2025 5th International Conference on Artificial Intelligence and Industrial Technology Applications (AIITA)

Automated Construction of 3D Model Knowledge Graphs and Knowledge Extraction Using Large Language Models

*Mengting Li
School of Mechano-Electronic Engineering
Xidian University
Xi'an, China
*23041212655@stu.xidian.edu.cn

Qianyong Hu
New York University
370 jay st, Brooklyn, NY
qh2252@nyu.edu

Xianguang Kong
School of Mechano-Electronic Engineering,
Xidian University
Xi'an, China
kongxiang uang@xidian.edu.cn

***Abstract*—In the field of design, the construction of a knowledge graph for 3D product models is crucial for obtaining design norms and successful cases. Traditional knowledge extraction methods rely on industry experts to manually define rules, which suffer from low recall rates, poor robustness, and difficult maintenance. To address these issues, this paper proposes an automatic construction system for a comprehensive knowledge graph of 3D product models. By leveraging large model technology, it simplifies automated semantic understanding and representation, significantly reducing the reliance on manual labor in traditional 3D product model knowledge extraction methods. It is the first to complete the automatic extraction of 3D model knowledge based on large language model technology. Additionally, this paper presents a workflow agent module for 3D model knowledge extraction. This module is plug-and-play without the need for additional training and supports further fine-tuning of the large language model to enhance the accuracy of knowledge extraction, demonstrating good generalization. Experiments show that this method is more efficient than manual extraction, providing an efficient and automated solution for constructing 3D model knowledge graphs.**

Keywords—Knowledge extraction; Large language model; Knowledge graph

I. INTRODUCTION

In the field of design, designers need to select and estimate connection structural components for different parts, choose parameters, and design models. A large part of the existing design specifications and successful design cases are presented in the form of 3D model structures. Successful examples can be used as design knowledge. After processing, the knowledge expression can assist professionals in design work. Knowledge graphs, as a tool for knowledge expression and management, can achieve structured storage and reuse of design knowledge. The construction of knowledge graphs relies on knowledge extraction technology. Knowledge extraction is based on existing knowledge modeling to extract knowledge from data of different sources and structures, forming structured data and storing it in the knowledge graph. The accuracy of knowledge extraction directly affects the quality of the knowledge graph. Therefore, a key step in constructing a 3D model knowledge graph is to extract knowledge from 3D models.

The research focus of knowledge extraction lies in how to construct an extraction model to extract various knowledge points from data sources in a certain form and store them in the knowledge base, with the aim of enhancing the reusability of knowledge information. Early knowledge extraction methods mainly relied on industry experts manually defining recognition rules and templates [1-4]. L.F. Rau [1] combined heuristic algorithms with manually constructed rules to build a model for automatically extracting company name entities from unstructured financial news, achieving an average accuracy rate of 95%, far higher than the accuracy rate of manual extraction. However, this method of constructing rules and templates has extremely high requirements for the business level and rule representation ability of industry personnel. Rule-based entity extraction models have problems such as low recall rate, poor robustness, and difficulty in maintenance.

Deep learning can automatically learn complex hidden features from less domain and experience knowledge data. Neural network models mainly rely on convolutional neural networks (CNN) or recurrent neural networks (RNN). Traditional CNNs cannot capture long-term information in sentences. In 2021, Jun [5] combined Attention and CNN and proposed a new convolutional neural network model ALL-CNN (ACNN), which uses CNNs with integrated different convolutional kernels and residual structures to capture context information at different scales, and then introduces the attention mechanism to enhance the capture ability. In addition, RNN models are also widely used in NER tasks. Huang [6] proposed using models such as LSTM, Bi-LSTM, and BiLSTM-CRF for sequence labeling, which can effectively utilize the context information of sequences. Bi-LSTM-CRF is currently the most mainstream method for solving NER tasks. In 2023, Yan [7] proposed a new nested named entity recognition method, namely the method based on local hypergraphs. This method is different from the previous method of encoding the entire text using a single hypergraph. It simplifies and efficiently processes nested structures by extracting entities with the same

979-8-3315-0976-7/25/$31.00 ©2025 IEEE

boundaries in local hypergraphs. Peng [8] extracted design knowledge elements and their relationships from text data through deep learning and semantic analysis. Then, these elements were mapped to the RFPC design knowledge ontology framework and stored in a graph database to support product design. Wang [9] studied a method for mechanical product conceptual design based on historical 3D numerical modeling design knowledge. The focus of this method is to standardize the representation, extraction, and storage of design knowledge obtained from historical 3D models. Guo [10] designed modeling methods suitable for part genes, product function and structure genes, product performance and failure genes based on the gene characteristics at different product levels. For those implicit rules and knowledge that are difficult to directly represent, the article expresses them in IF-THEN form and converts them into triples. Long [11] proposed the General Feature Design Flow (GFDF) framework to represent the design knowledge of various mechanical products from the initial design stage (including requirements analysis) to the final automatic generation of Computer-Aided Design (CAD) models. Although these methods effectively represent various forms of knowledge resources, the level of automation in the subsequent semantic knowledge extraction needs to be improved.

In 2017, Vaswani [12] introduced the Transformer model and built modern LLMs such as BERT [13] and GPT [14] based on it. Thanks to the extensive application of large model technology in the fields of NLG and NLU, many scholars have attempted to integrate large model technology to enhance the application of knowledge graphs in the knowledge graph field. Integrating models with natural language understanding capabilities into the automated construction process of knowledge graphs can significantly improve the accuracy of relationship and entity extraction in database schema design, thereby enhancing the overall efficiency of database management systems [15,16]. AutoAlign [17] can automatically identify the similarity of predicates in different knowledge graphs with the assistance of LLMs without the need for expensive manual seed creation, thus achieving the alignment of knowledge graphs. When dealing with multi-document question-answering tasks, Knowledge Graph Prompts (KGP) [18] can be introduced to construct and explore knowledge graphs, designing appropriate context environments to improve the accuracy and efficiency of question-answering systems. CP-KGC [19] uses the constraint prompts of large language models to complete the knowledge graph to improve and enhance the text descriptions in KGC datasets. CP-KGC utilizes simple and carefully designed prompts to help regenerate or supplement existing text descriptions, thereby enhancing the overall expressiveness and practicality of the data.

The innovations of this study are as follows:

1. The proposed automatic knowledge graph construction system has for the first time achieved the automatic extraction of knowledge from 3D models, significantly reducing the reliance on manually defined rules. Compared with traditional 3D model knowledge extraction methods, it has greatly improved the efficiency of knowledge extraction.

2. The proposed knowledge extraction agent (FSCKE agent) simplifies the semantic understanding and representation process and has good flexibility. FSCKE agent supports the fine-tuning of large language models, enabling more accurate knowledge extraction and better adaptation to different types of 3D modeling tasks and data.

II. METHOD

A. *Framework of 3D model knowledge graph construction*

The proposed framework is divided into four parts, as shown in Figure 1: 3D model preprocessing, function tree analysis, knowledge analysis and knowledge storage. In the 3D model pre-processing stage, the modeling order and structure naming method of the 3D model are standardized. Then, FSCKE agent generates function trees according to the tree structure knowledge extracted from the data, and establishes corresponding mapping relationships for these knowledge trees. These knowledge trees and their associated mapping relationships are represented in the form of triples. By integrating the knowledge tree and its mapping relationship, the entity, relation and attribute database are obtained. Finally, Neo4j is used to store and visualize the knowledge graph of the 3D model.

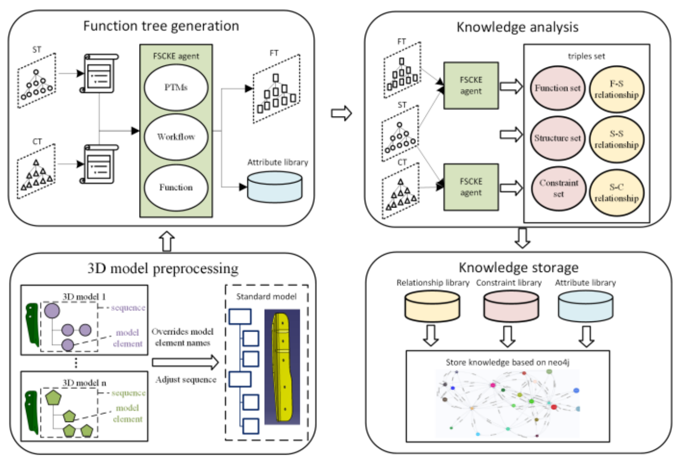

Figure 1 Framework of 3D model knowledge graph Construction

3D model preprocessing stage does some preparatory work for knowledge extraction based on FSC knowledge representation model, in order to assist the subsequent knowledge extraction process. Including knowledge normalization, model library import and access to some programming software API (application program interface) and other work. Based on the secondary development of 3D modeling software, the relatively hash knowledge information based on model level is obtained, including model name information, model reference shafting information, model entity set information, model constraint set information, etc. The hashing knowledge information collected by preprocessing, such as the model entity set, is constructed into structure tree and constraint tree.

The task of function tree generation is to integrate tree-shaped knowledge extracted in the preprocessing stage into text form and transfer it to the large model. Based on the semantic understanding ability of the large model, it analyzes the implicit relationships between structure and constraints and generates corresponding functional explanations. The task of knowledge analysis is to parse the mapping relationships between FT and ST, ST and CT, and internal elements of ST based on the semantic understanding ability of the large model, and generate a set of triples, including six parts: Function set, Structure set, Constraint set, F-S relationship, S-S relationship, and S-C relationship. FSCKE agent is a workflow for knowledge extraction of 3D models, shown in the Figure 2.

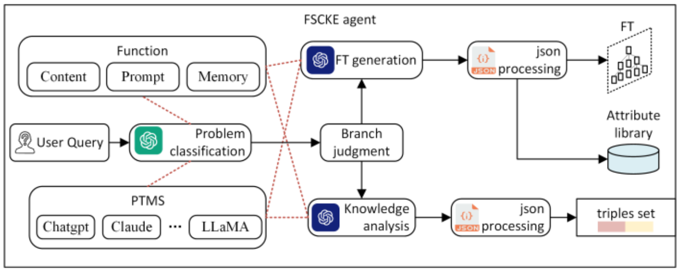

Figure 2 FSCKE agent

User Query consists of two parts: instruction and file. Instruction is the user's direct command, and file is the tree-shaped knowledge processed into text form. The instruction and knowledge are concatenated and input into the large model used for problem classification. In the FSCKE agent system, multiple pre-trained models can be selected for tasks such as problem classification, FT generation, and knowledge analysis, including chatgpt series, Claude series, Llama series. Various development tools are used to enrich the functions of the large model. Content includes the context length and context window of the large model, determining the range of information captured by the model and the upper limit of its processing capacity. Prompt is the key to guiding the LLM to generate the corresponding output. For LLMs capable of handling diverse tasks, a well-designed prompt can significantly affect the model's performance. Memory records the previous conversation context and uses this context as part of the prompt, passing it to the model in the latest call. After completing the classification task, a branch judgment is made to determine which task it belongs to. The large model's response is returned in a structured JSON format. Collect the JSON format model responses obtained from FT generation, integrate the corresponding functional elements of the same-level structure and constraint elements to form a function tree, and integrate the detailed explanations of the functional elements into the attribute set of the functional entity. Collect the JSON format responses obtained from knowledge analysis, and based on the generated FT, ST, and CT, mine the mapping relationships between the corresponding elements of the same level of FT, ST, and CT, as well as the mapping relationships between internal elements of FT, ST, and CT. The large model annotates the element relationships in the form of triples.

B. *Reference approach*

Knowledge extraction of 3D models based on functional-structure-constraint (FSC) knowledge representation models is the core part of the knowledge graph of 3D models constructed by wang [9]. The FSC knowledge representation model is shown in Figure 3. For a single part product, its internal structure is diverse, and the functions and constraints correspond to the structure. Therefore, function, structure and constraint are often presented in the form of sets in the product, such as function set, structure set, constraint set, etc. The design knowledge is mined through the relationship between elements and the mapping between elements to prepare for intelligent recommendation. The knowledge extraction method using FSC knowledge representation as template is shown in the formula1 to formula4, which relies on artificial construction of function tree and mapping relationship of each element.

$$ST = \{se_1, se_2, ..., se_i\} \quad (1)$$

$$CT = \{ce_1, ce_2, ..., ce_i\} \quad (2)$$

$$FT = G(ST, CT) = \{fe_1, fe_2, ... fe_i\} \quad (3)$$

$$G(ST, CT) = \{g(se_1, ce_1), ..., g(se_i, ce_i)\} \quad (4)$$

ST represents structure tree, CT represents constraint tree, FT represents function tree, G represents artificial generation, se_i represents structure element, ce_i represents constraint element, fe_i represents function element, g means artificial generator.

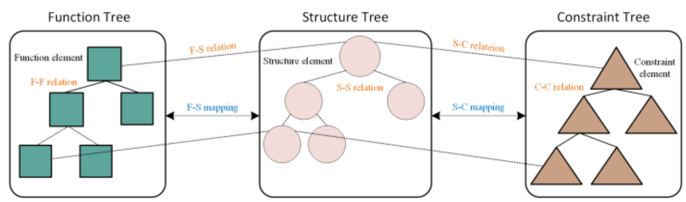

Figure 3 FSC knowledge represent model

III. EXPERIMENT

A. *Background Description*

In this study, we take a typical aircraft structural component L-shaped corner as an example to verify. The three-dimensional model of L-shaped corner piece is shown in Figure 4. L-shaped Angle parts are widely used in aircraft, involving fuselage structure connection, electrical wiring interconnection system, composite material structure and so on. The L-shaped corner piece consists of a folding net 、camber、Fillet corner and snip. The L-shaped Angle plate is in contact with the surface of the keel beam, the surface of the truss, and the skin of the wire, and is usually fixed by bolts or rivets.

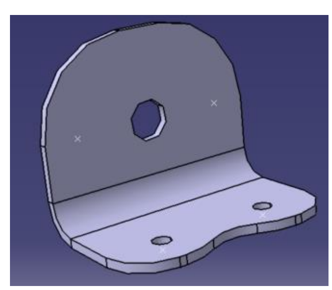

Figure 4 3D model of L-shaped corner piece

B. Knowledge extraction based on the proposed framework

The proposed framework for automatic extraction of 3D model knowledge based on large model technology is applied to the extraction of design knowledge from L-shaped corner piece 3D models. Firstly, the sample models are preprocessed. The preprocessing steps include standardizing the modeling process and naming the corresponding entities. The preprocessing effect is shown in Figure 5. Standardizing the modeling process ensures model consistency. Standardizing the naming of corresponding entities includes completing and replacing the corresponding entities. Clear and understandable entity naming enables quick identification of each part in the model, facilitating the search and reference of corresponding entities in subsequent modification, editing, or analysis processes.

Secondly, 3D graphic modeling data is collected based on the secondary development technology of modeling software, and the structure tree and constraint tree are organized, as shown in Figure 6. Folded grids are used to increase the strength and stability of the corner piece and also serve as a positioning function during the connection process, helping to ensure that the corner piece is correctly aligned during installation and maintains its shape when subjected to external forces. Bolts, screws, or other fasteners fix the corner piece to the aircraft shell through holes. Notches are used to reduce the weight of the corner piece and prevent wires or cables from being subjected to excessive stress.

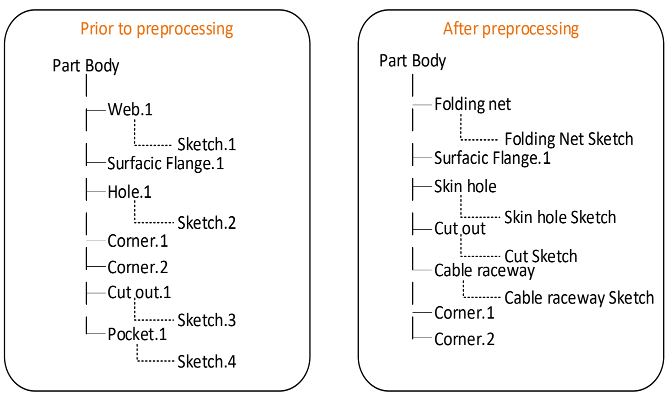

Figure 5 Comparison before and after preprocessing

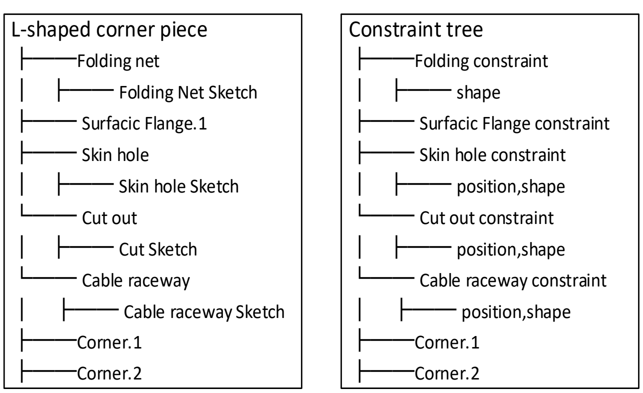

Figure 6 Knowledge extraction results of structure tree and function tree

Then, the structure tree and constraint tree are flattened and represented in text form and input into the built workflow agent. This workflow is orchestrated and developed based on the Dify platform and integrates multiple large model-assisted tasks, as shown in Figure 7.

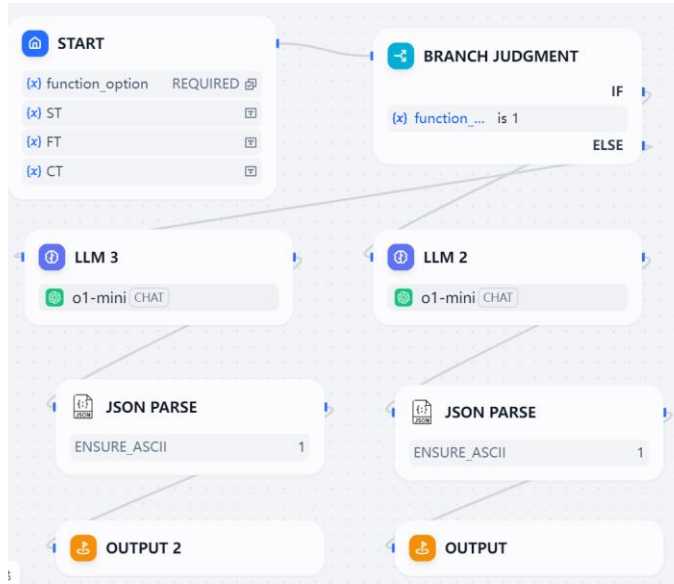

Figure 7 Dify workflow construction diagram

The flattened tree knowledge is uploaded and input into the workflow. The large model responsible for conditional branching makes judgments on the task type based on the predefined conditional branching prompt words, determining whether it belongs to the generation of the function tree or knowledge analysis. The conditional branching prompt words include the definition of the large model role, the explanation of the branch task, and the output standard.

The large model responsible for generating the function tree generates the corresponding functional elements for specific structural elements and constraint elements based on the predefined FT task prompt words, considering the overall situation of the structure tree and the function tree. The FT task prompt words include the definition of the large model role, the concepts of the structure tree, function tree, and constraint tree, and output standard examples.

The large model responsible for knowledge analysis summarizes the three sets based on the input three trees, namely the entity set, relationship set, and attribute set. The elements within the sets are represented in the form of triples. The knowledge analysis task prompt words include model role positioning, knowledge analysis examples, and the concepts of the three trees. Finally, the workflow plugin after packaging and release is integrated as a module into the 3D model knowledge extraction system, achieving the fully automatic 3D model knowledge extraction function based on the FSC knowledge representation model. The final extracted knowledge set is visualized based on neo4j, as shown in Figure 8.

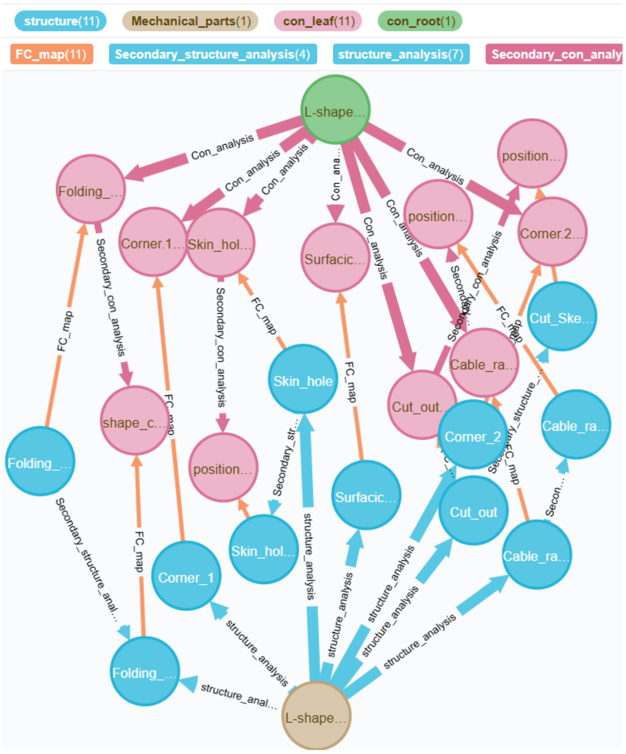

Figure 8 Triplet storage visualization

C. *Result and discussion*

In this study, 50 L-shaped 3D model samples were selected to test the accuracy and speed of knowledge extraction of the proposed method, and compared with the results of artificial knowledge extraction based on FSC knowledge representation model adopted by Wang [9]. Five experts were invited to score and evaluate the quality of triplet knowledge extraction, and the comparison results are shown in Table 1.

As can be seen from the experimental results, the proposed model, based on large model technology and with the assistance of few sample prompts, accelerates the speed of knowledge extraction of 3D models and increases the speed of knowledge extraction twice under the circumstance of relatively loose precision requirements.

Table 1 Comparison between the proposed model and the manual experiment

	Proposed framework	**FSC-Manual extraction**
accuracy	88%	97%
time	52min	143min
valid triplet	2997	3300

IV. Conclusion

This paper proposes an innovative method for automatic construction and knowledge extraction of 3D model knowledge graph based on large language model (llm). Previous knowledge extraction methods of 3D knowledge graph rely heavily on manual intervention. By integrating the LLM into the knowledge extraction process, a higher degree of automation is achieved and reliance on human intervention is reduced.

This method not only improves the accuracy and efficiency of knowledge extraction, but also introduces a plug-and-play workflow agent module, eliminating the need for additional training. This advance represents an important step in the field of knowledge graph construction of 3D models, providing designers and professionals with a more streamlined and efficient solution.

The experimental results show that the LLM-based method is faster than the manual method in knowledge extraction. With the ability to process and understand complex 3D model data, this approach has the potential to revolutionize how design knowledge is captured, stored, and utilized across industries.

The integration of the LLM. with the 3D model knowledge extraction process marks a new era in design knowledge management. It not only improves the technical capability of knowledge extraction, but also opens up new possibilities for the application of artificial intelligence in the field of design and engineering. Future research will continue to explore the full potential of this approach, refining the model and extending its application to a wider range of design scenarios.

References

[1] Rau L F. 1991. Extracting company names from text. In: Proceedings the Seventh IEEE Conference on Artificial Intelligence Application. IEEE Computer Society. 29-32.

[2] Humphreys K, Gaizauskas R, Azzam S, et al. 1998. University of Sheffield: Description of the LaSIE-II system as used for MUC-7. In: Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29-May 1, 1998.

[3] Collobert R, Weston J, Bottou L, et al. 2011. Natural language processing (almost) from scratch. Journal of machine learning research, 12: 2493-2537.

[4] Qiu J, Zhou Y, Wang Q, et al. 2019. Chinese clinical named entity recognition using residual dilated convolutional neural network with conditional random field. IEEE transactions on nanobioscience, 18(3): 306-315.

[5] Zhu J, Qiao J, Dai X, et al. 2017. Relation Classification via Target-Concentrated Attention CNNs. In: Cham:, Springer International Publishing. 137-146.

[6] Huang Z, Xu W, Yu K. 2015. Bidirectional LSTM-CRF models for sequence tagging. In: Conference on Empirical Methods in Natural Language Processing. pp.1064-1074.

[7] Yan Y, Cai B, Song S. 2023. Nested Named Entity Recognition as Building Local Hypergraphs. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, No. 11.

[8] Peng X, Yang J, Li Q, et al. 2022. Design map construction method based on RFPC conceptual design framework, involving the construction of a map application module, providing product design for realizing retrieval of knowledge in the process of conceptual design reusing, and supporting design scheme generation. CN115455196-A. 2022-12-25.

[9] Wang, Q. et al. (2024). Conceptual Design Method for Mechanical Products Based on Historical 3D Model Design Knowledge. In: Tan, J., Liu, Y., Huang, HZ., Yu, J., Wang, Z. (eds) Advances in Mechanical Design. ICMD 2023. Mechanisms and Machine Science, vol 155. Springer, Singapore. https://doi.org/10.1007/978-981-97-0922-9_24

[10] Yuyao Guo, Lei Wang, Zelin Zhang, Jianhua Cao, Xuhui Xia, Ying Liu, Integrated modeling for retired mechanical product genes in remanufacturing: A knowledge graph-based approach, Advanced Engineering Informatics, Volume 59, 2024, 102254, ISSN 1474-0346, https://doi.org/10.1016/j.aei.2023.102254.

[11] Xinjiani Long, Haitao Li, Yuefeng Du, Enrong Mao, Jianjian Tai, A knowledge-based automated design system for mechanical products based on a general knowledge framework, Expert Systems with Applications, Volume 178, 2021, 114960, ISSN 0957-4174. https://doi.org/10.1016/j.eswa.2021.114960.

[12] Vaswani A, Shazeer N, Parmar N, et al. 2017. Attention is all you need. Adv Neural Inform ProcSyst, 30.

[13] Lee J, Toutanova K. 2018. Pre-training of deep bidirectional transformers for language understanding. arXiv preprint.

[14] Brown T B. 2020. Language models are few-shot learners. arXiv preprint arXiv: 2020.

[15] He Y, Chen J Y, Antonyrajah D, et al. 2022. Bertmap: a bert-based ontology alignment system. In: AAAI, 36: 5684-5691.

[16] Zhong L, Wu J, Li Q, Peng H, Wu X. 2023. A comprehensive survey on automatic knowledge graph construction. ACM Comput Survey.56(4):1-62.

[17] Zhang R, Su Y, Trisedya B D, Zhao X, Yang M, Cheng H, Qi J. 2023. AutoAlign: Fully Automatic and Effective Knowledge Graph Alignment enabled by Large Language Models.

[18] Wang Y, Lipka N, Rossi R A, Siu A, Zhang R, Derr T. 2023. Knowledge Graph Prompting for Multi-Document Question Answering.

[19] Yang R, Zhu J, Man J, Fang L, Zhou Y. 2024. Enhancing Text-based Knowledge Graph Completion with Zero-Shot Large Language Models: A Focus on Semantic Enhancement.