Article

A Knowledge Graph Based Implicit Requirements Mining Method in Personalized Products Development

Zhenchong Mo , Lin Gong * , Jun Gao , Haoran Cui , Junde Lan

Posted Date: 12 July 2024

doi: 10.20944/preprints202407.1040.v1

Keywords: Implicit requirement; Knowledge graph; Requirement mining; Patent analysis; Engineering design; Generative design; Crowds innovation

Article

A Knowledge Graph Based Implicit Requirements Mining Method in Personalized Products Development

Zhenchong Mo [1], Lin Gong [1,2,*], Jun Gao [1], Haoran Cui [1] and Junde Lan [1]

[1] School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China; mozc@bit.edu.cn
[2] Yangtze Delta Region Academy of Beijing Institute of Technology, Jiaxing 314000, China
* Correspondence: gonglin@bit.edu.cn

Abstract: In the context of crowds innovation and the generative design driven by big language models, the personalized requirements has become the best engine for innovative design. The exploration of personalized requirements has become a key in significantly improving product innovation, concepts feasibility, and design interaction efficiency. To mine a large number of vague and unexpressed implicit requirements of personalized products, a domain knowledge graph-based method is proposed in this research. First, based on the classical theory of design science, the characteristics and categories of personalized implicit requirements are analyzed. Next, in order to improve the practicability and construction efficiency of the domain knowledge graph, a more informative ontology is constructed, and better performing NLP models are proposed. Then, a series of mining methods based on knowledge graph are proposed for the personalized implicit requirements of different categories. Finally, a platform was developed based on the technical solution proposed in this study, and an example verification was conducted in the field of electromechanical engineering. The efficiency improvement of the training model proposed in the research was analyzed, and the practicality of implicit requirement mining methods were discussed.

Keywords: implicit requirement; knowledge graph; requirement mining; patent analysis; engineering design; generative design; crowds innovation

1. Introduction

The field of engineering design, especially in product design, service system design, and complex system design, is being reshaped by two major trends: on the one hand, with the rise of large language models, generic-field generative intelligent design is gradually becoming mainstream[1]; On the other hand, driven by the Internet, human designers have formed new design models such as crowdsourcing design and group intelligence innovative design through collaborative organization [2–4]. The core goal of these development trends is to address the growing personalized requirements and achieve rapid iterative design of products and user-participation design.

In engineering design, the expression of personalized requirements has significant semi professional characteristics, which will be elaborated in detail in Chapter 3 "Nature of the Problem". This semi professional feature provides feasibility for directly extracting requirements elements from user descriptions, but also increases the importance and difficulty of implicit requirement mining. In the field of generative intelligent design, a major challenge is how to make machines empathetic [5], so that after users input personalized requirements, they can empathize with the design and product usage context, automate the completion of users' prompts, mine design problems that users have not anticipated or are difficult to express clearly, and improve the output quality of intelligent generative models. In terms of group intelligence innovation design, there are limitations in user understanding of products, which is not conducive to the emergence of innovation, but also becomes a fundamental problem that leads to the design process being repetitive, chaotic, and of low design quality[2–4]. Therefore, accurately mining and supplementing implicit needs that have not been clearly expressed in personalized requirements is of great significance for improving the innovation of engineering design, the completeness of product functions, and the feasibility of solutions.

Many studies have analyzed the design requirements of [6–8] through data-driven technologies such as machine learning. Although these studies extensively use data derived from user reviews on the Internet, user-generated content (UGC) has inherent limitations in complementing user needs [9,10]. In addition, the review data in the field of engineering design is difficult to meet the research needs in quantity and quality, and its timeliness is not in line with the development trend of engineering design. To overcome these limitations, this study introduced the engineering design knowledge contained in the patents.

This research presents a method to identify implicit personalized requirements using knowledge graphs. First, this study provides an in-depth analysis of semi professional, task-oriented personalized requirements from the perspective of design science and engineering theory to reveal the characteristics of implicit requirements. Then, it builds a domain knowledge graph by gathering product design knowledge from patents. TBy optimizing the ontology layer and stacking the pre-training model, the comprehensiveness and practicability of the knowledge graph are enhanced while improving its construction efficiency. Finally, taking the user's personalized description as input, extract the explicit requirement elements and match the corresponding knowledge entities, and realize the mining of multiple personalized implicit requirements based on the methods of structural similarity and link prediction.

This study makes several key contributions:

- It makes a detailed analysis of the user requirements of front-end design stage, clarifies the characteristics, forms and types of the personalized implicit requirements of users that drive product innovation, and enriches the theoretical basis of the related research of the implicit requirement mining for innovative design.
- It enriches the domain patent knowledge graph ontology layer, improves the comprehensiveness and practicability of the graph, and improves the accuracy and efficiency of the knowledge graph construction by optimizing the pre-training model of the entity layer.
- Aiming at the different characteristics of personalized implicit requirements driving product innovation, this paper puts forward a series of targeted implicit requirement mining method system based on the domain knowledge graph.

In the upcoming chapters, we will systematically explore and analyze personalized implicit requirement mining methods. Chapter 2 will closely review literature related to the study's topic, providing readers with the research background and theoretical basis in this field. Chapter 3 will deeply analyze the unique nature of the research object and problem and clarify the research's boundaries and purpose. In Chapter 4, we will propose a technical solution for building a domain knowledge graph. Chapter 5 will elaborate on the personalized implicit requirement mining methodology. Chapter 6 will demonstrate the platform we developed, verify the effectiveness and practicality of our method through case studies, and discuss the performance of the algorithmic model proposed during the study. Finally, in Chapter 7, we will comprehensively summarize the entire research and propose prospects for future research directions.

2. Related Works

2.1. User Requirements Analysis in Engineering Design

In order to obtain valuable user requirements from these data, there is an increasing number of studies on algorithms and platforms for parsing consumer opinion data, such as identifying high-quality consumer opinion data, intelligently detecting customer sentiment polarity and its corresponding opinion targets, and automatically ranking the importance of user requirements [11]. User requirement analysis theory can be divided into three categories: based on the Kano model, based on fuzzy set and rough set theory, and based on natural language processing algorithms. The research objects of related studies are mainly user online reviews of mass-produced consumer products.

2.1.1. Methods Based on the Kano Model

The Kano model is a user requirements analysis method proposed by Noriaki Kano in the 1980s [12]. This method divides user requirements into basic, expected, attractive, indifferent, and reverse requirements and shows the relationship between different types of requirements and customer satisfaction. Since its introduction, the Kano model has become one of the most commonly used requirement analysis models by marketing, management practitioners, and researchers [13]. Zhou F et al. [14] used FastText technology to filter out uninformative comments from Internet product reviews from the perspective of the product ecosystem. Then, a topic modeling technique was used to extract various topics related to customer requirements, and a rule-based sentiment analysis method was applied to predict the comments' sentiment and intensity values. Finally, based on the "dissatisfaction-satisfaction" pairs in sentiment analysis, the analytical Kano model was used to classify customer requirements related to the extracted topics. Budiarani V H et al. [15] implemented a Kano model for e-commerce. They used it to study user satisfaction with two widely used digital wallets in online shopping transactions during COVID-19. Janmejay Bhardwaj et al. [16] studied the 20 characteristics of products based on the Kano model to determine whether a specific requirement characteristic plays a decisive role in the customer's purchasing behavior.

2.1.2. Methods Based on Rough Sets and Fuzzy Sets

The theory of fuzzy sets was introduced by Zadeh LA in 1965. Fuzzy set theory is based on fuzzy mathematics and studies imprecise phenomena by establishing appropriate membership functions [17]. Rough set theory was first proposed by Polish scientist Pawlak Z [18] in 1982 and has been widely used in rule induction, feature selection, and other applications. Rough and fuzzy sets are two important mathematical tools to deal with uncertainty after probability theory, which play an important role in tackling the ambiguity and uncertainty of requirement. To determine the relative importance level of heterogeneous fuzzy customer requirements more objectively and accurately, Zhang P et al. [19] proposed a weighted interval rough number method. They addressed the problem of customer heterogeneity fusion by assigning different weights to customers. Chen Z et al. [20] presented a hybrid framework for identifying user service requirements and determining priorities to address the lack of systematic induction and assessment methods for intelligent services requirements in uncertain environments. Haber N et al. [21] combined the Kano model and the fuzzy analytic hierarchy process (Fuzzy AHP) to analyze and evaluate customer requirements and select a product service system that can improve product value. Zhang Meixia et al. [22] used fuzzy reasoning to derive the spatiotemporal distribution of requirements based on survey data and chain theory.

2.1.3. Methods Based on Natural Language Processing

Natural Language Processing (NLP) is a theory-driven computing technology that automatically analyzes and represents human language [23]. In recent years, neural networks based on dense vector representation have yielded excellent results in various NLP tasks. As NLP technology continues to advance, there has been a gradual increase in related studies focused on processing user requirements using this method. Li X et al. proposed a kind of context perception diversity knowledge recommendation method, through semantic analysis, context definition and perception, user portrait modeling to solve the item diversity, context diversity and user diversity, realize the diversity and accuracy of the recommended knowledge, meet the requirements of many stakeholders in the design process[24]. To leverage critical information in online review texts for automobile design and development, Zhang Guofang et al. utilized TF-IDF and dependency syntactic analysis methods to extract product features. They then used the BERT pre-trained model for text sentiment analysis and built a House of Quality to transform requirements into technical engineering features. This approach resulted in a product planning method driven by review data[25].

However, unlike general user requirements, personalized requirements do not often appear on ordinary e-commerce platforms but more often appear in crowdsourcing or collaborative community platforms in the form of product development tasks. This article will take this type of personalized requirements as the primary research object.

2.2. Mining and Completion of High Personalized Degree of User Requirements

Many scholars have proposed to explore the implicit needs through Empathic Design, observing the scenarios of users using products, letting users participate in the design process and based on use case reasoning. Zhou F et al. [26] proposed a two-layer model that combines sentiment analysis and case analogy reasoning to address the problem of potential user requirements often being hidden in user requirements semantics and difficult to recognize by typical text mining based methods. This model is used to identify explicit user requirements and deduce implicit features of potential user requirements.. Timoshenko A et al. [27] identified customer requirements from UGC by combining machine learning methods, using convolutional neural networks to filter out non-informative content, and using interview methods to confirm requirements that could not be discovered effectively. Wang Z et al. [28] predefined ontologies such as products, services, and scenarios, extracted requirements generated scenarios and targeted products and services from historical requirement data, constructed a requirement graph, and completed requirement elicitation based on the DeepWalk algorithm. Chen R et al. [29] proposed a domain-based requirement mining framework to promote improving mobile application quality. Zhang M et al. [30] proposed a deep learning-based method to identify and extract product innovation ideas from online review data, using the LSTM algorithm to identify sentences containing innovative ideas from reviews. Chen K et al. [31] extracted explicit and implicit features from user-generated text to obtain requirements. Then they used a multi-layer neural network to distinguish the impact of positive and negative opinions on each product feature.

From the above research, we can see that, firstly, in the process of identifying and improving the requirements, it is difficult to build a stable communication channel between the users and the designers, which is difficult to assist the designers to clarify and improve the user requirements, let alone further inspire the innovative design. Secondly, the above implicit requirement inspiration methods are primarily based on historical requirement data, user usage data and other UGC, which requires a large number to support use case reasoning. However, the number of requirements for the same personalized products is generally small, which makes it challenging to support accurate requirement identification and completion. Finally, the above methods still require a lot of subjective judgments, and can hardly solve the subjective limitations of personalized requirements for innovative design. Therefore, based on the existing analysis of personalized requirement data, it is urgent to integrate relevant domain knowledge to assist.

2.3. Knowledge Graph in Engineering Design

Concept design is a knowledge-intensive process. Design knowledge is a crucial design resource in motivating designers to inspire creativity in concept design. Design knowledge in multidisciplinary fields mainly exists in the form of natural language. The rapid development of NLP technologies, such as named entity recognition, topic discovery, embedding models, pre trained models, sentiment recognition, text similarity calculation, search recommendation, text generation, etc., provides opportunities for designers to process unstructured text and obtain design knowledge from it. L. Siddharth, Jianxi Luo, et al. reviewed the essential applications of natural language processing technology in concept design. He assigned corresponding natural language processing technology and models to different design stages under the existing conceptual design framework [32]. Jia J et al. proposed a method for capturing and reusing implicit knowledge in the design process. The design problem and design solution are represented by constructing a design knowledge graph. The implicit design knowledge is reused through relational learning, tensor decomposition, and other technologies [33]. Jianxi Luo et al. proposed a computer-aided design creative generation method based on the InnoGPS system. In response to the problems of traditional design creative generation relying on the knowledge or intuition of human experts and high uncertainty, combined with data-driven design methods, a cloud based computer-aided rapid creative process is provided. The InnoGPS proposed in this study integrates an experience network diagram of all technical fields based on international patent classification, with functions such as technology spatial map, technology positioning, neighboring domain recommendation, map browsing and domain discovery, and concept discovery within the domain[34]. Liu Q et al. proposed a function-structure concept network construction and analysis method to support an intelligent product design system, exploring the design information association containing explicit and implicit associations as a stimulus to stimulate creativity to support design conception[8]. Gong Lin et al. proposed a cross-domain knowledge discovery method based on knowledge graph and patent mining. Knowledge elements are classified through natural language processing-related technologies such as BERT and word2vec, and the correlation between cross-domain knowledge is mined[35]. Sarica S et al. proposed a knowledge-based expert system to guide data-driven design conception through knowledge distance[36].

As a hallmark of the fourth industrial revolution, Artificial General Intelligence (AGI) has brought about a paradigm shift in design. Large-scale language models such as GPT demonstrate significant integration of cross-domain knowledge and general reasoning capabilities through learning and training on large-scale data, exhibiting empathy and creativity in extremely high dimensions and sequences, and have broad application prospects in creative generation, design questioning and answering, etc., and stimulating the possibility of design change[37]. Xinyu Zhang et al. proposed a generative design method based on pre-trained language model to generate biological incentive design concepts in the form of natural language[38]. Qihao Zhu et al. explored an intelligent design idea generation method based on pre-trained language models, using near-field or far-field external knowledge as inspiration to generate concise and easy-to-understand design ideas[39]. Qihao Zhu et al. continued to explore the application of natural language generation (NLG) technology in concept generation in the early design stage. They proposed a method using the GPT-3 model to transform knowledge in text data into new concepts in the form of natural language [40].

As mentioned above, relevant research ignores the front-end design problems to be solved, that is, the mining and completion of user requirements, and pays too much attention to the concept generation method, leading to repeated iteration of the design process, and the output concept scheme needs to be improved in terms of innovation and feasibility[3–5].

3. Nature of the Problem and Research Framework

In order to clarify the essence of personalized implicit requirements mining, according to the classical theory of engineering design [41], the classification and representation of elements in personalized requirements are defined, and the differences between general user requirements and personalized requirements are compared, so as to further analyze the characteristics and categories of implicit personalized requirements.

3.1. Classification and Characterization of Elements of Personalized Requirements

In engineering design, the expression of personalized requirements often has a certain degree of professionalism, and can clearly describe the core function and structure of the product. This feature provides the feasibility of directly extracting requirement elements from user description. Before information extraction, we refine the requirement elements into two categories: functional elements and structural elements based on axiomatic design theory. Further, according to the behavior theory, we subdivide the functional elements into "function operation" and "function object" to facilitate the subsequent matching and supplement process.

In most descriptions of personalized requirements, users usually describe the specific functions and partial structures of the product. Using natural language processing (NLP) technology, we can identify, classify, and extract the functional and structural requirements from the original text. And the functional elements can be further divided into FA (verb) and FO (noun). For example, in a real user requirement case shown in Figure 1, we successfully identified functional requirements such as "loading and unloading" (FA) and "paper box" (FO), as well as structural requirements such as "six-axis robotic arm."

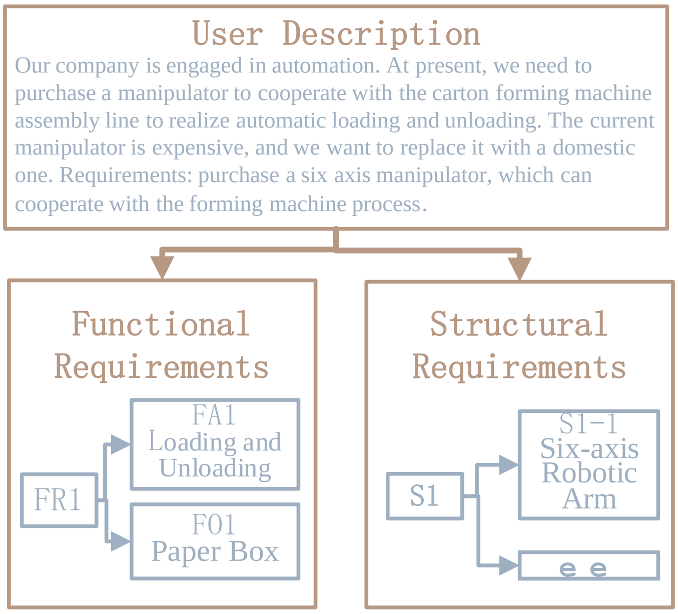

User Description
Our company is engaged in automation. At present, we need to purchase a manipulator to cooperate with the carton forming machine assembly line to realize automatic loading and unloading. The current manipulator is expensive, and we want to replace it with a domestic one. Requirements: purchase a six axis manipulator, which can cooperate with the forming machine process.
Functional Requirements
FR1
FA1 Loading and Unloading
F01 Paper Box
Structural Requirements
S1
S1-1 Six-axis Robotic Arm
e e

Figure 1. Examples of personalized requirements characterization.

3.2. Analysis on the Characteristics of Personalized Implicit Requirements

By comparing the characteristic differences between general user requirements and personalized requirements, the symbolic characteristics of personalized requirements are studied.

It can be seen from Table 1 that the biggest characteristic difference between general user requirements and personalized requirements is that the former is fuzzy and emotional, while the latter is relatively more professional. Therefore, the general user requirements are more vulnerable to the user's incomplete expression, so the implicit requirements are represented by the lack of standard function operation words, typical structure words and standard engineering parameters. However, personalized requirements are easily subject to the subjective limitations of users. The expressed requirements are often limited to specific functions or structural words. Implicit requirements are mostly actions related to the time series of FAs, sub/parent or very similar structures of FOs/structures. The characteristics, typical performance and requirement examples of the above implicit requirement are summarized in Table 2:

Table 1. Comparison of general user requirements and personalized requirements.

	General User Requirements	**Personalized Requirements**
Requirements Sources	User comments from ordinary e-commerce platform	Tasks from crowdsourcing/collaborative community platform
Targeted Products	Mass-produced consumer products	Mechanical, mechanical and electrical equipment, customized products, non-ordinary consumer products
User Characteristics	Ordinary consumers	Field enthusiasts or practitioners
Requirements Characteristics	Dynamics, complexity, concealment, unstructured description, emotion, ambiguity	Dynamics, complexity, concealment, unstructured description, semi-professional
Contents	Fuzzy expression of user requirements	Core functional requirements and even some structural requirements
Examples	I want a mobile phone with a large screen, thin body, clear pictures, especially at night, with not card, to use	Our company is engaged in automation. At present, we need to purchase a manipulator to cooperate with the carton forming machine assembly line to realize automatic loading and

	General User Requirements	Personalized Requirements
	for more than 3 years, to support 2 days without charging.	unloading. The current manipulator is expensive, and we want to replace it with a domestic one. Requirements: purchase a six axis manipulator, which can cooperate with the forming machine process.

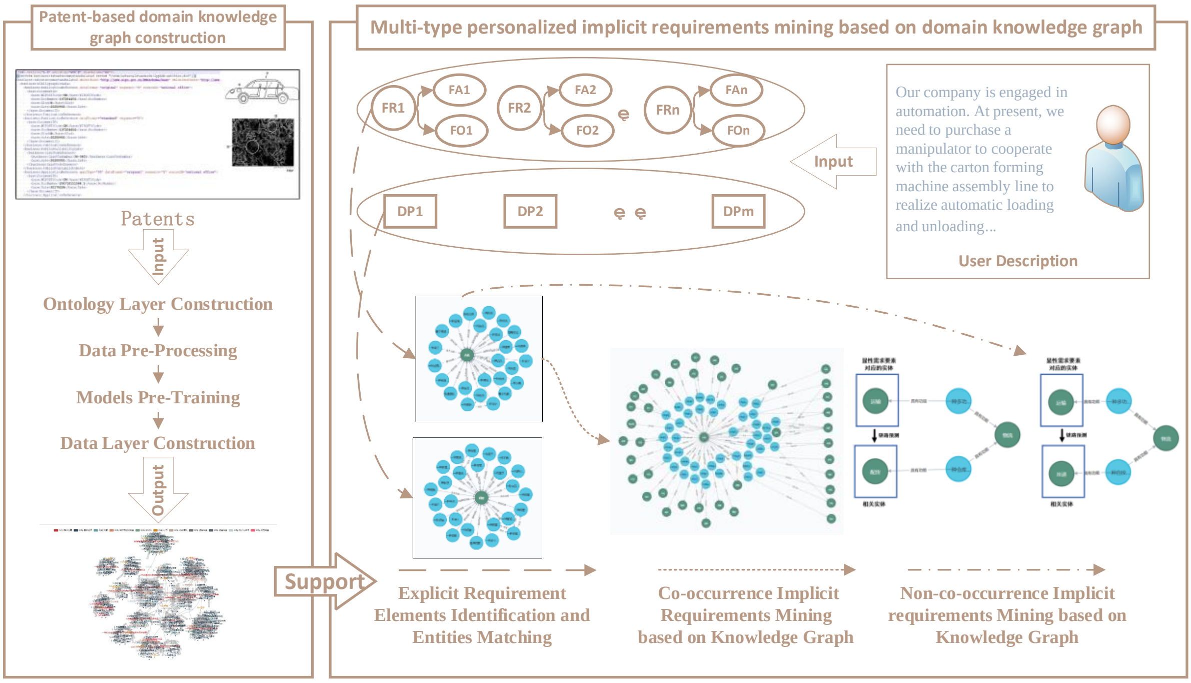
Table 2. The characteristics and typical manifestations of general user implicit requirements and personalized implicit requirements.

	General User Implicit Requirements	Personalized Implicit Requirements
Kano Types	Basic and Expected	Attractive
Characteristics	The incompleteness of the description	Subjective limitations of the users
Typical Manifestations	Standard function operation words, typical structure words and standard engineering parameters	Sub-actions of FAs, pre/post action of FAs, sub/parent or very similar structures of FOs/structures
Example	Smooth operation, HD camera, good durability, large capacity battery, etc	Ingredients, testing, grasping, transportation, sorting, robotic hand, robotic arm, etc

Therefore, the goal of this study is to mine the personalized implicit requirements of sub-actions of FAs, pre/post operations of FAs, sub/parent structures of FOs/structures, pre/post actions of sub-actions of FAs, and very similar structures of FOs/structures. Through the mining of these requirements, a more accurate personalized requirement completion model is constructed to improve the innovation, feasibility and interaction efficiency of generative design or swarm intelligence design.

3.3. Research Framework

For the nature of the problem in this study, combined with the patent knowledge in the field, the research framework of this paper is proposed as following:


Patent-based domain knowledge graph construction
Patents
Input
Ontology Layer Construction
Data Pre-Processing
Models Pre-Training
Data Layer Construction
Output
Support
Multi-type personalized implicit requirements mining based on domain knowledge graph
FR1
FA1
FO1
FR2
FA2
FO2
FRn
FAn
FOn
DP1
DP2
DPm
Input
User Description
Explicit Requirement Elements Identification and Entities Matching
Co-occurrence Implicit Requirements Mining based on Knowledge Graph
Non-co-occurrence Implicit requirements Mining based on Knowledge Graph

Our company is engaged in automation. At present, we need to purchase a manipulator to cooperate with the carton forming machine assembly line to realize automatic loading and unloading...

Figure 2. Research Framework.

doi:10.20944/preprints202407.1040.v1

4. Patent-Based Domain Knowledge Graph Construction Method

The process of constructing a domain knowledge graph based on patent text is illustrated in Figure 3. It consists of four main stages: ontology layer construction, data pre-processing, models pre-training, and data layer construction. In order to enhance the entity recognition and sentence sequence classification performance of the model in the field of patent text, this study conducts incremental pre-training on a large number of domain related patent text data collected, and constructs the domain knowledge pre-training models BERT-base-patent-fusion and BERT-wwm-ext-patent-fusion based on two base models BERT-base and BERT-wwm-ext[42] respectively. The pre-trained models marked with * in the figure are the models proposed in this study. The numbers and letters in brackets in the data layer construction represent the datasets and training model numbers referenced in the construction process.

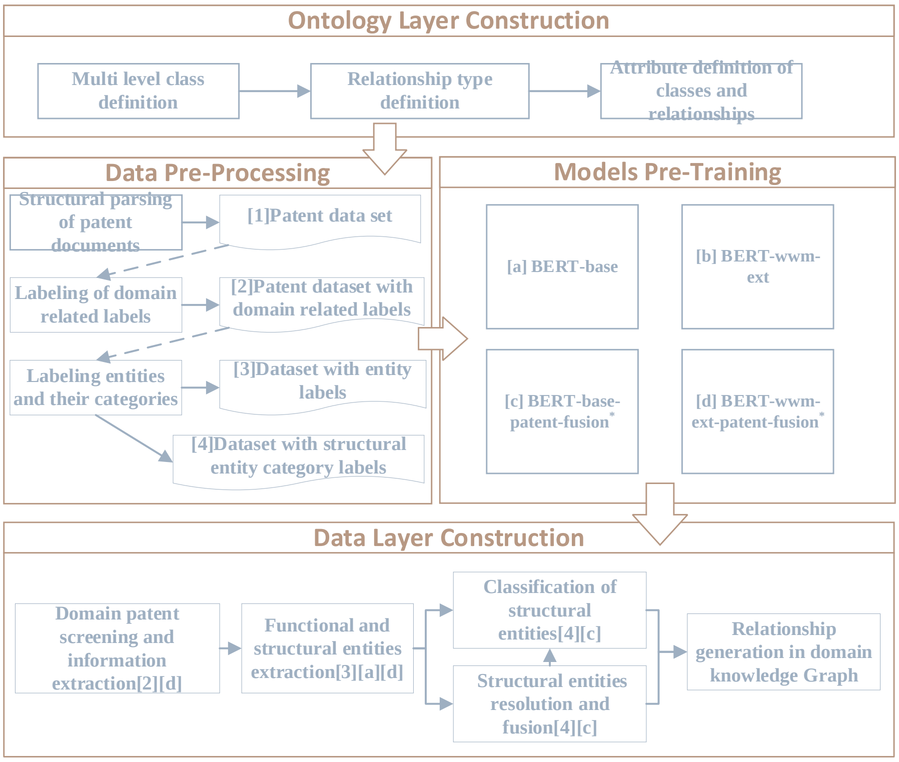

Ontology Layer Construction
Multi level class definition
Relationship type definition
Attribute definition of classes and relationships
Data Pre-Processing
Structural parsing of patent documents
[1]Patent data set
Labeling of domain related labels
[2]Patent dataset with domain related labels
Labeling entities and their categories
[3]Dataset with entity labels
[4]Dataset with structural entity category labels
Models Pre-Training
[a] BERT-base
[b] BERT-wwm-ext
[c] BERT-base-patent-fusion*
[d] BERT-wwm-ext-patent-fusion*
Data Layer Construction
Domain patent screening and information extraction[2][d]
Functional and structural entities extraction[3][a][d]
Classification of structural entities[4][c]
Structural entities resolution and fusion[4][c]
Relationship generation in domain knowledge Graph

Figure 3. Construction process of domain knowledge graph based on patent data.

4.1. Ontology Layer Construction

This paper uses ontology to standardize entities, relationships, and the types and properties of entities. The ontology layer schema of the domain knowledge graph can be expressed as <*Class, Property, Relation*>, where:

Class={*class*$_i$}, which specifies the node classes in the network and the hierarchical relationships between classes. The domain knowledge graph defines three types of entities, namely functional entities, structural entities, and patent entities; a total of 11 node classes are specified, among which: function-action class and function-object class are subclasses of the function class, and power component, transport component, control component, etc. are subclasses of the structure class.

Property={<*class, property_type, value*>} specifies the attributes that each entity in the knowledge graph should have and the range of its property value;

Relation={*class*$_A$*, rel_type, class*$_B$} specifies the relationships between various entities. Relationships can be divided into three categories: patent has functions, patent has structures and related relationships between patents due to similar functions or structures.

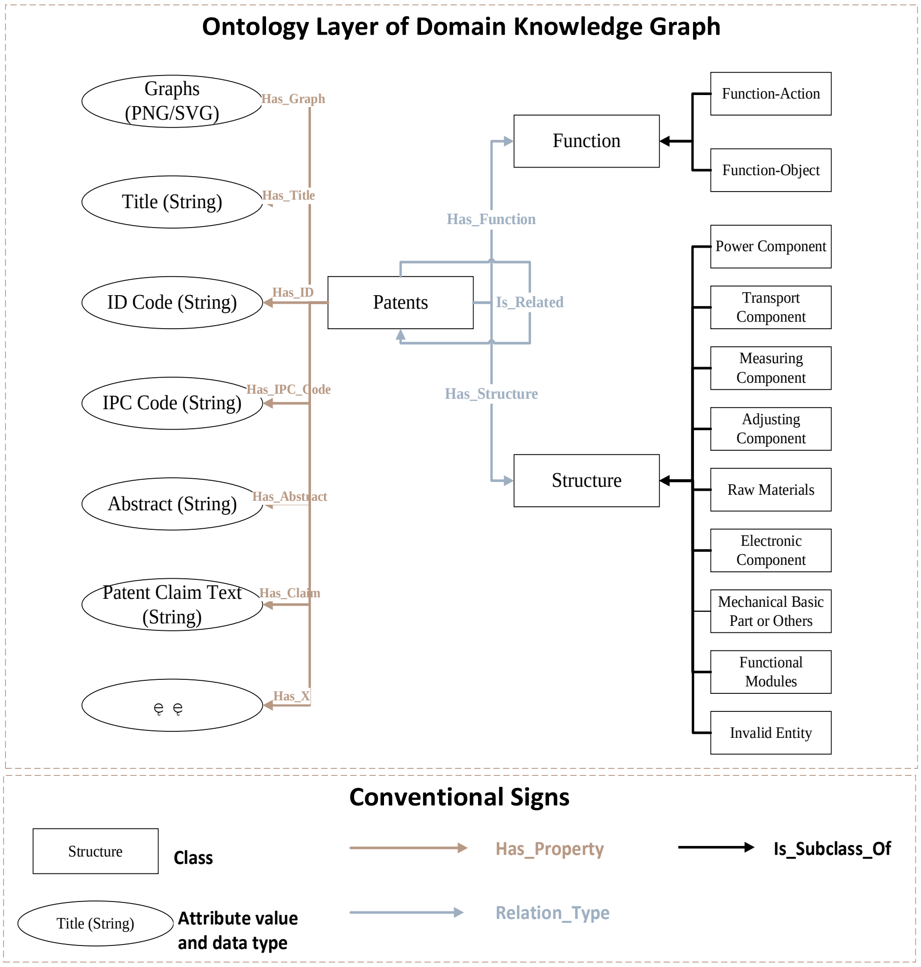


Figure 4. A schematic diagram of the domain knowledge graph ontology layer.

4.2. Data Pre-Processing

This paper focuses on Chinese patents as the research subject, and first of all, it needs to complete the structured parsing and storage of patent data. In addition to the patent data set, for the construction of the pre-training model, three data sets are extracted and labeled, including the patent data set with domain-related labels, data set with function and structural entity labels, and data set with structural entity category labels.

- Patent data set with domain-related labels is an annotation data set used to determine whether a patent is domain-related. The domain-related classification of patents is a sentence classification task. The domain features of the title and rights statement are significant. As the input data of the model, the patent data with domain-related labels is annotated to form the training and verification sets of the model.
- Data set with function and structural entity labels is constructed to fine-tune the named entity recognition model of patent text.The BIO labeling method is used to label the functional and structural entities in the patent title and claim text for the model pre-training of extracting functional and structural entities.
- Data set with structural entity category labels is used to build a classification model for structural entities, subdividing structural entities into subcategories such as power and transportation components.

4.3. Models Pre-Training

Patent text classification, information extraction and entity semantic similarity calculation all need NLP support. At present, the best SOTA (state of the art) model adopts the "pre train+fine tune" framework. Based on this framework, this paper superimposes patent text data for pre training process to enhance the model's ability to recognize and understand patent text, so as to obtain better results in patent text oriented NLP tasks.When Google released the BERT model in 2018, it included a Chinese model trained on the Chinese Wikipedia corpus. In 2019, the BERT-wwm-ext model was proposed by Harbin CuiY et al[43]. On the basis of Bert, the masked language model at the character level was improved to the whole word mask, and the corpus used for pre-training was expanded, which achieved significant optimization results in multiple tasks.

To improve the model's ability to recognize entities and classify sentence sequences in domain patent texts, this study conducted incremental pre-training using a large number of domain-related patent text data with two base models: BERT-base and BERT-wwm-ext. This led to the creation of pre-training models BERT-base-patent-fusion and BERT-wwm-ext-patent-fusion, which superimposed domain knowledge. For fine-tuning in subsequent tasks, it will be trained based on four models: BERT-base, BERT-base-knowl-fusion, BERT-wwm-ext, and BERT-wwm-ext-knowl-fusion.

4.4. Data Layer Construction

The data layer construction of the domain knowledge graph can be divided into five stages: domain patent screening and information extraction, functional and structural entity extraction, classification of structural entities, structural entities resolution and fusion, and relationship generation in the graph.

4.1.1. Domain Patent Screening and Information Extraction

In order to improve the information purity of domain knowledge graph, it is necessary to screen all patents related to the domain in all patent data. Since there are patents related to the domain under all IPC primary partitions, it is not possible to directly screen through the IPC partition information of patents. Therefore, through the filter based on NLP algorithm, this paper selects the domain patents for graph construction.

Based on NLP algorithm and data set with domain-related labels, this paper fine-tunes four pre-training models to construct the domain-related patent filtering model.

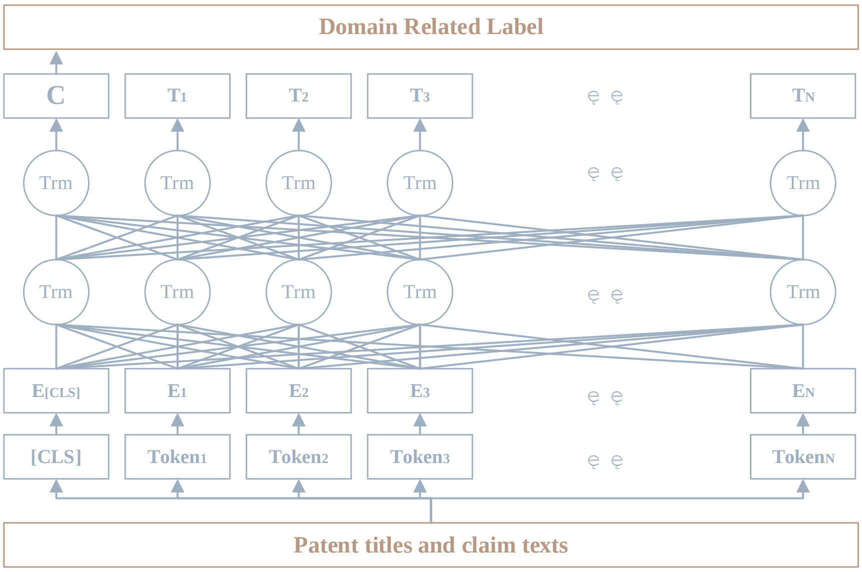


Figure 5. The sentence sequence classification principle based on the BERT-Transformer module.

As shown in Figure 6, in the patent title and claim text, the last hidden layer vector $C \in R^{H}$ output after the information extraction of the Transformers module in BERT[44] is used as the representation vector of the sentence, where H is the number of neurons in the last hidden layer. Finally, adjust the classification layer weight $W \in R^{K \times H}$, where $K=2$ is the number of labels, and the model training can be completed by calculating the classification error based on the Softmax result output by the following formula.

$$P=softmax(CW^T) \tag{1}$$

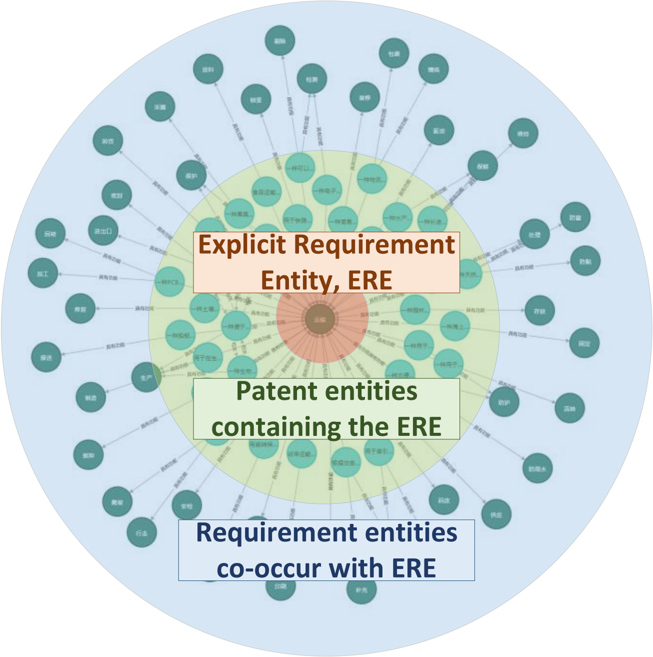

Explicit Requirement Entity, ERE
Patent entities containing the ERE
Requirement entities co-occur with ERE

Figure 6. Implicit requirement elements co-occur with "transport" (outermost entities / nodes).

Import domain-related patents and their attribute information, such as patent numbers, titles, IPC, etc., into the knowledge graph, to complete the construction of patent entities and extract their attribute information.

4.1.2. Functional and Structural Entity Extraction

Extracting functional and structural entities from unstructured patent text is a text sequence annotation problem [45]. Specifically, given a text sequence of length N

$$S=(w_1, w_2, \dots , w_N) \tag{2}$$

w_i represents a single word or token. The named entity recognition model returns a series of tuples $<I_s, I_e, t>$ ($I_s \in [1, N]$, $I_e \in [1, N]$), where I_s and I_e represent the start and end positions of a named entity, t represents the entity category which should be one of the entity categories predefined according to the actual problem model. The function and structure entity extraction model has three types of named entities, namely FA (Function-Action), FO (Function-Object) and S (Structure), so $t \in \{FA,FO,S\}$.

The patent title mainly includes the function of the patent and its overall structure. For example, *"A kind of stone grinding machine"* clearly states that its function is "grinding stone," where the function-action entity is "grinding," the function-object entity is "stone," and its overall structure is "stone grinding machine", which is a structural entity. The claim of the rights mainly contains the technical characteristics of the patent and expresses the scope of protection requested. The structural entity can be obtained using named entity recognition technology from the claim of rights.

The functional and structural entity extraction models based on patent title, and the structural entity extraction models based on patent rights claim are constructed. Based on the data set with patent functional and structural entity labels, we fine-tuned four pre-trained models, selected the best iteration rounds and models, and constructed three entity extraction models.

4.1.3. Classification of Structural Entities

In order to provide more detailed entity information, the structural entities are divided into 8 categories: power component, transport component, measuring component, adjusting component, raw materials, mechanical basic part or other component, electronic components and functional modules. In the process of named entity recognition, it is inevitable to introduce some wrong recognition entities such as non related fields, so invalid entity is also regarded as a category. Similarly, based on the data set with structural entity category label, the structural entity classification model is constructed by fine-tuning and testing the four pre-training models.

Table 3. Structural entity classification information table.

Abbreviation	Meaning	Examples
PC	Power Component	Traveling motor, turbine device
TC	Transport Component	Mud settling pipeline, discharge belt
MC	Measuring Component	Tank pressure monitor, low temperature sensor
AC	Adjusting Component	Pump motor stop button, lock control system host
RM	Raw Materials	Columnar metal, spherical catalyst
BP	Mechanical Basic Part Or Other Component	Screws, gears, boxes
EC	Electronic Component	FPGA logic controller, ripple generation circuit
FM	Functional Module	Disabled scooters, alloy production devices
IE	Invalid Entity	Triples, cells

4.1.4. Structural Entities Resolution and Fusion

Entity Resolution is mainly used to solve the problem of multiple references actually pointing to the same entity. In the domain knowledge graph, since the product described in the patent may have multiple identical structures, they will be distinguished by "Xth structure," "Structure X," etc. Therefore, it is necessary to do Co-referential resolution analysis on the structural entities and fuse the entities that point to the same.

A rule-based method is used for the structural entities resolution: first, regular expressions are used to determine whether the structural entity is expressed in the form of "Xth structure" or "Structure X". If so, the "Xth" prefix or "X" suffix is deleted to determine whether the new entity formed exists. If it exists, the entity is merged with the existing entity; otherwise, a new entity is created.

4.1.5. Relationship Generation in Domain Knowledge Graph

There are three types of relationships in the domain knowledge graph: "Has_Function," "Has_Structure," and "Is_Related". Among them, the "Has" type relationship is between the parent entity and the function and structure entities. In extracting the function and structure entities, this type of relationship has been obtained and can be directly constructed in the graph. The "Is_Related" is constructed based on the network structure similarity between entities. Based on the highly interpretable Jaccard similarity measure, the structural similarity between patent entities with more identical functional and structural entities is higher, and the "Is_Related" is constructed between patent entities above the similarity threshold.

5. Multi-Type Personalized Implicit Requirements Mining Method Based on Knowledge Graph

In view of the different categories of personalized requirements mentioned in Table 2, this paper implements entity-layer co-occurrence implicit requirement mining based on structural similarity, and constructs a co-reference embedding layer to realize non-co-occurrence implicit requirement mining based on link prediction, as summarized in Table 4:

Table 4. Methods of the multi-type personalized implicit requirements mining.

Types of Personalized Implicit Requirements	Relations with the Explicit Requirement Entities	Mining Method
Sub-actions of the FAs (Type Ⅰ)	Co-occurrence mostly	Entity-layer; Structural similarity
Pre/post action of FAs (Type Ⅱ)		
Sub/parent structures of FOs/structures (Type Ⅲ)		
Pre/post actions of sub-actions of FAs (Type Ⅳ)	Non-co-occurrence mostly	Co-reference embedding layer; Link prediction
Similar structures of FOs/structures (Type Ⅴ)	Non-co-occurrence	

Therefore, the process of mining personalized implicit requirements based on a domain knowledge graph mainly includes three stages: explicit requirement element identification and entity matching, co-occurrence implicit requirements mining of the entity layer, and non-co-occurrence implicit requirements mining of the co-reference embedding layer. As shown in the following formula, for the expression of explicit requirement elements in the domain knowledge graph, that is, each element in the knowledge entity set $KE = \{ke_1^{FA}, \dots, ke_{n_1}^{FA}, ke_1^{FO}, \dots, ke_{n_2}^{FO}, ke_1^{S}, \dots, ke_{n_3}^{S}\}$: If there is a knowledge entity of co-occurrence relationship, the structural similarity algorithm is used to measure its relativity with ; otherwise, its relativity with will be measured based on the entity embedding and link prediction algorithm. Based on the correlation $R(ke, e_i)$, the two types of entities are sorted to realize the personalized implicit requirements mining.

$$R(ke, e_i) = \begin{cases} GraphSim_{Jaccard}(ke, e_i) & \text{Entity layer} \\ GraphSim_{Node2Vec}(ke, e_i) & \text{Co} - \text{reference embedding layer} \end{cases} \tag{3}$$

5.1. Explicit Requirement Element Identification and Entity Matching

The model fine-tuned based on the four pre-training models, is used as the identification model for personalized explicit requirement elements, completing the mapping of unstructured RD (Requirement Description) to ERE (Explicit Requirement Entity). As shown in the following equation, where $ere_i^{FA}, ere_i^{FO}, ere_i^{S}$ represent the explicit FA, FO, and structural elements expressed by the user.

$$ND \rightarrow ERE = \{ere_1^{FA}, \dots, ere_{p_1}^{FA}, ere_1^{FO}, \dots, ere_{p_2}^{FO}, ere_1^{S}, \dots, ere_{p_3}^{S}\}$$
$$(p_1 + p_2 + p_3 > 0) \tag{4}$$

The matching process between explicit requirement elements and entities in the knowledge graph is completed based on the cosine similarity calculation of word embedding vectors.Each explicit requirement entity is input into the BERT model to obtain its semantic vector representation $V_{ere} \in \mathbb{R}^H$. Similarly, the semantic vector representation of functional and structural entities in the domain knowledge graph is obtained, that is, the matrix $M \in \mathbb{R}^{N \times H}$, where N is the number of functional and structural entities in the network. Then, based on the cosine similarity, the cosine similarity measures of and M are obtained to complete the matching of demand elements and knowledge entities, as follows:

$$SemSim(V_{ere}, M)_i = \frac{V_{ere} \cdot M_i}{|V_{ere}||M_i|} \quad \forall i \in [1, N], i \in \mathbb{N}^* \tag{5}$$

Based on the matching process between requirement elements and knowledge entities, the mapping process from the explicit requirement element set ERE to the entity KE set (Knowledge Entities) in the knowledge graph is completed, as shown in the following formula:

$$ERE = \{ere_1^{FA}, \dots, ere_{p_1}^{FA}, ere_1^{FO}, \dots, ere_{p_2}^{FO}, ere_1^{S}, \dots, ere_{p_3}^{S}\} \rightarrow KE \tag{6}$$

$$= \{ke_1^{FA}, \ldots, ke_{n_1}^{FA}, ke_1^{FO}, \ldots, ke_{n_2}^{FO}, ke_1^{S}, \ldots, ke_{n_3}^{S}\}$$

$$(n_1 + n_2 + n_3 \geq p_1 + p_2 + p_3)$$

5.2. Co-Occurrence Implicit Requirements Mining of the Entity Layer

For implicit requirement elements that co-occur with explicit requirement, the structural similarity algorithm can be used to mine them. With explicit requirement entities as input, output possible implicit requirement entities and their ranking as shown in Figure 6:

The mining of co-occurrence related entities with the same neighbors can be completed using the node similarity calculation method based on network theory. The node structure similarity calculation algorithms based on network theory mainly include Jaccard similarity, Sorensen-Dice similarity, Hub Promoted similarity, and Hub Depressed similarity, etc., which are all based on the neighbor nodes set calculation. This paper uses Jaccard similarity to calculate node similarity.

In the network *G=(V, E)*, *V* is the vertices set, and *E* is the edges set. For a vertex *x∈V*, *N(x)* represents the set of its neighbors. Then, for *x, y∈V*, the Jaccard similarity is:

$$GraphSim_{Jaccard}(x, y) = J(x, y) = \frac{|N(x) \cap N(y)|}{|N(x) \cup N(y)|} \tag{7}$$

Sort the output of the above formula, mine and recommend the co-occurrence layer implicit requirement entities.

5.3. Non-Co-Occurrence Implicit Requirements Mining of the Co-Reference Embedding Layer

Some personalized implicit requirements do not have a co-occurrence relationship with the explicit requirement elements in the graph. This study uses network node embedding and link prediction algorithms to mine these implicit requirements. Also take the explicit requirement entities as the input, and output the possible non-co-occurrence implicit requirement entities and their ranking. Taking the requirement "transportation" as an example, a kind of non-co-occurrence implicit requirements such as "Preparation according to order" and "delivery" are shown in Figure 7 (in addition to the "secondary co-occurrence" implicit requirements in the figure, our method can also mine the entities with other types of network structural similarities):

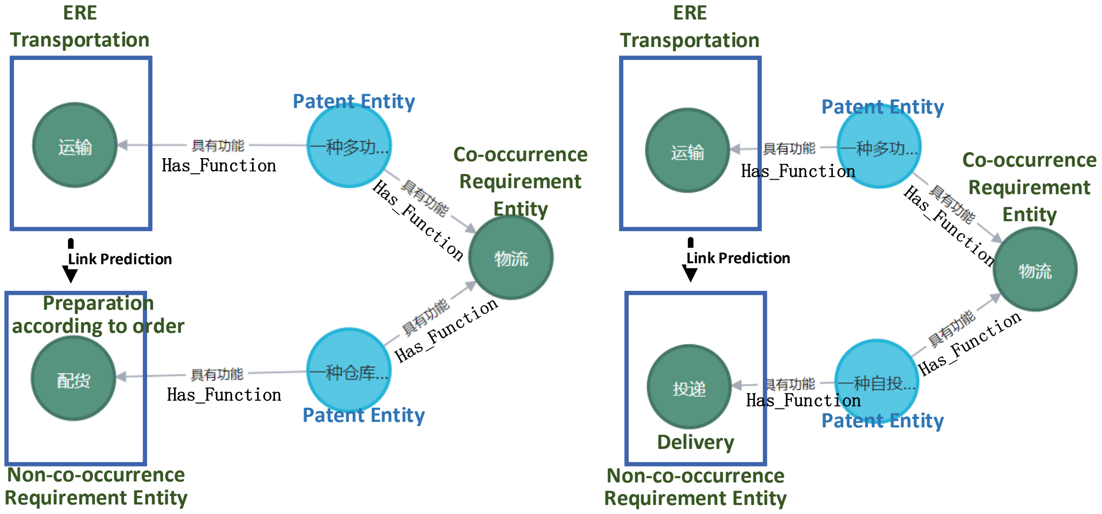


Figure 7. A kind of implicit requirement element that has a relevance but no co-occurrence relationship with "transport".

Link prediction is an algorithm that predicts the possibility of a new connection between two nodes in a given network. With the continuous advancement of complex network modeling technology, researchers have proposed many link prediction algorithms for different types of networks, including: simple similarity based, probability theory and maximum likelihood based,

doi:10.20944/preprints202407.1040.v1

dimension reduction based, and other methods. According to the research of Kumar A et al.[46], based on the AUPR (Area Under the Precision-Recall curve) , it was found that the network embedding algorithm Node2Vec based on the dimension reduction had the best performance in the test data set. Therefore, this paper uses the Node2Vec algorithm to complete the link prediction task of the knowledge graph to assist in the discovery of non-co-occurring related entities and support the mining of implicit requirements.The process of implicit requirements mining based on the Node2Vec algorithm includes the following three steps:

5.3.1. Build Co-Occurrence Networks of Functional and Structural Entities

If the sequence is collected directly based on the knowledge graph, it can only start from the patent entities, and the sequence length just can be 2, such as $(P_1, F_1), (P_2, S_1)$. Therefore, we construct an entity co-occurrence network, which can generate a more extended node sequence, such as $(F_1, F_2, F_3), (S_1, S_3, S_2)$, which is conducive to capturing the relationship between entities. Based on the co-occurrence of functional and structural entities, the two co-occurrence network are constructed respectively. The network nodes are still functional and structural entities in the knowledge graph, and the edge weight is the number of co-occurrences. This process is shown in Figure 8:

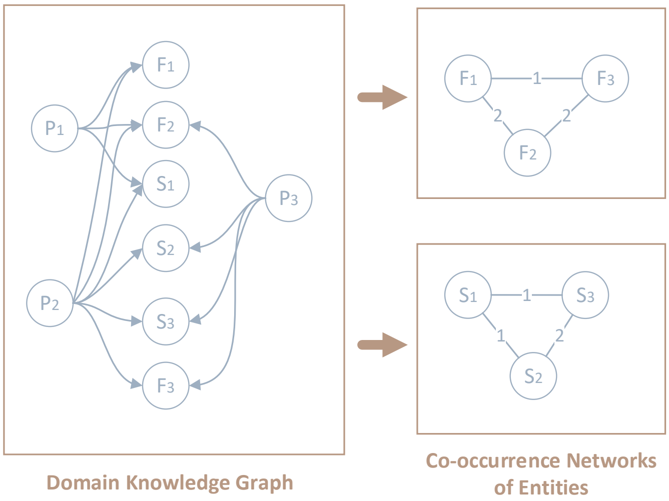


Figure 8. Generation of co-occurrence networks based on the domain knowledge graph.

5.3.2. Construct Node Sequences by Biased Random Walk

As shown in Figure 9, the Node2Vec algorithm draws on the method of generating word vectors. It first starts from any point in the network based on a biased random walk and transfers between connected nodes. In this way, a limited nodes sequence can be collected, and the SkipGram model [47] is used to train the embedding vector E based on these sequences.

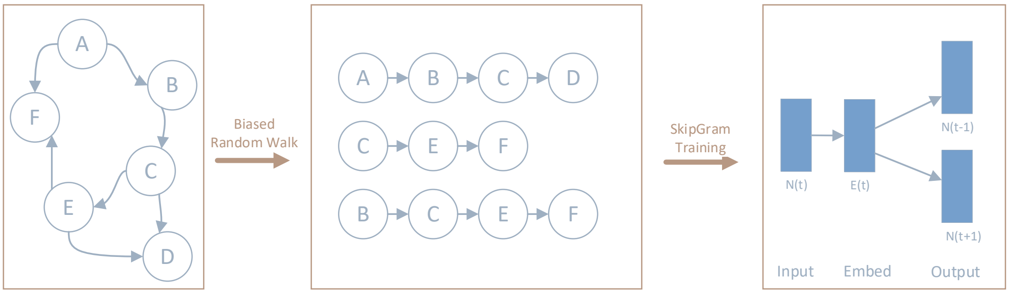


Figure 9. Node2Vec algorithm process.

The "Biased Random Walk" method proposed by Node2Vec uses two parameters to adjust the random walk process. As shown in Figure 10, nodes and belong to the same node community, showing the homogeneity of nodes; while and belong to two different communities, but have the same structure (or role) in these two communities, showing the structural equivalence of nodes. In order to adjust the weights of these two structural characteristics in the algorithm, Node2Vec generalizes DeepWalk[48]. By setting parameters, the bias of the breadth-first sampling strategy and the depth-first sampling strategy in the node sequence generation process can be selected to support the adjustment of the homogeneity and structural equivalence information reflected in the node embedding vector.

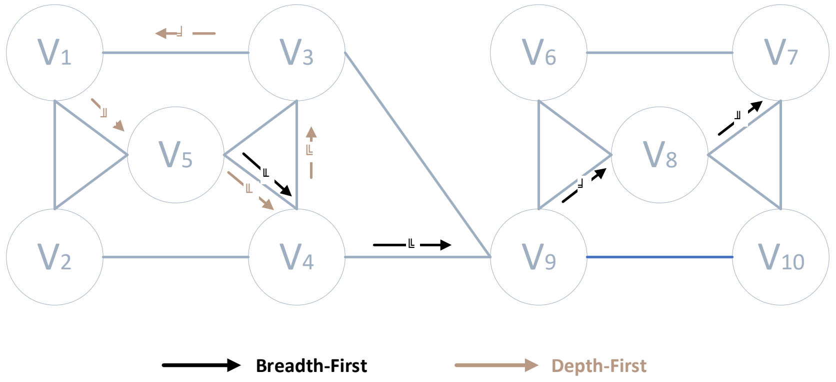


Figure 10. Walk strategies of breadth-first and depth-first.

Specifically, Node2Vec defines a random walk with parameters. Suppose a random walk has just walked through the edge (step ① of Figure 10), and the current position is at node v_4. Next, the random walk needs to make the next edge by evaluating the transition probability π_{v_4x}. Let the transition probability be

$$\pi_{v_4x} = \alpha_{pq}(v_4, x) \cdot \omega_{v_4x} \tag{8}$$

$$\alpha_{pq}(v_4, x) = \begin{cases} \frac{1}{p} & if\ d_{v_4x} = 0 \\ 1 & if\ d_{v_4x} = 1 \\ \frac{1}{q} & if\ d_{v_4x} = 2 \end{cases} \tag{9}$$

Where is the edge weight, and represents the shortest distance between and x. The parameters and can control the strategy of random walk exploration, where controls the possibility of a walk returning to the node previously reached. Setting it to a large value makes the walk more breadth first, and smaller encourages the depth first strategy.

5.3.3. Train the SkipGram Model to Obtain the Entity Embedding Vector, and Sort the Entities by Relevance

Based on the collected nodes sequences set, a node is regarded as a word, and a node sequence is regarded as a sentence. Based on the SkipGram method of the Word2Vec model, each node is represented as a low-dimensional dense vector E_i.

$$GraphSim_{Node2Vec}(x, y) = \frac{E_x \cdot E_y}{\|E_x\|\|E_y\|} \tag{10}$$

When mining non-co-occurring implicit requirement, the formula can be used to calculate and sort the relevance of all entities, giving possible implicit requirement entities.

6. Platform Development, Case Study and Discussion

6.1. Platform Development

Based on the domain knowledge graph construction and personalized implicit requirement mining technology mentioned above, a personalized implicit requirement mining platform is designed to provide knowledge entity retrieval, query, matching, and personalized requirement identification and completion functions.

6.1.1. Platform Development and Operation Environment

The development and operation software environment and functional framework are shown in the figure. The programming language is Python 3.6. The structured patent data is stored in the MySQL database, while the knowledge network uses the unstructured database Neo4j for storage. The Web backend development is mainly based on the Flask framework. The presentation layer uses the Bootstrap framework and JavaScript to provide interactive operation.

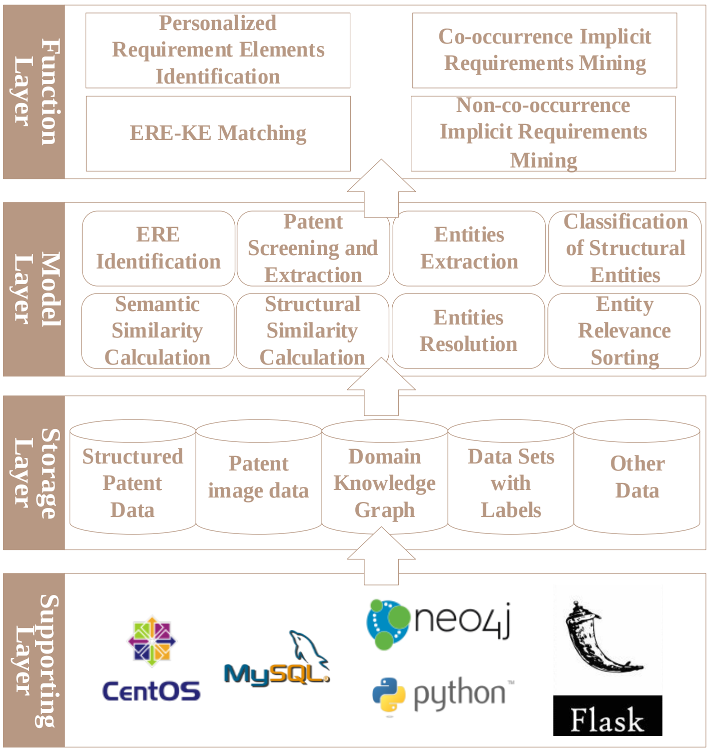

Function Layer
Personalized Requirement Elements Identification
Co-occurrence Implicit Requirements Mining
ERE-KE Matching
Non-co-occurrence Implicit Requirements Mining
Model Layer
ERE Identification
Patent Screening and Extraction
Entities Extraction
Classification of Structural Entities
Semantic Similarity Calculation
Structural Similarity Calculation
Entities Resolution
Entity Relevance Sorting
Storage Layer
Structured Patent Data
Patent image data
Domain Knowledge Graph
Data Sets with Labels
Other Data
Supporting Layer
CentOS
MySQL
neo4j
python
Flask

Figure 11. The software environment and functional framework of personalized implicit requirement mining platform.

6.1.2. Construction of the Knowledge Graph in the Electromechanical Domain

The data is sourced from the experimental system of the State Intellectual Property Office of China, and it consists of XML format files containing patent text information and JPEG or TIFF format files containing patent image information. The XML files are parsed into structured tables based on the format and stored in the database along with the path of the corresponding JPEG or TIFF files. Based on the technical process of constructing the domain knowledge graph, this study acquired, parsed and stored 212,100 pieces of structured data of Chinese patents, annotated 100 domain-related tags and 3,000 entities, including 1,000 structural entities, forming a pre-training data set of 62.01 million words.

The study selected sentences with a length of 100 to 500 characters separated by Chinese periods to create a pre-training data set. Using unlabeled patent texts, characters were masked with a 15% probability of performing the Masked Language Model training task, completing the pre-training of the model with domain-specific knowledge. The classification of structural entities in the labeled data set is shown in Table 5:

Table 5. Structural entities classification information in the domain knowledge graph.

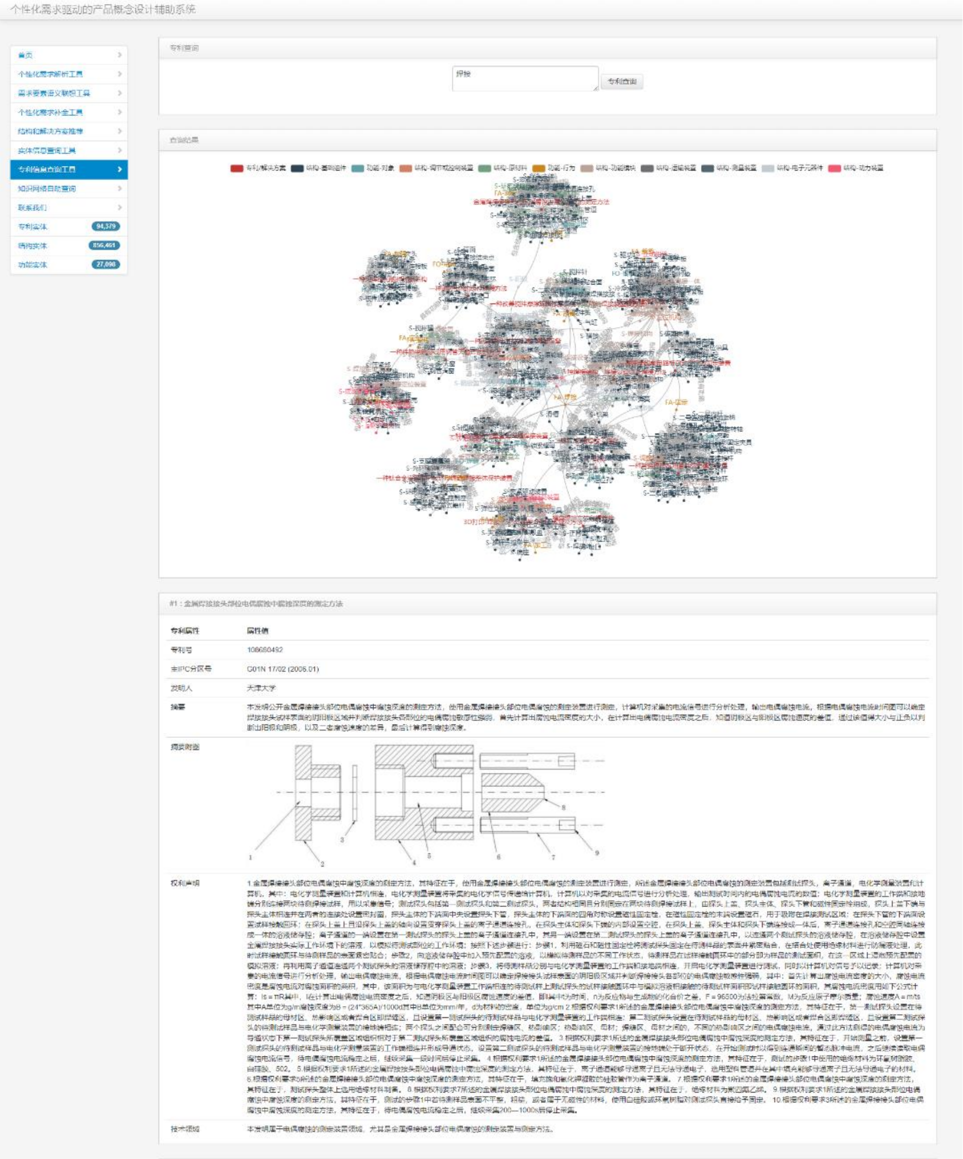
Labels	PC	TC	MC	AC	RM	BP	EC	FM	IE
Proportion	3.10%	7.00%	1.70%	3.00%	3.40%	46.30%	12.50%	6.80%	16.20%

The training information of the four models is shown in Table 6. The * mark represents the model proposed in this paper.

Table 6. Structural entities classification information in the domain knowledge graph.

Models	BERT-base	BERT-base-patent-fusion*	BERT-wwm-ext	BERT-wwm-ext-patent-fusion*
Mask	Character	Character	Whole Word	Character
Data Source	Chinese Wiki	Patent Data	Chinese Wiki and extension	Patent Data
Number of Words	40 Million	62.01 Million	5 Billion	62.01 Million
Initialization Mode	Random	BERT-base	BERT-base	BERT-wwm-ext

Finally, a knowledge graph of electromechanical domain was constructed, which included 94,379 patent entities, 27,098 functional entities, and 856,461 structural entities. A partial screenshot of the graph is shown in Figure 12:



#1：金属焊接接头部位电偶腐蚀中腐蚀深度的测定方法

专利属性	属性值
专利号	108680492
主IPC分区号	G01N 17/02 (2006.01)
发明人	天津大学
摘要	本发明公开金属焊接接头部位电偶腐蚀中腐蚀深度的测定方法，使用金属焊接接头部位电偶腐蚀的测定装置进行测定，计算机对采集的电流信号进行分析处理，输出电偶腐蚀电流，根据电偶腐蚀电流时间图可以确定焊接接头试样表面的阴阳极区域并判断焊接接头各部位的电偶腐蚀敏感性强弱。首先计算出腐蚀电流密度的大小，在计算出电偶腐蚀电流密度之后，知道阴极区与阳极区腐蚀速度的差值，通过该值得大小与正负以判断出阳极和阴极，以及二者腐蚀速度的差异，最后计算得到腐蚀深度。
摘要附图	
权利声明	1.金属焊接接头部位电偶腐蚀中腐蚀深度的测定方法，其特征在于，使用金属焊接接头部位电偶腐蚀的测定装置进行测定，所述金属焊接接头部位电偶腐蚀的测定装置包括测试探头，离子通道，电化学测量装置和计算机，其中：电化学测量装置和计算机相连，电化学测量装置将采集的电化学信号传递给计算机，计算机以对采集的电流信号进行分析处理，输出测试时间内的电偶腐蚀电流的数值；电化学测量装置的工作端和接地端分别连接两块待测焊接试样，用以采集信号；测试探头包括第一测试探头和第二测试探头，两者结构相同且分别固定在两块待测焊接试样上，由探头上盖、探头主体、探头下管和磁性固定栓组成，探头上盖下端与探头主体相连并在两者的连接处设置密封圈，探头主体的下端面中央设置探头下管，探头主体的下端面的四角对称设置磁性固定栓，在磁性固定栓的末端设置磁石，用于吸附在焊接测试区域；在探头下管的下端面设置试样接触圆环；在探头上盖上且沿探头上盖的轴向设置贯穿探头上盖的离子通道连接孔，在探头主体和探头下管的内部设置空腔，在探头上盖、探头主体和探头下管连接成一体后，离子通道连接孔和空腔同轴连接成一体的溶液储存腔；离子通道的一端设置在第一测试探头的探头上盖的离子通道连接孔中，其另一端设置在第二测试探头的探头上盖的离子通道连接孔中，以连通两个测试探头的溶液储存腔，在溶液储存腔中设置金属焊接接头实际工作环境下的溶液，以模拟待测试部位的工作环境；按照下述步骤进行：步骤1，利用磁石和磁性固定栓将测试探头固定在待测样品的表面并紧密贴合，在结合处使用绝缘材料进行防漏液处理，此时试样接触圆环与待测样品的表面紧密贴合；步骤2，向溶液储存腔中加入预先配置的溶液，以模拟待测样品的不同工作状态，待测样品在试样接触圆环中的部分即为样品的测试面积，在这一区域上添加预先配置的模拟溶液；再利用离子通道连通两个测试探头的溶液储存腔中的溶液；步骤3，将待测样品分别与电化学测量装置的工作端和接地端相连，开启电化学测量装置进行测试，同时以计算机对信号予以记录；计算机对采集的电流信号进行分析处理，输出电偶腐蚀电流，根据电偶腐蚀电流时间图可以确定焊接接头试样表面的阴阳极区域并判断焊接接头各部位的电偶腐蚀敏感性强弱，其中：首先计算出腐蚀电流密度的大小，腐蚀电流密度是腐蚀电流对腐蚀面积的商积，其中，该面积为与电化学测量装置工作端相连的待测试样上测试探头的试样接触圆环中与模拟溶液相接触的待测试样面积即试样接触圆环的面积，其腐蚀电流密度用如下公式计算：Is = πR其中，I在计算出电偶腐蚀电流密度之后，知道阴极区与阳极区腐蚀速度的差值，即I其中t为时间，n为反应物与生成物的化合价之差，F = 96500为法拉第常数，M为反应原子摩尔质量；腐蚀速度A = m/ts其中A单位为g/m腐蚀深度为B = (24*365A)/1000d其中B单位为mm/年，d为材料的密度，单位为g/cm 2.根据权利要求1所述的金属焊接接头部位电偶腐蚀中腐蚀深度的测定方法，其特征在于，第一测试探头设置在待测试样品的母材区、热影响区或者焊合区即焊缝区，且设置第一测试探头的待测试样品与电化学测量装置的工作端相连；第二测试探头设置在待测试样品的母材区、热影响区或者焊合区即焊缝区，且设置第二测试探头的待测试样品与电化学测量装置的接地端相连；两个探头之间配合可分别测定焊缝区、热影响区；热影响区、母材；焊缝区、母材之间的，不同的热影响区之间的电偶腐蚀电流，通过此方法测得的电偶腐蚀电流为导通状态下第一测试探头所覆盖区域组织相对于第二测试探头所覆盖区域组织的腐蚀电流的差值。 3.根据权利要求1所述的金属焊接接头部位电偶腐蚀中腐蚀深度的测定方法，其特征在于，开始测量之前，设置第一测试探头的待测试样品与电化学测量装置的工作端相连并形成导通状态，设置第二测试探头的待测试样品与电化学测量装置的接地端处于断开状态，在开始测试时以得到连通瞬间的暂态脉冲电流，之后继续读取电偶腐蚀电流信号，待电偶腐蚀电流稳定之后，继续采集一段时间后停止采集。 4.根据权利要求1所述的金属焊接接头部位电偶腐蚀中腐蚀深度的测定方法，其特征在于，测试的步骤1中使用的绝缘材料为环氧树脂胶、白硅胶、502。 5.根据权利要求1所述的金属焊接接头部位电偶腐蚀中腐蚀深度的测定方法，其特征在于，离子通道能够导通离子且无法导通电子，选用塑料管道并在其中填充能够导通离子且无法导通电子的材料。 6.根据权利要求5所述的金属焊接接头部位电偶腐蚀中腐蚀深度的测定方法，其特征在于，填充饱和氯化钾凝胶的硅胶管作为离子通道。 7.根据权利要求1所述的金属焊接接头部位电偶腐蚀中腐蚀深度的测定方法，其特征在于，测试探头整体上选用绝缘材料制备。 8.根据权利要求7所述的金属焊接接头部位电偶腐蚀中腐蚀深度的测定方法，其特征在于，绝缘材料为聚四氟乙烯。 9.根据权利要求1所述的金属焊接接头部位电偶腐蚀中腐蚀深度的测定方法，其特征在于，测试的步骤1中若待测样品表面不平整，粗糙，或者属于无磁性的材料，使用白硅胶或环氧树脂对测试探头直接给予固定。 10.根据权利要求3所述的金属焊接接头部位电偶腐蚀中腐蚀深度的测定方法，其特征在于，待电偶腐蚀电流稳定之后，继续采集200—1000s后停止采集。
技术领域	本发明属于电偶腐蚀的测定装置领域，尤其是金属焊接接头部位电偶腐蚀的测定装置与测定方法。

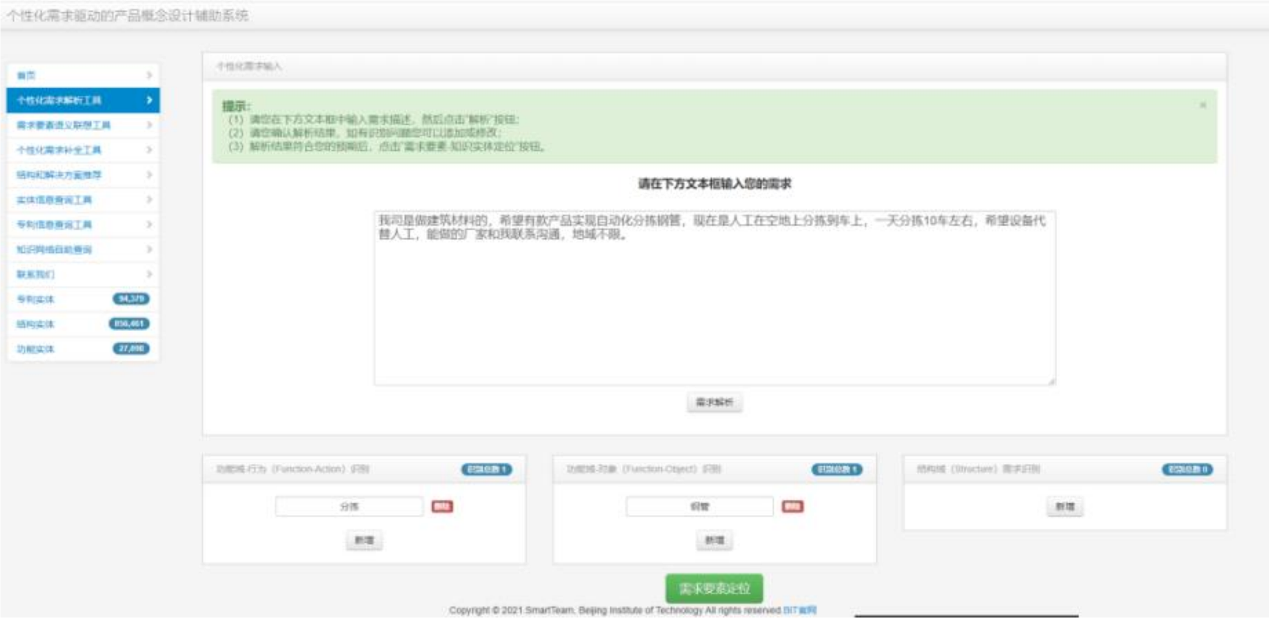
Figure 12. Screenshots of some domain knowledge graph and details of patent entities.

6.1.3. Function Modules of the Platform

Functional modules of the platform include three parts: requirement elements identification, requirement elements - knowledge entities matching, and implicit requirement mining.

- **Requirement elements Identification**

The requirement element identification interface is shown in the figure. Users can directly enter personalized requirements in the text box and then click the "Requirement Analysis" button to obtain the domain identification of requirement elements. Users can add and delete them.



个性化需求驱动的产品概念设计辅助系统

个性化需求输入

提示：
（1）请您在下方文本框中输入需求描述，然后点击"解析"按钮；
（2）请您确认解析结果，如有识别问题您可以添加或修改；
（3）解析结果符合您的预期后，点击"需求要素-知识实体定位"按钮。

请在下方文本框输入您的需求

我司是做建筑材料的，希望有款产品实现自动化分拣钢管，现在是人工在空地上分拣到车上，一天分拣10车左右，希望设备代替人工，能做的厂家和我联系沟通，地域不限。

功能域-行为（Function-Action）识别 已识别数 1

分拣

功能域-对象（Function-Object）识别 已识别数 1

钢管

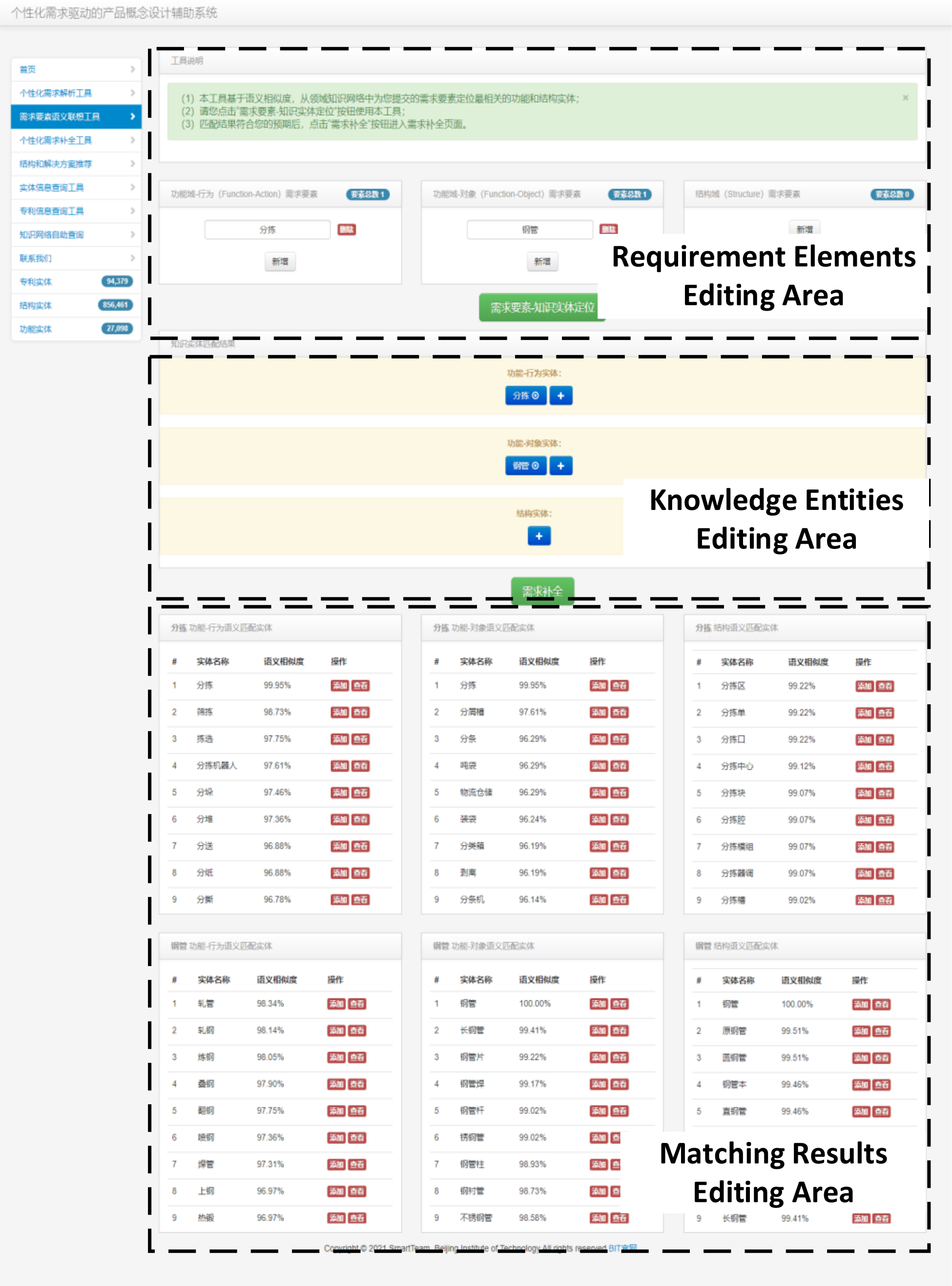
结构域（Structure）需求识别 已识别数 0

Copyright © 2021 SmartTeam. Beijing Institute of Technology All rights reserved BIT官网

Figure 13. Requirement elements identification interface.

- **Requirement elements - knowledge entities matching**

The requirement elements - knowledge entity matching page is shown in the figure. Users can adjust and update the requirement elements, add and delete matched knowledge entities. The entity-matching results are presented in three parts: "FAs", "FOs", and "Structures". Users can select a knowledge entity by clicking the "Add" button or the "View" button to view the entity's information.



（1）本工具基于语义相似度，从领域知识网络中为您提交的需求要素定位最相关的功能和结构实体；
（2）请您点击"需求要素-知识实体定位"按钮使用本工具；
（3）匹配结果符合您的预期后，点击"需求补全"按钮进入需求补全页面。

分拣 功能-行为语义匹配实体

#	实体名称	语义相似度	操作
1	分拣	99.95%	添加 查看
2	筛拣	98.73%	添加 查看
3	拣选	97.75%	添加 查看
4	分拣机器人	97.61%	添加 查看
5	分垛	97.46%	添加 查看
6	分堆	97.36%	添加 查看
7	分送	96.88%	添加 查看
8	分纸	96.88%	添加 查看
9	分断	96.78%	添加 查看

分拣 功能-对象语义匹配实体

#	实体名称	语义相似度	操作
1	分拣	99.95%	添加 查看
2	分隔槽	97.61%	添加 查看
3	分条	96.29%	添加 查看
4	吨袋	96.29%	添加 查看
5	物流仓储	96.29%	添加 查看
6	装袋	96.24%	添加 查看
7	分类箱	96.19%	添加 查看
8	剥离	96.19%	添加 查看
9	分条机	96.14%	添加 查看

分拣 结构语义匹配实体

#	实体名称	语义相似度	操作
1	分拣区	99.22%	添加 查看
2	分拣单	99.22%	添加 查看
3	分拣口	99.22%	添加 查看
4	分拣中心	99.12%	添加 查看
5	分拣块	99.07%	添加 查看
6	分拣腔	99.07%	添加 查看
7	分拣模组	99.07%	添加 查看
8	分拣器调	99.07%	添加 查看
9	分拣槽	99.02%	添加 查看

钢管 功能-行为语义匹配实体

#	实体名称	语义相似度	操作
1	轧管	98.34%	添加 查看
2	轧钢	98.14%	添加 查看
3	炼钢	98.05%	添加 查看
4	叠钢	97.90%	添加 查看
5	翻钢	97.75%	添加 查看
6	喷钢	97.36%	添加 查看
7	焊管	97.31%	添加 查看
8	上钢	96.97%	添加 查看
9	热锻	96.97%	添加 查看

钢管 功能-对象语义匹配实体

#	实体名称	语义相似度	操作
1	钢管	100.00%	添加 查看
2	长钢管	99.41%	添加 查看
3	钢管片	99.22%	添加 查看
4	钢管焊	99.17%	添加 查看
5	钢管杆	99.02%	添加 查看
6	锈钢管	99.02%	添加 查
7	钢管柱	98.93%	添加 查
8	钢衬管	98.73%	添加 查
9	不锈钢管	98.58%	添加 查看

钢管 结构语义匹配实体

#	实体名称	语义相似度	操作
1	钢管	100.00%	添加 查看
2	原钢管	99.51%	添加 查看
3	圆钢管	99.51%	添加 查看
4	钢管本	99.46%	添加 查看
5	直钢管	99.46%	添加 查看
9	长钢管	99.41%	添加 查看

Copyright © 2021 SmartTeam, Beijing Institute of Technology All rights reserved BIT官网

Figure 13. Requirement elements - knowledge entities matching interface.

- **Implicit requirement mining**

Based on the mining methods proposed in this paper, the platform lists all the implicit requirements that have been mined, and users can click the "Add" button to select. The interface for implicit requirement mining will be presented in the Case Study.

6.2. Case Study

In order to verify the feasibility of the proposed personalized implicit requirement mining method, this paper takes the personalized requirement of mechanical automation equipment on a crowdsourcing platform as an example. The original requirement description proposed by the user is:

"My company is engaged in building materials. I hope there will be a product that can realize the automatic sorting of steel pipes. Now, we sort them manually on the open ground and put them on the truck.

We sort about ten trucks a day. I hope that the equipment can replace the manual work. Manufacturers who can do it can contact me to communicate. There is no limit on the region."

6.2.1. Explicit Requirement Elements Identification

Take the result of the explicit requirement elements identified by the algorithm as: FA element "sorting" and FO element "steel pipe" as examples.

6.2.2. Requirement Elements - Knowledge Entities Matching

According to the results returned by semantic similarity, the "sorting" and "steel pipe" entities already exist in the knowledge graph, and the matching is completed directly.

6.2.3. Implicit Requirement Mining

The co-occurrence and non-co-occurrence entities of "sorting" and "steel pipe" are calculated. In the link prediction process, since the mining of implicit requirements focuses on discovering node homogeneity, the selection of parameters p and q should focus on depth priority, so *p=0.25* and *q=4* are selected as the parameters for controlling biased random walk.

Table 7. Node2Vec parameter setting.

Parameters	Parameter Meaning	Values
dim	The dimensions number of generated representation vector	128
number-walks	Number of random walks at the beginning of each node	10
walk-length	Step size of each random walk at the beginning of each node	30
workers	Number of parallel operations of algorithm	10
window-size	Window size of SkipGram model	10
p	Control parameters of biased random walk	0.25
q		4

The elements identification, entities matching and requirements mining results are shown in Figure 14 and Table 8:

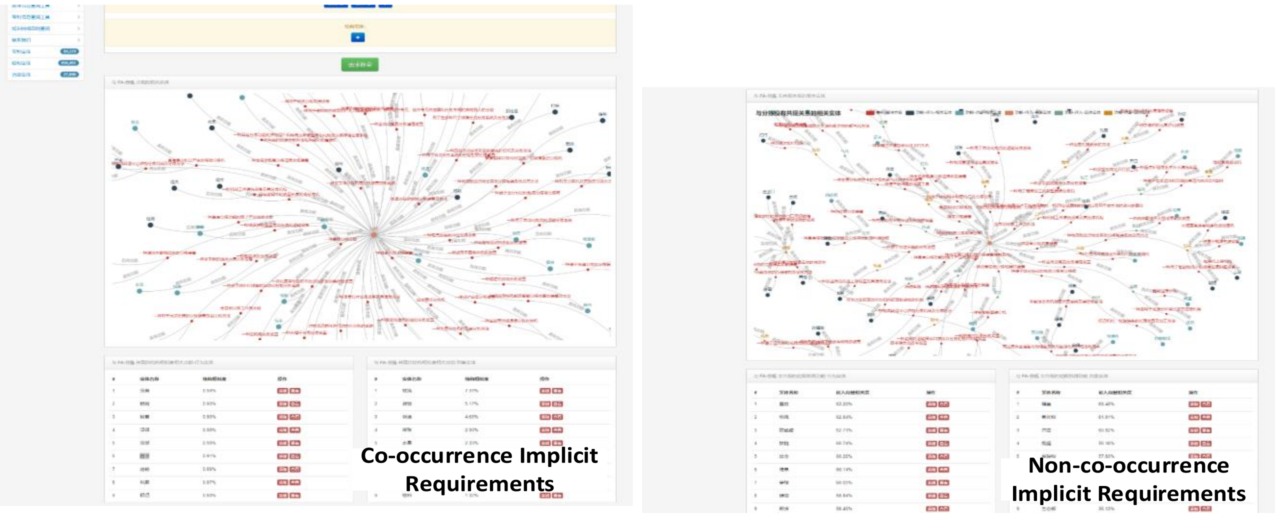

Co-occurrence Implicit Requirements
Non-co-occurrence Implicit Requirements

Figure 14. Personalized implicit requirements mining results of case study.

Table 8. Implicit requirements mined in case study.

ERE	Implicit Requirements	Mining Method and Calculation Results			Types	Whether to Adopt
		Co-occurrence Frequency	Jaccard Similarity	Cosine Similarity		

Sorting	Transport	4	0.89%	/	II	**Yes**
	Classification	2	0.94%	/	I	**Yes**
	Detection	2	0.93%	/	I	**Yes**
	Marking	1	0.93%	/	II	**Yes**
	Check	1	0.93%	/	I	**Yes**
	Assignment	1	0.93%	/	I	No
	Tin dipping	1	0.93%	/	II	No
	Prevent jamming	1	0.91%	/	II	No
	Shuttle	/	/	65.52%	Ⅳ	No
	Delivery	/	/	65.12%	Ⅳ	No
	Logistics	/	/	64.85%	Ⅳ	No
	Distribution	/	/	63.99%	Ⅳ	No
	Store	/	/	63.46%	Ⅳ	No
	Place	/	/	62.91%	Ⅳ	**Yes**
Steel Pipe	Inner wall	3	2.24%	/	Ⅲ	**Yes**
	Port	2	2.04%	/	Ⅲ	**Yes**
	Surface	4	1.22%	/	Ⅲ	**Yes**
	Transport height	1	1.14%	/	Ⅲ	No
	Side formwork	1	1.14%	/	Ⅲ	No
	Building materials	1	0.96%	/	Ⅲ	**Yes**
	Steel ring	/	/	69.85%	V	No
	Short pipe	/	/	64.62%	V	**Yes**
	Body of ships	/	/	64.29%	Ⅲ	No
	Billet rod	/	/	64.25%	V	No
	Head end	/	/	64.02%	Ⅲ	**Yes**

Note: The specific meaning of the categories from I to V is shown in Table 4.

After exploring implicit requirements, the completed user requirements can be described as follows:

"Our company is engaged in building materials. We hope to have a product that can automatically sort ***building materials*** *steel pipes. The equipment is required to* ***classify*** *steel pipes by* ***detecting short pipes****. In some scenarios, it is necessary to verify the* ***marking*** *information on the* ***inner wall****,* ***port*** *or* ***surface*** *of the steel pipe. After sorting, the* ***head ends*** *of the steel pipes requirement to be* ***aligned*** *and* ***placed****, and then dispatched to the vehicle for* ***transportation****. Now it is manually sorted on the open ground and sorted to the vehicle. About 10 vehicles are sorted a day. I hope that the equipment can replace manual labor. Manufacturers who can do it can contact me for communication. There is no limit on the region."*

6.3. Discussions

First, we will focus on the training performance of the two NLP models proposed in this study. The models were fine-tuned at different stages of the graph construction, and evaluated using as the standard and by 5-fold cross-validation.

6.3.1. Domain Patent Classification Model

The results are shown in Figure 15. When the training reached the sixth round, the of each model was stable at around 90%. Starting from the seventh round, some models began to overfit, and the on the validation set showed a downward trend. Therefore, based on the performance of each model when the training round was 6, the best training performance at this time was BERT-wwm-ext-patent-fusion (=92.56%).

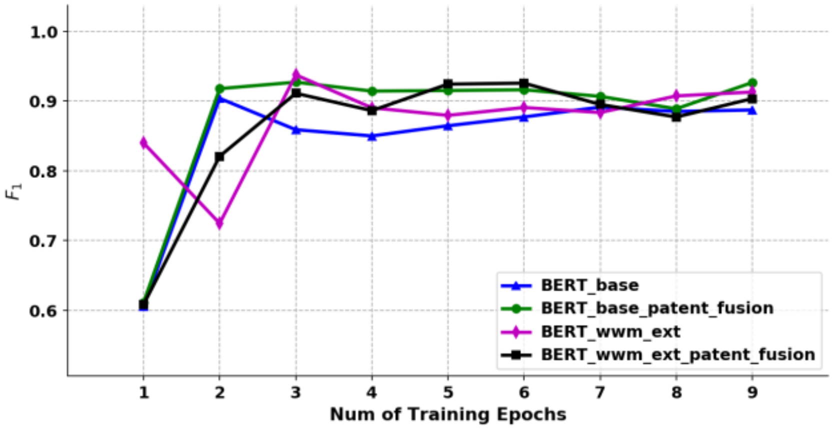


Figure 15. Personalized implicit requirements mining results of case study.

6.3.2. Functional and Structural Entities Identification Model

Different from the classification model, in the effect evaluation of the entity identification model, this study adopts accurate matching evaluation in the calculation of quasi-call rate, that is, only when the boundary and category are completely matched with the manually marked entity, it is recorded as True Positives (TP). And Micro- is used, that is, the method of calculating the total accuracy rate and recall rate of all categories.

The results are shown in Figures 16 and 17. The value of identifying the functional entities from the patent titles is about 70%, which is mainly because the patent titles are short texts, contain less context information, and have obvious unstructured characteristics, which increases the difficulty of obtaining information from the model. While the structural entities identification model in the patent claim text is relatively high, up to 94.6%, and the best performance of the above two model training is BERT-wwm-ext-patent-fusion.

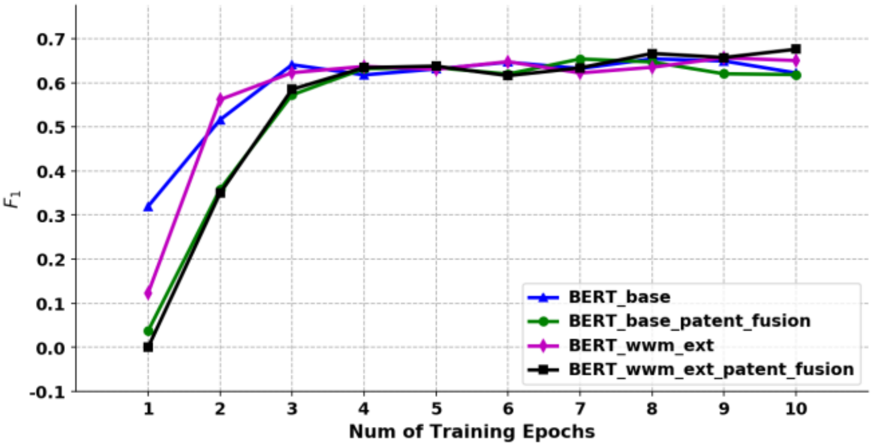


Figure 16. Training performance of the functional entities identification model in the patent titles.

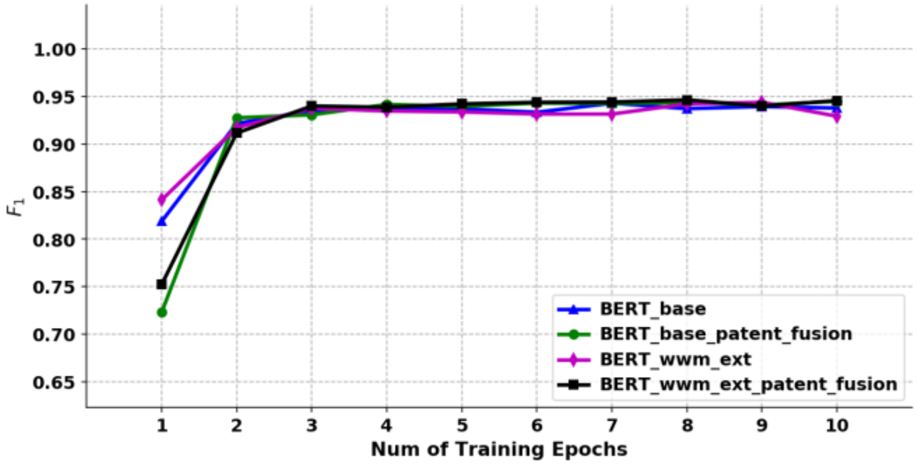


Figure 17. Training performance of the structural entities identification model in the patent claim.

6.3.3. Structural Entities Classification Model

As shown in Figure 18, the value of the pre-trained model proposed in this study is significantly better than the benchmark BERT model by over 2 percentage points, and the BERT-base-patent-fusion model was used as the classification model of structural entities.

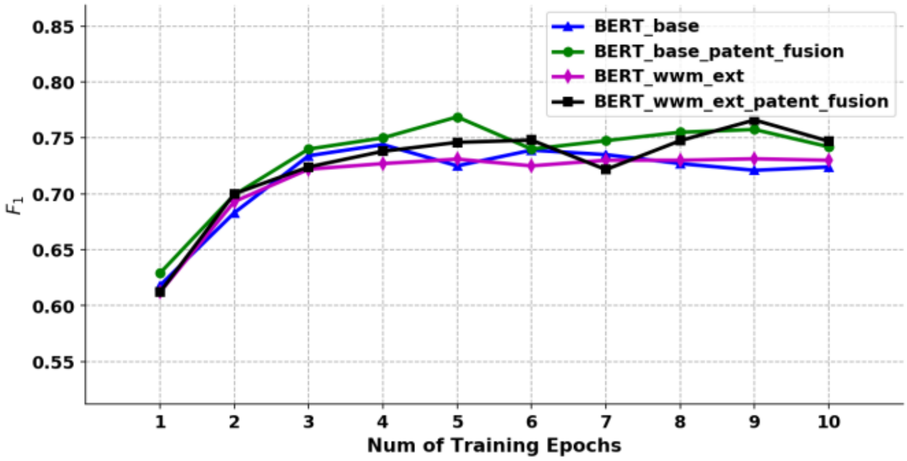


Figure 18. Training performance of the structural entities classification model.

The performance of the four pre-trained models is summarized in Table 9

Table 9. Performance of each pre-trained model on each problem of knowledge graph construction.

Problems	Pre-training Model							
	BERT-base		BERT-wwm-ext		BERT-base-patent-fusion*		BERT-wwm-ext-patent-fusion*	
	Round	F-Score	Round	F-Score	Round	F-Score	Round	F-Score
Domain Patent Classification	6	87.71%±6.41%	6	89.06%±8.69%	6	91.61%±6.54%	6	92.56%±7.30%
Functional Entities Identification	8	65.42%±3.39%	9	65.65%±3.54%	7	65.39%±2.42%	10	67.58%±2.92%
Structural Entities Identification	7	94.28%±3.59%	9	94.38%±2.68%	10	94.48%±3.49%	8	94.64%±2.20%

Structural Entities Classification	4	74.40%± 4.26%	9	73.13%±3.6 6%	5	**76.88%±3.80%**	9	76.60%±2.54%

The results indicate that the pre-training models proposed in this study outperformed the other models at different stages of graph construction.

In addition, the final output of the research content of this paper, that is, the results of personalized implicit requirements mining, should be discussed. From the case study, it can be seen that the personalized requirements are complemented through the mining of implicit requirements. The complemented user requirements can significantly improve the quality of prompt, and the interaction efficiency between users and experts in the process of group intelligence innovation.

7. Conclusions

The Internet's rapid development has led to a surge in requirement for personalized product design and development. However, meeting implicit requirements that drive product innovation remains a challenge. To address this, a study has proposed a method for mining personalized implicit requirements based on a domain knowledge graph. This method involves analyzing personalized requirements and knowledge from domain patents. A patent knowledge ontology layer representation method has been proposed, leading to the construction of a domain knowledge graph. This expands the information contained in the knowledge graph. The two NLP models proposed during the study successfully improves the efficiency of graph construction. Two implicit requirement mining methods are proposed: one based on structural similarity and another based on link prediction. These methods aim to overcome subjective professional limitations and broaden the requirement space for product innovation design.

However, this study has identified some shortcomings that require further exploration. Future research could focus on combing the user context during personalized requirement analysis, and mining implicit requirements in specific usage scenarios. Additionally, in practical engineering application, how to further expand the knowledge source and build a knowledge graph with both breadth and depth under knowledge integration remains to be further studied.

Author Contributions: Conceptualization, ZC.M and L.G.; methodology, ZC.M and J.G; software, ZC.M and J.G; validation, ZC.M and J.G.; formal analysis, J.G; investigation, ZC.M; data curation, J.G.; writing—original draft preparation, ZC.M and J.G; writing—review and editing, HR.C and JD.L; visualization, ZC.M; supervision, L.G; project administration, L.G; funding acquisition, L.G.

Funding: This research was funded by National Natural Science Foundation of China, grant number No.52375229, Theory and Method of Collective Intelligence Innovative Design Space Collaborative Exploration and Innovation Generation.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

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

References

1. Luo, J. Designing the future of the fourth industrial revolution. J Eng Design 2023, 34, 779-785, doi:10.1080/09544828.2023.2268848.
2. Wei, G.; Zhen, W.; Hongyu, S.; Lei, W.; Lin, G.; Shusong, Y.U.; Yixiong, F.; Xinming, W.; Jianqin, L. Crowdsourcing design theory and key technology development. Computer Integrated Manufacturing System 2022, 28, 2650.
3. Wei, G.; Yixiong, F.; Lei, W. crowdsourcing design theory and method. China Machine Press: Beijing,China, 2023; p 306.

4. Shanghua, M.I.; Zhaoxi, H.; Yixiong, F.; Shanhe, L.; Shaomei, F.; Kangqu, Z.; Tifan, X.; Wei, G.; Jianrong, T. Process control key technologies of crowdsourcing design for product innovation, research and development. Computer Integrated Manufacturing System 2022, 28, 2666.
5. Zhu, Q.; Luo, J. Toward artificial empathy for human-centered design. J Mech Design 2024, 146, 61401.
6. Liu, Q.; Wang, K.; Li, Y.; Liu, Y. Data-driven concept network for inspiring designers' idea generation. J Comput Inf Sci Eng 2020, 20, 31004.
7. Li, X.; Chen, C.; Zheng, P.; Wang, Z.; Jiang, Z.; Jiang, Z. A knowledge graph-aided concept–knowledge approach for evolutionary smart product–service system development. J Mech Design 2020, 142, 101403.
8. Liu, Q.; Wang, K.; Li, Y.; Chen, C.; Li, W. A novel function-structure concept network construction and analysis method for a smart product design system. Adv Eng Inform 2022, 51, 101502.
9. Tan, L.; Zhang, H. An approach to user knowledge acquisition in product design. Adv Eng Inform 2021, 50, 101408.
10. Goldberg, D.M.; Abrahams, A.S. Sourcing product innovation intelligence from online reviews. Decis Support Syst 2022, 157, 113751.
11. Jin, J.; Liu, Y.; Ji, P.; Kwong, C.K. Review on recent advances in information mining from big consumer opinion data for product design. J Comput Inf Sci Eng 2019, 19, 10801.
12. Kano, N.; Seraku, N.; Takahashi, F.; Tsuji, S. Attractive quality and must-be quality. 1984.
13. Mikulić, J.; Prebežac, D. A critical review of techniques for classifying quality attributes in the Kano model. Managing Service Quality: An International Journal 2011, 21, 46-66.
14. Zhou, F.; Ayoub, J.; Xu, Q.; Jessie Yang, X. A machine learning approach to customer needs analysis for product ecosystems. J Mech Design 2020, 142, 11101.
15. Budiarani, V.H.; Maulidan, R.; Setianto, D.P.; Widayanti, I. The kano model: How the pandemic influences customer satisfaction with digital wallet services in Indonesia. Journal of Indonesian Economy and Business (Jieb) 2021, 36, 61-82.
16. Bhardwaj, J.; Yadav, A.; Chauhan, M.S.; Chauhan, A.S. Kano model analysis for enhancing customer satisfaction of an automotive product for Indian market. Materials Today: Proceedings 2021, 46, 10996-11001.
17. Zadeh, L.A. Fuzzy sets. Information and Control 1965, 8, 338-353.
18. Pawlak, Z. Rough sets. International Journal of Computer & Information Sciences 1982, 11, 341-356.
19. Zheng, P.; Xu, X.; Xie, S.Q. A weighted interval rough number based method to determine relative importance ratings of customer requirements in QFD product planning. J Intell Manuf 2019, 30, 3-16.
20. Chen, Z.; Ming, X.; Zhou, T.; Chang, Y.; Sun, Z. A hybrid framework integrating rough-fuzzy best-worst method to identify and evaluate user activity-oriented service requirement for smart product service system. J Clean Prod 2020, 253, 119954.
21. Haber, N.; Fargnoli, M.; Sakao, T. Integrating QFD for product-service systems with the Kano model and fuzzy AHP. Total Qual Manag Bus 2020, 31, 929-954.
22. Meixia, Z.; Yahui, C.; Xiu, Y.; Li, L. Charging demand distribution analysis method of household electric vehicles considering users' charging difference. Electric Power Automation Equipment/Dianli Zidonghua Shebei 2020, 40.
23. Young, T.; Hazarika, D.; Poria, S.; Cambria, E. Recent trends in deep learning based natural language processing. Ieee Comput Intell M 2018, 13, 55-75.
24. Li, X.; Chen, C.; Zheng, P.; Jiang, Z.; Wang, L. A context-aware diversity-oriented knowledge recommendation approach for smart engineering solution design. Knowl-Based Syst 2021, 215, 106739.
25. Guofang, Z.; Jiaojiao, K.; Linghua, C. Web review of a text-driven vehicle design planning approach. Machine Design 2021, 38, 139-144.
26. Zhou, F.; Jianxin Jiao, R.; Linsey, J.S. Latent customer needs elicitation by use case analogical reasoning from sentiment analysis of online product reviews. J Mech Design 2015, 137, 71401.
27. Timoshenko, A.; Hauser, J.R. Identifying customer needs from user-generated content. Market Sci 2019, 38, 1-20.
28. Wang, Z.; Chen, C.; Zheng, P.; Li, X.; Khoo, L.P. A graph-based context-aware requirement elicitation approach in smart product-service systems. Int J Prod Res 2021, 59, 635-651.
29. Chen, R.; Wang, Q.; Xu, W. Mining user requirements to facilitate mobile app quality upgrades with big data. Electron Commer R A 2019, 38, 100889.
30. Zhang, M.; Fan, B.; Zhang, N.; Wang, W.; Fan, W. Mining product innovation ideas from online reviews. Inform Process Manag 2021, 58, 102389.
31. Chen, K.; Jin, J.; Luo, J. Big consumer opinion data understanding for Kano categorization in new product development. J Amb Intel Hum Comp 2022, 1-20.
32. Siddharth, L.; Blessing, L.; Luo, J. Natural language processing in-and-for design research. Des Sci 2022, 8, e21.
33. Jia, J.; Zhang, Y.; Saad, M. An approach to capturing and reusing tacit design knowledge using relational learning for knowledge graphs. Adv Eng Inform 2022, 51, 101505.

34. Luo, J.; Sarica, S.; Wood, K.L. in Computer-aided design ideation using InnoGPS, International design engineering technical conferences and computers and information in engineering conference, 2019; American Society of Mechanical Engineers: 2019; pp V02AT03A011.
35. Ye, F.; Fu, T.; Gong, L.; Gao, J. in Cross-domain knowledge discovery based on knowledge graph and patent mining, Journal of Physics: Conference Series, 2021; IOP Publishing: 2021; p 42155.
36. Luo, J.; Sarica, S.; Wood, K.L. Guiding data-driven design ideation by knowledge distance. Knowl-Based Syst 2021, 218, 106873.
37. Verganti, R.; Vendraminelli, L.; Iansiti, M. Innovation and design in the age of artificial intelligence. J Prod Innovat Manag 2020, 37, 212-227.
38. Zhu, Q.; Zhang, X.; Luo, J. Biologically inspired design concept generation using generative pre-trained transformers. J Mech Design 2023, 145, 41409.
39. Zhu, Q.; Luo, J. in Generative design ideation: a natural language generation approach, International Conference on-Design Computing and Cognition, 2022; Springer: 2022; pp 39-50.
40. Zhu, Q.; Luo, J. Generative transformers for design concept generation. J Comput Inf Sci Eng 2023, 23, 41003.
41. Li, Z.; Tate, D.; Lane, C.; Adams, C. A framework for automatic TRIZ level of invention estimation of patents using natural language processing, knowledge-transfer and patent citation metrics. Comput Aided Design 2012, 44, 987-1010.
42. Hulth, A. in Improved automatic keyword extraction given more linguistic knowledge, Proceedings of the 2003 conference on Empirical methods in natural language processing, 2003; 2003; pp 216-223.
43. Cui, Y.; Che, W.; Liu, T.; Qin, B.; Yang, Z. Pre-training with whole word masking for chinese bert. Ieee/Acm Transactions On Audio, Speech, and Language Processing 2021, 29, 3504-3514.
44. Lample, G.; Ballesteros, M.; Subramanian, S.; Kawakami, K.; Dyer, C. Neural architectures for named entity recognition. Arxiv Preprint Arxiv:1603.01360 2016.
45. Daud, N.N.; Ab Hamid, S.H.; Saadoon, M.; Sahran, F.; Anuar, N.B. Applications of link prediction in social networks: A review. J Netw Comput Appl 2020, 166, 102716.
46. Kumar, A.; Singh, S.S.; Singh, K.; Biswas, B. Link prediction techniques, applications, and performance: A survey. Physica a: Statistical Mechanics and its Applications 2020, 553, 124289.
47. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems 2013, 26.
48. Perozzi, B.; Al-Rfou, R.; Skiena, S. in Deepwalk: Online learning of social representations, Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, 2014; 2014; pp 701-710.

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.