FabKG: A Knowledge graph of Manufacturing Science domain utilizing structured and unconventional unstructured knowledge source

Aman Kumar akumar33@ncsu.edu

Akshay G Bharadwaj abharad3@ncsu.edu

Binil Starly bstarly@ncsu.edu

Collin Lynch cflynch@ncsu.edu

Abstract

As the demands for large-scale information processing have grown, knowledge graph-based approaches have gained prominence for representing general and domain knowledge. The development of such general representations is essential, particularly in domains such as manufacturing which intelligent processes and adaptive education can enhance. Despite the continuous accumulation of text in these domains, the lack of structured data has created information extraction and knowledge transfer barriers. In this paper, we report on work towards developing robust knowledge graphs based upon entity and relation data for both commercial and educational uses. To create the FabKG (Manufacturing knowledge graph), we have utilized textbook index words, research paper keywords, FabNER (manufacturing NER), to extract a sub knowledge base contained within Wikidata. Moreover, we propose a novel crowdsourcing method for KG creation by leveraging student notes, which contain invaluable information but are not captured as meaningful information, excluding their use in personal preparation for learning and written exams. We have created a knowledge graph containing 65000+ triples using all data sources. We have also shown the use case of domain-specific question answering and expression/formula-based question answering for educational purposes.

1 Introduction

In recent years, the advancement of artificial intelligence applications has grown multifold. Many areas such as natural language processing, digital twins (Liu et al., 2021), and chatbots (Chen et al., 2021) have become very popular for their ability to record and use information from unstructured sources efficiently. One such application is Knowledge Graph (KG), which has gained popularity in various domains due to its potential applications. A Knowledge Graph is a graph meant to accumulate and impart real-world knowledge, with nodes representing entities of interest and edges representing potentially diverse relations between the entities. A KG has varied applications in recommendations, search, question answering and many more. Most importantly, a KG can be used to make decisions based on inferences.

The use of a knowledge graph is of high value in making design and manufacturing-related decisions. As there has been an explosion of knowledge addition in various design considerations and manufacturing decisions, most of the knowledge is with Small and medium-sized enterprises (SMEs). The decision-making in design and production could be significantly improved using knowledge graphs (Buchgeher et al., 2021). It can benefit not only small and medium manufacturers (Li et al., 2021), but also hardware-based entrepreneurs and help boost self-sustaining product development (Li et al., 2020).

A number of prior researchers have started developing manufacturing related knowledge graphs based on specific problem areas such as machining process planning (Yang et al., 2019; Ye et al., 2018), workshop resource KG (Zhou et al., 2021; Sun and Wang, 2019), intelligent manufacturing (Yan et al., 2020), faults (Liang et al., 2022; Wang and Yang, 2019), maintenance (Hossayni et al., 2020) and industry 4.0 (Garofalo et al., 2018; Bader et al., 2020; Kraft and Eibeck, 2020). However, none of these graphs represent fundamental knowledge of manufacturing concepts, processes, process parameters, characterization, materials, applications, and various other basic aspects of manufacturing domain education. A large amount of such fragmented knowledge can be integrated to assist the learners in intuitively and easily connecting with the knowledge system by leveraging the nodes and relationships. Such knowledge integration will also assist in intelligent question answering that can accelerate knowledge discovery and search.

Google bases part of its Knowledge Vault on the

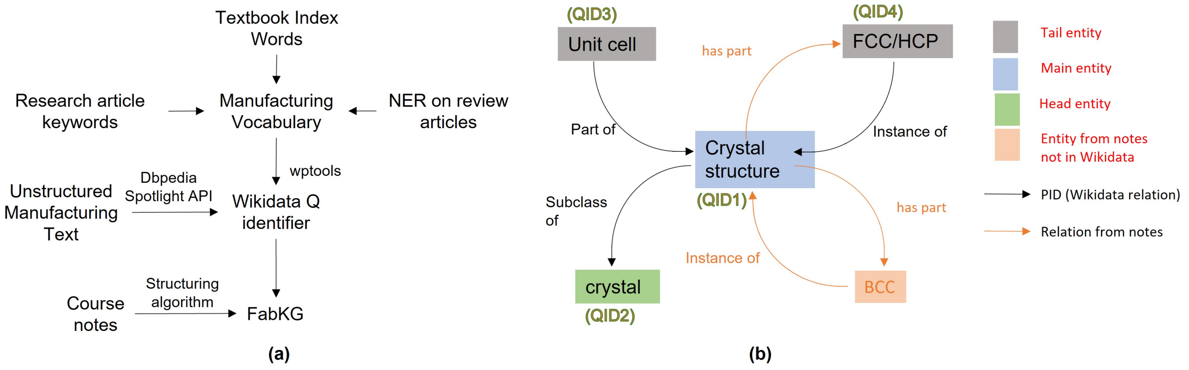


Figure 1: (a) Manufacturing Knowledge Graph construction methodology (b) Use of SPARQLWrapper to fetch Wikidata items associated with 'crystal structure' in two step forward and two step backward. This image also shows addition of entities from student notes.

well-known Wikidata knowledge base (Ringler and Paulheim, 2017). Even though Wikidata has a large amount of information from Wikipedia, there is a dearth of standardized knowledge regarding many important entities related to the Manufacturing domain. For instance, the term 'additive manufacturing' is present as '3d printing'; while there have been substantial developments in the field of 'metal additive manufacturing' (metal AM) over the last decade, it is not present as a subclass of '3d printing' in Wikidata. Moreover, within metal additive manufacturing (Frazier, 2014), sub-classifications such as DMLS, EBAM, and PBF are not present in Wikidata. One reason for this is the volunteer-driven nature of Wikidata as a knowledge base; this has led to a limited amount of specialist terminology and information regarding the manufacturing domain. Therefore, Wikidata cannot provide direct answers to questions that are very specific to this domain. To understand the basic concepts in the context of manufacturing we focus on formulating answers to some basic questions such as, 'What are some precision finishing manufacturing process?', 'What are some tools for machining copper?' etc. The purpose of creating such a knowledge graph of manufacturing using Wikidata is to provide a starting point for a structured manufacturing knowledge base, which can be amalgamated with knowledge from other sources such as textbook (Rahdari et al., 2020) and research articles (Wang et al., 2020b).

To tackle the challenges in developing the knowledge graph from scratch for manufacturing science, we consider various methodologies for creating accurate graphs. We propose a merged knowledge graph that combines the existing structured Wikidata knowledge graph with a novel semi-supervised knowledge graph extracted from textbook data. For extracting graph triples from Wikidata, as mentioned in Figure 1, we have adopted two methods for the approach: (1) Vocabulary-based and (2) Based on Unstructured text. Former includes fetching Wikidata items using a collection of manufacturing vocabulary terms through the utilization of textbook index words, keywords from research papers, and named entity recognition using FabNER (Kumar and Starly, 2021), followed by the use of DBpedia (Mendes et al., 2011) to find Wikidata items. The latter is a semi-supervised approach that utilizes students' notes, considering standard textbooks as the reference. The most significant purpose of the latter method is to make use of textbook knowledge structured by humans, thereby increasing the quality of the knowledge base. The following sections elaborate on the details of the methodology and implementation.

2 Manufacturing Knowledge Graph Construction

2.1 KG construction using Wikidata

Wikidata is a knowledge base maintained collaboratively by the community to represent information in machine readable format. Since no such knowledge base exists for the manufacturing domain, we decided first to extract existing Wikidata knowledge and then merge this with the knowledge contained within manufacturing textbooks.

Wikidata's knowledge graph has Q and P identifiers where Q represents entities, and P represents relations (Hernández et al., 2015). Currently, Wikidata is limited to very few relevant relations between entities for manufacturing domain-specific entities. We have taken about 10 unique relations based on all P identifiers attached with relevant Q identifiers identified by us. The relevant relations in Wikidata include 'Instance of', 'Subclass of', 'Use', 'Color', 'Part of', 'Uses', 'Has quality', 'Has cause', 'Has part', 'Facet of', 'Different from'.

In order to find manufacturing-specific entities in Wikidata, we used the following methods:

2.1.1 Entities extraction from index words of textbooks

Index words located at the end of textbooks are a list of all topics and entities provided to assist readers in finding the location of the text (Kumar and Dinakaran, 2021). These are important terms that are often overlooked but are a good collection of domain-specific entities. We utilized easily accessible 5 diverse ebooks related to manufacturing (Groover, 2020), digital manufacturing (Zhou et al., 2012), manufacturing process (El-Hofy, 2005) welding technology (Kou, 2003) and additive manufacturing (Gibson et al., 2021), and extracted the index entities mentioned at the end of the book to expand the list of relevant entities. We found about 3500 relevant entities from various books and added those to our vocabulary.

2.1.2 Keywords from research papers

We used 500k+ abstracts to create the corpus for manufacturing, as mentioned in FabNER. While extracting the abstracts, we accumulated the keywords mentioned in the abstract, removed duplicates, and normalized many of the words (using Levenshtein distance). There are many words written with some variation in the spelling. E.g., Landau-Ginzburg-Devonshire, Landau-Ginsburg-Devonshire, Landau-Ginsberg-Devonshire, are the same entities with variation in the way it is written in different abstract keywords by various authors. Overall, we found about 4500 relevant entities from a sample of 5000 abstracts.

2.1.3 Named entity recognition on unstructured text

We utilized review articles related to manufacturing to find the most frequent and diverse terms since it generally mention most of the past work and technologies developed in the succinct text. Ten full review articles (Wong and Hernandez, 2012; ElMaraghy et al., 2012; Zhu et al., 2013; Frazier, 2014; Oztemel and Gursev, 2020; Yan et al., 2018; Stuart et al., 2010; Rajurkar et al., 2017; Wang et al., 2020a; Kaur and Singh, 2019) for this part were selected, which were processed using a trained neural network model consisting of BERT (Devlin et al., 2018) and GloVe (Pennington et al., 2014) stacked embeddings through Flair framework (Akbik et al., 2019). Next, we employed BiLSTM and CRF (Consoli and Vieira, 2019) architecture to identify 12 category entities in the review articles with F-score of 83%. Overall, we found about 2000 entities from diverse review articles related to manufacturing.

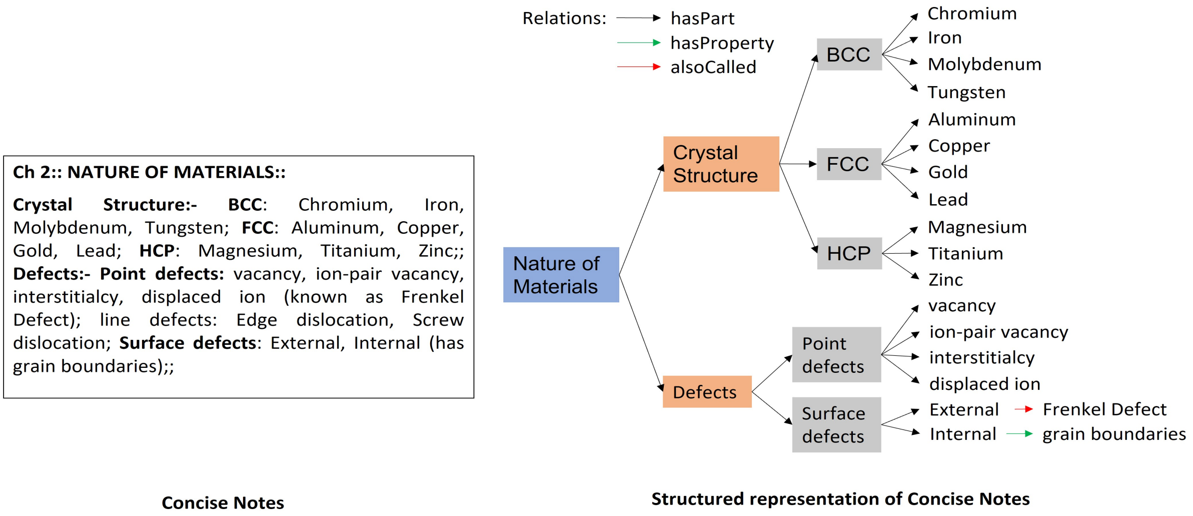
Table 1: Named entity recognition performance for the Manufacturing dataset

Model	Precision	Recall	F1
BERT+BiLSTM+CRF	0.8185	0.8429	0.8306

Using text and vocabulary of entities from all the above sources, i.e., index words, research paper keywords, and NER on review articles, we further employed two methods for finding existing Wikidata items. As depicted in Fig. 1, in the first method, we used DBpedia spotlight API to find Wikidata items associated with the unstructured text directly based on a 0.5 confidence value. In the second, we provide manufacturing vocabulary terms as the input to wptools python library to fetch Wikidata items as the output. We find all manufacturing relevant Wikidata items to extract a subgraph from Wikidata and later merge this relatively bigger knowledge graph with textbook knowledge (explained in the next section). Upon availability of some Wikidata items, we further used SPARQLWrapper (uses Wikidata SPARQL endpoint) and relations list (P identifiers) to fetch forward (head from the primary entity) as well as backward (tail from the primary entity) entities associated with the item. We performed the same for two linked steps forward and two linked steps backward to find most of the nodes that are connected with each other.

2.2 KG construction using Exam cheatsheet/notes for Manufacturing

We propose a novel approach for creating triples utilizing human knowledge. Qualifying exams (or course exams) are part of any doctoral degree program. In some schools, written exams are conducted for a few courses. In some specific courses, cheatsheets/concise notes are allowed for students to bring into the exam to enable the student to remember important points. In most cases, the cheat-



Ch 2:: NATURE OF MATERIALS::

Crystal Structure:- BCC: Chromium, Iron, Molybdenum, Tungsten; **FCC**: Aluminum, Copper, Gold, Lead; **HCP**: Magnesium, Titanium, Zinc;; **Defects:- Point defects:** vacancy, ion-pair vacancy, interstitialcy, displaced ion (known as Frenkel Defect); line defects: Edge dislocation, Screw dislocation; **Surface defects**: External, Internal (has grain boundaries);;

Figure 2: Conversion of concise notes to structured graph

sheet (or notes) developed for the exam are useless when the exams are over. This also means that the verified knowledge written by a student to remember the essential facts is lost or left unutilized for future references.

We devised a strategy for making these short, concise notes be useful input for building connected entities within FabKG. We created optional advice on cheatsheet generation for students to follow prior to the exam at our institution so that they could participate to the task of knowledge base enhancement in the Manufacturing area. Students can only write crucial details from various textbook chapters, assuming that the number of pages allowed in the exam is limited. There is a title within each chapter, followed by several subtitles, each of which contains some entities and context, which is potentially a good knowledge source. The guidelines were kept simple so that students would not have to spend much time referring to them. It mentioned the title, subtitle, and content hierarchy and a precise technique for separating them.

The following guidelines are provided for example purposes only:

a) The chapter name is preserved as the top title, followed by a distinctive symbol, making it easier to distinguish between chapters.

b) Within a chapter, many sub-topics are separated by another unique symbol, such as a double semi-colon ';;'. Two sub-topics are shown in Fig. 2, for example, (1) Defects and (2) Crystal structure

c) If there is a further subtopic within a subtopic, it is separated by a symbol such as ':' followed by some relevant points. A single semi-colon separates multiple subtopics.

d) Explanations or additional information about any term are retained in brackets as an attribute of a relational entity. For example, displaced ion (Frenkel defect) denotes that a point defect with a displaced ion is also known as a Frenkel defect.

Use of some symbols patterns when creating the notes aided in the design of regex patterns for quickly extracting entities and their obvious relationships. We were able to extract over 1200 distinct entities, 25 unique relations, and 4200 unique triples using this method. Fig. 2 depicts the notes in their raw and structured state. The student notes in both unstructured and structured form was verified by human supervision. Indirect crowdsourcing is the crucial aspect that has made this element of the project possible. However, the intention was to use note takers' knowledge. It should be emphasized that even though some previous work has mentioned the use of notes (Denny et al., 2015) for developing a knowledge map, however, on a larger scale and for educational applications, this type of knowledge source has not been studied. This method might be used with little effort for any domain-specific textual material.

Despite the small number of entities/relations discovered, this method allows textbook knowledge to be converted into useable knowledge, which aids in developing a knowledge graph for educational purposes. In general, for automatic extraction of directed relation, it is often difficult to determine which entities are related to each other when more than 2 entities are present in a sentence. This is also because, on multiple occasions, no relation exists between the entities. It becomes a challenge to employ a NER and detect directed relations between entities automatically which we solve by this semi-supervised method. Based on the analysis of the notes, some of the crucial relations found include: 'has', 'hasProperty', 'uses', 'usedTo', 'usedIn', 'causes', 'producedBy', 'makes', 'hasExpression', 'hasPart', 'addedWith', 'hasValue', 'includes', 'partOf', 'alsoCalled', 'dueTo', 'instanceOf', 'isAbbrev', 'isAcronym', 'hasComparator'.

2.3 Fusion of structured and unstructured knowledge

All triples found with the above-mentioned methods were aggregated together to create a knowledge graph of about 65000 triples. Fig1(b) depicts the merger of Wikidata and textbook knowledge. We created a collection of possible synonyms for various entities to enable us to merge Wikidata entities with textbook entities. We found that out of 1200 textbook entities, about 25% were present in Wikidata. We also found some links between entities which otherwise were not present in Wikidata due to limited relations.

3 Knowledge driven QA

3.1 Domain specific question answering

The Knowledge Graph for manufacturing (FabKG) is suitable for answering questions and powering a chatbot to answer questions. The FabKG is a directed graph G = (V, E) where the node v ∈ V denotes named entities of manufacturing, numeric literal or expression, and the edge e ∈ E denotes directed relation between the nodes.

Given a natural language question as input, the entities are categorized in their respective classes. Based on the subject and predicate most similar object (highest cosine similarity) to the category in the knowledge base is queried.

Some of the common domain specific questions could not be answered using general purpose search engines. Examples of questions that could be answered by FabKG are:

a. Which tool geometry is used for planning?

b. Which material has more hardness, cermet or alumina? Note: We have used a hasComparator relation specifying various comparison values in our KG that could answer the 'more' and 'less' inference question.

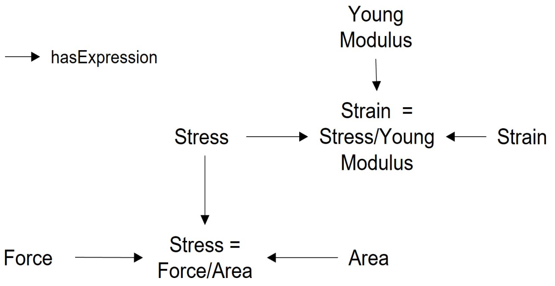


Figure 3: A small subgraph showing the links of entities connected with other expressions for ease of calculation, making the system think like humans.

c. What is the composition of Tungsten in cast cobalt?

d. Which nontraditional manufacturing process is used for coining operations?

e. What is the length to depth ratio for discontinuous fibers?

3.2 Expression based question answering

We have included some manufacturing-specific formulas/expressions in the knowledge graph to enable inference-based calculations. Since we have captured some formulae linked with entities using 'hasExpression' relation, traversing for the formula node in the graph is easy. We have also included a simple rule for calculation-type questions. Here is an example question below:

> Calculate the strain on the cylinder given the area 1 cm^2, 10N force, and Young's modulus for steel 200 GPa.

Given the question above, we have some 'formula entities': area, force, and young modulus of steel. These entities are queried in the KG for any available linked expression. Similar to MathGraph (Zhao et al., 2019), we utilize SymPy (Meurer et al., 2017) to convert the queried expression into a mathematical equation with variables, and to perform the calculation, we use some basic rules of precedence to fetch the results. As shown in fig. 3, we can find strain using Young's modulus and stress; however, since stress is not known, we calculate stress as the first step using force and area. This process depicts the way human thinks while answering a question with some inputs and related expressions.

Some other examples of questions forms that are easier than the above-mentioned questions: a. Calculate material removal rate given feed rate, cutting speed, and depth of cut. HINT: We can calculate the Material removal rate using (feed rate)*(cutting speed)*(depth of cut). b. Calculate measuring length of roughness given cutoff length of 0.8. HINT: measuring length of roughness = 0.5 * cutoff length.

4 Conclusion and future work

We have developed FabKG – a knowledge graph for product design and manufacturing, which utilizes two critical sources of knowledge, (1) Wikidata and (2) Human constructed notes, that combine structured/unstructured knowledge towards answering question-related to product development and manufacturing. Using this KG, students, product developers, and knowledge seekers can get good insights into various concepts and fundamentals about various topics in this domain. Using all the methods described above, we have found 65000+ triples in 12 entity categories.

In the future, we plan to use the heterogeneous knowledge graph for directed relation prediction in the bigger corpus, performing graph embedding and link prediction. Moreover, lecture presentations with succinct text could also be utilized for finding entities and relations. Generally, the title/topic of the presentation symbolizes the subject, with some entities either written directly or placed after another subtopic. Furthermore, 'property/attribute' of relation through the specific value of entities such as the strength of materials, carbon content, Brinell hardness, Etc., currently available in tabular form in books and other resources, can be added to the KG. The same could be represented using a hypergraph by combining multimodal data. Therefore, the new graph structure would have not only an 'entity-relation-entity' type graph but also an 'entity-attribute-value' graph. Finally, this knowledge graph could help link to global knowledge by contributing to existing Wikidata knowledge with the help of Wikimapper.

References

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. Flair: An easy-to-use framework for state-of-the-art nlp. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.

Sebastian R Bader, Irlan Grangel-Gonzalez, Priyanka Nanjappa, Maria-Esther Vidal, and Maria Maleshkova. 2020. A knowledge graph for industry 4.0. In *European Semantic Web Conference*, pages 465–480. Springer.

Georg Buchgeher, David Gabauer, Jorge Martinez-Gil, and Lisa Ehrlinger. 2021. Knowledge graphs in manufacturing and production: A systematic literature review. *IEEE Access*, 9:55537–55554.

Tzu-Yu Chen, Yu-Ching Chiu, Nanyi Bi, and Richard Tzong-Han Tsai. 2021. Multi-modal chatbot in intelligent manufacturing. *IEEE Access*, 9:82118–82129.

Bernardo Consoli and Renata Vieira. 2019. Multidomain contextual embeddings for named entity recognition. In *Proceedings of the Iberian Languages Evaluation Forum*, volume 2421, pages 434–441.

Joshua C Denny, Anderson Spickard III, Peter J Speltz, Renee Porier, Donna E Rosenstiel, and James S Powers. 2015. Using natural language processing to provide personalized learning opportunities from trainee clinical notes. *Journal of biomedical informatics*, 56:292–299.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Hassan Abdel-Gawad El-Hofy. 2005. *Advanced machining processes: nontraditional and hybrid machining processes*. McGraw Hill Professional.

Waguih ElMaraghy, Hoda ElMaraghy, Tetsuo Tomiyama, and Laszlo Monostori. 2012. Complexity in engineering design and manufacturing. *CIRP annals*, 61(2):793–814.

William E Frazier. 2014. Metal additive manufacturing: a review. *Journal of Materials Engineering and performance*, 23(6):1917–1928.

Martina Garofalo, Maria Angela Pellegrino, Abdulrahman Altabba, and Michael Cochez. 2018. Leveraging knowledge graph embedding techniques for industry 4.0 use cases. In *Cyber Defence in Industry 4.0 Systems and Related Logistics and IT Infrastructures*, pages 10–26. IOS Press.

Ian Gibson, David W Rosen, Brent Stucker, Mahyar Khorasani, David Rosen, Brent Stucker, and Mahyar Khorasani. 2021. *Additive manufacturing technologies*, volume 17. Springer.

Mikell P Groover. 2020. *Fundamentals of modern manufacturing: materials, processes, and systems*. John Wiley & Sons.

Daniel Hernández, Aidan Hogan, and Markus Krötzsch. 2015. Reifying rdf: What works well with wikidata? *SSWS@ ISWC*, 1457:32–47.

Hicham Hossayni, Imran Khan, Mohammad Aazam, Amin Taleghani-Isfahani, and Noel Crespi. 2020. Semkore: Improving machine maintenance in industrial iot with semantic knowledge graphs. *Applied Sciences*, 10(18):6325.

Manmeet Kaur and K Singh. 2019. Review on titanium and titanium based alloys as biomaterials for orthopaedic applications. *Materials Science and Engineering: C*, 102:844–862.

Sindo Kou. 2003. Welding metallurgy. *New Jersey, USA*, 431(446):223–225.

Markus Kraft and Andreas Eibeck. 2020. J-park simulator: Knowledge graph for industry 4.0. *Chemie Ingenieur Technik*, 92(7):967–977.

Aman Kumar and Swathi Dinakaran. 2021. Textbook to triples: Creating knowledge graph in the form of triples from ai textbook. *arXiv preprint arXiv:2111.10692*.

Aman Kumar and Binil Starly. 2021. "fabner": information extraction from manufacturing process science domain literature using named entity recognition. *Journal of Intelligent Manufacturing*, pages 1–15.

Xinyu Li, Chun-Hsien Chen, Pai Zheng, Zuoxu Wang, Zuhua Jiang, and Zhixing Jiang. 2020. A knowledge graph-aided concept–knowledge approach for evolutionary smart product–service system development. *Journal of Mechanical Design*, 142(10):101403.

Yunqing Li, Shivakumar Raman, Paul Cohen, and Binil Starly. 2021. Design of knowledge graph in manufacturing services discovery. In *International Manufacturing Science and Engineering Conference*, volume 85079, page V002T07A010. American Society of Mechanical Engineers.

Kun Liang, Baoxian Zhou, Yiying Zhang, Yiping Li, Bo Zhang, and Xiankun Zhang. 2022. Pf2rm: A power fault retrieval and recommendation model based on knowledge graph. *Energies*, 15(5):1810.

Shimin Liu, Yuqian Lu, Jie Li, Dengqiang Song, Xuemin Sun, and Jinsong Bao. 2021. Multi-scale evolution mechanism and knowledge construction of a digital twin mimic model. *Robotics and Computer-Integrated Manufacturing*, 71:102123.

Pablo N Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. 2011. Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th international conference on semantic systems*, pages 1–8.

Aaron Meurer, Christopher P Smith, Mateusz Paprocki, Ondřej Čertík, Sergey B Kirpichev, Matthew Rocklin, AMiT Kumar, Sergiu Ivanov, Jason K Moore, Sartaj Singh, et al. 2017. Sympy: symbolic computing in python. *PeerJ Computer Science*, 3:e103.

Ercan Oztemel and Samet Gursev. 2020. Literature review of industry 4.0 and related technologies. *Journal of Intelligent Manufacturing*, 31(1):127–182.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Behnam Rahdari, Peter Brusilovsky, Khushboo Thaker, and Jordan Barria-Pineda. 2020. Using knowledge graph for explainable recommendation of external content in electronic textbooks. In *iTextbooks@ AIED*.

KP Rajurkar, H Hadidi, J Pariti, and GC Reddy. 2017. Review of sustainability issues in non-traditional machining processes. *Procedia Manufacturing*, 7:714–720.

Daniel Ringler and Heiko Paulheim. 2017. One knowledge graph to rule them all? analyzing the differences between dbpedia, yago, wikidata & co. In *Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz)*, pages 366–372. Springer.

Martien A Cohen Stuart, Wilhelm TS Huck, Jan Genzer, Marcus Müller, Christopher Ober, Manfred Stamm, Gleb B Sukhorukov, Igal Szleifer, Vladimir V Tsukruk, Marek Urban, et al. 2010. Emerging applications of stimuli-responsive polymer materials. *Nature materials*, 9(2):101–113.

Tao Sun and Qi Wang. 2019. Multi-source fault detection and diagnosis based on multi-level knowledge graph and bayesian theory reasoning (s). In *SEKE*, pages 177–232.

Baicun Wang, S Jack Hu, Lei Sun, and Theodor Freiheit. 2020a. Intelligent welding system technologies: State-of-the-art review and perspectives. *Journal of Manufacturing Systems*, 56:373–391.

Qingyun Wang, Manling Li, Xuan Wang, Nikolaus Parulian, Guangxing Han, Jiawei Ma, Jingxuan Tu, Ying Lin, Haoran Zhang, Weili Liu, et al. 2020b. Covid-19 literature knowledge graph construction and drug repurposing report generation. *arXiv preprint arXiv:2007.00576*.

XiuQing Wang and ShunKun Yang. 2019. A tutorial and survey on fault knowledge graph. *Cyberspace Data and Intelligence, and Cyber-Living, Syndrome, and Health*, pages 256–271.

Kaufui V Wong and Aldo Hernandez. 2012. A review of additive manufacturing. *International scholarly research notices*, 2012.

Hehua Yan, Jun Yang, and Jiafu Wan. 2020. Knowime: a system to construct a knowledge graph for intelligent manufacturing equipment. *Ieee Access*, 8:41805–41813.

Qian Yan, Hanhua Dong, Jin Su, Jianhua Han, Bo Song, Qingsong Wei, and Yusheng Shi. 2018. A review of 3d printing technology for medical applications. *Engineering*, 4(5):729–742.

Yan Yang, Tianliang Hu, Yingxin Ye, Wenbin Gao, and Chengrui Zhang. 2019. A knowledge generation mechanism of machining process planning using cloud technology. *Journal of Ambient Intelligence and Humanized Computing*, 10(3):1081–1092.

Yingxin Ye, Tianliang Hu, Chengrui Zhang, and Weichao Luo. 2018. Design and development of a cnc machining process knowledge base using cloud technology. *The International Journal of Advanced Manufacturing Technology*, 94(9):3413–3425.

Tianyu Zhao, Yan Huang, Songfan Yang, Yuyu Luo, Jianhua Feng, Yong Wang, Haitao Yuan, Kang Pan, Kaiyu Li, Haoda Li, et al. 2019. Mathgraph: A knowledge graph for automatically solving mathematical exercises. In *International conference on database systems for advanced applications*, pages 760–776. Springer.

Bin Zhou, Jinsong Bao, Jie Li, Yuqian Lu, Tianyuan Liu, and Qiwan Zhang. 2021. A novel knowledge graph-based optimization approach for resource allocation in discrete manufacturing workshops. *Robotics and Computer-Integrated Manufacturing*, 71:102160.

Zude Zhou, Shane Xie, and Dejun Chen. 2012. *Fundamentals of digital manufacturing science*. Springer.

Zicheng Zhu, Vimal G Dhokia, Aydin Nassehi, and Stephen T Newman. 2013. A review of hybrid manufacturing processes–state of the art and future perspectives. *International Journal of Computer Integrated Manufacturing*, 26(7):596–615.