Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 142 (2026) 310–315

www.elsevier.com/locate/procedia

36th CIRP Design Conference (CIRP Design 2026)

Language Models and Design Theory for Efficient Engineering Knowledge Retrieval

Tomas Mikael Elmgren[a], Axel Götling[b], Adam Tiger[b], Marvin Carl May[c], Haluk Akay[d,*]

[a]School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden
[b]Nordisk Elkraftteknik AB, Stockholm, Sweden
[c]School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore
[d]Faculty of Mechanical Engineering, Delft University of Technology, Delft, the Netherlands

* Corresponding author. *E-mail address:* h.j.akay@tudelft.nl

Abstract

Large Language Models (LLMs) are increasingly utilized as a tool for connecting engineers to design knowledge because of the scale and speed with which they automate information retrieval tasks during early-stage design. However, if engineering documentation is input to LLMs without preserving design-specific structures of information, this risks unexplainable and inefficient knowledge retrieval. In this work, we incorporate design theory into the process by which design knowledge is represented for storage and retrieval by LLMs. We construct design-inspired data representations of engineering knowledge with the aim of improving their explainability and performance. We demonstrate these methods on an industry case study of the design of high voltage power transmission structures. We present an implementation integrating LLMs into a design theory-based framework for knowledge representation. We validate the system experimentally with human subject matter experts and conduct a time complexity analysis of the search efficiency of the design theory-based representations in comparison to baseline methods. We conclude that design theory-based representations result in improved computational search efficiency, but that establishing ground truth in engineering design for validation purposes is challenging.

Keywords: design representation; large language models; information retrieval

1. Introduction

Engineering designers have access to an unprecedented scale of prior knowledge upon which to base early-stage decision-making due to widespread digitalization of user needs identification, concept development, testing, and post hoc documentation. To keep pace with available knowledge, information retrieval tools have evolved in parallel from keyword-matching methods to semantic search technology powered by Large Language Models (LLMs) trained to map queries to answers based on meaning derived from context.

LLMs are increasingly utilized in early-stage design where a substantial share of information is expressed with language. LLMs have been applied to infer user's goals and fundamental needs [1], stimulate concept-stage ideation [2] and integrate product design with data science [3]. LLMs are also now used in design documentation and dataset construction, such as the synthesis of textual datasets of functional requirements and solutions [4] and labeling mechanical components in Computer-Aided Design (CAD) databases with natural language descriptions [5].

For the application to design knowledge retrieval, a paradigm for implementing LLM-based semantic search involves representing documents as a list of segmented text chunks, encoding the meaning of each chunk into a vector embedding, and then using retrieval-augmented generation

10.1016/j.procir.2026.05.266

(RAG) to formulate answers based on semantically similar chunks [6,7]. The key benefit of this approach is the decoupling from matching queries to keywords and resulting gains in information retrieval efficiency.

LLM applications to design suffer because, despite the prevalence of textual data, engineering design is not a natural language akin to what LLMs are trained upon, and it must be computationally represented through an appropriate method before integration with Artificial Intelligence (AI)-based technology. Prior to the advent of contemporary AI, experts of engineering design did formalize repeatable innovation methods through design theories. Design theory provides a framework with which to represent the identification of meaningful problems and conceptualize functional solutions.

We propose that design theory contains elegant structures for representing design knowledge for computational storage and retrieval. In this work, we present a system for constructing such design theory-inspired representations of design and then implement and experimentally validate the method on an industry case study. The source code for implementing this proposed method on any design data is available on GitHub[8].

2. Background and Related Work

This section provides background information in the domains of both engineering design theory and natural language processing and introduces related research at the intersection of these fields working to apply LLMs to design knowledge retrieval.

2.1. Engineering Design Theory

Engineering design theories have been proposed with the goal of making the creative human practice of design a more repeatable endeavor. Various theories such as TRIZ [9], Axiomatic Design [10], and Product Design and Development [11] exist, each with strengths and drawbacks depending on context. In this work, we primarily draw upon Axiomatic Design (AD). AD introduces a useful method for representing the key elements of a design in a structure which is relevant for computation [12].

2.2. Natural Language Processing

The field of natural language processing (NLP) is concerned with representing language for computational purposes. Unlike numerical or digital images, textual data is comprised of characters with no inherent quantitative value apart from alphabetic order. Contemporary statistical language models have been developed to learn the significant features of language from examples of context. These language models, with many parameters, can represent text in vector form within this semantic feature space defining language meaning. These vector embeddings are useful quantitative representations of language for measuring similarity and other applied tasks.

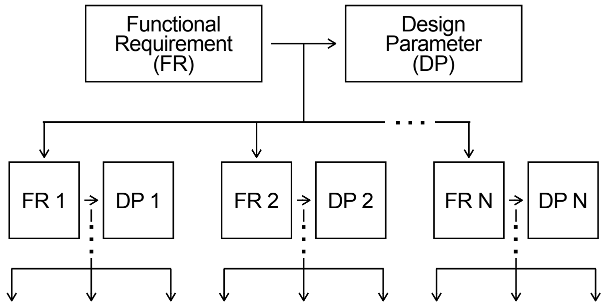

Fig. 1. Hierarchical tree representation of design information

Recent advances in NLP include the development of the "transformer" neural network architecture [13] which decouples the semantic relationship between words from the distance between them in a sequence. Transformer-based LLM applications such as the popularized conversational model ChatGPT [14] have been integrated into design education [15] and manufacturing knowledge management [16], and maintenance ontologies [17] among other engineering applications.

2.3. Design Theory and Large Language Models

Related work has been conducted at the intersection of design theory and LLMs to support engineering design. TRIZ (the Theory of Inventive Problem Solving) is a design theory which has been integrated with LLMs with various methods given the textual nature of its principles. Auto-TRIZ [18] takes as input a problem statement and utilizes an LLM-based framework for generating a solution report through a TRIZ-guided reasoning process.

From the perspective of developing structured representations of engineering documentation for NLP-enabled knowledge retrieval, related work involving the use of knowledge graphs (KGs) has been conducted. Methodology for representing supplier capability data through ontology-driven graph construction processes was developed for the purpose of LLM-based processing [19]. LLMs have also been used for the inverse purpose, to help construct enterprise-level KGs by automating ontological expertise which then constitute the structured repository for knowledge retrieval [20].

We have previously developed methodology [21] for utilizing LLM-based question-answering to take as input a paragraph of free text describing a product, and extractively output the FRs and DPs defining its design. This automated the structured dataset construction task, but the implementation was limited by shorter maximum context lengths of prior LLMs which prohibited the processing of actual full-length design documents. Furthermore, the extractive nature of the question-answering process resulted in extracted text spans which could be ungrammatical language chunks difficult to understand out of context. In this work, we improve upon this method and validate with design documentation from industry.

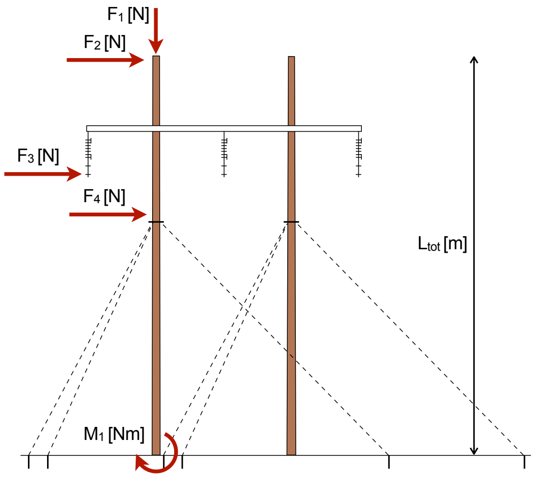

Fig. 2. Supporting structure for high voltage power line

3. Method

We integrate design theory with language models to represent functional knowledge through a structured means. In this section, the case study used for experimentation is introduced, and the method for constructing these data structures is described.

3.1. Industry Design Case: Power Line Design

The methods described in this work are deployed and tested on engineering design documents from the Swedish energy infrastructure design company Nordisk Elkraftteknik AB (NEKTAB). Specifically, the case study focuses on the structural design of power line supports for high voltage (130 – 400 kV) regional transmission lines, shown in Figure 2. Functional requirements for this design case primarily include withstanding mechanical loads due to environmental factors. Such power lines are distributed across Sweden, including in flat open terrain near the sea in the south, over water bodies, and through residential areas. A specifically extreme design environment is in the north of Sweden where variables such as wind, temperature, terrain, snow, and ice impose structural integrity design challenges.

For the purposes of this work, five documents were shared from an internal knowledge base of best practices concerning the design of these high voltage power line structures. This repository had been built incrementally by senior engineers as a guide for early-stage design decision making. The five documents used in this case study describe steps for verifying pole placement in regards to environmental variables measured in field visits and adjusting pole placement to minimize both forces due to lift and insulator deflection angles for various load cases.

The five documents are primarily written in Swedish and cannot be publicly shared in their entirety as the designs are categorized as critical infrastructure and potential targets due to threat of regional conflict. However, redacted excerpts translated to English are shared in examples, and the software implementation of the method developed is available for the public to use with their own documents.

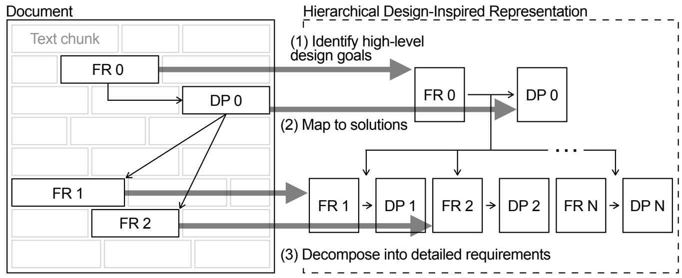

Fig. 3. Procedure for extracting hierarchical tree representations of functional requirements and design parameters from unstructured text

3.2. Constructing Design-Inspired Data Structures

The core module of the proposed system is tasked with taking as input an unstructured textual description of design knowledge, and automatically extracting key information being functional requirements (FRs) and design parameters (DPs) and placing this information appropriately in a hierarchical tree structure inspired by design representation theory.

Key considerations when developing this module were scalability and performance. Despite only applying the method to a limited case study of five documents, internal knowledge bases at engineering organizations documenting decades of design can be large, so methods for restructuring such data at scale must be fully automated to be feasible.

In terms of performance, in order to extract the correct design information with acceptable precision and recall (standard metrics for information retrieval discussed in further detail in Section 4.2), implementing a semantic model capable of identifying functional design information based on language meaning rather than keyword matching was of great importance. Given these considerations, we implement a Large Language Model (LLM)-powered routine to semantically identify high-level design goals, map to physical solutions, and decompose into detailed requirements. The routine is based on the recursive algorithm we developed previously [21], and follows the procedure illustrated in Figure 3. The specific LLM implemented was GPT-4o, a version of the Generative Pretrained Transformer family of models developed by OpenAI. The LLM was conditioned on an instructional prompt explaining the task and then called recursively to utilize prior extracted information to inform more detailed prompts during the extraction process. These exact prompts are enumerated in Table 1. Due to the LLM's tendency to return long lists of requirements, a maximum of 3 highest-level requirements was set during the initialization of the decomposition for each document. A total of 75 FRs and DPs were extracted from the 5 documents.

Table 1. Instructional prompts used to guide LLM-based extractions

Prompt	Intention
You are extracting functional requirements from engineering documents: {context}. Provide a bullet-point list.	Instructional prompt to give context for the extraction tasks.
What is the aim of this document? List all functional requirements in bullet points.	Initialization of decomposition to extract highest-level FRs.
How does this achieve {FR}	Mapping the extracted FRs to DPs, using the previous extracted FR.
What is required for {DP} to achieve {FR}?	Decomposing the previously extracted FR and DP into more detailed FRs.

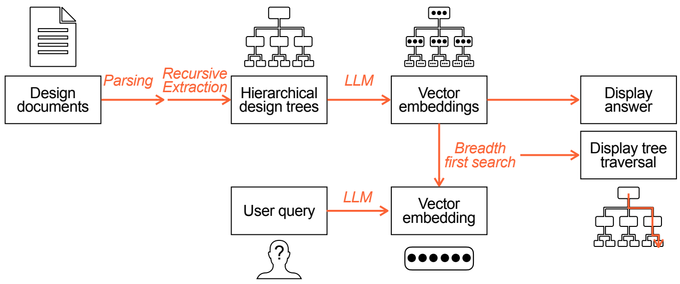

Fig. 4. Information pipeline abstraction for semantically connecting queries to design knowledge

3.3. System-level design knowledge processing

Around the core module for transforming design documents into hierarchical tree data structures, we develop a pipeline for adding, updating, and preserving design knowledge for the actual engineering team in our industrial case study. The overarching information pipeline is illustrated in Figure 4. The initial step involved identifying relevant design documents to include in the corpus of design knowledge. Like many engineering companies, documents in our industry case study are categorized according to organizational teams. In this study, we take 5 documents from the high voltage power line design department. Next, these documents are parsed to remove formatting and nonstandard textual characters and then passed for restructuring into hierarchical trees. The nodes of each tree are encoded into vector embeddings, completing the structured dataset construction process. User queries are then encoded into vector embeddings.

4. Results and Analysis

In this section, we conduct a quantitative theoretical analysis of the search performance of the design theory-inspired data structures, and experimental validation of the LLM-constructed design tree representations with human domain experts.

4.1. Analysis of search performance

We compare the efficiency of information retrieval between the hierarchical design-inspired tree structures and baseline methods of processing documentation with LLMs. A key metric for efficiency of information retrieval is time complexity, typically expressed in Big O notation, representing how much time an algorithm takes to execute.

The baseline method commonly utilized for integrating semantic search capabilities of LLMs with minimal alterations to a knowledge corpus involves the following. Documents are segmented into smaller sentence-level “chunks” of text and each chunk is represented by the LLM as a vector embedding. The vectors and corresponding human-readable text are stored in a relational database, and a retrieval-augmented generation (RAG) model scans the database, comparing to queries to find semantically similar language. For applications in engineering design, where documentation may include up to M number of design-irrelevant text chunks, and N design-relevant candidate answers, performing such a linear scan on an unsorted array has

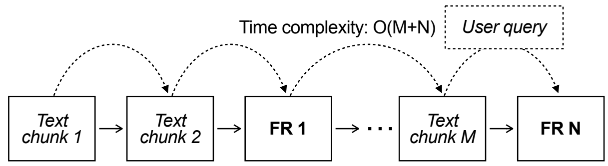

Fig. 5. Time complexity of linear scan of an unstructured design document sparsely containing functional requirements

asymptotic time complexity of O(M+N), which scales unfavorably for larger datasets. The time complexity for this information retrieval baseline is visualized in Figure 5.

The design-inspired hierarchical data structures are constructed by extracting curated information from the design documentation, resulting in some chunks being excluded in the tree. We expect nodes closer to the root to contain higher-level, general design needs, and deeper nodes to contain detailed technical requirements. This structure would allow for various algorithms to improve search performance beyond the baseline linear scan, such as a Breadth-First Search (BFS) combined with heuristics to prune the search space. The BFS method was considered for the theoretical analysis but not implemented for the case study. The time complexity of these methods depends on the dimensions of the constructed tree, but for a balanced tree with N FRs and a depth K, the time complexity would be O(NK), illustrated in Figure 6. Given the sparsity of long-form design documentation, we can assume M >> N and given the depth of trees extracted in the case study, we expect K not to exceed 3, indicating a favorable decrease in time complexity. The limitation of representing the information with this tree structure is that some chunks of text which are relevant may be omitted.

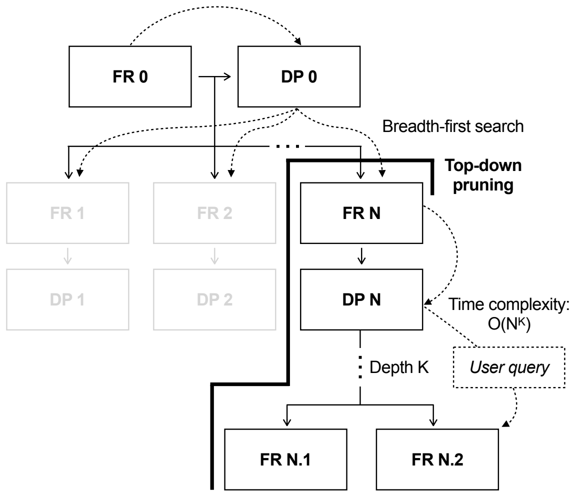

Fig. 6. Time complexity of tree traversal to search design information in a design theory-inspired hierarchical representation

4.2. Experimental validation with experts

A key consideration when transforming unstructured documentation into design-inspired hierarchical tree representations is the performance of the extraction system in terms of its precision and recall. In this case of design dataset construction, precision is measured as the share of functionally relevant information out of the total information retrieved.

Recall is measured as the share of functionally relevant information retrieved out of the total relevant functional information existing in the whole corpus. These two metrics can be calculated respectively using equations (1) and (2), based on the number of True Positives (TP), False Positives (FP) included in the hierarchical design representations, and False Negatives (FN) being information incorrectly left out.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

Ground truth is necessary to estimate these metrics. Therefore, the first validation experiment we conducted involved 2 subject matter experts manually performing the same task as the machine, to identify up to 3 key FRs, map to DPs, and decompose into detailed FRs resulting in the same tree structure for each of the 5 design documents. The subject matter experts were given no time constraints and worked together to complete the task over the course of 2 days.

The trees were compared by measuring the semantic similarity between the machine and subject matter expert-extracted design information. The semantic similarity was estimated based on the cosine similarity between vector embeddings representing each extraction method, obtained from a separate statistical language model [22]. A threshold of 0.4 was taken as the minimum semantic similarity needed to indicate a functional match. Although the documents and extracted trees cannot be entirely declassified due to their descriptions of critical infrastructure design, in Table 2, certain redacted examples of machine extractions and corresponding human extractions, all translated into English with semantic similarity scores are shared.

In total, the machine extractions resulted in 5 trees with 75 total nodes, while the human experts extracted 58 nodes. Between these two extractions, 14 nodes were shared based on the semantic similarity criterion (True Positives). In 9 instances, the human experts identified more nodes on a document's tree (False Negatives). In 61 instances, either the machine extraction did not match the human extraction, or the machine extractions resulted in nodes on a document's tree where human experts did not identify a node (False Positives). The semantic similarity between the 5 machine and human extracted trees are visualized in Figure 7. The resultant precision calculated is 0.187, and recall is 0.609.

Table 2. Examples of machine-extracted design information and corresponding human extractions

Machine extraction	Human expert extraction	Semantic Similarity
Perform a ground assessment at each pole location.	Make ground assessment at each pole position.	0.825
Identify lift in poles	Calculate lift in supporting insulators	0.448
Calculate the [redacted]-dimension (deflection measurement) for angle poles with hanging chains using the [redacted] program.	Calculate [redacted]-dimension	0.367
During the field visit, each pole location is visited, and measurements are taken using the provided code list to supplement existing laser data.	Measure uncertain and new objects and terrain information which wasn't present in the early stage design.	0.260

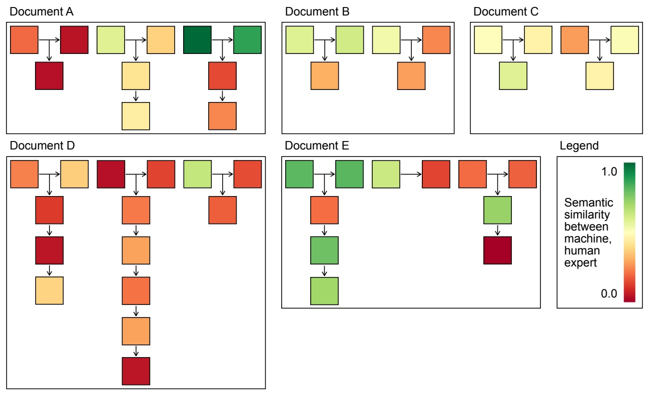

Fig. 7. Validation Experiment #1: comparing machine extractions to human expert extractions in terms of semantic similarity

This validation method results in lower metrics for precision and higher metrics for recall. The limitation of this method is that it assumes the human experts' extraction represents a comprehensive ground truth without omission. Therefore, in cases where the machine extracts a valid node which the human experts did not consider, this is still counted as a False Positive, skewing the precision metric unfavorably.

To better understand the true accuracy of the machine-extracted trees, a second round of validation experiments was performed. In this round, 2 human subject matter experts independently reviewed the machine-extracted trees without consulting each other or referring to the prior human extractions. In this round, the human experts were asked to rate each machine-extracted node as either "valid," "partially valid," or "invalid." The results of this second round of validations are shown in Figure 8. With the partially valid nodes given 0.5 the weight of valid nodes, this resulted in an average share of 0.607 nodes considered valid by the experts, which is significantly higher than the precision metric estimated during the prior round of expert validation. This discrepancy highlights the challenge of establishing ground truth for validating data-driven applications of design theory in industry.

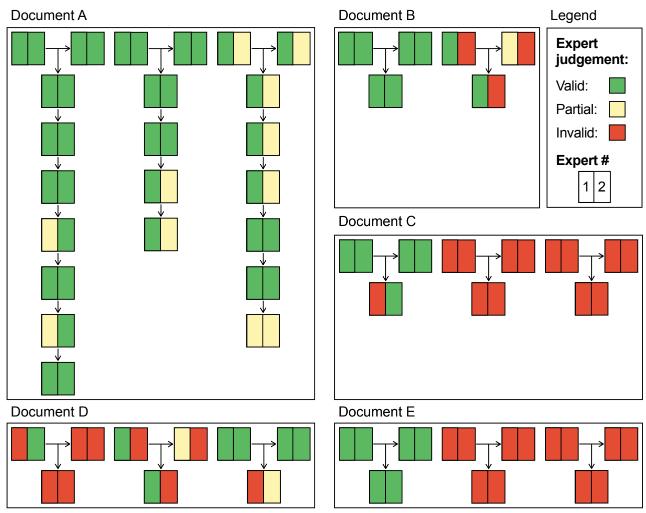

Fig. 8. Validation Experiment #2: human experts judging machine extractions for validity

5. Discussion

Based on the time complexity analysis conducted in this work, the results indicate that the design theory-inspired tree representations of design knowledge allow for superior information retrieval efficiency. The benefits of this representation method would scale with the sparsity of design documents included in the knowledge base. Because the procedure is automated, this system can be deployed to large industrial-scale instances of unstructured design knowledge.

Experimental results with human experts indicate low precision and higher recall for the independent validation study, indicating the LLM-based model returns a high share of the same knowledge found by humans, but that of the returned knowledge there existed many which were different than those found by humans. The second study then found that a high share of the knowledge returned by the model was found to be valid by the humans. The discrepancy between the lower precision and the subsequent validation suggests the model is discovering knowledge which humans independently did not find themselves.

A limitation of this work was the small number of subject matter experts recruited for validation experiments. A larger expert pool would result in a greater number of diverse perspectives, helping to establish ground truth. A limitation of the hierarchical representation method is that it does not guarantee balanced trees, which could result in unfavorable time complexity during information retrieval. Future work will involve further validation with more engineering design cases to explore how design theory-inspired data representations generalize across a wider range of products and services.

6. Conclusion

Textual data is prevalent in early-stage design documentation, making applications of LLMs a natural use case. However, the structure of engineering design does not necessarily match with the language that LLMs are trained upon. Representation of engineering language is required, and design theory has specific framework for facilitating this.

In this work, we present a method for representing unstructured textual documentation as a hierarchical data tree, inspired by design theory. We demonstrate that, for documentation sparse with design information, this tree structure results in more efficient information retrieval than current baselines. We validate the accuracy of the hierarchical representations through validation experiments on design documents from industry and measure the precision and recall of machine extracted trees with respect to human subject matter expert extractions. As LLMs continue to be integrated into more aspects of language use in engineering design, this research is intended to serve as a foundation for grounding AI integration in fundamental design theory.

Acknowledgements
This research is supported by a Digital Futures Industrial Innovation Project.

References

[1] Zhu, Q., Chong, L., Yang, M., & Luo, J. (2025). Reading Users' Minds With Large Language Models: Mental Inference for Artificial Empathy in Design. Journal of Mechanical Design, 147(6), 061401.
[2] Zhu, Q., & Luo, J. (2023). Generative transformers for design concept generation. Journal of Computing and Information Science in Engineering, 23(4), 041003.
[3] Liu, A., Lu, S., Tao, F., & Anwer, N. (2024). Integration of data science with product design towards data-driven design. CIRP annals, 73(2), 509-532.
[4] Qiu, Y., & Jin, Y. (2025). A Method for Synthesizing Ontology-Based Textual Design Datasets: Evaluating the Potential of Large Language Model in Domain-Specific Dataset Generation. Journal of Mechanical Design, 147(4), 041707.
[5] Sun, Y., Li, X., & Sha, Z. (2025). Large Language Models for Computer-Aided Design Fine Tuned: Dataset and Experiments. Journal of Mechanical Design, 147(4), 041710.
[6] Siddharth, L., & Luo, J. (2024). Retrieval augmented generation using engineering design knowledge. Knowledge-Based Systems, 303.
[7] Gammack, J., Akay, H., Ceylan, C., & Kim, S. G. (2022). Semantic knowledge management system for design documentation with heterogeneous data using machine learning. Procedia CIRP, 109, 95-100.
[8] Elmgren, T. (2025). Structured Functional Representation of Design. GitHub. https://github.com/tomaselmgren/Structured-Functional-Representation-of-Design
[9] Altshuller, G. S., & Shapiro, R. B. (1956). Psychology of inventive creativity. Issues of Psychology, 6, 37-49.
[10] Suh, N. P. "Design of systems." CIRP Annals 46, no. 1 (1997): 75-80.
[11] Ulrich, K. T., & Eppinger, S.D. (2016) Product design and development. McGraw-hill.
[12] Akay, H., & Kim, S-G. (2021, August). Artificial intelligence tools for better use of axiomatic design. In IOP Conference Series: Materials Science and Engineering (Vol. 1174, No. 1, p. 012005). IOP Publishing.
[13] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Lukasz, K., and Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.
[14] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. Advances in neural information processing systems, 33, 1877-1901.
[15] LaGrotta, V., & Shahbazi, Z. (2024, November). Natural Language Processing Applications in Mechanical Engineering Education. In ASME International Mechanical Engineering Congress and Exposition (Vol. 88650, p. V007T09A016). American Society of Mechanical Engineers.
[16] Wang, X., Anwer, N., Dai, Y., & Liu, A. (2023). ChatGPT for design, manufacturing, and education. Procedia CIRP, 119, 7-14.
[17] Wang, P., Karigiannis, J., & Gao, R. X. (2024). Ontology-integrated tuning of large language model for intelligent maintenance. CIRP annals, 73(1), 361-364.
[18] Jiang, S., & Luo, J. (2024, August). Autotriz: Artificial ideation with triz and large language models. In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (Vol. 88377, p. V03BT03A055). American Society of Mechanical Engineers.
[19] Li, Y., Ko, H., & Ameri, F. (2025). Integrating Graph Retrieval-Augmented Generation With Large Language Models for Supplier Discovery. Journal of Computing and Information Science in Engineering, 25(2), 021010.
[20] Laurenzi, E., Mathys, A., & Martin, A. (2024, May). An LLM-aided enterprise knowledge graph (EKG) engineering process. In Proceedings of the AAAI Symposium Series (Vol. 3, No. 1, pp. 148-156).
[21] Akay, H., Yang, M., & Kim, S. G. (2021, August). Automating design requirement extraction from text with deep learning. In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (Vol. 85390, p. V03BT03A035). American Society of Mechanical Engineers.
[22] Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) (p. 3982). Association for Computational Linguistics.