Can we know what they know? Prompting Large Language Models for Concept Dependency Graph Extraction

Dominik Glandorf
6007407
dominik.glandorf@student.uni-tuebingen.de
Anastasiia Alekseeva
5994775
anastasiia.alekseeva@student.uni-tuebingen.de
University of Tübingen

Abstract

Concept dependency graphs are vital knowledge representations for educational applications. Several different approaches to their extraction using automated or manual methods and textual information sources have been evaluated. Recent advancements in Large Language Models might allow for knowledge graph mining via prompt engineering. This work compares a novel information extraction method from language models called output re-feeding to textbook and encyclopedic baselines regarding the precision of the extracted dependencies. The quality of this output is promising yet far from perfect. Due to the focus on precision, recall is not evaluated in this work but requires further consideration to ensure high-quality graphs. The heuristic encyclopedic baseline extraction approach may be suitable for automatically fine-tuning language models in new domains.

1 Introduction

Large Language Models (LLMs) are trained on immense corpora of text and build on included factual information when performing well in downstream tasks such as question answering. Accessing this knowledge, which is represented by billions of parameters and the network's architecture, has given rise to the research field of Knowledge Extraction from LLMs (Cohen et al., 2023). This field rests on the assumption that language models can help retrieve information on the relation between entities. This work contributes to the field by addressing the question of whether LLMs can provide precise educational knowledge in the form of prerequisites. Apart from the area of knowledge extraction, which is part of computational linguistics, our work contributes to applied education sciences by providing data for computer-assisted education and creating new perspectives for educational knowledge engineering.

Effective and efficient instruction does not only incorporate what to teach but also in which order. Prior knowledge of prerequisites or exposure to them has been found to enhance learning, giving rise to the theory of *Instructional Sequencing* (Morrison et al., 2019). Within learning contents, Merrill (1983) differentiated facts, concepts, principles, rules, procedures, interpersonal skills, attitudes, and their sole recall from their application. For simplicity, we only focus on learning concepts such as "derivative" in mathematics, which possibly represent a keyword in a lecture or textbook, and do not distinguish between recalling and applying knowledge. If one concept is a prerequisite of another, we refer to this asymmetric relation as a *concept dependency*. For example, to understand the concept of a derivative of a function, knowing the concept of a limit will facilitate or even enable learning. Importantly, the dependency relation is distinguished from the similarity, hierarchy, and reference relation (Gordon et al., 2016). Besides that, the terms prerequisite and concept dependency are used interchangeably. When the concepts are considered as nodes in a graph and the dependencies as directed edges between them, a Concept Dependency Graph (CDG) emerges, which is a special type of knowledge graph (Wang et al., 2016). This graph is also called *concept map* in the field of Learning Sciences. The CDG can be used to generate ordered reading lists (Gordon et al., 2016), and hence advance curriculum planning (Yang et al., 2015), especially for new topics that are uncovered in courses learned before. Another use case is automated performance assessment, where a CDG allows inferring performance in concept dependencies as well instead of only the concept that is evaluated in a certain task (Wang et al., 2016).

In this work, we address the questions of how to extract this particular knowledge graph and how to evaluate its quality.

- RQ1: What input to a state-of-the-art LLM can provide a relatively robust and reasonable output for extracting concept dependencies?
- RQ2: What are simple but effective baseline CDG extraction methods based on traditional educational knowledge sources?
- RQ3: Are the baselines from RQ2 appropriate for automatic evaluation of the CDG extracted from LLM?

Our main contributions are the following: First, there is little research on how to automatically evaluate the precision of extracted concept dependencies, which can be used to train and fine-tune models for this task. Therefore, we propose a set of methods to create baselines for evaluation from two existing unstructured knowledge sources, namely Wikipedia and textbooks. Due to the heuristic characteristics of these methods, we conducted a manual assessment to test their suitability for automatic evaluation. The resulting dataset can be used as a baseline for further research. Second, the emerging field of prompt engineering (Liu et al., 2021) provides a constellation of inputs to query LLMs, the majority of which are experimental and cannot be considered reliable. We propose a method called *output re-feeding* that allows mining the educational concept dependencies by sequentially querying the LLM and transforming its answers into a knowledge graph.

1.1 Related work

Concept dependencies have been conceptualized as semantic relations in a variety of ways. Following Talukdar and Cohen (2012), concept A is a prerequisite of concept B, if A is helpful to comprehend B. This coincides with the operationalization of a concept dependency by Gordon et al. (2016) in terms of the helpfulness of learning about another concept and their modeling of information flow to determine the relation. Vuong et al. (2011) defined the relation in terms of performance. More specifically, prior knowledge that leads to a better graduation rate is considered a prerequisite, which is an unsuitable definition when analyzing content only. Concepts are often equated with Wikipedia articles (Wang and Liu, 2016) but also with results of topic modeling (Gordon et al., 2016).

Gordon et al. (2016) presented two information-theoretic approaches to extract the dependency relation from a corpus of scientific documents, in which concepts were obtained by a topic model and filtered by human judges. The first, entropy-based approach predicts a dependency of concept A on concept B to the extent that the distribution of B can predict the distribution of A. Their second approach, called information flow, is also based on co-occurrences of topics in documents but simulates navigation through the documents and considers transitions to documents about other topics as dependencies. Word similarity, citations, and hierarchy served as baselines where word similarity performed better or equally well as the proposed approaches in terms of precision and recall. This result indicates that their modeled relation is rather symmetric and hence not helpful for sequential curriculum planning. Despite the algebraic simplicity of the approach, interpreting the results and identifying error sources is hard.

As an alternative, Liang et al. (2015) defined a *reference distance metric*, which is asymmetric and reflects the extent to which concepts related to concept A (for example by hyperlinks) refer to concepts related to concept B compared to vice versa. If concepts related to A do this more strongly than the other way around, concept A is considered dependent on B. Their proposed measure on Wikipedia concepts performed better than the maximum-entropy classifier on the CrowdComp dataset by Talukdar and Cohen (2012) and their Courses dataset.

More recent approaches already employed neural networks. Li et al. (2021) used variational autoencoders for link prediction, whereas Gasparetti (2022) entered embeddings for the concepts into a number of popular binary classifiers and outperformed the previously referenced methods. Sun et al. (2022) also worked with concept embeddings but fed them together with a concept dependency graph into a graph neural network to predict the dependency structure between concepts.

Multiple approaches of prompt engineering for graph creation have been recently developed. Few-shot prompting proved successful for the extraction of several kinds of relationship for a seed entity (Cohen et al., 2023). Overall, the reviewed literature developed different methods to estimate the strength of prerequisite relation between concepts but was often limited to this set of nodes. Our approach can also discover required nodes outside of a pre-specified set.

Prerequisites can also be behaviorally inferred from learning paths by testing learners' performance after being presented with different instructional sequences (Pavlik Jr. et al., 2008; Vuong et al., 2011) or by analyzing human navigation through information sources (Gordon et al., 2016). However, these approaches usually lack data, and creating this data has the disadvantage of disengaging users with difficult concepts before teaching easier or more necessary ones. An alternative approach to creating a concept map involves expert knowledge, which incurs a relatively high cost. The automated extraction of concept dependencies would be a more efficient method of solving the non-trivial task.

2 Method

In this section, we will first detail our research design and then the characteristics of our information sources as well as the methods that we used to generate the concept dependency graph. All code is available on GitHub[1].

2.1 Research design

The choice of methodology reflects that our research is located at the intersection of education sciences and computational linguistics. On the one hand, we want to achieve a high performance of a state-of-the-art LLM on the downstream task of knowledge graph extraction. On the other hand, we consider a complex educational problem that cannot be purely simplified to classical machine learning metrics but requires additional qualitative assessment to ensure that our results are sound in terms of internal and external validity.

First of all, we restricted the concepts for that we extracted dependencies to the domain of linear algebra. This choice was driven by two assumptions. First, as a highly structured domain, linear algebra provides rich and numerous conceptual dependency relations. Second, as a rather mature branch of mathematics, there exist abundant resources for baseline testing and for LLM training. The central importance of linear algebra across a multitude of quantitative disciplines ensures that our results are of potential interest to a large audience of learners.

Baseline CDGs are necessary to evaluate an LLM's performance but no large-scale ground-truth CDG was available to us, which is unsurprising given the problem complexity (see Section 4). This baseline graph ideally only contains a small number of edges per node to have a number of direct prerequisites small enough for educational use (a student does not want to review 20 concepts to learn another). The reviewed literature has only predicted the relation given two concepts but did not produce a comprehensive CDG originating from a given set of concepts. We hence selected two common sources of educational knowledge to serve as our knowledge base, namely textbooks, and Wikipedia. Due to their end-user-facing format, further processing of the input data (PDFs and webpages) was required to extract CDGs from them. To make the graphs comparable with each other, dependencies were only extracted for the union of the concepts indicated in the textbooks' indices. This set of concepts was then used to prompt the LLM, whose responses were further processed to create a knowledge graph.

After creating the baseline and LLM-based CDGs, we used an interactive visualization dashboard to evaluate the resulting dependency structures. This extra step was introduced to get an impression of the general suitability of all three extraction pipelines and identify potential reasons for performance losses. Then a random sample of edges from all graphs was manually rated by multiple raters to not only measure the precision of the three methods but also assess the human inter-rater agreement on the problem (which reflects its complexity). We finally compared the LLM graph against the baselines to give suggestions for a scalable, low-cost evaluation system for LLMs. The former expert-based, human evaluations are rather typical in education research whereas the latter enables the creation of large-scale

[1] https://github.com/mlcolab/learning-dependencies

datasets, potentially suitable to train and finetune machine learning models.

2.2 Baseline extraction

2.2.1 Wikipedia

Wikipedia has a graph structure induced by the hyperlinks between articles, which makes it suitable for educational sequencing purposes as empirical studies have shown (Wang and Liu, 2016). Wikipedia covers most of the undergraduate-level concepts (Yang et al., 2015), with 284 pages in the category "Linear Algebra" as of March 2023 that abundantly cover the principal undergraduate textbook topics: vector spaces, determinants, matrices and linear maps, diagonalization, factorizations, and linear systems. The focus on dependencies required for understanding a concept makes an encyclopedia an ideal source because its articles are usually written to enable a relatively quick grasp of the article's topic. An online encyclopedia has a significant advantage over the non-digital one in that digital hyperlinks are programmatically actionable, which makes data collection less error-prone, faster, and easier. In Wikipedia, each content article begins with a summary, often serving as a definition and situating the topic into its context, before in-depth information follows. In this summary text, we assume an implicit order from most important to least important concepts that are used to explain the topic without digressing to other concepts or unimportant information. The involved concepts are inferred to be dependent on the article's concept. To prevent direct cycles between two concepts that involve each other in their explanation, the one that refers earlier to the other is considered to be dependent on the latter. Taking into account the straightforward and standardized structure of Wikipedia pages, we operationalized the dependency relationships in the following way: every link in the first content paragraph that contains links is a candidate dependency.

The process of mining dependencies from Wikipedia articles is the following. First, we removed the links to general categories or fields such as "Mathematics" or "Linear Algebra" based on a manually curated blacklist (a more sophisticated, automatic pruning could make use of Wikipedia's category pages). Then, target links to articles about persons were filtered out since these do not represent educational concepts. The same applies to the target links to pages that contain irrelevant or confusing content such as disambiguation pages. Subsequently, cyclic dependencies are resolved as described above using both character positions of the symmetric links in the text. If the article about A links at the 20th character to the article about B, but B's article links to A's at the 10th character, B is considered dependent on A instead of vice versa. The first five concepts mentioned in the paragraph are kept, which is based on the main assumption of decreasing importance from the beginning on and the constraint of some sparsity of the resulting graph. This set of concepts is then taken as the concept's dependencies.

The procedure was run for every concept in the set resulting from the textbook baseline. Although Wikipedia theoretically offers a larger number of articles about the topic, we decided to stick to the limited set to maintain comparability. However, we decided against restricting the dependencies to concepts in the set to be able to evaluate the recall of the methods.

The procedure is efficient and inexpensive, and most time is spent on network requests to Wikipedia servers for the concepts and the links. The overall time was hence reduced by caching the responses to these requests. The technical implementation in Python required engineering our own Wikipedia API using the requests library together with BeautifulSoup[2]. The tools are used to download the page content and conveniently access elements such as links or special HTML tags indicating that an article is about a person.

2.2.2 Textbook corpus

We chose student textbooks as a complementary information source besides Wikipedia, the general purpose knowledge base, because they are written by domain experts specifically for educational knowledge transfer to learners while Wikipedia is used more as a reference resource. This does not only lead to comprehensive coverage of carefully selected concepts that are considered relevant by different authors but also to a specific structure that our baseline extraction method exploits, namely a didactic sequencing of topics. More precisely, the authors are expected to first introduce concepts that later-introduced concepts build upon. Another useful structure in textbooks is the index. As a rule, the index is developed by specially trained professionals and provides a comprehensive list of concepts defined, with indications of their locations where they are introduced and, in some cases, also on which pages they appear again.

We selected a corpus of ten suitable textbooks on linear algebra with freely available PDF files (in some cases after optical character recognition of a printed source) that are listed in Appendix A. The books comprised 584 pages on average (652,297 characters) and disposed of a median of 382 index entries (and a mean of 485 due to one book with 1668 entries).

First, the textbooks were converted from PDF to unformatted text. To identify all concepts of interest covered in a book, the pages containing the index were identified and each index entry with its corresponding pages was extracted in a semi-manual fashion. This required setting parameters such as the title, position of the text columns, and the types of nesting and delimiters. To enable comparing the knowledge graph across the baselines, we decided to use the set of Wikipedia articles as the set of possible concepts, meaning that only a Wikipedia article title can serve as a node in the knowledge graph. This required to map each index entry to the closest Wikipedia article title. For this disambiguation, the index entry was introduced into the Wikipedia search engine suffixed with the field term `(Mathematics)` to hint at the general area. Due to the high diversity of the resulting ten articles, their titles were string-matched with the search term (excluding the suffix) using the Levenshtein distance (provided by the Python library `levenshtein`)[3]. A small distance indicates similarity. The closest article title (excluding disambiguation pages) was the result of our *Wikisearch disambiguation*.

For dependency extraction, we came up with two distinct strategies. The first is solely based on indices whereas the latter also uses the full text. The first strategy is called *order pruning* and is focused on unlikely dependencies on

[2] https://beautiful-soup-4.readthedocs.io/en/latest

[3] https://pypi.org/project/Levenshtein

the graph. It checks whether two concepts are introduced at least twice in a specific order and rules out the dependency that is not suggested by this order. For example, if concept A is introduced by at least two different authors before concept B, we deem A unlikely to depend on B. To create a sparse graph, this strategy requires a massive amount of books, which was not available to us, but it can help to prune dependencies and refine a candidate graph.

This candidate graph is the output of the second heuristic, which we called *common introductory usage*. This is based on a similar rationale as the Wikipedia heuristic because it figures out which other concepts are used to introduce a new concept. The first and rather structured source of information is the index, again. Some books list multiple pages for index entries, indicating that the concept occurs again at a later point in a significant way. If concept A appears for the second time on the first page for concept B, A is likely to be a dependency of B. However, we found the amount of evidence available through indexes to be insufficient and decided to analyze the full texts of the books. Here, recognizing the referenced concepts plays a crucial role because not every word is a concept of interest and not every concept occurs with its canonical title in natural text.

One convenient method for entity recognition is provided by the Wikifier (Brank et al., 2017). The underlying algorithm looks for links in the entire Wikipedia with the link text matching a word in the input. Naturally, the same text can link to different articles, which requires disambiguation based on the context. Wikifier solves this by creating a graph based on all potential links and uses a concept-link matrix that reflects concept similarities to calculate the score for each link target. We restricted links to correspond to a link text of at least three characters to get rid of abbreviations that were contained in formulas randomly. We used a minimum link score of 0.00005 as a threshold after a manual inspection which aimed at a good recall of links but also good precision. The Wikifier was called via its online API[4] since its source code is not available. All detected link targets on the first page of an index entry serve as candidate dependencies which are then ranked by the following three criteria: the number of books in which it occurs in the area of introduction, the average number of occurrences in these books, and the score of the link. As with the Wikipedia dependencies, only the five highest-ranking candidates are preserved as dependencies in the knowledge graph. This procedure was carried out for all index entries across all books.

The final graph is the union of dependencies identified by the two common introductory usage sources (indices and full text), pruned by the unlikely dependencies from order pruning.

2.3 Dependency Extraction from LLM

Due to the closed-source character of the very powerful and successful language model GPT3 by OpenAI (Brown et al., 2020) and the unavailability of an API for its most recent application ChatGPT at project time, we decided to use the open-source model BLOOM by BigScience (BigScience, 2023). BLOOM is a joint effort by hundreds of researchers around the world, is based on the same Transformer architecture as GPT3, and has a comparable amount of parameters. Because of resource

[4] https://wikifier.org/info.html

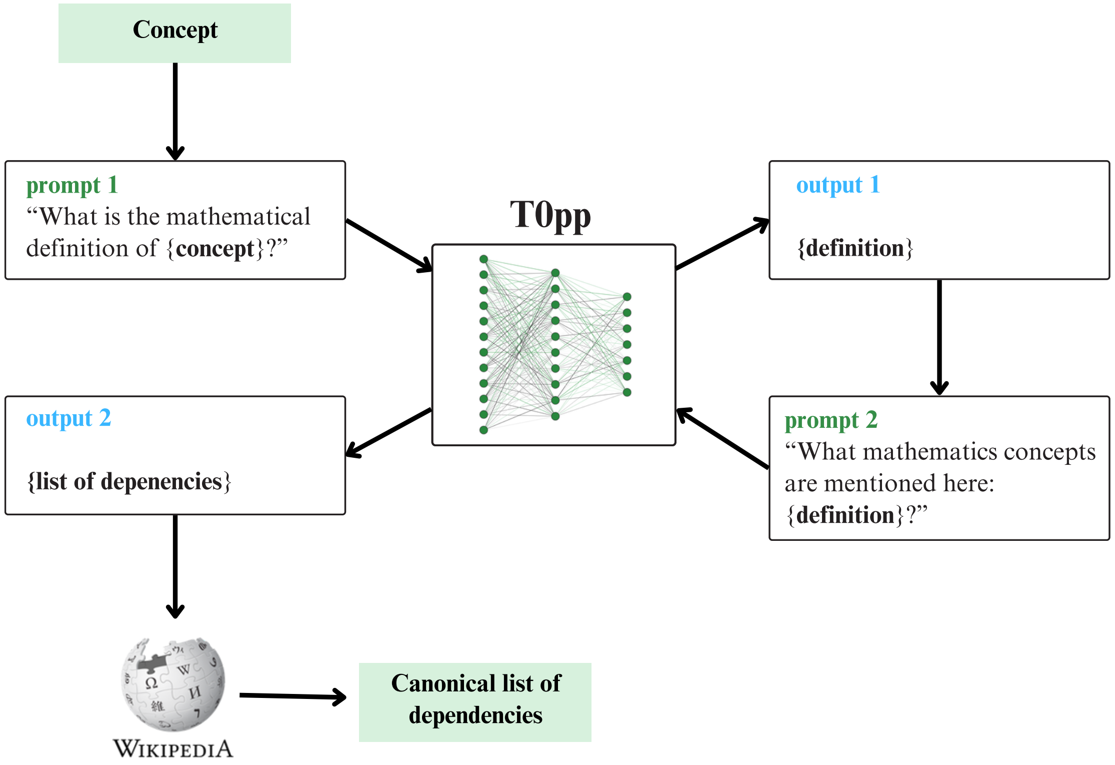

Concept
prompt 1
“What is the mathematical definition of {concept}?”
T0pp
output 1
{definition}
prompt 2
“What mathematics concepts are mentioned here: {definition}?”
output 2
{list of depenencies}
WIKIPEDIA
Canonical list of dependencies

Figure 1: The diagram illustrates the process of concept dependency extraction from the LLM (T0pp) via our output re-feeding prompting procedure.

limitations and our specific use case, we used a derived model called *T0pp*[5], which only has 11 billion parameters and is fine-tuned for zero-shot task generalization, especially for question answering. The dataset used for training T0pp is also available under the name P3 (Public Pool of Prompts)[6]. We set up the language model on the cloud infrastructure of the *Excellence Cluster ML: New Perspectives in Science* and ran it in generation mode (no training) with a maximum response length of 50 tokens on a node with up to 36 CPU cores of type Intel Xeon Gold 6240 CPU @ 2.60GHz. The model consumed approximately 45GB of RAM and a typical request took five to ten seconds to be answered. To save time loading the model into memory, we set up an HTTP server to conveniently query the model via GET requests from our local machines.

Based on our manual inspection of our chosen and other LLMs' responses such as Galactica by Taylor et al. (2022), we designed the following procedure to extract dependencies from the knowledge base of T0pp, which we call *output re-feeding* (see also Fig. 1). It is based on the assumption explained previously that understanding a concept requires understanding other concepts involved in its definition. Given a concept of interest, we therefore first prompt the model for a definition of the concept: "What is the mathematical definition of {concept}?" The response is defined as {definition} and involved in the second prompt: "What mathematical concepts are mentioned here: {definition}". The model

[5] https://huggingface.co/bigscience/T0pp

[6] https://huggingface.co/datasets/bigscience/P3

usually responds with a comma-separated list whose items we disambiguate in the same way as the index entries in the textbook baseline, i.e. via Wikisearch. Finally, self-dependencies are discarded. This procedure is run for the same set of concepts as the Wikipedia baseline.

2.4 Manual inspection

As indicated in our design, this educational problem requires also a qualitative evaluation. We hence developed a dashboard via the Python graphing library Plotly[7] (see Fig. 2). The application provides a convenient environment to explore the inferred CDGs. It allows displaying the concept dependencies graphs for three knowledge sources for different depths of dependencies (e.g. first- and second-order dependencies). The dashboard makes the output more approachable to educators and may facilitate communication between them and CDG developers.

2.5 Manual evaluation

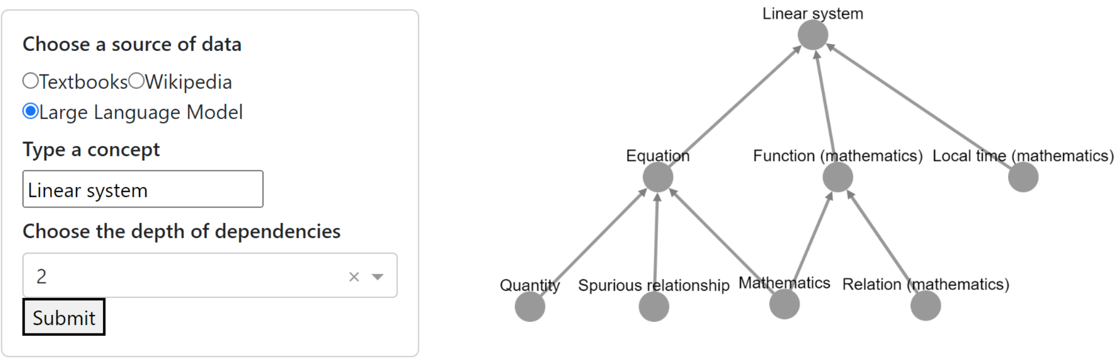
A quantitative perspective was directed at how good our baselines are by human standards, and to which extent humans agree on dependencies at all. The novel characteristic of our baselines and the resulting uncertainty about their appropriateness to evaluate the LLM-based performance motivated us to evaluate all three knowledge graphs independently before checking the LLM graph against the two baselines. Therefore, four raters were each presented with 100 concepts, which were sampled randomly, and one random dependency from each graph (300 ratings in total). Raters were asked "Is understanding the dependency necessary to understand the concept? (Yes/No)". The source of the dependency (i.e. which graph it belongs to) was blinded during the human evaluation. The ratio of correct dependency predictions of each method gives an estimate of the precision (true positives among all true predictions). We only evaluated the quality of dependencies for concepts that occurred at least in 50% of the textbooks to ensure that there is enough data for the textbook baseline and that we assess the quality for important concepts only. For the human ratings, we calculated Cohen's Kappa (Cohen, 1960) as a measure of interrater reliability. This should not only guarantee the confidence of the performance evaluation but also assess the difficulty of the problems for human raters.

As Cohen et al. (2023) pointed out for an open information extraction problem, the precision is of higher interest if the information extraction results in a small number of facts. The manual raters would theoretically need to judge millions of Wikipedia articles for us to be able to estimate the recall in a reliable way. The design of our information extraction procedures inherently reflects the goal of a sparse graph to not overwhelm users with useless information.

2.6 Automated evaluation

Agreement of the LLM with the baselines is the only scalable measure because the baselines can be created automatically and hence be used to evaluate large numbers of LLM outputs or even fine-tune the model for this specific downstream task. For this evaluation, we assumed our baselines to be the ground truth, i.e. all dependencies to be correct. The first-order agreement is fulfilled for a concept dependency extracted from the LLM if it is

[7] https://plotly.com/python/



Choose a source of data

○Textbooks○Wikipedia

●Large Language Model

Type a concept

Linear system

Choose the depth of dependencies

2

Figure 2: A screenshot of the interactive application developed for manual inspection of dependencies.

a direct dependency for this concept also in the baseline. The second-order agreement is fulfilled if a dependency is among the direct dependencies or the dependencies of dependencies in the baseline. For first and higher-order consistency, we can report precision calculated as follows:

$$\text{precision}_{C_i} = \frac{|D_{1,LLM} \cap D_{i,baseline}|}{|D_{1,LLM}|}$$

where C_i is a concept, $D_{1,LLM}$ the predicted dependencies and $D_{i,baseline}$ the baseline dependencies up to the i-th order. To be able to judge if the baseline is a good indicator of the CDG quality, the precision should be comparable to the manual evaluation.

3 Results

Using Wikipedia disambiguation, we identified 1,464 distinct concepts in the textbooks' indices. Table 1 summarizes the extracted graph in terms of distinct nodes (that emerged from extracting dependencies) and the average number of dependencies per concept. The Wikipedia CDG includes the largest number of distinct concepts although it does not predict more dependencies than the LLM on average. The textbook method predicts the largest average number of dependencies although the number of distinct concepts is only slightly larger than in the LLM graph.

Graph	N Nodes	Avg. N Edges
Wikipedia	3,041	2.91
Textbook	1,914	5.66
LLM	1,823	2.75

Table 1: Summary of the graphs in terms of distinct nodes and the average number of dependencies per concept.

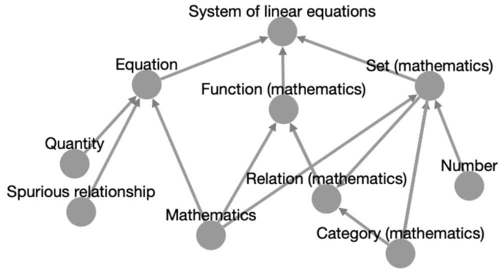


Figure 3: The dependencies of System of linear equations, extracted from T0pp and manually rearranged for better readability.

3.1 Case Study

As an example, we choose the concept "System of linear equations". The visualization of the LLM-based graph (Figure 3) shows that each node has two to three dependencies and that the concepts quickly become abstract. All three direct dependencies depend on "Mathe-

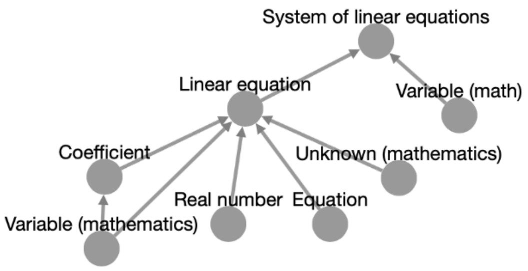


Figure 4: The dependencies of System of linear equations, extracted from Wikipedia and manually rearranged for better readability.

matics" (which was not filtered as a category, in contrast to the Wikipedia baseline). The dependencies "Number" and "Quantity" are also abstract entities, known to learners at a very early stage. Despite these flaws, it is hard to reject most of the dependencies as entirely wrong. Only, "Spurious relationship" does not have to be comprehended to understand "Equation", which, in turn, is absolutely necessary to understand the concept of "System of linear equations".

Figure 4 shows the dependencies for the same concept but extracted from Wikipedia. The overall number of first- and second-level dependencies is comparable, which might be caused by undetected dependencies for "Variable (math)". This dependency is obviously an undetected duplicate of "Variable (mathematics)". In contrast to the LLM output, we see that the two direct dependencies are more specific. E.g., we observe the dependency chain "System of linear equations" -> "Linear equation" -> "Equation", whereas T0pp directly outputs "Equation". Also "Real Number" is more specific than "Number", even though it may depend on the pedagogical focus if this specificity is required to understand a linear equation. The inclusion of the certainly useful dependencies "Coefficient" and "Unknown" might indicate that Wikipedia has a better recall than the language model.

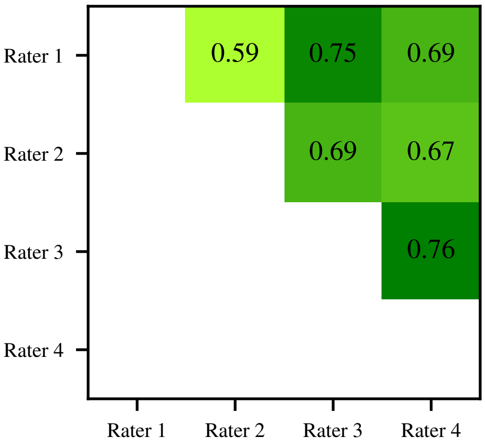


Figure 5: Cohen's Kappa indications of interrater reliability for concept dependency judgements. The coefficient is symmetric and ranges from 0 to 1.

3.2 Manual evaluation

The four raters showed pairwise agreements indicated by Cohen's kappa coefficient of at least 0.6 which is typically interpreted as substantial agreement (see Figure 5). However, there is no coefficient exceeding 0.8, which indicates that dependency recognition is also a hard problem for humans.

Figure 6 summarizes the individual raters' judgments per method as well as their mean. The Wikipedia graph performs best with around three out of four dependencies validated as actual dependencies. The LLM-based graph had an average precision of almost 50%, whereas the textbook-based method was rated as correct in only about one third of the cases.

3.3 Automated evaluation

The quality of the LLM's output judged by the two baselines is plotted in Figure 7, which provides the distribution of the precision metrics over the concepts for both baselines. The results are consistent with the fact that the Wikipedia ground truth should have a higher

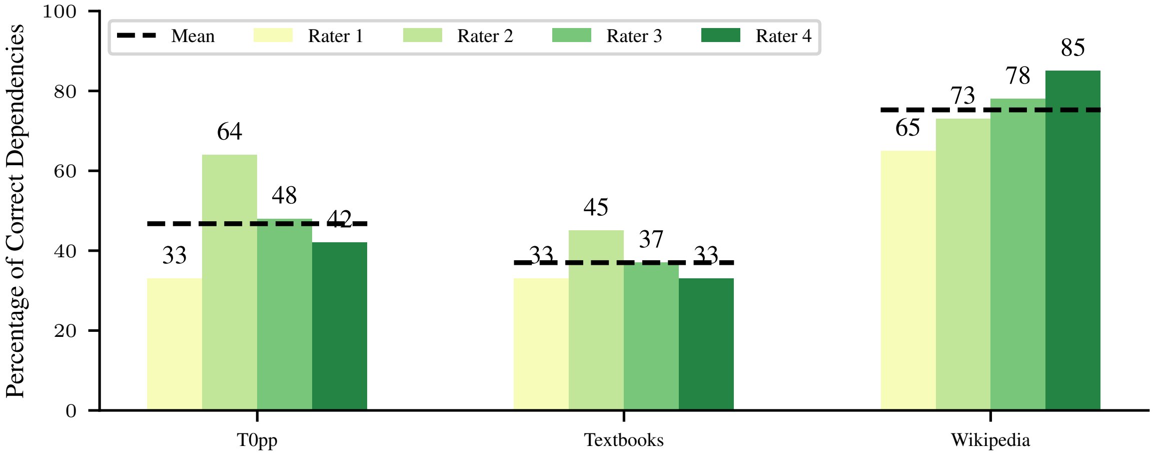


Figure 6: Results of the manual evaluation for all methods and raters. Wikipedia outperforms T0pp and Textbooks in terms of the precision of the extracted graphs.

quality of facts and hence should true predictions detect more reliably. Adding the second-order dependencies yields a higher precision, as expected. However, the third-order dependencies for Wikipedia dependencies do not add much compared to the textbooks. This might indicate that Wikipedia and the LLM do output dependencies at the same abstractness level whereas textbooks have a more fine-grained level structure (e.g., predicting square matrix as a prerequisite of determinant instead of matrix). Overall, the precision is rather low.

4 Discussion

In terms of an explicit knowledge graph for concept dependencies, we can know what an LLM can reveal about its internal representation with a reasonable prompt strategy. The novel method, output re-feeding, yields rapidly a concise list of concepts from T0pp, which was mapped to Wikipedia articles and evaluated. Due to the limited number of returned concepts and introduced ambiguity in this mapping, the resulting graph is unlikely to be comprehensive. This coincides with previous research showing that LLMs are still imperfect when creating such graphs (Hwang and Bhagavatula, 2021). Nevertheless, the strategy is promising, especially for its simplicity and in light of anticipated future developments of LLMs.

Moreover, we employed a simple heuristic that uses the free encyclopedia Wikipedia as an information source to obtain an automatic training set for fine-tuning LLMs for the downstream task of prerequisite mining. This coincides with the method of knowledge graph extraction suggested by West et al. (2021). Although not perfect, the quality of the knowledge graph might be enough to teach a model about the characteristics of prerequisites. This would enable it to discover textual relationships that indicate this relation between concepts. Moreover, it can bootstrap a recommender system for dependencies that is based on the success of user learning on the graph, which is strongly mediated by the order of presentation of learning items.

Our method for textbook knowledge graph extraction performed worst and we would argue against its use for fine-tuning in its current form. Due to the complex pipeline, we have not yet identified the crucial points for the loss

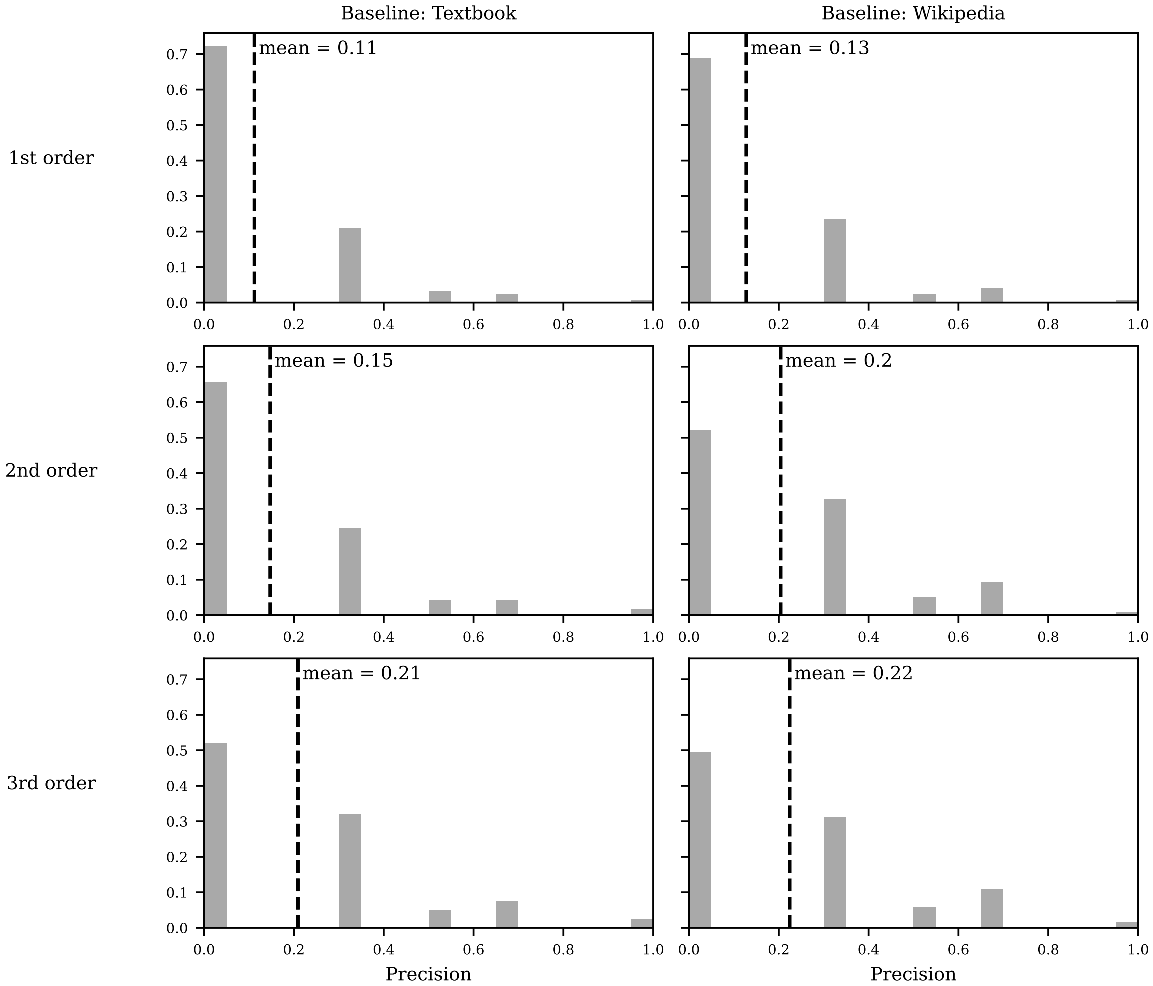


Figure 7: Distribution of the concepts' precision for the LLM with respect to the two baselines. The metric is calculated for the first-, second-, and third-order baseline dependencies.

of quality because textbooks are specifically designed to be ordered in reasonable instructional sequences, which let us expect a much higher quality of the extracted dependencies.

4.1 Limitations

In contrast to Wikipedia-based extraction, textbooks and T0pp are not inherently restricted to output a set of Wikipedia articles as dependencies. The required mapping to Wikipedia articles is error-prone for two reasons. First, the concept granularity might not exist in Wikipedia (for example there is only a joint article on Eigenvalue and Eigenvector). Second, the disambiguation might fail because the combination of Wikipedia's search engine and the subsequent string matching do not identify the closest possible article. We did not quantitatively evaluate the disambiguation quality. Some handpicked instances revealed several problems. E.g., permuting whole words leads to a high Levenshtein distance which is error-prone. This might be a reason for the weaker results of the two methods in human evaluation. One possible way would be to not restrict the concept and dependencies to a certain canonical form, which in turn would make the comparison more difficult. Due to this restriction, we might introduce harmful interdependencies between the methods, which impact the later evaluation. Another solution might be a more sophisticated disambiguation method which is also based on content related to the titles such as embeddings. However, this would be much more computationally costly.

For the textbooks, we developed a rather long pipeline to extract the knowledge graph. In each step (such as PDF to text, index extraction) noise can be introduced and will be propagated until the end of the pipeline. Although our reasoning assumes a certain amount of noise and tries to compensate for it by taking into account multiple books, this might lead to the poor results of the textbook-based graph. We might need order of magnitudes more books to make this approach robust. Especially, the order pruning will greatly benefit from a larger number of sources because the resulting graph of unlikely dependencies has been sparse so far. On the one hand, this is due to the common topic of all books, that will have some concepts always appear at the beginning and hence contribute less information about them. On the other hand, some concept pairs only appeared in one book and we can't clearly exclude a dependency with this little information about them.

Another issue is related to our extraction strategy and the subsequent evaluation. We only judged direct dependencies in the human evaluation and second and third-order dependencies when comparing the LLM to the baseline. In contrast, we did not evaluate the global graph structure in terms of its topology, a task for which several metrics have been proposed (Wills and Meyer, 2020). In our case, this is not reasonable because we did not recursively extract dependencies. Instead, we stopped after the direct dependencies. Higher-order dependencies are in our design only a consequence of dependencies, that are part of the initial set of concepts. Although the recursive extraction of dependencies makes especially sense if prerequisite chains are of interest, it would be unclear when to stop the chain. In the LLM-based CDG, the concepts already have become quite abstract and would become even more abstract.

Apart from these rather technical issues, the educational problem of finding prerequisite concepts might have to be reconsidered after our initial attempts to solve it. More precisely, a prerequisite structure is likely to depend on the individual learner or on the pedagogical approach taken by an instructor. A line of research, commencing with the theory of *cognitive entry behaviors* by Bloom (1976), has underlined a complex network of individual factors regarding "prior knowledge" involved in successive learning performance (Dochy et al., 2002). This serves as an argument against universal prerequisite structures, which originate from an ontological view of the subject matter. As a consequence, the CDG might be extended to support multiple pathways to learning a concept instead of one potential way.

Another theoretical question is directed at the nature of the LLM's knowledge. The research goal to make its internal knowledge representation explicit assumes some implicit structure. This assumption is likely to be caused by the overwhelming performance in various downstream tasks, which humans cannot think of without having a suitable representation of the underlying structures of the world. One could argue that the interaction with the model only via the textual completion of textual input is analog to Plato's cave, where the true origin of vivid shadows on the wall forever remains unexplained. As long as we are not able to trace back the origin of the model's output, we cannot judge with absolute certainty if it only echoes similar content that is included in the training corpus and incomprehensibly encoded in its parameters or if this corresponds to true generalization. One would have to design experiments to rule out the character of the model's responses as fused outputs from various sources but prove an underlying structure.

4.2 Future work

Due to the scalable character of the Wikipedia-based evaluation, the approach should be scaled and re-evaluated. This could also happen using more recent or future LLMs such as GPT-4 and LLaMA (Touvron et al., 2023). The prompting strategy can also be adapted to directly ask for graph edges. This method is based on a GitHub demo by Varun Shenoy[8]. Appendix B shows an example prompt and response for this strategy. In the example, the type of relations is not restricted. Therefore, it includes relations such as "function of" or "denoted as". Mapping them to the dependency relation is necessary and could be obtained by categorizing them into dependency-indicating relationships and other types. Disambiguation of the entities might also pose a challenge. Overall, the amount of extracted relations is remarkable compared to the length of the input. This shows that pre-learned knowledge is likely to be incorporated into the response.

A popular criticism of LLMs is their undecidable confidence in the output and the inability to estimate the confidence in their answer (Cohen et al., 2023). Due to our focus on precision, certainty information would help in thresholding only high-quality answers and discarding answers, for which the model does not have enough evidence to claim them definitely. Training a model to output "I don't know" if there is not enough evidence to come up with an answer could overcome this limitation.

[8] https://github.com/varunshenoy/GraphGPT

Another potential advancement is the interactive connection to a knowledge source. If the LLM could use Google Search or browse through Wikipedia to solve the quest for prerequisites of a specific concept, the quality of the answers might also rise. In this case, the pre-trained model could build on its language comprehension ability on the one hand, but also actively request more information in the case of missing evidence. Unfortunately, this might require too much computation time to be suitable for practical use.

5 Conclusion

Concept Dependency Graph extraction from text is possible with a moderate quality, either using a knowledge source such as Wikipedia or by prompting the Large Language Model directly. It remains challenging due to the ambiguity of referenced entities and the end-user-facing input formats. Fine-tuning an LLM with the Wikipedia-based baseline CDG may be a promising endeavor. Further investigation is required to extract a higher-quality CDG from textbooks to exploit the incorporated expert knowledge more effectively. Future advances in generative language models show a great potential to directly output CDGs. Overall, the educational problem is complex due to individual differences in learning and didactic approaches.

6 Acknowledgements

We sincerely thank our project supervisors, Álvaro Tejero-Cantero and Hanqi Zhou from the Machine Learning Science Colaboratory at the University of Tübingen, for their support and advice.

References

BigScience. 2023. Bloom: A 176b-parameter open-access multilingual language model.

Benjamin S. Bloom. 1976. *Human characteristics and school learning*. McGraw-Hill, New York City, NY.

Janez Brank, Gregor Leban, and Marko Grobelnik. 2017. Annotating documents with relevant Wikipedia concepts. *Proceedings of SiKDD*.

Tom B. Brown, Benjamin Mann, and Nick Ryder. 2020. Language models are few-shot learners.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Roi Cohen, Mor Geva, Jonathan Berant, and Amir Globerson. 2023. Crawling the internal knowledge-base of language models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*.

Filip Dochy, Catherine De Rijdt, and Walter Dyck. 2002. Cognitive prerequisites and learning: How far have we progressed since Bloom? Implications for educational practice and teaching. *Active Learning in Higher Education*, 3(3):265–284.

Fabio Gasparetti. 2022. Discovering prerequisite relations from educational documents through word embeddings. *Future Generation Computer Systems*, 127:31–41.

Jonathan Gordon, Linhong Zhu, Aram Galstyan, Prem Natarajan, and Gully Burns. 2016. Modeling concept dependencies in a scientific corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 866–875, Berlin, Germany. Association for Computational Linguistics.

Jena D. Hwang and Chandra Bhagavatula. 2021. (COMET-)ATOMIC 2020: On symbolic and neural commonsense knowledge graphs. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence*, pages 6384–6392.

Irene Li, Vanessa Yan, and Dragomir Radev. 2021. Efficient variational graph autoencoders for unsupervised cross-domain prerequisite chains.

Chen Liang, Zhaohui Wu, Wenyi Huang, and C Lee Giles. 2015. Measuring prerequisite relations among concepts. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1668–1674.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *CoRR*.

M. David Merrill. 1983. Component display theory. *Instructional-design theories and models: An overview of their current status*, 1:282–333.

Gary R. Morrison, Steven J. Ross, Jennifer R. Morrison, and Howard K. Kalman. 2019. *Designing effective instruction*. John Wiley & Sons, Hoboken, New Jersey.

Philip I. Pavlik Jr., Hao Cen, Lili Wu, and Kenneth R. Koedinger. 2008. Using item-type performance covariance to improve the skill model of an existing tutor. In *Proceedings of The First International Conference on Educational Data Mining*.

Hao Sun, Yuntao Li, and Yan Zhang. 2022. ConLearn: Contextual-knowledge-aware concept prerequisite relation learning with graph neural network. In *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*, pages 118–126. SIAM.

Partha Talukdar and William Cohen. 2012. Crowdsourced comprehension: predicting prerequisite structure in wikipedia. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 307–315.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A large language model for science.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and efficient foundation language models.

Annalies Vuong, Tristan Nixon, and Brendon Towle. 2011. A method for finding prerequisites within a curriculum. In *Proceedings of The Fourth International Conference on Educational Data Mining*, pages 211–216.

Shuting Wang and Lei Liu. 2016. Prerequisite concept maps extraction for automatic assessment. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 519–521.

Shuting Wang, Alexander Ororbia II, Zhaohui Wu, Kyle Williams, Chen Liang, Bart Pursel, and C Lee Giles. 2016. Using prerequisites to extract concept maps from textbooks. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*, pages 317–326.

Peter West, Chandra Bhagavatula, Jack Hessel, Jena D. Hwang, Liwei Jiang, Ronan Le Bras, Ximing Lu, Sean Welleck, and Yejin Choi. 2021. Symbolic knowledge distillation: from general language models to commonsense models.

Peter Wills and François G. Meyer. 2020. Metrics for graph comparison: A practitioner's guide. *PLOS ONE*, 15(2):1–54.

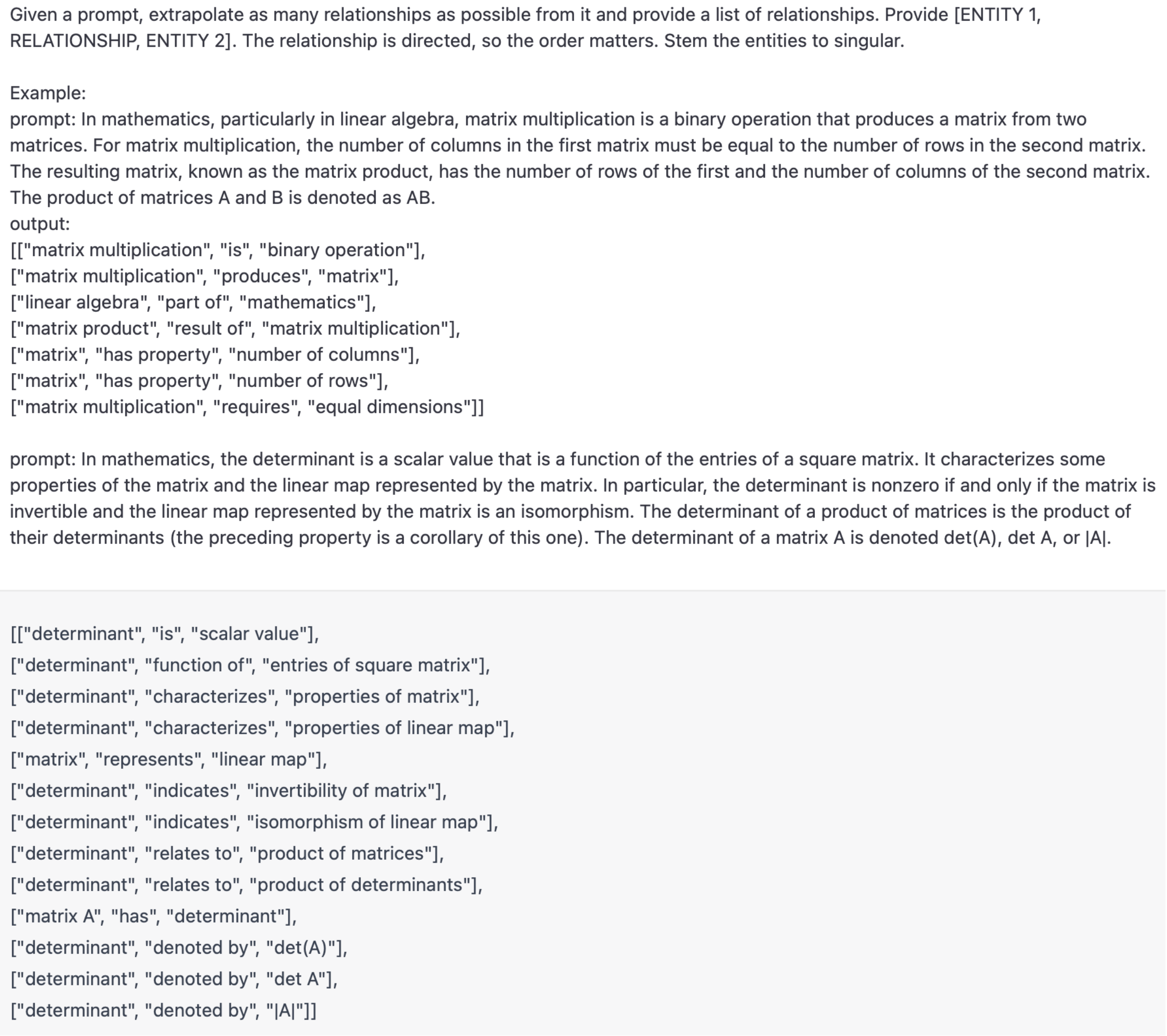
Y. Yang, H. Liu, J. Carbonell, and W. Ma. 2015. Concept graph learning from educational data. In *Proceedings of the Concept Graph Learning from Educational Data*, pages 159–168.

A List of Textbooks

1. Sheldon Axler. 2015. Linear Algebra Done Right. Springer.
2. Robert A. Beezer. 2015. A First Course in Linear Algebra. University of Puget Sound. URL: https://open.umn.edu/opentextbooks/textbooks/5
3. David Cherney, Tom Denton, Rohit Thomas and Andrew Waldron. 2013. Linear Algebra. URL: https://www.math.ucdavis.edu/~linear/
4. Sean Fitzpatrick. 2020. Math 1410. Elementary Linear Algebra. University of Lethbridge. URL: https://www.cs.uleth.ca/~fitzpat/Textbooks/Math1410_ebook.pdf
5. Jim Hefferon. 2020. Linear Algebra. URL: https://joshua.smcvt.edu/linearalgebra/
6. Kenneth Hoffman and Ray Kunze. 1971. Linear Algebra. PRENTICE-HALL, INC., Englewood Cliffs, New Jersey.
7. K. Kuttler. 2017. A First Course in Linear Algebra. Lyryx. URL: https://lyryx.com/first-course-linear-algebra/
8. Dan Margalit and Joseph Rabinoff. 2019. Interactive Linear Algebra. Georgia Institute of Technology. URL: https://services.math.duke.edu/~jdr/ila/ila.pdf
9. W. Keith Nicholson. 2019. Linear Algebra with Applications. Lyryx. URL: https://lyryx.com/linear-algebra-applications/
10. Sharon North and Jay Abramson. 2020. College Algebra with Corequisite Support. OpenStax. URL: https://openstax.org/details/books/college-algebra-corequisite-support-2e

B GPT-4 Showcase

This appendix showcases an idea for future developments of prompt-based graph extraction (see Section4.2).



Given a prompt, extrapolate as many relationships as possible from it and provide a list of relationships. Provide [ENTITY 1, RELATIONSHIP, ENTITY 2]. The relationship is directed, so the order matters. Stem the entities to singular.

Example:
prompt: In mathematics, particularly in linear algebra, matrix multiplication is a binary operation that produces a matrix from two matrices. For matrix multiplication, the number of columns in the first matrix must be equal to the number of rows in the second matrix. The resulting matrix, known as the matrix product, has the number of rows of the first and the number of columns of the second matrix. The product of matrices A and B is denoted as AB.
output:

```
[["matrix multiplication", "is", "binary operation"],
["matrix multiplication", "produces", "matrix"],
["linear algebra", "part of", "mathematics"],
["matrix product", "result of", "matrix multiplication"],
["matrix", "has property", "number of columns"],
["matrix", "has property", "number of rows"],
["matrix multiplication", "requires", "equal dimensions"]]
```

prompt: In mathematics, the determinant is a scalar value that is a function of the entries of a square matrix. It characterizes some properties of the matrix and the linear map represented by the matrix. In particular, the determinant is nonzero if and only if the matrix is invertible and the linear map represented by the matrix is an isomorphism. The determinant of a product of matrices is the product of their determinants (the preceding property is a corollary of this one). The determinant of a matrix A is denoted det(A), det A, or |A|.

```
[["determinant", "is", "scalar value"],
["determinant", "function of", "entries of square matrix"],
["determinant", "characterizes", "properties of matrix"],
["determinant", "characterizes", "properties of linear map"],
["matrix", "represents", "linear map"],
["determinant", "indicates", "invertibility of matrix"],
["determinant", "indicates", "isomorphism of linear map"],
["determinant", "relates to", "product of matrices"],
["determinant", "relates to", "product of determinants"],
["matrix A", "has", "determinant"],
["determinant", "denoted by", "det(A)"],
["determinant", "denoted by", "det A"],
["determinant", "denoted by", "|A|"]]
```

Figure 8: A screenshot of an interaction with GPT-4 to directly create a knowledge graph.