machines

Article

Development and Application of Knowledge Graphs for the Injection Molding Process

Zhe-Wei Zhou [1], Yu-Hung Ting [1,2,3,*], Wen-Ren Jong [1,2,3], Shia-Chung Chen [1,2,3] and Ming-Chien Chiu [4]

[1] Department of Mechanical Engineering, Chung Yuan Christian University, Taoyuan City 320, Taiwan
[2] R&D Center for Smart Manufacturing, Chung Yuan Christian University, Taoyuan City 320, Taiwan
[3] R&D Center for Semiconductor Carrier, Chung Yuan Christian University, Taoyuan City 320, Taiwan
[4] Gudeng Precision Industrial Co., Ltd., New Taipei City 236, Taiwan
[*] Correspondence: august@cycu.edu.tw

Abstract: Injection molding, the most common method used to process plastics, is a technique with a high knowledge content; however, relevant knowledge has not been systematically organized, and as a result, there have been many bottlenecks in talent cultivation. Moreover, most of the knowledge stored in books and online articles remains in the form of unstructured data, while some even remains unwritten, resulting in many difficulties in the construction of knowledge bases. Therefore, how to extract knowledge from unstructured data and engineers' statements is a common goal of many enterprises. This study introduced the concept of a Knowledge Graph, a triplet extraction model based on bidirectional encoder representations from transformers (BERT) which was used to extract injection molding knowledge entities from text data, as well as the relationships between such entities, which were then stored in the form of knowledge graphs after entity alignment and classification with sentence-bidirectional encoder representations from transformers. In a test, the triplet extraction model achieved an F1 score of 0.899, while the entity alignment model and the entity classification model achieved accuracies of 0.92 and 0.93, respectively. Finally, a web platform was built to integrate the functions to allow engineers to expand the knowledge graphs by inputting learning statements.

Keywords: knowledge graph; injection molding; BERT; knowledge management

check for updates

Citation: Zhou, Z.-W.; Ting, Y.-H.; Jong, W.-R.; Chen, S.-C.; Chiu, M.-C. Development and Application of Knowledge Graphs for the Injection Molding Process. *Machines* **2023**, *11*, 271. https://doi.org/10.3390/machines11020271

Academic Editors: Paulo Eigi Miyagi, Flávio Soares Corrêa Da Silva, Alexandre Da Silva Simões, Fabrício Junqueira, Marcosiris Amorim de Oliveira Pessoa, Izabel Fernanda Machado and Lilian Berton

Received: 3 January 2023
Revised: 6 February 2023
Accepted: 9 February 2023
Published: 10 February 2023

1. Introduction

Due to easy processes, high precision, low cost, quick molding, and scalability for practical applications, the plastic injection industry is widely used to create products from daily necessities to medical equipment and even precision consumer electronics components. As a result, the industry has scaled up and requires more talent. However, the current injection molding industry primarily inherits knowledge in the form of mentoring and written records, which makes it difficult for novices to clearly understand the relationships between various process factors and product quality, and this will have adverse effects on the industry in the long-term.

Among the studies on injection molding knowledge management, Jong et al. [1,2] used computer-aided design (CAD) secondary development tools and a historical knowledge base to establish a mold design navigating system that would aid decision-makers regarding subsequent products and mold designs. It was based on the practical knowledge accumulated in previous mold design processes, which would effectively reduce the time spent on mold design and prevent losses caused by improper design. Khosravani et al. [3] applied case-based reasoning (CBR) and realized an intelligent injection machine fault detection system that could analyze the features of a current fault, search previous similar cases, and determine the cause of the current machine malfunction, which would effectively reduce the downtime of injection machines. In order to solve molding quality problems encountered during the injection process, Mikos et al. [4] found previous similar cases through CBR, and they applied the solutions found as the basis for solving current problems.

As shown in the above-referenced literature, it can be found that currently, in the injection molding industry (in terms of mold design, machine troubleshooting, and product defect resolution), decisions are largely made based on past experience. Most engineers lack a clear basis for decision-making and cannot foresee the possible influences of it. At the same time, this decision-making method will be difficult to pass down. Moreover, as product lifecycles are gradually shortened and more frequent product design changes occur, operators' experiences are often insufficient for solving the problems in contemporary design and manufacturing processes.

In terms of the application of knowledge graphs, Sebastian et al. [5] established a knowledge graph for Industry 4.0, which recorded relevant standards and regulations for Industry 4.0 and helped users understand how to implement Industry 4.0. Chi et al. [6] constructed a knowledge graph by integrating healthy diet information on the Internet, which provided a retrieval system for users to quickly and comprehensively gain healthy diet knowledge. Lee et al. [7] collected open-source data held by government institutions and constructed a knowledge graph that showed the relationships between government data in order to improve users' accessibility to government data and allow users to acquire relevant data with a single query, which greatly reduced obstacles for citizens looking to acquire necessary data. Jiang et al. [8] constructed the Knowledge Graph of Construction Safety Standards (KGCSS) to guide the development and revision of Construction Safety Standards (CSS), and they equipped it with a question input and query function, which enhanced the analyses, queries, and sharing of information regarding construction safety standards. Gao et al. [9] introduced a method to construct a knowledge graph for power system dispatching which stated the relationships between different power facilities and the influence of failures, investigated the application of the knowledge graph in power system dispatching practices, and proved that the knowledge graph could help dispatchers make decisions in the event of emergencies. Zhang et al. [10] constructed a knowledge graph to show the relationship between enterprises, and based on this knowledge graph and the target enterprise, they identified correlated enterprises, calculated the credit risk of the target enterprise using Graph Neural Networks (GNN), and took the results as the basis for investors to make investments. Hossayni et al. [11] collected machine failure and maintenance records to construct a knowledge graph so that users would be able to find solutions by searching the knowledge graph when they encountered machine failure in the future, thereby speeding up the elimination of machine breakdowns.

As shown in the above-referenced literature, a knowledge graph can clearly express the interactions between data, which will help personnel easily understand complex injection molding expertise. However, injection molding knowledge is primarily stored in the form of text, and significant personnel costs and time would be required to extract the knowledge from the text data and manually build an injection molding knowledge graph. In addition, some of the existing injection molding knowledge is stored in the minds of the experts. Because these experts do not have the professional backgrounds that lend themselves to using knowledge graphs, it will be difficult to transfer their expertise into knowledge graph. Therefore, how to extract knowledge from texts and expert statements and automatically construct an injection molding knowledge graph is the key to the progress of knowledge management in the injection molding industry.

Among the studies on knowledge extraction from texts and other unstructured data, Harnoune et al. [12] used the bidirectional encoder representation from transformers (BERT) model and conditional random field (CRF) to extract knowledge from clinical biomedical records, and this data was made into a medical knowledge graph to provide visible medical knowledge and help medical staff quickly find the information they needed. Chansai et al. [13] used BERT to extract semantic triples from dental textbooks and made them into a knowledge graph to help dental students learn important concepts in textbooks. Wei et al. [14] proposed a Novel Cascade Binary Tagging Framework and used BERT as a text encoder to extract triples from texts, and the F1 scores [15] of this model in the NYT (New York Times Annotated Corpus) public dataset [16] and the WebNLG public dataset [17] were 89.6 and 91.8, respectively. Yang et al. [18] built a DeNERT-KG model based on BERT and the Deep Q-Network to extract the subject, subject type, object, object type, and the relationship between the subject and object from sentences and construct a knowledge graph, and the model's F1 score in the TACRED (TAC Relation Extraction Dataset) dataset [19] was 72.4. Dang et al. [20] used sentence-BERT (SBERT) to integrate the knowledge of the different states of evaluation indicators for power facilities, and they constructed a knowledge graph of the evaluation indicators to help engineers evaluate the status of power facilities. Zheng et al. [21] used the BERT-CRF model to recognize the named entities and relevant terms in the records of hazardous chemical accidents, and they constructed a knowledge graph for the management and control of hazardous chemicals. Yu et al. [22] proposed a power safety named entity recognition model based on BERT-BiLSTM-CRF, and they extracted power safety-related terms from unstructured data and verified the performance of the model. The results showed that the model was superior to BiLSTM-CRF and other models.

As shown in the above-referenced literature, as BERT has an excellent performance in natural language processing and can help extract knowledge from text data, it is widely used in the construction of knowledge graphs. This study used the BERT-based entity relation extraction model to extract knowledge from the literature on injection molding and constructed an industrial knowledge graph where the entity alignment and entity classification tasks were performed by Sentence-BERT to ensure the performance. This industrial knowledge graph showed the relationships between mold design, finished product design, and process parameters in a theoretical and visible manner, which can serve as the basis for decision-making and facilitate new staff training.

In this section, we have summarized the past applications of knowledge in injection molding to illustrate the deficiencies in the industry's knowledge management. At the same time, by introducing the application of knowledge graphs in different fields, we demonstrated the potential of knowledge graphs to be used for knowledge management. Our previous research [23] has consolidated injection molding knowledge to build a knowledge graph and conducted related applications. In addition, there have been many studies [24,25] dedicated to applying knowledge graphs to the field of injection molding. Injection molding has a development history of several decades, and a lot of its professional knowledge is saved in text form. However, to the best of our knowledge, the current literature on the application of knowledge graphs in injection molding rarely explores how to extract knowledge from text data and build a knowledge graph through subsequent processing. Inspired by the literature wherein BERT was used to extract knowledge from text data, this research, based on the relevant application technology of BERT, proposes a method for building a knowledge graph from injection molding-related text data as an example of integrating text knowledge in the injection molding industry.

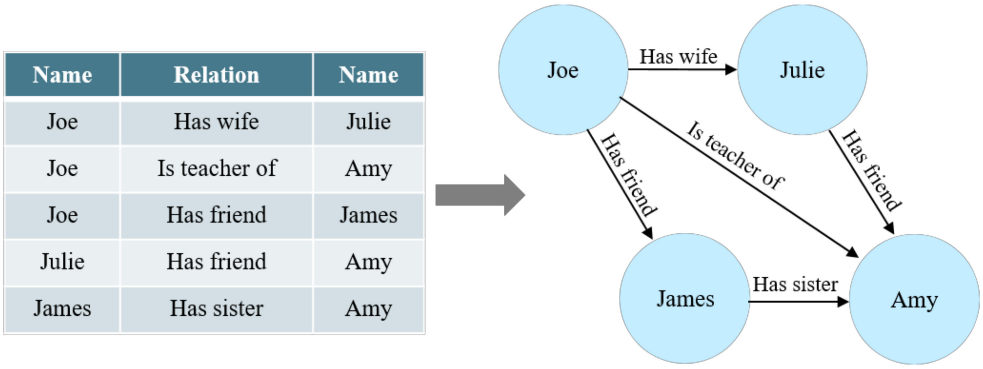
We will introduce the background knowledge of this study in Section 2, explain our methodology in Section 3, demonstrate the operation of the system through cases in Section 4, and summarize the results obtained in this research in Section 5.

2. Background Knowledge

2.1. Knowledge Graph

Google put forward the concept of knowledge graphs in 2012 [26], and it was first applied to search engines to describe the knowledge data of entity relations. Semantic triples, namely, entities, relations, and attributes, are the basic units of a knowledge graph. Entities are the nodes in a knowledge graph, and they can be used to express specific persons and things, as well as objects and abstract concepts. This study linked entities with entity relations, and thus, entity relations refers to the edges between nodes in a knowledge graph. By giving directionality to the edges between nodes, the subject and object in a relationship can be described. Moreover, attributes can be added to nodes and edges for more detailed descriptions.

As shown in Figure 1, compared with traditional tabular data, a knowledge graph can connect different entities to make information understandable from a relational perspective.



Name	Relation	Name
Joe	Has wife	Julie
Joe	Is teacher of	Amy
Joe	Has friend	James
Julie	Has friend	Amy
James	Has sister	Amy

Figure 1. An example of a knowledge graph.

In our previous research [23], we attempted to use a knowledge graph to represent knowledge related to injection molding, and we achieved preliminary results.

2.2. Database

As a traditional SQL database stores very large and complicated data, it requires a lot of time to write computer programs that describe data relations. Neo4j is a property graph database built with the Java and Scala programming languages that is specifically used for accessing graph structure data and information relations. Neo4j can quickly describe data relationships that the traditional SQL language cannot express with its built-in Cypher [27] and it can visually show graphs through D3.js, and thus, users can clearly review and edit graphs. Therefore, this study used Neo4j to store the industrial knowledge graph for injection molding.

2.3. BERT

BERT [28], which is a pre-trained natural language model put forward by Google in 2018, is composed of several transformer encoders [29], and it is a model trained using a massive amount of text data. Before reading a text, BERT processes the text through a tokenizer [28] which converts the text into token embeddings, segment embeddings, and position embeddings, and it separately records the character codes, sentence fragment annotations, and position information of each character in the text. In this way, the features of the text are extracted by these three embeddings as the input to BERT. The most significant difference between BERT and the traditional NLP (natural language processing) model is that BERT considers the position of a character in the text, as well as its corresponding context, while extracting the feature of that character in the text, and thus, BERT has an excellent performance when reading text data.

This study used a BERT-based model to extract text data features. As a simplified version of the BERT model, this BERT-based model was primarily composed of 12 layers of transformer encoders, had 110 million weight parameters, and could extract 768 dimensions of semantic features from the text data input. We used the Python programming language to import the open-source model architecture and the weights provided by Google, and we introduced the BERT-base model into the current program for subsequent applications.

3. Methodology

3.1. Knowledge Extraction

This study used an exemplar model, bert4keras [30], to extract the knowledge in the text data, and the model architecture was as shown in Figure 2. After receiving the text data processed by the tokenizer, the model extracted text features with BERT which were then input into the fully connected layer after being processed by layer normalization with a sigmoid function as the activation function to generate a two-dimensional matrix to express the predicted results of the head and tail positions of the subject. By predicting the head and tail positions of the subject, we were able to determine that the subject of 'Excessive flow resistance can lead to short shorts' is 'Excessive flow resistance'.

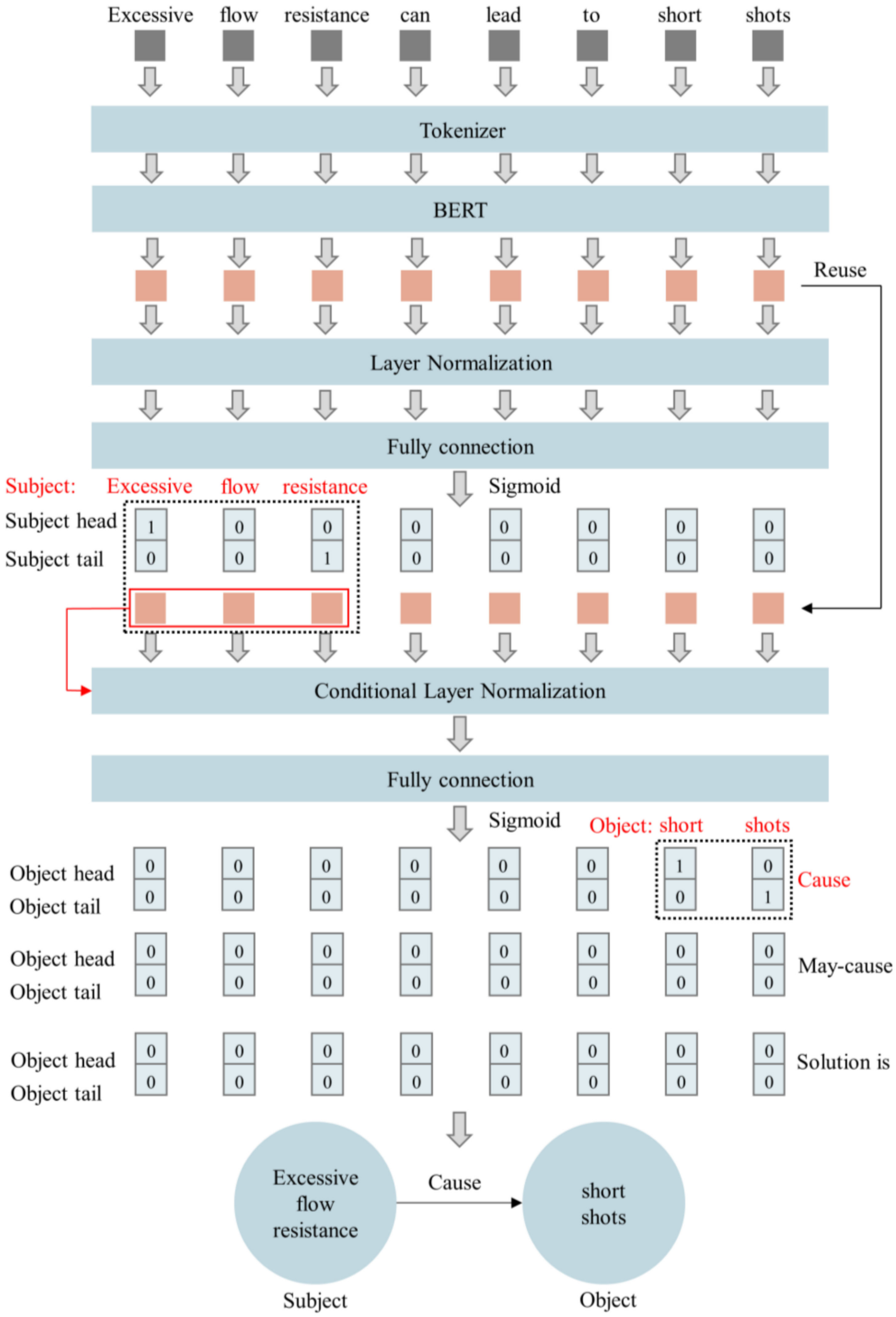


Figure 2. Architecture of the knowledge extraction model.

After predicting the head and tail positions of the subject, the model obtained the text features of the subject from the text features generated by BERT based on the head and tail positions, and it input them as the input of the conditional function in the conditional layer normalization to process the text data features generated by BERT. Then, the completed features were input into the fully connected layer, using Sigmoid as the activation function, to generate a three-dimensional matrix to express the head and tail positions of the object corresponding to the subject in different relations. Conditional layer normalization allowed the model to have different understandings of the text data based on different inputs and to handle a situation where there were multiple subjects in the text data. From the model's prediction results, it could be concluded that the subject "Excessive flow resistance" corresponded to the object "short shorts" and the type of relationship between them was "Cause".

This study prepared 1000 sets of professional knowledge text data for injection molding, which were collected from sentences in related articles and experts' descriptions of injection molding knowledge and annotated through manual means. We divided the data into 900 sets of training data and 100 sets of verification data. Binary cross entropy was used to calculate the loss value, the loss values of the subject and object were summed as the loss value of the whole model, and the F1 score was used as the indicator to evaluate the performance of the model. With a learning rate of 0.00001, the model had an F1 score of 0.899 in the dataset verification after 200 training sessions. We believed that this amount of training data would enable the model to handle a majority of the knowledge extraction tasks, but if the model were to face sentences with more complex grammatical structures in the future, a better performance could only be achieved with more training data.

3.2. Entity Alignment

In the text data of injection molding, there are often situations where different words have the same semantic meaning. When building a knowledge graph, if multiple entities with the same meaning are added, it will lead to a sparse structure of the knowledge graph, resulting in poor performance of the graph.

In the past, when comparing whether different sentences held similar meanings, a Levenshtein distance was often used as a judgment basis. A Levenshtein distance refers to the minimum number of operations required to convert one string into another string through insertion, deletion, and substitution between two strings. However, entities such as "Flash" and "Burrs" with high Levenshtein distances have the same meaning, and "increase injection pressure" and "decrease injection pressure" are two entities with low Levenshtein distances that have different meanings. Therefore, we cannot completely judge whether two entities have similar meanings based on the differences between the strings.

This study used SBERT (Sentence-BERT) to identify entities with similar semantic meanings. Figure 3 shows the architecture of the SBERT model. After receiving an entity pair, SBERT extracted the text features of each entity and projected the text features to the vector space through mean pooling, and then it calculated the cosine similarity of the coordinates of the entity pairs in the vector space as the basis for determining whether the meanings of the two entities were similar. Cosine similarity is an algorithm that quantifies the similarity between two vectors by measuring the cosine of the angle between them. In the vector space, entities with similar meanings have a higher cosine similarity, and vice versa.

This study prepared 1500 pairs of entity alignment data, which were obtained by collecting common terms in injection molding, dividing them into pairs, and annotating whether they had similar meanings through manual means. We divided the data set into 1400 training data and 100 verification data, and iteratively trained the model 200 times with the mean square error (MSE) of the cosine similarity as the loss value, AdamW as the optimizer, and 0.001 as the learning rate. During data collection, we also validated the performance of the model when trained with different amounts of training data. We found that when the number of training data reached 1400, the model showed satisfactory performance in validation. However, the validation results of the model may have differed from the actual applications. When encountering less common or misspelled terms, more training data may be required for better performance.

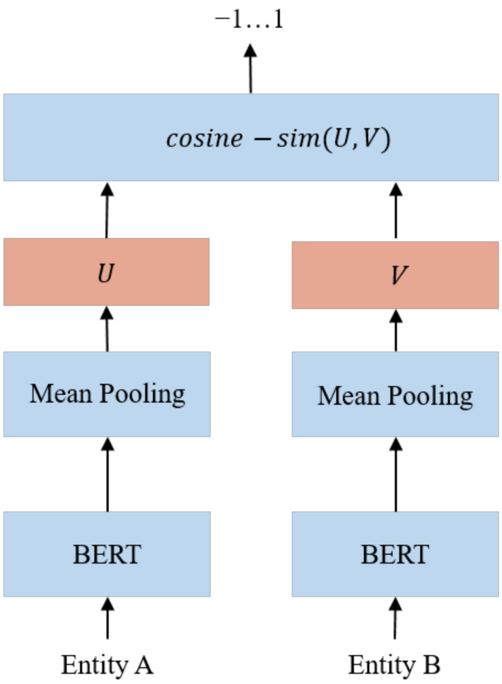


Figure 3. Architecture of the SBERT model.

During model verification, 0.7 was used as the threshold of the cosine similarity. When the cosine similarity between entities was higher than the threshold, it was concluded that the two entities had similar semantic meanings. The trained model had an accuracy rate of 0.92 for the 100 pieces of verification data.

3.3. Entity Classification

As the size of knowledge graphs is scaled up day by day, users encounter increasing difficulties in maintaining and reviewing the graphs. If entities in a knowledge graph can be classified, the cost of graph maintenance and the difficulty in graph reviewing will be greatly reduced. In previous research [23], the standard for entity classification was based on the relationship between the entity and the product defects. However, this method of classification can easily cause confusion because an entity is related to multiple defects. Therefore, this study divided the entities into eight classifications according to the domains they belonged to in injection molding, as detailed in Figure 4.

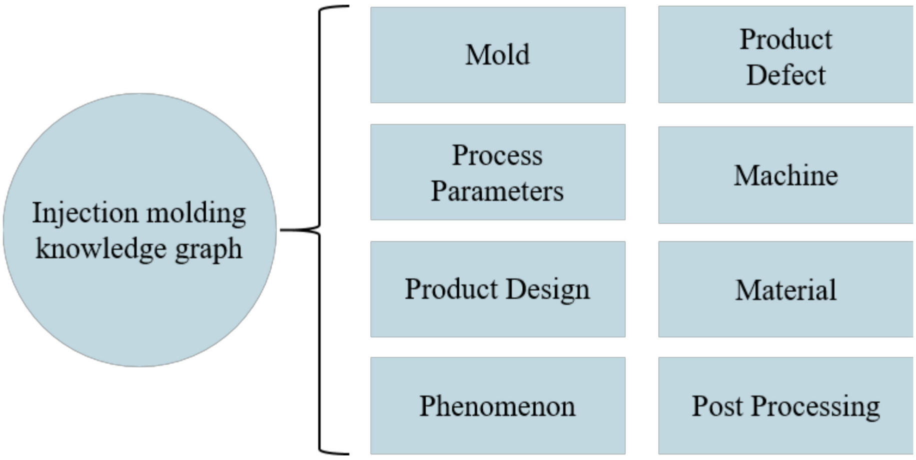


Figure 4. Entity classifications.

This study used SBERT for entity classification. As shown in Figure 5, the model first used BERT and mean pooling to project the category titles into the vector space as category coordinates, and then it projected the entity it received to the vector space in the same way, calculated the cosine similarity between the entity in the vector space and the coordinates of the different categories, and took the category with the highest similarity as the prediction result.

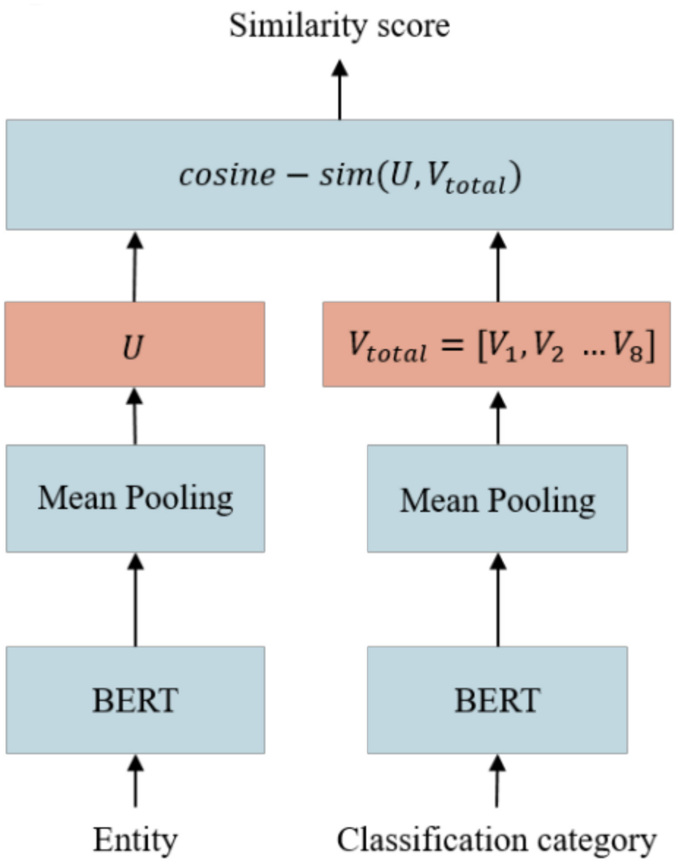


Figure 5. Architecture of the entity classification model.

This study prepares 1900 entity classification data which were obtained by collecting common terms in injection molding and categorizing them through manual means. We divided the data into 1800 pieces of training data and 100 pieces of verification data, and iteratively trained the model 200 times with the mean square error (MSE) of the cosine similarity as the loss value, AdamW as the optimizer, and 0.001 as the learning rate. The trained model had an accuracy of 0.93 for the 100 pieces of verification data. This number of training data allowed the model to effectively classify common injection molding related terms; however, when encountering less common or misspelled terms in the future, more training data may be required for the model.

4. Discussion

A significant portion of injection molding knowledge is stored in the minds of experts; however, these experts do not have backgrounds that would lend themselves to creating knowledge graphs, which results in knowledge that cannot be imported into a knowledge graph. Therefore, this study built a web platform for users to import knowledge into graphs by inputting knowledge statements. The overall operation flow is shown in Figure 6. After the system received the user's input sentence, it first used a knowledge extraction model to find the subject, the object, and the predicate between them in a sentence. After the user confirmed that the extraction result was correct, the system then used an entity alignment model to compare whether the entities in the result were similar in semantic meaning to the existing entities in the knowledge graph, and then the system presented it to the user for confirmation. Finally, through the entity classification model, the system classified the entities based on their semantic meaning. The knowledge processed through the above procedures would then be imported into the graph.

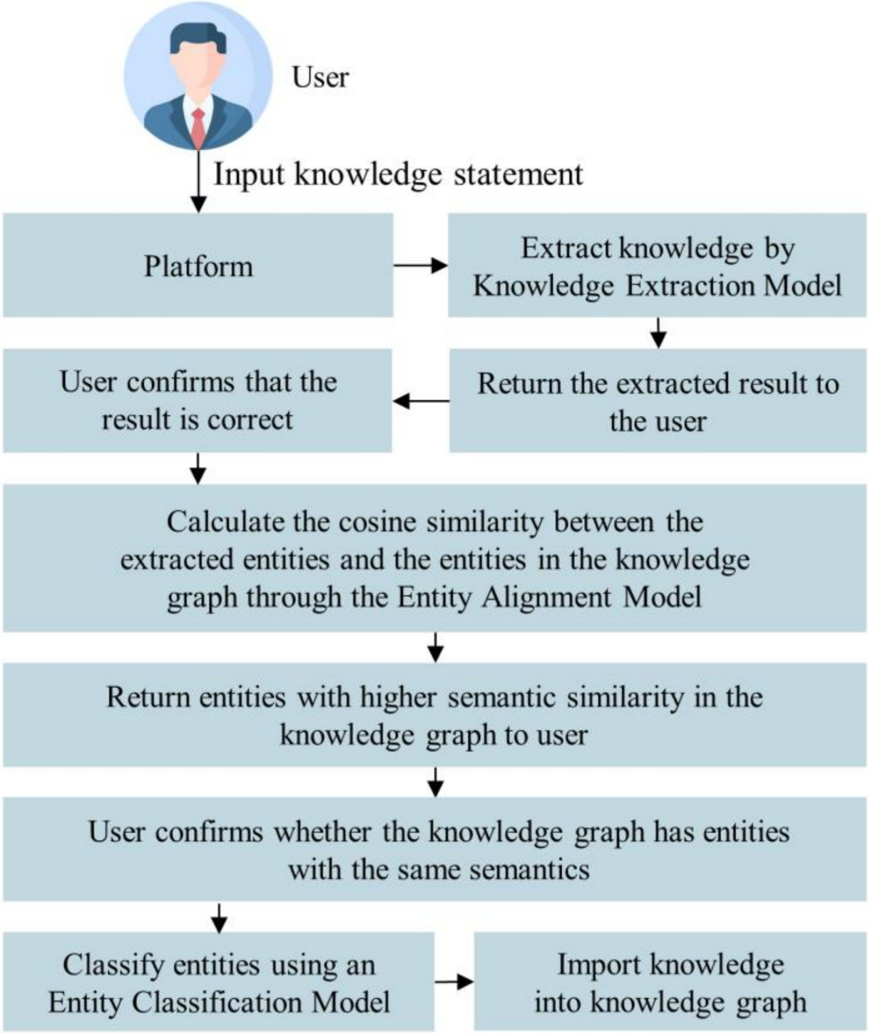
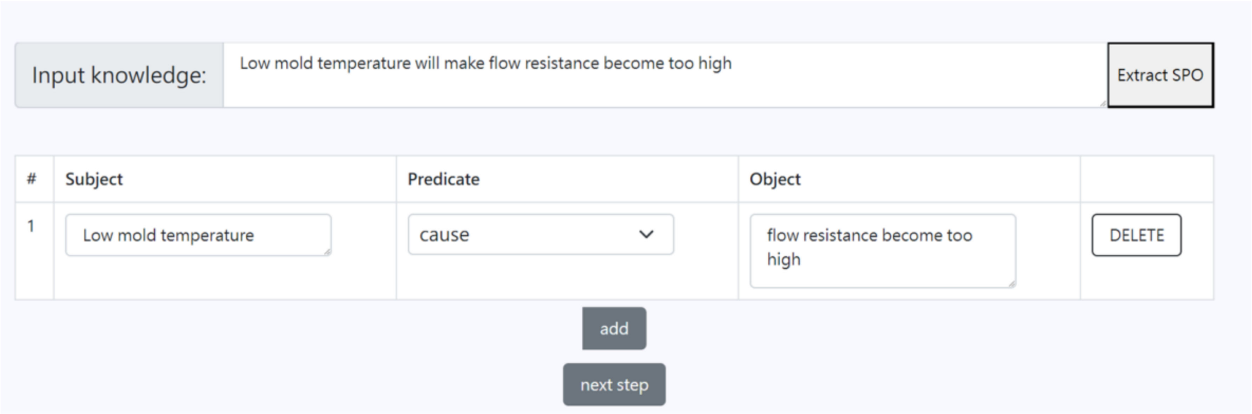

User
Input knowledge statement
Platform
Extract knowledge by Knowledge Extraction Model
Return the extracted result to the user
User confirms that the result is correct
Calculate the cosine similarity between the extracted entities and the entities in the knowledge graph through the Entity Alignment Model
Return entities with higher semantic similarity in the knowledge graph to user
User confirms whether the knowledge graph has entities with the same semantics
Classify entities using an Entity Classification Model
Import knowledge into knowledge graph

Figure 6. Process flow diagram of importing knowledge into a knowledge graph.

4.1. Knowledge Extraction Results

Figure 7 show the results of the knowledge extraction model's extraction of the subject, the object, and their predicate from the knowledge statement sentence. Users could review the extracted results to determine if they were correct and make any necessary corrections. When the system detected that a user had made corrections, it would store that sentence and the corrected results for model re-training.



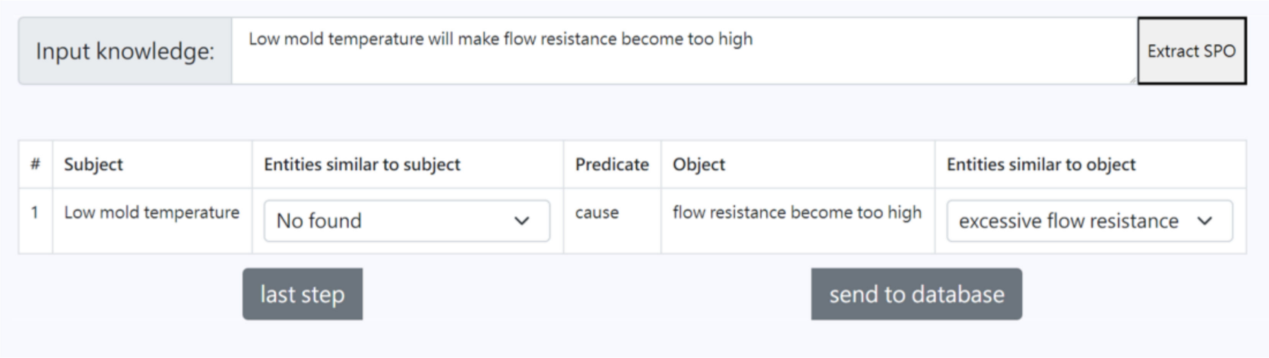
Input knowledge: Low mold temperature will make flow resistance become too high

#	Subject	Predicate	Object	
1	Low mold temperature	cause	flow resistance become too high	DELETE

Figure 7. Real-time picture of knowledge extraction.

4.2. Entity Alignment Results

As shown in Figure 8, after confirming that the knowledge extraction results were correct, the system calculated the cosine similarity between the entities in the extracted results and the existing entities in the knowledge graph with the entity alignment model. Afterwards, the top three entities with similarities higher than 0.7 were returned to the page for the users to confirm whether there were entities with the same semantic meaning in the knowledge graph.



Input knowledge: Low mold temperature will make flow resistance become too high

#	Subject	Entities similar to subject	Predicate	Object	Entities similar to object
1	Low mold temperature	No found	cause	flow resistance become too high	excessive flow resistance

Figure 8. Picture of entity alignment.

4.3. Entity Classification Results

After the entity alignment process, the knowledge extraction results were classified by an entity classification model based on the semantic meaning of the entities before being imported into the knowledge graph. The entities in the extracted results were then classified according to their semantic meaning.

Figure 9 shows the real-time picture of a triple being imported into Neo4j after entity classification, where the model regarded "Excessive flow resistance" as a node in the phenomenon domain and concluded that "Low mold temperature" belonged to the node in the process parameter domain.

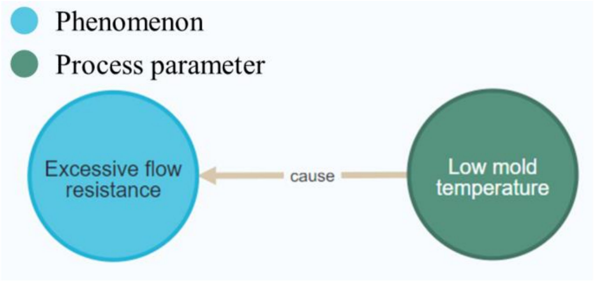


Figure 9. Picture of entity classification.

4.4. Practical Application

In this study, five injection molding text data were selected as actual application examples, as shown in Table 1. The system extracted knowledge from text data with the knowledge extraction model, organized the knowledge, and constructed the knowledge graph with the entity alignment and entity classification models. Figure 10 shows a real-time picture of the first two data being imported into Neo4j after being processed by the system, where the system concluded that "High flow resistance" had a semantic meaning similar to "Excessive flow resistance" in the graph, and thus, it did not import "High flow resistance" into the graph. Instead, the system imported entities in the text data related to "High flow resistance" into the knowledge graph and linked them to "Excessive flow resistance", and thus, the second piece of text data was processed after successfully constructing the knowledge graph with the text data.

Table 1. Injection molding text data.

Index	Text
1	Insufficient injection pressure and excessive flow resistance cause short shots.
2	High flow resistance can be improved by reducing the melt viscosity.
3	Too-low mold temperatures and too-slow injection speeds cause the solidified layer to be too thick, which further leads to excessive flow resistance.
4	Too-low melting temperatures cause excessive melt viscosity, which further causes excessive flow resistance.
5	Reason for excessive flow resistance: (1) inappropriate gate location, (2) nozzle size is too small, or (3) product is too thin.

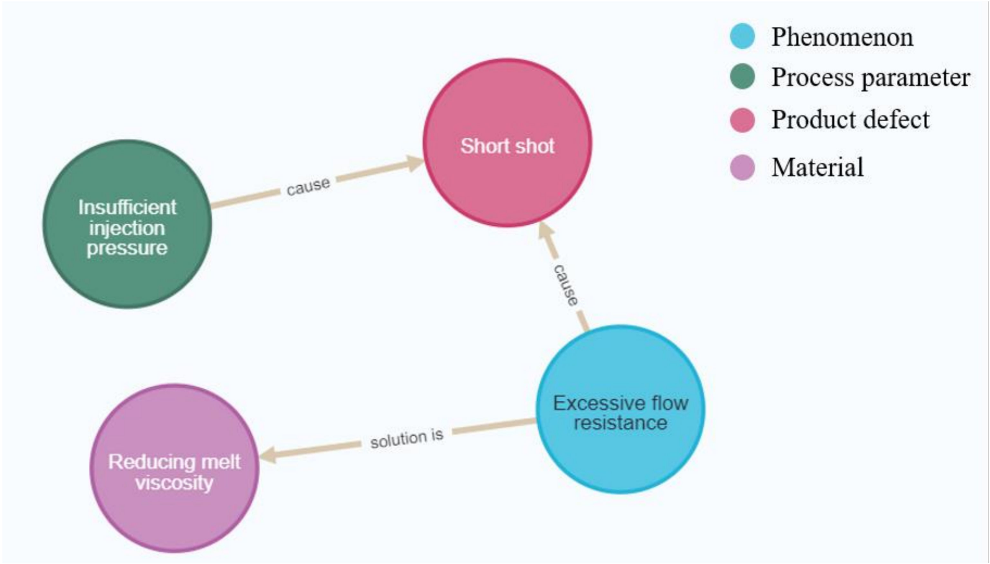


Figure 10. Results of importing the first two data sets.

Figure 11 shows the knowledge graph that was composed of all the text data through the system, and it verified that the system was able to integrate the knowledge contained in multiple pieces of text data and store it in the form of a knowledge graph.

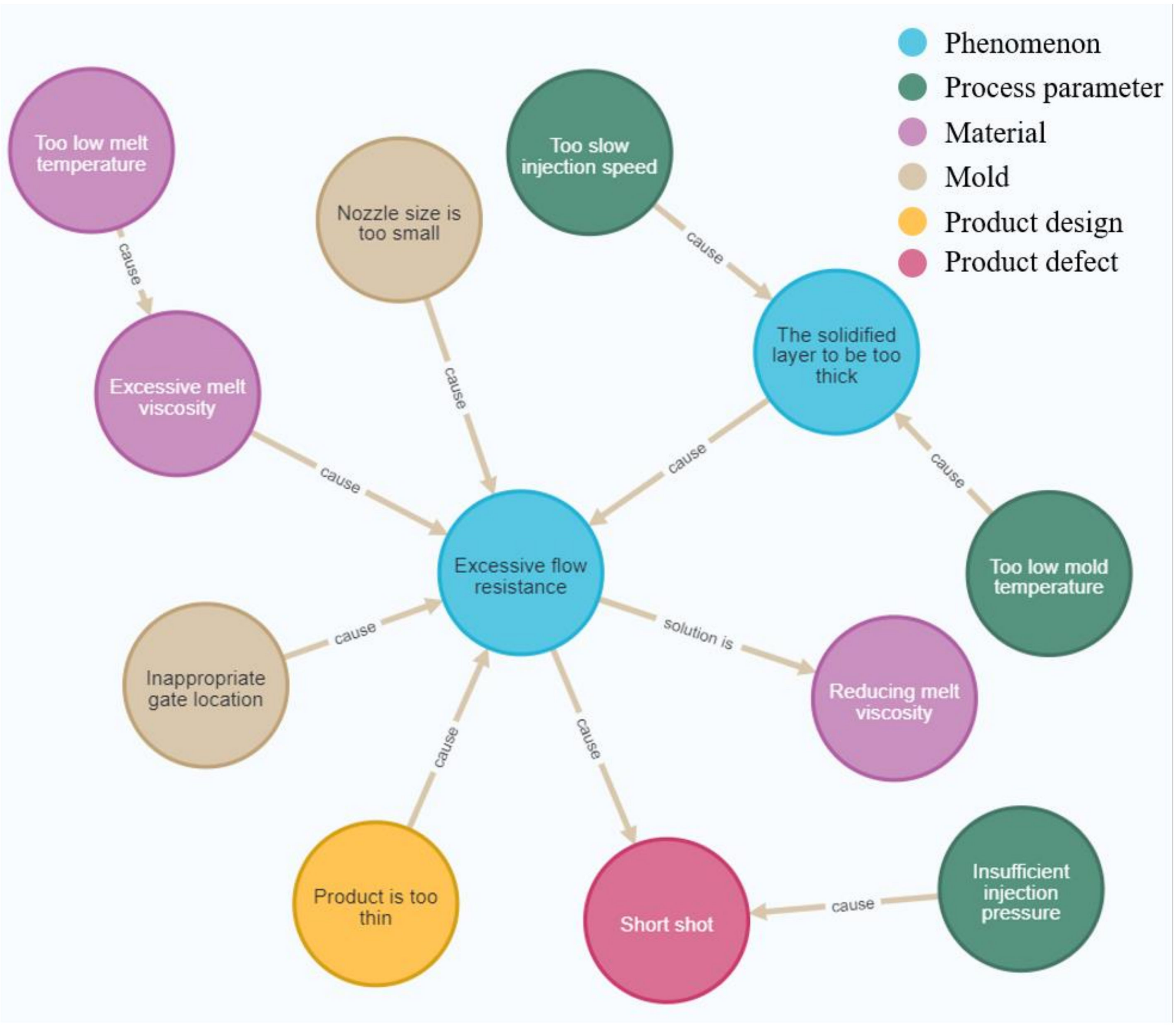


Figure 11. Results of all imported data sets.

This study prepared 300 text data sets related to injection molding, and it used a system to extract the knowledge contained in them to construct a knowledge graph. During the process of constructing the knowledge graph, there were five text data sets where the system was unable to completely extract the subject and the object and their relationship from the sentence, requiring the users to revise the extraction results to ensure the completeness of the knowledge being imported.

Figure 12 shows an injection molding knowledge graph created by the system using the 300 text data sets. The graph has 207 entities and 258 relationships. After manual verification, we did not find any semantically similar entities in the knowledge graph, and we also confirmed that all the entity classification results were correct. However, the system can be improved. After the system's implementation, the users reported that when dealing with less common terms, there were problems with entity alignment and classification errors. These issues can be improved in the future by increasing the amount of training data for the model.

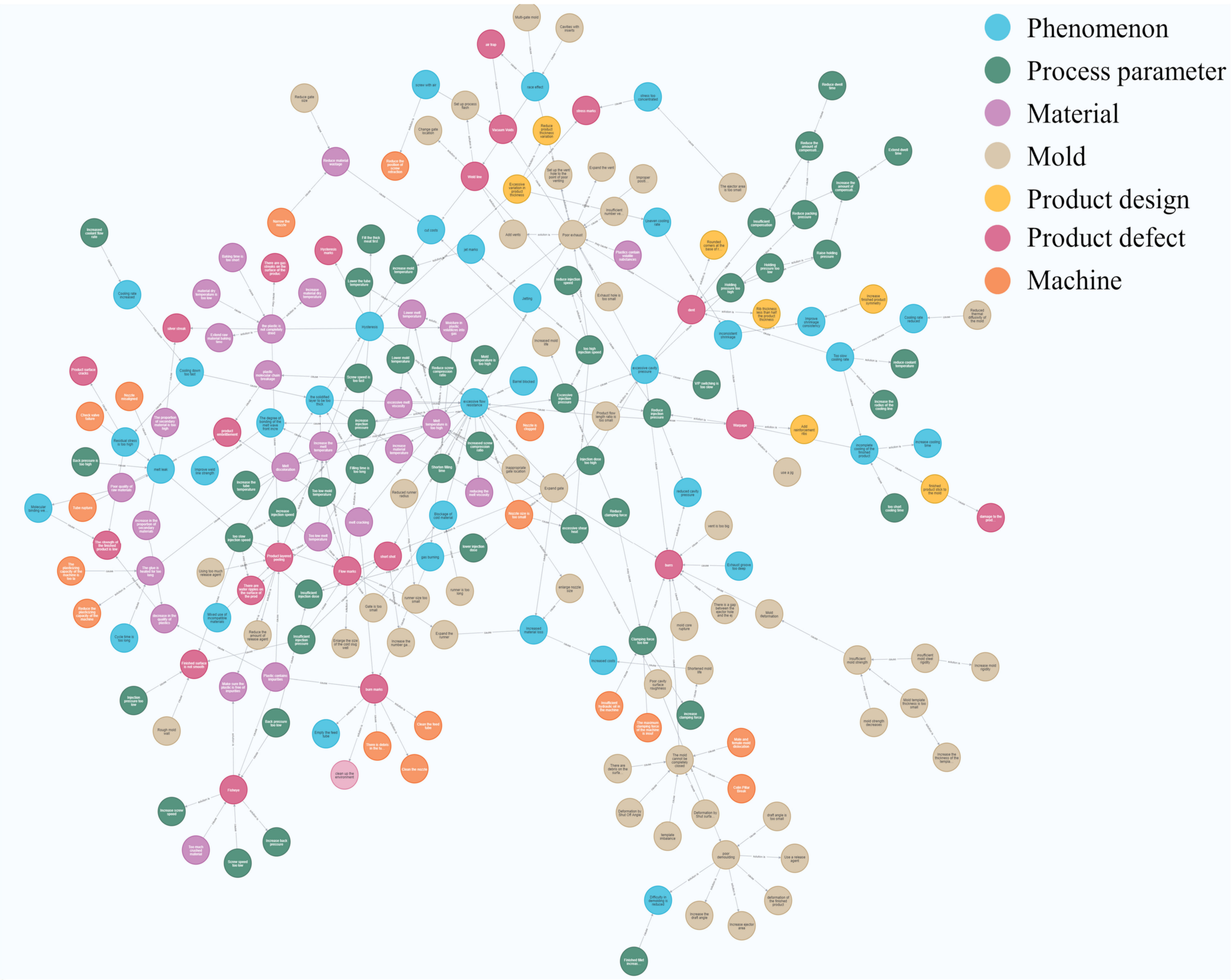


Figure 12. Injection molding knowledge graph.

5. Conclusions

This study used a knowledge extraction model, an entity alignment model, and an entity classification model to extract injection molding knowledge contained in the text data, and I constructed a web platform to integrate these models. The validation data for our prepared knowledge extraction model contained various common grammatical structures, while both the entity alignment model and the classification model validation data included commonly used terms in casting. Thus, the system obtained an F1 score of 0.899 and accuracy rates of 0.92 and 0.93, respectively, from the three models in the validation data, which proved that their performance had reached a level that could be practically applied. Verification through practical cases showed that the platform could extract knowledge from injection molding related text data and import it into a knowledge graph. In our previous study [23], an injection molding knowledge graph was manually constructed by converting knowledge in expert interviews and text data into triples, and the subsequent entity alignment and entity classification tasks were also completed manually, resulting in a process of building a knowledge graph where a large amount of human resources was required. Furthermore, as the number of nodes in a knowledge graph increases, it can become difficult to perform entity alignment through manual means. The platform developed in this study can automate the tasks of knowledge extraction, entity alignment, and entity classification through natural language technology, significantly reducing the human resources required to build the knowledge graph, while also solving the difficulty problem of entity alignment tasks increasing as the number of nodes in the knowledge graph increases. Finally, we built an injection molding knowledge graph through the platform, expressing and storing the intricate injection molding knowledge in different types of nodes and edges. Through subsequent applications, the injection molding knowledge graph will be able to help in personnel training and decision-making.

Author Contributions: Conceptualization, Z.-W.Z., Y.-H.T. and W.-R.J.; methodology, Z.-W.Z., Y.-H.T., W.-R.J. and S.-C.C.; software, Z.-W.Z.; validation, Z.-W.Z., Y.-H.T. and W.-R.J.; writing—original draft preparation, Z.-W.Z. and W.-R.J.; writing—review and editing, Y.-H.T., S.-C.C. and M.-C.C.; visualization, Z.-W.Z.; supervision, S.-C.C. and M.-C.C.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data are not publicly available due to subsequent research needs.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jong, W.R.; Wu, C.H.; Li, M.Y. Web-based navigating system for conceptual mould design with knowledge management. *Int. J. Prod. Res.* **2010**, *49*, 553–567. [CrossRef]
2. Jong, W.R.; Ting, Y.H.; Li, T.C.; Chen, K.Y. An integrated application for historical knowledge management with mould design navigating process. *Int. J. Prod. Res.* **2012**, *51*, 3191–3205. [CrossRef]
3. Khosravani, M.R.; Nasiri, S.; Weinberg, K. Application of case-based reasoning in a fault detection system on production of drippers. *Appl. Soft Comput.* **2018**, *75*, 227–232. [CrossRef]
4. Mikos, W.L.; Ferreira, J.C.E.; Gomes, F.G.C. A distributed system for rapid determination of nonconformance causes and solutions for the thermoplastic injection molding process: A Case-Based Reasoning Agents approach. In Proceedings of the IEEE International Conference on Automation Science and Engineering, Trieste, Italy, 24–27 August 2011.
5. Sebastian, R.B.; Irlan, G.G.; Priyanka, N.; Maria-Esther, V.; Maria, M. A Knowledge Graph for Industry 4.0. *Semant. Web* **2020**, *1212*, 465–480.
6. Chi, Y.; Yu, C.; Qi, X.; Xu, H. Knowledge Management in Healthcare Sustainability: A Smart Healthy Diet Assistant in Traditional Chinese Medicine Culture. *Sustainability* **2018**, *10*, 4197. [CrossRef]
7. Lee, J.W.; Park, J. An Approach to Constructing a Knowledge Graph Based on Korean Open-Government Data. *Appl. Sci.* **2019**, *9*, 4095. [CrossRef]
8. Jiang, Y.; Gao, X.; Su, W.; Li, J. Systematic Knowledge Management of Construction Safety Standards Based on Knowledge Graphs: A Case Study in China. *Environ. Res. Public Health* **2021**, *18*, 10692. [CrossRef] [PubMed]
9. Gao, H.; Miao, L.; Liu, J.; Dong, K.; Lin, X. Construction and Application of Knowledge Graph for Power System Dispatching. In Proceedings of the International Forum on Electrical Engineering and Automation, Hefei, China, 25–27 September 2020.
10. Zhang, J.; Song, Z. Research on knowledge graph for quantification of relationship between enterprises and recognition of potential risks. In Proceedings of the IEEE International Conference on Cybernetics, Beijing, China, 5–7 July 2019.
11. Hossayni, H.; Khan, I.; Aazam, M.; Taleghani-Isfahani, A.; Crespi, N. SemKoRe: Improving Machine Maintenance in Industrial IoT with Semantic Knowledge Graphs. *Appl. Sci.* **2020**, *10*, 6325. [CrossRef]
12. Harnoune, A.; Rhanoui, M.; Mikram, M.; Yousfi, S.; Elkaimbillah, Z.; Asri, B.E. BERT based clinical knowledge extraction for biomedical knowledge graph construction and analysis. *Comput. Methods Programs Biomed. Update* **2021**, *1*, 100042. [CrossRef]
13. Chansai, T.; Rojpaismkit, R.; Boriboonsub, T.; Tuarob, S.; Yin, M.S.; Haddawy, P.; Hassan, S.; Pomarlan, M. Automatic Cause-Effect Relation Extraction from Dental Textbooks Using BERT. In Proceedings of the 23rd International Conference on Asia-Pacific Digital Libraries, Virtual Event, 1–3 December 2021.
14. Wei, Z.; Su, J.; Wang, Y.; Tian, Y.; Chang, Y. A Novel Cascade Binary Tagging Framework for Relational Triple Extraction. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Virtual Event, 5–10 July 2020.
15. Yutaka, S. *The Truth of the F-Measure*; Technical Report; University of Manchester: Manchester, UK, 2007.
16. Evan, S. The New York Times Annotated Corpus LDC2008T19. In *Web Download*; Linguistic Data Consortium: Philadelphia, PA, USA, 2008.
17. Gardent, C.; Shimorina, A.; Narayan, S.; Perez-Beltrachini, L. Creating Training Corpora for NLG Micro-Planners. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July 2017.
18. Yang, S.; Yoo, S.; Jeong, O. DeNERT-KG: Named Entity and Relation Extraction Model Using DQN, Knowledge Graph, and BERT. *Appl. Sci.* **2020**, *10*, 6429. [CrossRef]
19. Yuhao, Z.; Zhong, V.; Chen, D.; Angeli, G.; Manning, C.D. Position-aware Attention and Supervised Data Improve Slot Filling. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 9–11 September 2017.

20. Dang, J.; Wei, H.; Yan, Y.; Jia, R.; Li, Y. Construction of Knowledge Graph of Electrical Equipment Based on Sentence—BERT. In Proceedings of the 2021 International Conference on Advanced Electrical Equipment and Reliable Operation, Beijing, China, 15 October 2021.
21. Zheng, X.; Wang, B.; Zhao, Y.; Mao, S.; Tang, Y. A knowledge graph method for hazardous chemical management: Ontology design and entity identification. *Neurocomputing* **2020**, *430*, 104–111. [CrossRef]
22. Yu, J.; Sun, J.; Dong, Y.; Zhao, D.; Chen, X.; Chen, X. Entity recognition model of power safety regulations knowledge graph based on BERT-BiLSTM-CRF. In Proceedings of the 2021 IEEE International Conference on Power Electronics, Computer Applications, Shenyang, China, 22–24 January 2021.
23. Zhou, Z.-W.; Ting, Y.-H.; Jong, W.-R.; Chiu, M.-C. Knowledge Management for Injection Molding Defects by a Knowledge Graph. *Appl. Sci.* **2022**, *12*, 11888. [CrossRef]
24. Bachhofner, S.; Kiesling, E.; Revoredo, K.; Waibel, P.; Polleres, A. Automated Process Knowledge Graph Construction from BPMN Models. In Proceedings of the International Conference on Database and Expert Systems Applications, Database and Expert Systems Applications, Vienna, Austria, 22–24 August 2022.
25. Wang, Y.; Zou, J.; Wang, K.; Yuan, X.; Xie, S. Injection Molding Knowledge Graph Based on Ontology Guidance and Its Application to Quality Diagnosis. *J. Electron. Inf. Technol.* **2022**, *44*, 1521–1529.
26. Singhal, A. Introducing the Knowledge Graph: Things, Not Strings. 2021. Available online: https://googleblog.blogspot.co.at/2012/05/introducing-knowledge-graph-things-not (accessed on 6 August 2022).
27. Guia, J.; Soares, V.G.; Bernardino, J. Graph Databases: Neo4j Analysis. In Proceedings of the International Conference on Enterprise Information Systems, Porto, Portugal, 26–29 April 2017.
28. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
29. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
30. Su, J. bert4keras. Available online: https://github.com/bojone/bert4keras (accessed on 23 October 2022).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.