level Triple Extraction.
- We engineered our prompts to recover meaningful information, including descriptions for entities and predicates and multiple entity types.
- We also evaluated and quantified LLM *hallucinations* in the information extraction scenario.

The rest of the paper is organized as follows: Section 2 reports the related work; in Section 3, we describe our methodology, whereas Section 4 shows the experimental results. Section 5 ends the paper with the conclusions.

2. Related Work

Automated KG generation, aimed at creating a structured knowledge representation from heterogeneous sources without any human effort, has been widely investigated in recent years. Traditional Knowledge Graph construction pipelines comprise several NLP techniques, such as Named Entity Recognition [6], Relation Extraction [16], and Entity Resolution [8]. For example, the pipeline of [10] combined co-reference resolution, named entities recognition, and Semantic Role Labeling to build a financial news KG. In [15], the authors presented an end-to-end KG construction system that exploits a Deep Learning-based predicate mapping model to map the extracted entities to the DBpedia namespace.

However, classical approaches present several weaknesses. On the one hand, they usually depend on specific ontologies or other semantic resources. Our proposal addresses this limitation, as we do not rely on external resources. On the other hand, numerous methods are often limited to a predefined set of entities and relationships, mainly annotated by humans to address the specific use case they semantically represent. Such approaches require extensive human manual annotation and, therefore, are difficult to scale up and generalize to out-of-domain contexts. An emerging strategy to address this issue is exploiting pre-trained language models [19, 11]. Technological advancements and increasing data availability have led to a severe escalation in developing Large Language Models (LMMs). New LLMs, such as GPT-3.5, have shown outstanding zero or few-shot Information Extraction capabilities, as proved by recent studies [2, 20, 14], in which ChatGPT performance have been systemically assessed in various tasks. Such stud-

ies pointed out that ChatGPT, in an Open Domain scenario, is particularly effective in Information Extraction (e.g., entity recognition or entity typing) but, conversely, is inadequate for more complex tasks, such as Relation Extraction. Although Wan et al. [18] addressed this problem through an In-Context Learning strategy, their method only extracts relations at the sentence level. It is highly dependent upon a gold standard to find the best demonstrations. Instead, our approach finds relations between long portions of text and is not dependent on any external resource.

3. Methodology

In the following, we describe our methodology, including detailing the pipeline step by step. The pipeline implementation and the generated data are available on a proper GitHub repository[1].

3.1. Overview

LLM Querying Strategy. Our proposal relies on integrating an innovative task-specific prompting strategy aimed at exploiting the capability of LLMs to analyze, elaborate, and generate human-like text. The challenge is to define well-formed prompts to properly structure knowledge and obtain:

- **A proper entity characterization**, aiming to explore further the simple identification of text spans representing an entity *mention*, enriching it with (i) an *entity label*, not necessarily corresponding to an exact text span, (ii) a proper entity *description*, and (iii) a list of *types* or *hypernyms*.
- **An appropriate characterization of triples and predicates**, representing a relation between two entities (i.e., *subjet* and *object*) with (i) a relevant predicate defined by a suitable label, not necessarily corresponding to an exact text span, and (ii) a general description of the relationship.

Our strategy may provide a more informative representation of the extracted concepts, enhancing the classical methods typically based on extracting entities or predicates corresponding to exact text segments from the input document. We deem that the descriptions and entity types generated by our approach provide a detailed context for each mention, leading us to an actual *semantification*.

Architecture. The proposed strategy may be represented by the high-level architecture in Figure 1, which depicts the main steps of the process.

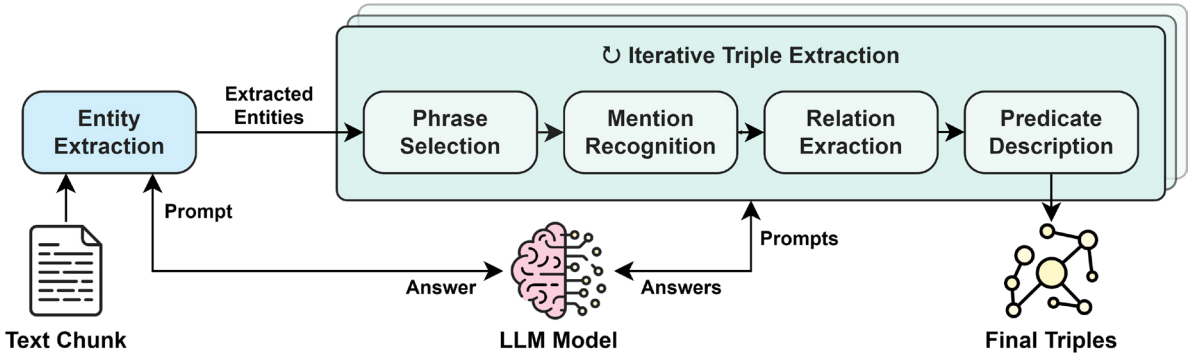

Fig. 1. High-level architecture

[1] https://github.com/SandroGT/KG-LLM-Prompting.git

Let us remark that, in this work, we adopt the GPT-3.5 model (in particular GPT-3.5-turbo-0301[2]) aiming for a deterministic setting[3]. Nevertheless, the reasoning required in each step could be performed by any sufficiently powerful LLM.

In detail, the process involves two main stages: *entity extraction*, which aims to identify relevant entities from the input text, and *iterative triple extraction*, which generates the final triples. The two-stage approach can limit LLMs *hallucinations*, as the triple generation adheres to only existing entities in the text. The following Sections detail the two aforementioned stages.

3.2. Entity Extraction

We query the LLM with a straightforward system prompt to identify relevant entities. This prompt briefly defines what we mean by "entity" and then requests the retrieval of entity mentions from the input text. We include the additional requirement of providing a description and a list of types for each mention, along with a specific output format. We denote as E the set of entities extracted. Let us point out that although the LLM probably "knows" what an entity is, in this context, preliminary studies showed that providing the aforementioned details guides toward a greater focus on concrete nouns and named entities better detailed in the text chunk.

3.3. Iterative Triple Extraction

The set of retrieved entities E is used as a constraint to extract the relations between the entities from each text chunk content T. Generating triples that explicitly refer to E is challenging for the LLM. When E is highly populated, the LLM fails at considering the complete set and generates triples involving entities not included in E. Although some of such triples can represent a correct statement, they are somewhat useless since they contain entities for which we do not have a description, which is an essential requirement of our approach.

To this end, we developed an iterative strategy that focuses on a different entity $e_i \in E$ at each iteration i. The aim is to simplify the task complexity at every iteration, making selecting only entities in E easier. We discuss the steps in each iteration in the following paragraphs.

Phrase Selection. To avoid the use of the whole text T, we aim to summarize the information of each entity e_i extracted by T by a target text excerpt T_i^G, which defines a smaller context than T, thus reducing the complexity of the following steps and increasing the reliability of the LLM. Among different techniques, including text summarization approaches, we bypass the explicit extraction of T_i^G. Indeed, we successfully tested a simple but effective solution, i.e., we consider the entity descriptions generated during the *Entity extraction* stage as T_i^G.

Mention Recognition. We aim to detect which entities e_i are mentioned in T_i^G, thus defining the subset $E_i^G \subseteq E$. In doing so, we instruct the LLM with a proper prompt to identify the mentions of entities in the specific text. We provide the complete list E and the text T_i^G, asking the LLM to reply with the same list E in which each entity e_i is annotated with a "yes/no" answer. The entities for which we have a "yes" answer define the set E_i^G of interest.

Relation Extraction. We perform the relation extraction using the narrowed content given by T_i^G and E_i^G, generated in the previous two steps. The LLM is queried with a suitable prompt, asking for the generation of entity relations between the entities in E_i^G that are expressed in T_i^G, representing them in the form of RDF triples, i.e., <subject, predicate, object>, and producing the set of triples R_i^G. The system prompt clarifies what we mean by an *expressive predicate*, guiding the LLM to generate predicates that effectively represent the relationship between the two entities without being too specific, as it would make the predicate hardly reusable and observable in other triples, aiming for a sort of predicate *canonicalization*.

[2] https://platform.openai.com/docs/models/gpt-3-5-turbo

[3] To aim for a deterministic behavior (i.e., to get the same message in response to the same input prompts), we set the GPT-3.5 API *temperature* parameter equal to zero.

Predicate Description. The final step is the generation of proper predicate descriptions. In doing so, we defined an appropriate prompt aimed at asking the LLM to return the description of each unique predicate. Each prompt embodies the underlying text excerpt T_i^G and the triples R_i^G generated from it. The model returns a list of <*predicate, description*> pairs.

4. Experiments

This section describes the experiments aimed at assessing the effectiveness of our approach, describing the selected datasets and evaluation metrics, and reporting the final results.

4.1. Datasets

Let us remark that our tool can extract entities and predicates along with enriched descriptions obtained from in-text knowledge. Therefore, such a task cannot be performed consistently if the text lacks the necessary details and contextual information, as often occurs with most available datasets, typically based on corpora composed of single sentences or short paragraphs. To this end, we built an adequate dataset, denoted as `ST`, by gathering entire webpages from the English version of the *Sardegna Turismo* website[4], a portal describing tourist destinations and points of interest in Sardinia, defining a set of 44 documents (888 sentences) covering a specific topic (culture and tourism). Moreover, to assess the KG enrichment capability of our approach, we randomly selected an additional set of annotated Wikipedia pages covering various topics from the `REBEL` dataset[5] [5] collecting 148 documents (2,803 sentences). Let us point out that we did not exploit the entire dataset to limit the GPT charges. Moreover, we selected a subset of 20 documents (denoted as `REBEL_20`) for manual evaluation, as it requires considerable human effort.

4.2. Evaluation

We conducted two types of evaluations, a human assessment (for `REBEL_20` and `ST` datasets) and an automated evaluation (for `REBEL` dataset), the latter aimed at assessing the amount and quality of additional information retrieved by our strategy.

4.2.1. Human Assessment

Human assessors annotated as *correct* or *incorrect* each *entity* (correct if it is relevant to the textual context and mentioned in the input text), *entity type* (correct if it captures the actual class and context of the entity), and *triple* (correct if the predicate label accurately expresses the relation and description and the two entities have been labeled as correct). Moreover, each correct entity and triple is labeled with a boolean annotation a_σ to evaluate whether GPT-3.5 retrieves the information from the text ($a_\sigma = true$) or draws on its knowledge ($a_\sigma = false$). Furthermore, the assessors identified a list of *missed entities*, i.e., relevant entities included in the input text but not retrieved by the model, by considering all entity types having at least two associated entities to create a reference schema. Also, for the `REBEL` dataset texts, the assessors introduced an additional boolean annotation a_γ to evaluate whether each GPT-3.5 entity and triple corresponds to a concept mentioned in the annotations of the dataset, even if in a different form ($a_\gamma = true$) or it is not ($a_\gamma = false$).

We derive classical confusion matrix entries from such annotations. Each correct component is a *true positive* (TP), an incorrect item is a *false positive* (FP), and a missed entity is a *false negative* (FN). To identify missing entities, we solely rely on human annotations, disregarding the `REBEL` annotations. The `REBEL` annotations may not align with the schema implicitly adopted by GPT-3.5, and it would be unfair to count some of them as missed entities. Such entries permit us to compute the well-known metrics of *precision* (P), *recall* (R), and *F-score* (F_1), as shown in eq. (1), eq. (2) and eq. (3) respectively[6].

[4] https://www.sardegnaturismo.it/en/

[5] https://osf.io/4x3r9/?view_only=87e7af84c0564bd1b3eadff23e4b7e54

[6] Let us note that R and F_1 can be computed only for entities, as, in this preliminary work, we asked assessors to annotate only missing entities.

$$P = \frac{TP}{TP + FP} \quad (1) \qquad R = \frac{TP}{TP + FN} \quad (2) \qquad F_1 = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \quad (3)$$

We also assess the capability of the model to infer additional information using the a_σ annotations, defining a score σ as in eq. (4) corresponding to the percentage of information coming from the GPT-3.5 internal knowledge among all the returned correct information. Furthermore, exploiting the a_γ annotations on the REBEL dataset, we estimated the *knowledge enhancement* capability of our tool by devising a score γ as in eq. (5), equivalent to the percentage of correct information not mentioned in the underlying dataset.

$$\sigma = \frac{TP \wedge \neg a_\sigma}{TP} \quad (4) \qquad \gamma = \frac{TP \wedge \neg a_\gamma}{TP} \quad (5)$$

Summarizing, we can compute the following metrics:

- P^E, R^E, F_1^E: precision, recall, and F-score of the entity extraction;
- P_a^T, P_f^T: entity typing precision, accounting for all the extracted types of each entity, and only for the first respectively[7];
- P^R: precision of the relation extraction;
- σ^E, σ^R: σ-scores of entity description and relation extraction;
- γ^E, γ^R: γ-scores of entity extraction and relation extraction.

4.2.2. Automated Assessment

We performed a more extensive evaluation to estimate the knowledge enrichment potential by exploiting the REBEL's Ground Truth. Unlike human evaluation, we cannot rely on classical confusion matrix metrics, as they rely on an exact match between what is generated by the model and what exists in the ground truth. Indeed, triples generated by LLMs may still be correct and informative but be represented differently from the triples contained in the reference dataset. With traditional metrics, they would be evaluated as false positives. To address this, we implemented two metrics proposed by Jiang et al. [13].

The first metric is the *Topical Similarity Score* (T), which measures the information abundance of the extracted triples compared to the source text. It relies on a Latent Dirichlet Allocation (LDA) model [3] for topic modeling to generate a probability distribution representing a text's alignment with a set of N abstract topics (LDA_N). All triples, represented as strings, are concatenated together to obtain a text T_D, and its LDA representation is compared with that of the source text D as in eq. (6). A high score of T means that the triples and the text have a similar alignment with the same abstract topics, indicating an effective information extraction.

$$T = \exp\left(-\sum_{i=1}^{N} LDA_N(D)_i \cdot \log\left(\frac{LDA_N(D)_i}{LDA_N(T_D)_i}\right)\right) \quad (6)$$

The second metric is the *Uniqueness Score* (U), which assesses the diversity of the extracted triples by evaluating the percentage of triple pairs that are adequately different. Each n extracted triples, represented as a string, is encoded in a vector v through embedding. We consider a pair of triples (v_i, v_j) to differ if their cosine similarity is below a given threshold θ. The final score is evaluated as in eq. (7). A high score should indicate that all the triples are conveying different and not overlapping information.

[7] We ask the LLM a list of types: we can consider the entire list or only its first entry.

$$U = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{i \neq j}^{n} \left(\begin{cases} 1 & \text{if } CosSim\left(v_i, v_j\right) < \theta \\ 0 & \text{otherwise} \end{cases} \right) \quad (7)$$

Using the proposed equations, we compare the scores of our triples (T_L and U_L) against the scores obtained by the ground truth triples (T_G and U_G). The actual implementation of the described metrics can be found on the aforementioned project repository[8].

4.3. Results

Our tool extracted a total of 761 entities and 640 triples from the ST dataset, 2,845 entities and 2,024 triples from the REBEL dataset, and 379 entities and 329 triples from REBEL_20. Our experiments assess our tools exclusively. Indeed, considering the peculiar extraction technique performed by our tool, to the best of our knowledge, no other similar state-of-the-art tools can be directly compared. We report the manual evaluation results in Table 1.

Dataset	P^E	R^E	F_1^E	P_a^T	P_f^T	P^R	σ^E	σ^R	γ^E	γ^R
ST	0.974	0.943	0.958	0.857	0.951	0.753	0.094	0.000	-	-
REBEL_20	0.976	0.896	0.934	0.777	0.939	0.836	0.322	0.065	0.451	0.945

Table 1. Evaluation results from manual annotations

Results prove the effectiveness of our KG generation approach, particularly for identifying meaningful entities, as highlighted by the high values of P^E, R^E, and F_1^E for both datasets. Likewise, although the system returns an adequate types list for each entity (P_a^T), the performance increases significantly considering the top-ranked type in the list (P_f^T), meaning that our prompting strategy is effective in assigning an entity type. Moreover, also the relation extraction provides satisfactory results, as highlighted by P^R. Furthermore, the efforts in guiding GPT-3.5 towards a text-centric focus have been particularly successful with the ST dataset, where we observed additional details in only 9.20% of entity descriptions. The REBEL dataset exhibited higher values (33.8% for entity descriptions and 21.9% for relations), probably due to the nature of its texts, where mentioned entities often lack useful details for generating a proper description. We deem that tuning the entity extraction prompt and defining a proper phrase selection strategy should limit the issue without affecting the devised pipeline. Finally, the γ scores highlighted how our methodology significantly enhances an existent KG, especially for identifying new relevant entities not included in the underlying knowledge representation. Concerning the automated evaluation of the REBEL dataset, Table 2 reports the *Topical Similarity* scores for the extracted triples and the ground truth, varying the number of LDA topics (N). Results emphasize how our methodology can better align the extracted information with the main topics of input texts.

	$N = 5$	$N = 10$	$N = 20$	$N = 30$	$N = 40$	$N = 50$	$N = 75$	$N = 100$
T_L	**0.703**	**0.624**	**0.582**	**0.454**	**0.437**	**0.399**	**0.296**	**0.251**
T_G	0.636	0.502	0.513	0.359	0.309	0.292	0.186	0.154

Table 2. Topical similarity results: extracted triples (T_L) vs. ground truth triples (T_G) on the REBEL dataset

Finally, Table 3 reports the *Uniqueness* score for the extracted triples and the ground truth, varying the similarity threshold θ. We extracted a total of 2,024 triples against the 615 triples annotated in the ground truth. However, the Uniqueness scores are higher for our extracted triples, highlighting how our tool provides a more complete and diverse perspective on information that is still unique also for low values of θ.

[8] https://github.com/SandroGT/KG-LLM-Prompting/tree/main/dataset

	$\theta = 0.70$	$\theta = 0.75$	$\theta = 0.80$	$\theta = 0.85$	$\theta = 0.90$	$\theta = 0.95$
U_L	**0.836**	**0.885**	**0.922**	**0.954**	**0.981**	**0.994**
U_G	0.668	0.768	0.837	0.877	0.933	0.981

Table 3. Uniqueness results: extracted triples (U_L) vs. ground truth triples (U_G) on the REBEL dataset

For the sake of clarity, Figure 2 depicts an example of a sub-graph extracted ST, in which entities are the yellow nodes and the pink ones are the types.

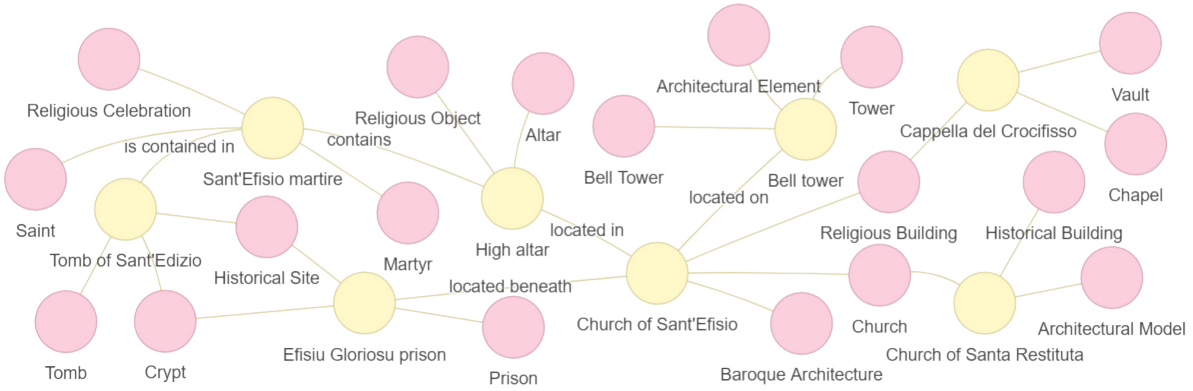

Fig. 2. Example of a sub-graph.

5. Conclusion

This paper proposes a novel LLM prompting-based strategy for generating and enriching Knowledge Graphs. Our proposal leverages the zero-shot generative and comprehension capabilities of the GPT-3.5 model to address the typical issues yielded by the lack of well-established datasets and highly reliable methods. Our approach extracts entities and triples, including multiple entity types, extended entity descriptions, and proper explanations of predicates, providing a proper *semantification* of the extracted elements. The experiments performed on domain-specific datasets confirm the potential of our approach for scalable and versatile knowledge engineering.

Acknowledgements

We acknowledge financial support under the National Recovery and Resilience Plan (NRRP), Mission 4 Component 2 Investment 1.5 - Call for tender No.3277 published on December 30, 2021 by the Italian Ministry of University and Research (MUR) funded by the European Union – NextGenerationEU. Project Code ECS0000038 – Project Title eINS Ecosystem of Innovation for Next Generation Sardinia – CUP F53C22000430001- Grant Assignment Decree No. 1056 adopted on June 23, 2022 by the Italian Ministry of University and Research (MUR). This work has also been partially carried out thanks to the Ministerial Decree no. 351 of 9th April 2022, based on the NRRP – funded by the European Union - NextGenerationEU - Mission 4 “Education and Research”, Component 1 “Enhancement of the offer of educational services: from nurseries to universities” - Investment 4.1, that provided financial support for the Leonardo Piano’s doctoral pathway.

References

[1] Abu-Salih, B., AL-Qurishi, M., Alweshah, M., AL-Smadi, M., Alfayez, R., Saadeh, H., 2023. Healthcare knowledge graph construction: A systematic review of the state-of-the-art, open issues, and opportunities. Journal of Big Data 10. doi:10.1186/s40537-023-00774-9.

[2] Agrawal, M., Hegselmann, S., Lang, H., Kim, Y., Sontag, D., 2022. Large language models are few-shot clinical information extractors, in: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pp. 1998–2022.
[3] Blei, D.M., Ng, A.Y., Jordan, M.I., 2003. Latent dirichlet allocation. Journal of machine Learning research 3, 993–1022.
[4] Bonifacio, L., Abonizio, H., Fadaee, M., Nogueira, R., 2022. Inpars: Data augmentation for information retrieval using large language models. arXiv:2202.05144.
[5] Cabot, P.L.H., Navigli, R., 2021. Rebel: Relation extraction by end-to-end language generation, in: Conference on Empirical Methods in Natural Language Processing. URL: https://api.semanticscholar.org/CorpusID:244119726.
[6] Chiu, J.P., Nichols, E., 2016. Named Entity Recognition with Bidirectional LSTM-CNNs. Transactions of the Association for Computational Linguistics 4, 357–370. URL: https://doi.org/10.1162/tacl_a_00104, doi:10.1162/tacl_a_00104, arXiv:https://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00104/1567392/tacl_a_00104.pdf.
[7] Du, H., Tang, Y., Cheng, Z., 2023. An efficient joint framework for interacting knowledge graph and item recommendation. Knowledge and Information Systems 65, 1685 – 1712. doi:10.1007/s10115-022-01808-z.
[8] Ebraheem, M., Thirumuruganathan, S., Joty, S.R., Ouzzani, M., Tang, N., 2017. Deeper - deep entity resolution. ArXiv abs/1710.00597.
[9] Ehrlinger, L., Wöß, W., 2016. Towards a definition of knowledge graphs., in: SEMANTiCS (Posters, Demos, SuCCESS).
[10] Elhammadi, S., Lakshmanan, L.V., Ng, R., Simpson, M., Huai, B., Wang, Z., Wang, L., 2020. A high precision pipeline for financial knowledge graph construction, in: Proceedings of the 28th international conference on computational linguistics, pp. 967–977.
[11] Hao, S., Tan, B., Tang, K., Zhang, H., Xing, E.P., Hu, Z., 2022. Bertnet: Harvesting knowledge graphs from pretrained language models. arXiv preprint arXiv:2206.14268 .
[12] Hou, X., 2022. Design and application of intelligent financial accounting model based on knowledge graph. Mobile Information Systems 2022. doi:10.1155/2022/8353937.
[13] Jiang, P., Lin, J., Wang, Z., Sun, J., Han, J., 2024. Genres: Rethinking evaluation for generative relation extraction in the era of large language models. arXiv preprint arXiv:2402.10744 .
[14] Li, B., Fang, G., Yang, Y., Wang, Q., Ye, W., Zhao, W., Zhang, S., 2023. Evaluating chatgpt's information extraction capabilities: An assessment of performance, explainability, calibration, and faithfulness. ArXiv abs/2304.11633.
[15] Mehta, A., Singhal, A., Karlapalem, K., 2019. Scalable knowledge graph construction over text using deep learning based predicate mapping, in: Companion Proceedings of The 2019 World Wide Web Conference, pp. 705–713.
[16] Nguyen, T.H., Grishman, R., 2015. Relation extraction: Perspective from convolutional neural networks, in: VS@HLT-NAACL.
[17] Peng, N., 2022. Controllable text generation for open-domain creativity and fairness. arXiv:2209.12099.
[18] Wan, Z., Cheng, F., Mao, Z., Liu, Q., Song, H., Li, J., Kurohashi, S., 2023. Gpt-re: In-context learning for relation extraction using large language models. arXiv preprint arXiv:2305.02105 .
[19] Wang, C., Liu, X., Song, D., 2020. Language models are open knowledge graphs. arXiv preprint arXiv:2010.11967 .
[20] Wei, X., Cui, X., Cheng, N., Wang, X., Zhang, X., Huang, S., Xie, P., Xu, J., Chen, Y., Zhang, M., Jiang, Y., Han, W., 2023. Zero-shot information extraction via chatting with chatgpt. ArXiv abs/2302.10205.
[21] Wu, Q., Fu, D., Shen, B., Chen, Y., 2020. Semantic service search in it crowdsourcing platform: A knowledge graph-based approach. International Journal of Software Engineering and Knowledge Engineering 30, 765 – 783. doi:10.1142/S0218194020400069.
[22] Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T.L., Cao, Y., Narasimhan, K., 2023. Tree of thoughts: Deliberate problem solving with large language models. arXiv:2305.10601.
[23] Zhang, H., Liang, H., Zhang, Y., Zhan, L.M., Wu, X.M., Lu, X., Lam, A., 2022. Fine-tuning pre-trained language models for few-shot intent detection: Supervised pre-training and isotropization, in: Carpuat, M., de Marneffe, M.C., Meza Ruiz, I.V. (Eds.), Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Seattle, United States. pp. 532–542. URL: https://aclanthology.org/2022.naacl-main.39, doi:10.18653/v1/2022.naacl-main.39.
[24] Zhang, T., Ladhak, F., Durmus, E., Liang, P., McKeown, K., Hashimoto, T.B., 2023. Benchmarking large language models for news summarization. arXiv:2301.13848.