SN Computer Science (2024) 5:1029
https://doi.org/10.1007/s42979-024-03341-y

ORIGINAL RESEARCH

Core Concept Identification in Educational Resources via Knowledge Graphs and Large Language Models

Daniel Reales[1] · **Rubén Manrique**[1] · **Christian Grévisse**[2]

Received: 26 August 2024 / Accepted: 12 September 2024

Abstract
The growing demand for online education raises the question of which learning resources should be included in online programs to ensure students achieve their desired learning outcomes. By automatically identifying the core concepts in educational materials, teachers can select coherent and relevant resources for their courses. This work explores the use of Large Language Models (LLMs) to identify core concepts in educational resources. We propose three different pipelines for building knowledge graphs from lecture transcripts using LLMs and ontologies such as DBpedia. These knowledge graphs are then utilized to determine the central concepts (nodes) within the educational resources. Results show that LLM-constructed knowledge graphs when guided by ontologies, achieve state-of-the-art performance in core concept identification.

Keywords Knowledge graphs · Large language models · Core concept identification

Introduction

Knowledge graphs (KGs) have become a fundamental tool for advancing education. Their widespread application in adaptive and personalized learning, curriculum design, educational analytics, and knowledge tracing underscores their significance in the educational domain [1]. The accurate automatic identification of core concepts in educational material is a significant achievement, impacting the relevance of the resources students learn from and enabling tailored recommendations based on their current knowledge levels. Previous research has explored the use of open knowledge graphs to identify core concepts in educational resources [2, 8]. However, these methods, which exploit open knowledge graphs such as DBpedia,[1] primarily focus on identifying relevant entities within the lesson text that are present in the ontology. They then build the graph by selecting a subset of the relations these entities have in the original database. Consequently, the knowledge graph serving as the semantic representation of the lesson is always a sub-graph of the open knowledge graph.

While relying solely on the relationships contained in the original database is useful, this approach becomes unsuitable when the relations among relevant entities are absent in the base ontology. Other methods, such as those employing Deep Learning techniques and Data Mining [3], have been proposed for constructing KGs for educational purposes. However, the triple extraction (TE) process in these methods requires labeled datasets, which can be costly to create.

We propose the use of Large Language Models (LLMs) assisted by entities found in open knowledge graphs to construct a semantic representation of lesson text that effectively identifies its core concepts. A growing body of literature explores the capabilities of LLMs in performing domain-specific entity recognition (ER) [4], relation extraction (RE) [6, 11, 12], and triple extraction (TE) [5, 10], often achieving performance that surpasses previous state-of-the-art benchmarks in certain settings.

Guided by these findings, we explore three different pipelines for building knowledge graphs to identify the core concepts of lessons using LLMs. We begin by examining a Zero-Shot triple extraction setting with the LLM, where each concept pair and the relation between them in

✉ Daniel Reales
da.reales@uniandes.edu.co

Rubén Manrique
rf.manrique@uniandes.edu.co

Christian Grévisse
christian.grevisse@uni.lu

[1] Universidad de los Andes, Bogotá, Colombia

[2] University of Luxembourg, Esch-sur-Alzette, Luxembourg

[1] https://www.dbpedia.org.

Published online: 06 November 2024

the knowledge graph is identified by the model without any external information. After extracting the triples, we use DBpedia Spotlight-a disambiguation and annotation service that links entity mentions in text to DBpedia resources-to disambiguate the concepts before building the graph. Once the graph is constructed, we apply PageRank, as suggested by [8], to compute the importance of each concept (node) within the resource representation, thereby identifying the core concepts of the lesson.

Using the Zero-Shot results as a baseline, we enhance the model's performance by annotating the DBpedia resources present in the lesson text. With these annotations, we instruct the model to extract relationships exclusively between those annotated concepts. This modification significantly improves the precision and recall in retrieving the core concepts of the lesson.

Finally, building on the observations in [4, 8] that adjacent relevant concepts to the lesson topic may be present but not explicitly mentioned, we use the LLM to generate a summary of the lesson text. We then annotate both the summary and the original text using DBpedia Spotlight. The concepts (DBpedia resources) identified in the summary and the original text are provided to the LLM, which now has access to the summary, lesson text, and this augmented set of concepts for the triple extraction task. Among all the proposed pipelines, this approach yields the best results, surpassing previous benchmarks for the evaluation dataset [7].

The remainder of the work is organized as follows: "Related Work" section reviews related work on the use of large language models in tasks related to knowledge graph construction. "Knowledge Graph Construction" section details the proposed method for using large language models to identify the core concepts in lesson texts. "Evaluation" section presents the experimental results for the suggested pipelines using four large language models: Claude-3-Opus, GPT-4-0613, GPT-4-Turbo, GPT-4o. Finally, "Conclusion" section discusses the findings and suggests directions for future research.

Related Work

Assessing Large Language Model's capabilities to extract complex structured information from text has become an important research area. In the context of knowledge graph construction, tasks such as entity recognition, relation extraction, and triple extraction serve as the building blocks to transform text into the desired target representation.

LLMs have been studied for their effectiveness in extracting the basic constituents of knowledge graphs: nodes (entities) and edges (relations). de Paiva et al. use ChatGPT to extract Category Theory concepts present in academic papers [4]. They conclude the model is able to extract mathematical concepts from text. However, they highlight the inadequacy of the model to identify concepts that are not explicitly mentioned in the text but that a human would infer by context. Concerning relation extraction, Wadhwa et al. find that providing GPT-3 with around 12–20 training examples in the context window results in performance similar to state-of-the-art fully supervised models for datasets such as CoNLL04, NYT, DocRED, and ADE [11]. Papaluca et al. explore the capabilities of LLMs to perform Zero-Shot and Few-Shot graph triple extraction evaluating the performance of various models with datasets such as NYT and WebNLG [10]. They find that Zero-Zhot and Few-Show settings are insufficient to match classical end-to-end models that hold state-of-the-art performances. The authors suggest the use of Knowledge Bases to provide more context information to the LLM via triplets found therein. The results show that the Knowledge Base augmented pipeline proposed by the authors improves performance even surpassing old Bi-LSTM baselines. However, it does not surpass current state-of-the-art models.

The primary contribution of our work is the use of large language models for knowledge graph construction, resulting in a more accurate semantic representation of textual resources for use in core concept identification. Building on previous research that demonstrates the feasibility of applying these models to knowledge graph construction, we use existing ontologies to guide the models in identifying relevant entities from the text for inclusion in the final representation.

Knowledge Graph Construction

We semantically represent the lesson text using a directed weighted graph, where concepts are depicted as nodes and their relationships as edges. We extract triples, each consisting of a source concept, a relationship, and a target concept. Weights are assigned based on the frequency of concept co-occurrences, regardless of the specific nature of their relationships. In this section, we detail three methodologies for constructing these knowledge graphs using large language models (LLMs).

Zero-Shot Triple Extraction

The first proposed method for building the knowledge graph involves prompting the model to extract triples from the source text directly. The proposed pipeline for this method is illustrated in Fig. 1.

A lesson transcript is split into pieces each around 10 to 15 sentences long so that the context window of the model is not exceeded. Each of these pieces is then passed to the model which is instructed to extract relationships

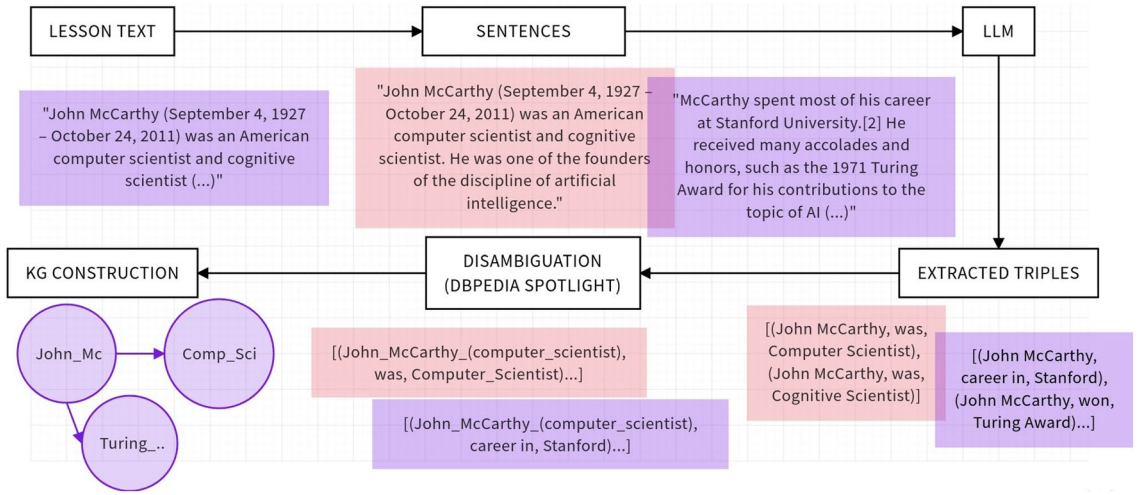

Fig. 1 Zero-shot knowledge graph construction via LLM usage and DBpedia disambiguation

between the concepts that are found in the text. The model is instructed to output only a valid JSON that includes the identified triples. For the Zero-Shot prompt, no other information other than the format of the expected JSON and the instruction to extract the triples is provided to the model as can be seen in the used prompt captured in Fig. 2. In cases in which the model is able to generate structured output (such as chat models from the OpenAI API) this option is used to reduce the probability that an incorrect output format is generated.

After the inference from the model, relationships are dropped and each of the concepts is sent to DBpedia Spotlight for disambiguation. As suggested by [8], we configure DBpedia Spotlight parameters of confidence and support with values of 0.35 and 5, respectively. For concepts that are not mapped to any DBpedia resource successfully, the original text output extracted by the LLM is preserved as the node. Finally, the representation is built using the extracted, disambiguated pairs. It is important to note that, as pointed out by [2, 8], no correction after the disambiguation process

Fig. 2 Prompt used for zero-shot triple extraction

```
You have a text delimited between the following symbols: ||.
Your task is to extract all the relationships between the concepts found in
the text.
You must extract the concepts and the relations that connect them in the
format delimited between three backticks: ``` .
You must output only a valid JSON that follows the JSON format delimited
between three asterisks: ***.
TEXT:
||
{text}
||
CONCEPT RELATIONSHIPS EXTRACTION FORMAT:
```
(concept, relation, concept)
```
JSON OUTPUT:
***
{{
    "concept_relations": Array<String>
}}
***
Remember to only output valid JSON and no other text, but the JSON.

JSON OUTPUT:
```

is performed. Therefore, incomplete (no found resource) or incorrect disambiguation is possible as there are no guarantees that the mapped resource is sensible in the context of the supplied text.

Concept Supported Triple Extraction

Following observations from previous work discussed earlier, large language models perform better on triple extraction tasks when guided with complementary information about the text the model is performing extraction upon. We, therefore, propose to explicitly instruct the model to extract relationships among pre-identified entities that are present in the text. The entity identification task is done via the DBpedia Spotlight API service that was used also in the previous step for disambiguation purposes. In the proposed pipeline (see Fig. 3), the original lesson transcript is chunked. Then, the entity recognition step is performed with the DBpedia service. Both the identified entities and the text are then passed to the model with the instruction to only extract relationships in the text that include concepts in the list of identified entities as can be appreciated in the used prompt presented in Fig. 4. As in the previous pipeline, the extracted triples are used to build the knowledge graph.

Expanded Concept List Triple Extraction

Both [4, 8] identify a significant limitation in the automatic process of building knowledge graphs from text: the omission of relevant concepts. Specifically, de Paiva et al. [4] highlights the inability of large language models (LLMs) to infer related but not explicitly mentioned concepts that humans would consider relevant. To address this, Manrique et al. [8] suggests including adjacent concepts by retrieving connected concepts from a reference ontology containing the originally identified ones. However, as noted in the introduction of this work, this approach may be unsuitable when the desired relationships are absent in the source ontology. We propose an alternative method where the LLM summarizes the current chunk of the lesson and then DBpedia Spotlight is used to extract new entities from the generated summary. The prompt to extract the new concepts is shown in Fig. 6. This pipeline is similar to the concept-supported triple extraction method, with the key difference being that entities included in the prompt are now supplemented with those identified in the generated summary. Therefore, after the new entities are extracted from the LLM generated summary, the same prompt as in the previous proposed pipeline (see Fig. 4) is used. That is, the original text of the lesson in addition to the entities identified both in the lesson text and the generated summary are supplied in the triple extraction step.

The proposed third pipeline is illustrated in Fig. 5. Notably, the implementation of this pipeline in an automated manner is facilitated by the model's ability to generate structured outputs, enabling LLM calls to be chained sequentially. This is crucial as 2 inferences are needed. The first one corresponds to the generation of the summary and the second one to the extraction of the triples given the lesson text and the list of concepts extracted from both the original lesson text and the generated summary (Fig. 6).

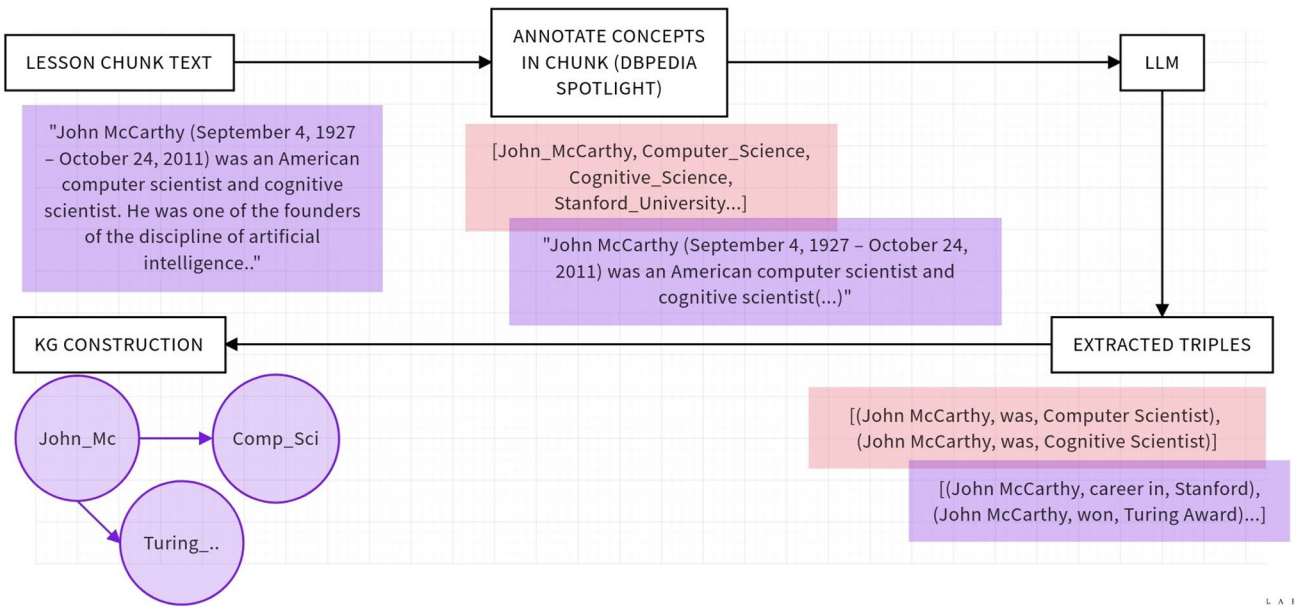

Fig. 3 Knowledge graph construction supported with concept annotations

```
You have a list of concepts delimited between the following symbols: ###.
You have a text delimited between the following symbols: ||.
Your task is to extract the relationships between the concepts that are within ### in
the text.
You must extract triples of concepts-relationship in the format delimited between
three backticks: ``` .
The concepts you include in the triples must be ONLY from the list of concepts that
are within ###.
You must output only a valid JSON that follows the JSON format delimited between
three asterisks: ***.
LIST OF CONCEPTS:
###
{concepts}
###
TEXT:
||
{text}
||
CONCEPT RELATIONSHIPS EXTRACTION FORMAT:
```
(concept, relation, concept)
```
JSON OUTPUT:
***
{{
    "concept_relations": Array<String>
}}
***
Remember to only output valid JSON and no other text, but the JSON.

JSON OUTPUT:
```

Fig. 4 Prompt used for concept supported triple extraction

Evaluation

Experimental Setup

To evaluate the effectiveness of the three proposed pipelines in identifying core concepts in educational material, we utilize the LERECCI dataset [7]. This dataset consists of 192 Coursera lesson transcripts, each annotated by experts with the concepts they consider are core in the lesson. The ground truth annotations for core concepts are mapped to selected DBpedia resources. Following the method suggested by [8], once the knowledge graph is built, PageRank is computed to rank the importance of the concepts within the semantic representation. These core concepts are then compared to the ground truth annotations.

For our evaluation, we focus on the subset of 135 lessons available in English, as they span a diverse array of domains, including Physics, Biology, Music, and History (Table 1).

Results

The results using four different large language models and each of the three proposed pipelines are presented below. We calculate the ranked F_1 score at k for $k \in \{3, 10\}$, as reported by previous state-of-the-art (SOTA) unsupervised approaches for this dataset in [9]. The Zero Shot, Concept Annotated and Concept Expanded labels correspond to the previously suggested methodologies to build the knowledge graph. For each construction method, the highest performing model is highlighted in red.[2] As shown in both Figs. 7 and 8, the LLM suggested pipeline surpasses previous state-of-the-art approaches.

Shown in Tables 2 and 3 are the performances of the Concept Expanded pipeline for the different chosen models

[2] The state-of-the-art value is taken as the highest F_1 score reported in the results section of [9] for the LERECCI dataset.

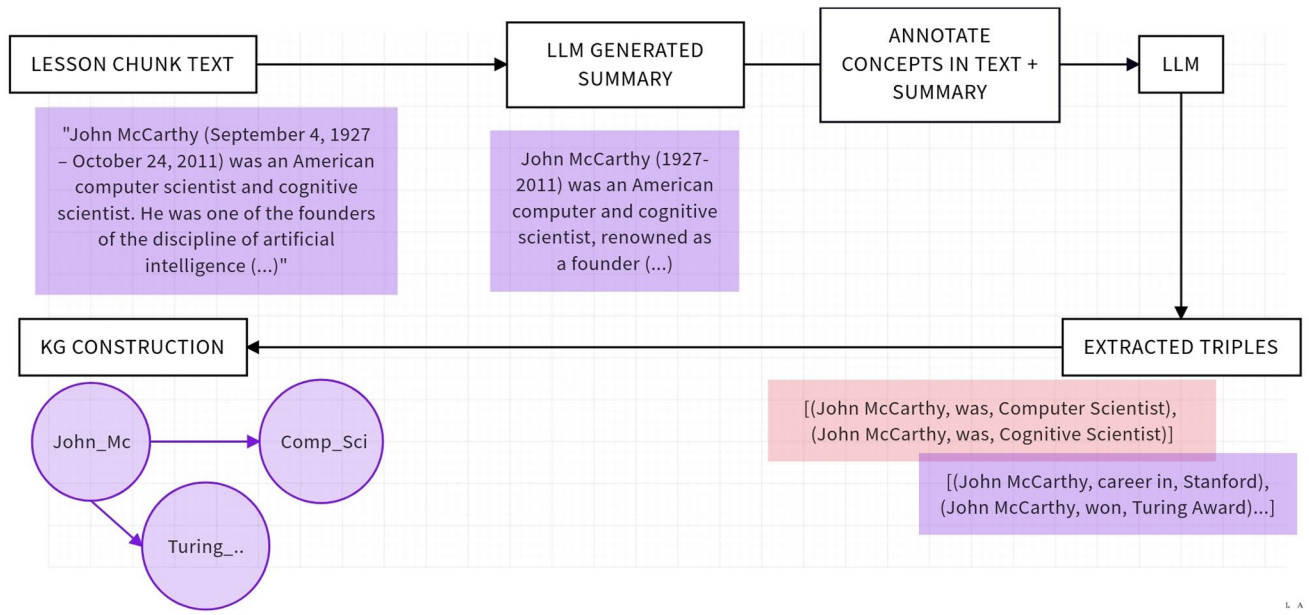

Fig. 5 Knowledge graph construction supported with expanded concept annotations

Fig. 6 Prompt used to summarize the lesson text

```
You have a text delimited between the following symbols: ||.
Your task is to explain the text including related concepts to those found in the
text. Be thorough in your explanation.
You must output only a valid JSON that follows the JSON format delimited between
three asterisks: ***.
TEXT:
||
{context_text}
||
JSON OUTPUT:
***
{{
    "explanation": String
}}
***
Remember to only output valid JSON and no other text, but the JSON.
JSON OUTPUT:
```

Table 1 Number of lessons by educational domain in the evaluation dataset

Domain	History	Music	Physics	Biology
Lessons	48	18	33	36

by domain. The best performing model score for a particular domain is highlighted in bold. As expected, for a particular model, variation in the performance of the core concept identification retrieval task is evidenced by domain. However, the performance tends not to deviate by a margin of more than 15% of the F_1 computed on the whole dataset. Although more research is needed in this direction, results seems to indicate this method's performance is stable across domains.

Discussion

As anticipated based on previous research, the Zero-Shot Core Concept identification pipeline is not competitive with prior state-of-the-art methods. However, instructing the model to extract relationships among pre-identified entities present in the text leads to a significant improvement in performance. Similarly, incorporating new concepts through the summarization strategy enhances the results even further yielding the best results.

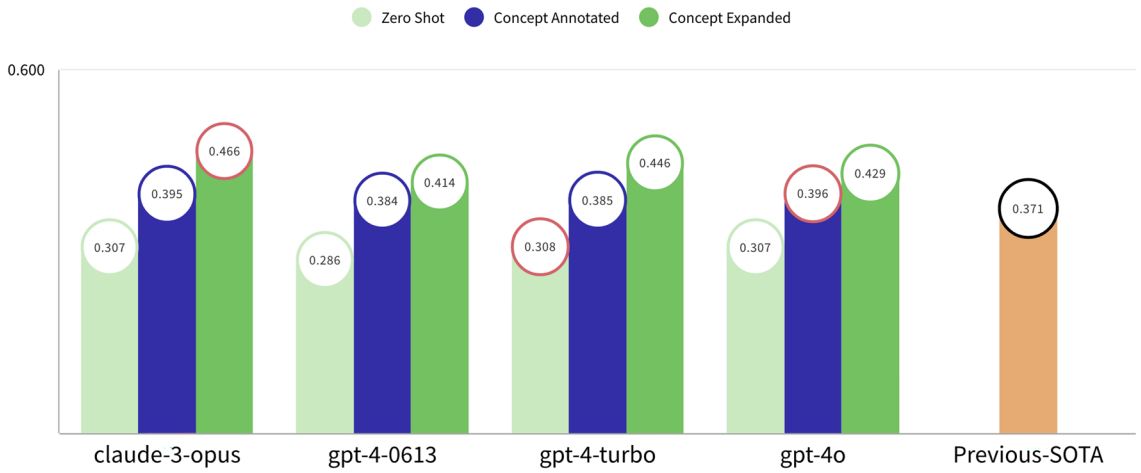

Fig. 7 F1@k, k = 3

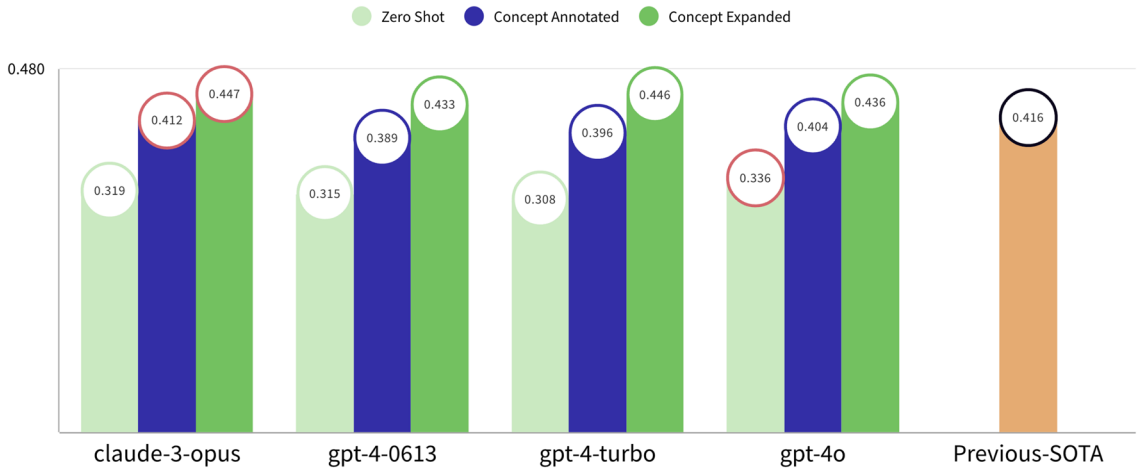

Fig. 8 F1@k, k = 10

Table 2 F1@k, k = 3

Model/domain	History	Music	Physics	Biology
claude-3-opus	**0.470**	**0.515**	0.441	**0.436**
gpt-4-0613	0.430	0.437	0.377	0.393
gpt-4-turbo	0.442	0.443	**0.454**	0.426
gpt-4o	0.416	0.443	0.424	0.423

Model versus Domain Performance under the Concept Expanded Pipeline

Table 3 F1@k, k = 10

Model/Domain	History	Music	Physics	Biology
claude-3-opus	**0.451**	**0.426**	**0.396**	0.487
gpt-4-0613	0.417	0.386	0.373	0.518
gpt-4-turbo	0.426	0.397	0.392	**0.535**
gpt-4o	0.423	0.383	0.382	0.513

Model versus Domain Performance under the Concept Expanded Pipeline

The performance improvement observed when providing guidance to the model on which entities to focus on suggests that building semantic representations with large language models is best approached as a multi-step process. These models excel in extracting relations, which is often the most challenging aspect of triple extraction.

Similarly, improvement observed when entities are provided as context in the prompt to the language model raises the question of how to effectively select these entities to ensure that the final semantic representation includes only the most relevant ones, thereby reducing noise in the core concept identification strategy. Additionally, it is

Table 4 Average inference cost per lesson for concept expanded pipeline

Model	claude-3-opus	gpt-4-0613	gpt-4-turbo	gpt-4o
Cost (US$)	0.595	0.664	0.389	0.170

important to note that both the concept-annotated and concept-expanded pipelines offer the advantage that if the model can detect incorrectly annotated entities based on the context of the text, the final representation will be of higher quality. This is because no relations will be expected in the source text for a resource that has been incorrectly disambiguated.

Not only the suggested pipeline based on concept expansion yields better results than previous SOTA methods, but for particular use cases in which expert annotations or labeled dataset construction is required it may be cheaper. Costs of inference for different models for the most expensive pipeline (the one labeled as Concept Expanded which is also the better performing) are shown in Table 4.[3]

Finally, this pipeline can be implemented without the use of ontologies, as the model expects at most a list of concepts and a text to perform extraction from. This decoupling from the entity recognition and knowledge graph building may significantly reduce associated costs of building (or depending on) comprehensive ontologies that must contain detailed relationships between the entities involved.

Conclusion

This work explored the application of large language models to construct semantic representations of educational lessons from text and leverage these representations to extract the core concepts addressed in the lesson. By utilizing knowledge graphs as the semantic framework, the configuration of relationships among concepts is harnessed to identify central nodes within the network. The results demonstrate that providing guidance on which concepts to prioritize during the triple extraction process yields superior results compared to a zero-shot approach.

The success of accurately extracting triples from text leads to future research questions regarding the model's ability to maintain this capability in multilingual settings. This could prove advantageous given that ontologies and open knowledge graphs in non-English languages are often less developed than their English counterparts. Similarly, it raises questions about the model's capacity to perform triple extraction across different levels of material complexity and within various domains of knowledge.

[3] Reported costs are in US$ based on API rates established by OpenAI and Anthropic during the period of time comprised between the last week of May and first of June of 2024 for the selected models.

Author contributions All authors contributed to the study's conception and design. DR performed evaluation tests. DR and RM spearheaded the article writing process. All authors participated in writing the article and analyzing the results. All authors read and approved the final manuscript.

Funding Open Access funding provided by Colombia Consortium. No funding was received to assist in the preparation of this manuscript.

Data availability No data was obtained for this report.

Declarations

Conflict of interest The authors declare that they have no Conflict of interest.

References

1. Abu-Salih B, Alotaibi S. A systematic literature review of knowledge graph construction and application in education. Heliyon. 2024;10(3):e25383. https://doi.org/10.1016/j.heliyon.2024.e25383.
2. Ain QU, Chatti MA, Bakar KGC, et al. Automatic construction of educational knowledge graphs: a word embedding-based approach. Information. 2023;14(10):526. https://doi.org/10.3390/info14100526.
3. Chen P, Lu Y, Zheng VW, et al. KnowEdu: a system to construct knowledge graph for education. IEEE Access. 2018;6:31553–63. https://doi.org/10.1109/ACCESS.2018.2839607.
4. de Paiva V, Gao Q, Kovalev P, et al. Extracting mathematical concepts with large language models. 2023. https://doi.org/10.48550/arXiv.2309.00642
5. Ding Z, Huang W, Liang J, et al. Improving recall of large language models: a model collaboration approach for relational triple extraction. 2024. https://doi.org/10.48550/arXiv.2404.09593
6. Li G, Wang P, Ke W. Revisiting large language models as zero-shot relation extractors. 2023. https://doi.org/10.48550/arXiv.2310.05028
7. Manrique R. LERECCI dataset. 2024. https://github.com/Ruframpi/LERECCI, Online. Accessed 01 July 2024
8. Manrique R, Grévisse C, Mariño O, et al. Knowledge graph-based core concept identification in learning resources. In: Ichise R, Lecue F, Kawamura T, et al., editors. Semantic Technology. Cham: Springer International Publishing; 2018. p. 36–51. https://doi.org/10.1007/978-3-030-04284-4_3.

9. Manrique Piramanrique RF. Towards automatic learning resources organization via knowledge graphs, PhD thesis. Bogotá: Universidad de los Andes; 2019. http://hdl.handle.net/1992/41293
10. Papaluca A, Krefl D, Rodriguez SM, et al. Zero- and few-shots knowledge graph triplet extraction with large language models. 2023. https://doi.org/10.48550/arXiv.2312.01954,
11. Wadhwa S, Amir S, Wallace B. Revisiting Relation Extraction in the era of Large Language Models. In: Rogers A, Boyd-Graber J, Okazaki N, editors. Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Toronto: Association for Computational Linguistics; 2023. https://doi.org/10.18653/v1/2023.acl-long.868
12. Zhang K, Gutiérrez BJ, Su Y. Aligning instruction tasks unlocks large language models as zero-shot relation extractors. 2023. https://doi.org/10.48550/arXiv.2305.11159,