Research on Version Difference Detection Method of Metrology Base Platform Based on Knowledge Graph

Runzhou Ma[1], Rongxin Zhu[1*]
[1]School of Computer Science, Hainan University, Haikou, 570228, China
Corresponding author: Rongxin Zhu (e-mail: :rongxin.zhu@hainanu.edu.cn)

ABSTRACT The metrology base platform serves as the core infrastructure for data management, business operation and standard implementation in the metrology field. Iterative updates of the platform generate multiple versions, and differences in functions, data and rules between versions directly affect the accuracy and stability of metrology work. Traditional version detection methods mainly rely on manual comparison and text retrieval, which suffer from low efficiency, high omission rate and inability to sort out correlation relationships. Taking knowledge graph technology as the core, this paper proposes a version difference detection method for metrology base platforms. This study first analyzes the iteration characteristics of metrology base platform versions and the drawbacks of traditional detection methods, then designs the construction process of a domain-specific knowledge graph for metrology. Combined with the comparison logic of graph nodes, relationships and attributes, a version difference detection model is established, and an experimental scheme is designed to verify the proposed method. The experimental results show that this method can automatically identify differences in data structures, business rules and functional modules across different versions of the metrology base platform, with detection efficiency and accuracy significantly outperforming traditional manual detection and general text comparison methods. This research provides technical references for version management, system operation and maintenance, and standard compliance verification of metrology platforms.

INDEX TERMS Knowledge Graph, Metrology Base Platform, Version Detection, Difference Analysis, System Operation and Maintenance.

I. INTRODUCTION

A. RESEARCH BACKGROUND

Metrology is a fundamental work to guarantee the accuracy and uniformity of measurement data across all industries, widely applied in industrial production, public services, market supervision and other fields[1]. As an integrated platform that consolidates metrology standards, data, business processes and management rules, the metrology base platform acts as the technical carrier for the normal operation of the entire metrology system[2]. Driven by continuous revisions of metrology standards, growing business demands and the upgrading of digital technologies, the metrology base platform undergoes regular version iterations[3]. Each version update brings adjustments to platform module functions, data fields, verification rules and business processes[4].

In actual operation and maintenance scenarios, staff need to quickly identify specific differences between new and old versions. On the one hand, version changes affect the operating habits of frontline staff and alter the logic of data submission and result verification[5]. On the other hand, some version modifications are associated with metrology compliance requirements. Failure to detect relevant differences in a timely manner may lead to data errors, business violations and system docking failures[6].

At present, version difference detection for metrology base platforms in most domestic institutions still depends on manual checking[7]. Staff compare system contents module by module, field by field and rule by field between two versions, which is extremely time-consuming. For large-scale metrology base platforms, manual detection is prone to human-induced omission and misjudgment[8]. Some institutions adopt general text comparison tools, yet such tools can only identify literal discrepancies. They fail to understand the inherent business logic of metrology, distinguish between trivial text modifications and core business rule changes, or sort out logical connections among different differences[9].

Knowledge graph technology structurally represents entities, attributes and correlation relationships within a domain, converting unstructured and semi-structured platform contents into visualized and computable graph data[10]. Leveraging the correlation analysis capability of knowledge graphs for version comparison can break through the limitations of traditional detection methods[11]. Against this backdrop, this paper conducts research on version difference detection methods for metrology base platforms based on knowledge graphs, which possesses high practical application value.

B. RESEARCH SIGNIFICANCE

1) Theoretical Significance

Current research on version difference detection mainly focuses on general software, internet platforms and office systems, while studies targeting version detection of dedicated metrology base platforms are relatively scarce. Meanwhile, existing applications of knowledge graphs in the metrology field are mostly concentrated on standard retrieval, knowledge push and fault diagnosis, and the integration of knowledge graphs with system version difference detection is still in the exploratory stage.

Combined with the business characteristics of the metrology industry, this paper constructs a knowledge graph architecture adapted to metrology base platforms and designs a graph-based difference detection process. The research findings enrich the application scenarios of knowledge graphs in the operation and maintenance of industry-specific systems, and also provide new ideas for version detection research on industrial software, government affairs platforms and other professional systems.

2) Practical Significance

First, it improves the efficiency of version detection. The proposed method realizes automatic difference identification, reduces manual workload and shortens the verification cycle before version release.

Second, it enhances the completeness of detection results. Relying on the correlation logic of knowledge graphs, the method can fully explore explicit differences and implicit associated differences, and reduce the omission rate.

Third, it standardizes version management of metrology base platforms. Structured and clear difference results help operation and maintenance personnel as well as business staff quickly understand version modifications, ensure the smooth transition of metrology business, and maintain the accuracy and compliance of metrology data.

C. RESEARCH STATUS

1) Version Difference Detection

Domestic relevant research falls into two categories[12]. The first is code-level version detection, which serves software development enterprises and focuses on comparing changes in program codes and interface parameters for software development and vulnerability investigation. The second is comparison of documents and system configurations, which is widely used in government affairs systems and office platforms.

A common defect of existing methods is that they only focus on isolated content differences while ignoring business correlations[13]. When a system involves complex business logic, the detection results cannot reflect the cascading impacts brought by modifications.

2) Industrial Applications of Knowledge Graphs

After being introduced into China, knowledge graph technology was first applied in internet fields such as search engines and intelligent question answering[14]. In recent years, it has gradually penetrated into traditional industries. In the metrology field, existing studies mainly use knowledge graphs to integrate knowledge including metrology standards, measuring instruments and detection methods, build metrology knowledge service platforms, and realize functions such as standard inquiry, knowledge popularization and auxiliary detection[15].

In the field of system operation and maintenance and platform management, scholars have applied knowledge graphs to fault location, equipment management and process sorting[16]. However, there are few cases of applying knowledge graphs to system version difference detection. Overall, the integration of knowledge graphs and version detection for industry-specific systems still has great room for research and application.

D. RESEARCH CONTENTS

The research contents of this paper are divided into four parts:

1.Analyze the structural characteristics and version iteration rules of metrology base platforms, and summarize the defects of traditional version detection methods.

2.Design the construction scheme of the knowledge graph for metrology base platforms combined with metrology business contents, and clarify the division rules of entities, relationships and attributes.

3.Establish a version difference detection model and complete workflow based on knowledge graph comparison, and classify and count different types of differences.

4.Build an experimental environment and conduct comparative experiments using real version data of metrology base platforms to verify the effectiveness of the proposed method.

E. RESEARCH INNOVATIONS

1.Scenario Innovation: This paper applies knowledge graph technology to the dedicated industry platform of metrology base platforms, and designs graph structures and detection logic tailored to the unique rules of metrology business, filling the application gap in this segmented scenario.

2.Method Innovation: Different from superficial text comparison and code comparison, this method leverages the entity correlation of knowledge graphs to identify both explicit differences and implicit associated differences simultaneously.

3.Application Innovation: A standardized detection process and result classification mode are formulated. The method is easy to understand and adapts to the operation and maintenance scenarios of grassroots metrology institutions, enabling large-scale promotion and application.

II. BASIC THEORIES AND TECHNOLOGIES

A. OVERVIEW OF METROLOGY BASE PLATFORM

The metrology base platform is a comprehensive management platform covering all metrology businesses. The metrology base platform studied in this paper consists of five core components: data storage module, metrology standard module, business process module, rule verification module and user operation module.

Version iterations of the metrology base platform are divided into minor updates and major updates. Minor updates mainly involve modifying individual data fields, correcting simple verification rules and optimizing page displays. Major updates include adding new business modules, adjusting overall processes, updating complete sets of metrology standards and reconstructing data structures. Different types of version updates lead to differences with varying scopes and impacts.

B. BASIC CONCEPTS OF VERSION DIFFERENCE DETECTION

Version difference detection refers to the process of identifying all discrepancies between two different versions of the same software or platform via technical means. In terms of detection objects, it can be divided into code layer detection, configuration file detection, business content detection and interface function detection.

This paper mainly targets business contents, configuration rules and data structures of metrology base platforms, excluding underlying program codes. In terms of manifestation, version differences are classified into three categories:
1.Newly added content: Entities, attributes or relationships that exist in the new version but not in the old one.
2.Deleted content: Content that exists in the old version but is removed in the new version.
3.Modified content: Entities remain, while attribute values or correlation relationships are changed. These three categories are also the main basis for subsequent detection and classification in this paper.

C. CORE TECHNOLOGIES OF KNOWLEDGE GRAPHS

The complete construction process of a knowledge graph includes seven steps: data collection, data preprocessing, entity extraction, relationship extraction, attribute extraction, graph fusion and graph storage. The constructed knowledge graph supports computational operations such as node comparison, relationship traversal and attribute comparison, which serve as the technical foundation for version difference detection.

D. OVERVIEW OF METROLOGY BASE PLATFORM

The core of comparing two homologous knowledge graphs is to match nodes, relationships and attributes one by one. First, match entity nodes to check whether identical entities exist in the two graphs. Second, compare the correlation relationships between entities to judge whether connection modes have changed. Finally, match various attributes of entities to verify changes in attribute values.

Through three layers of comparison, newly added, deleted and modified differences can be fully identified. Meanwhile, relying on the graph network, the impact of changes to one entity on associated entities can be tracked, realizing the mining of implicit differences.

III. ANALYSIS OF VERSION CHARACTERISTICS OF METROLOGY BASE PLATFORMS AND TRADITIONAL DETECTION METHODS

A. ITERATION CHARACTERISTICS OF METROLOGY BASE PLATFORM VERSIONS

To clarify detection objectives, this paper sorts out version update records of in-service metrology base platforms in multiple regions over the past three years and summarizes four typical iteration characteristics:
1.Stable update frequency: Metrology standards are revised regularly in accordance with national requirements. Thus, minor versions of the metrology base platform are released every 1 to 3 months on average, and 1 to 2 major versions are launched every year.
2.Scattered distribution of differences: A single version update usually involves modifications to multiple parts including data fields, verification rules, business processes and standard documents, with differences distributed across all areas of the platform.
3.Correlation of differences: Modifications to a core rule will trigger synchronous changes to multiple data fields and detection items. For example, adjustments to metrology accuracy standards will lead to changes in corresponding threshold fields, judgment rules and detection processes, forming a correlated network of modifications.
4.Varied importance of differences: Some modifications such as page text and style adjustments are low-impact differences; while changes to metrology judgment rules and core data structures are high-impact differences that directly determine the compliance of metrology results. Detection staff need to distinguish the impact levels of differences.

B. CLASSIFICATION OF TRADITIONAL VERSION DIFFERENCE DETECTION METHODS

Currently, three main methods are adopted for version detection of metrology base platforms in the industry: manual item-by-item comparison, general text comparison and configuration file comparison. Their working processes and applicable scenarios are analyzed as follows.

1) Manual Item-by-Item Comparison

As the most widely used method at present, manual comparison requires staff to log into both the new and old versions of the metrology base platform, check pages, data items, verification rules and business processes module by module, and manually record all discrepancies. Cross-verification is also required after initial comparison to avoid errors.

This method enables staff to judge the impact of differences based on professional experience. However, it has obvious drawbacks: extremely low working efficiency (a complete comparison of two versions of a medium-sized metrology base platform takes 3 to 5 working days); frequent omission and misjudgment caused by long-time manual operation; and inability to quickly sort out correlations between differences.

2) General Text Comparison

This method exports rule documents, field descriptions, standard texts and other contents of the metrology base platform into plain text files, and uses universal text comparison tools to scan and mark literal discrepancies.

It improves the comparison speed of text contents but has prominent limitations: it can only identify literal differences and cannot understand metrology business logic or distinguish between trivial text modifications and core rule changes; it is unable to detect non-text contents such as data field types, rule thresholds and business process relationships; and it lacks correlation analysis capability to discover cascading impacts of differences.

3) Configuration File Comparison

Operating parameters, field definitions and rule parameters of the metrology base platform are stored in background configuration files. This method directly compares configuration files of new and old versions, which can accurately identify modifications to parameters and is mainly used by technical staff to troubleshoot underlying configuration changes.

Its limitations include: high usage threshold, as configuration files are incomprehensible to ordinary business staff and can only be operated by professional technicians; failure to detect non-configuration contents such as front-end pages, business processes and standard documents; and scattered modification entries in complex configuration files, making detection results hard to convert into business-readable information.

C. SUMMARY OF COMMON PROBLEMS OF TRADITIONAL METHODS

Combined with the above analysis, the three traditional methods share four major problems:

1.Incomplete detection coverage: A single method can only compare specific types of contents and fails to realize unified detection across all modules.

2.Low automation: Most processes still rely on manual intervention, resulting in high labor costs.

3.Lack of correlation analysis capability: Only isolated differences can be found, while cascading impacts caused by modifications cannot be explored.

4.Poor readability of results: Some detection results adopt professional formats that are difficult for business staff to understand, hindering the application of new versions.

These problems severely affect the efficiency and quality of version management for metrology base platforms. Therefore, this paper introduces knowledge graph technology to design a new detection method to address the pain points of traditional approaches.

IV. DESIGN OF VERSION DIFFERENCE DETECTION METHOD BASED ON KNOWLEDGE GRAPH

A. OVERALL DESIGN IDEA

The proposed method consists of two major stages:

1.Knowledge graph construction stage: Extract business data, rules, processes and standards from the old and new versions of the metrology base platform respectively to build two independent knowledge graphs.

2.Graph comparison and difference detection stage: Compare entities, relationships and attributes of the two graphs layer by layer to identify newly added, deleted and modified differences. Finally, classify, count and grade the differences to output standardized detection reports.

The whole process is dominated by automatic processing, with manual work only involved in preliminary data sorting and subsequent result review.

B. CONSTRUCTION OF KNOWLEDGE GRAPH FOR METROLOGY BASE PLATFORM

1) Data Collection

Data collection is the first step of graph construction. Five types of data sources covering all business contents of the metrology base platform are collected:

1.Structured data: Database data fields, field types, field definitions and data table correlations.

2.Rule data: Verification rules, judgment thresholds, compliance standards and calculation logic.

3.Process data: Business steps, sequence of steps and associated operation items.

4.Document data: Metrology standard texts, operation instructions and remarks.

5.Functional module data: Platform menus, function entrances and module divisions.

Two collection methods are adopted: structured data and configuration rules are exported in batches via background interfaces; documents and page contents are collected by combining web crawlers and manual sorting. The same collection operations are performed on both old and new versions to obtain two sets of original datasets.

2) Data Preprocessing

Original collected data usually has problems such as disordered formats, duplicate contents, invalid characters and missing information, which require preprocessing including four tasks:

1.Data cleaning: Remove empty data, duplicate data, garbled characters and meaningless remarks.

2.Format unification: Convert documents and configuration files in different formats into standard formats for subsequent extraction.
3.Data completion: Supplement missing key information based on original platform contents.
4.Data classification: Classify data by modules and business types to reduce the difficulty of subsequent extraction.

Two sets of standardized datasets are obtained after preprocessing.

3) Definition and Extraction of Entities, Relationships and Attributes

Combined with the business characteristics of the metrology base platform, this paper defines the three core elements of the knowledge graph and formulates corresponding extraction rules.

1. Entity Definition and Extraction

Entities refer to independent business objects in the metrology base platform, which are divided into six categories (Table 1).

TABLE 1. Entity Classification of Knowledge Graph for Metrology Base Platform

Entity Category	Specific Contents	Examples
Module Entity	Platform functional modules, primary and secondary menus	Metrology verification module, data submission module
Field Entity	All database data fields	Instrument number, detection temperature, measurement error
Rule Entity	Verification rules, judgment rules, threshold rules	Error judgment rule, data format verification rule
Process Entity	Business processes and process steps	Verification process, result review step
Standard Entity	Metrology standards and specification documents	Pressure metrology standard, length detection specification
Operation Entity	Page operations and function buttons	Data submission, result saving, record query

Entities are automatically extracted from preprocessed data by combining rule-based extraction and keyword matching. Each entity is assigned a unique ID to avoid duplication.

2. Relationship Definition and Extraction

Relationships connect two entities and reflect business correlations. Eight core relationships are summarized based on metrology business logic (Table 2).

TABLE 2. Core Relationships of Knowledge Graph for Metrology Base Platform

Relationship Name	Meaning	Connected Entity Example
Contain	A module contains multiple sub-modules/fields	Verification Module - Contain - Detection Field
Follow	Rules/processes follow specific standards	Judgment Rule - Follow - Metrology Standard
Restrict	Rules restrict corresponding data fields	Threshold Rule - Restrict - Measurement Error Field
Execute	Processes perform corresponding operations	Review Process - Execute - Data Check Operation
Associate	Different business entities have business linkages	Instrument Information - Associate - Verification Item
Precede	Process steps have a sequential order	Data Entry - Precede - Result Calculation
Belong to	Fields/rules belong to corresponding modules	Submission Field - Belong to - Data Submission Module
Reference	Documents quote other standards	Operation Manual - Reference - General Metrology Specification

Correlation relationships between entities are automatically identified according to the above rules, with entity IDs at both ends and relationship types recorded.

3. Attribute Definition and Extraction

Attributes describe the specific characteristics of entities, and different types of entities correspond to different core attributes (Table 3).

TABLE 3. Core Attributes of Various Entities

Entity Category	Core Attributes	Attribute Description
Module Entity	Module name, module ID, activation status	Identify module identity and operating status
Field Entity	Field name, data type, field length, default value	Describe characteristics of database fields
Rule Entity	Rule name, threshold, judgment logic, effective time	Describe verification and judgment rules
Process Entity	Step name, execution sequence, time limit	Describe characteristics of business processes
Standard Entity	Standard number, release time, scope of application	Distinguish different metrology standards
Operation Entity	Operation name, operation permission, trigger condition	Describe characteristics of page operations

Attributes are extracted from structured data and rule data, and attribute values are bound to corresponding entities.

4) Graph Storage

After the extraction of entities, relationships and attributes is completed, the knowledge graph is stored in a graph database. A graph database adapts to the network structure of knowledge graphs and supports efficient node query, relationship traversal and attribute comparison. Finally, two knowledge graphs denoted as G1 (old version) and G2 (new version) are generated with a unified architecture, entity classification and relationship rules, laying a foundation for subsequent comparison.

C. DESIGN OF VERSION DIFFERENCE DETECTION MODEL

Taking two homologous knowledge graphs G1 and G2 as inputs, the model adopts a three-layer comparison logic: entity layer comparison, relationship layer comparison and attribute layer comparison, to identify differences layer by layer.

1) The First Layer: Entity Layer Comparison

As the basic nodes of the graph, entities are matched first. Entities in G1 and G2 are matched bidirectionally by entity ID and entity name, with three matching results:
1.Entities exist in G1 but not in G2: identified as entity deletion differences.
2.Entities exist in G2 but not in G1: identified as entity addition differences.
3.Entities are fully matched on both sides: retain the entities and proceed to relationship comparison.

Entity layer comparison directly identifies the addition and deletion of core objects such as platform modules, fields, rules and processes, which are the most intuitive version differences.

2) The Second Layer: Relationship Layer Comparison

For entities successfully matched in the first layer, their correlation relationships are compared. All relationships of

each entity are traversed to check changes in relationship types and connected peer entities:
1.Disappearance of original relationships: identified as relationship deletion.
2.Emergence of brand-new relationships: identified as relationship addition.
3.Changes in relationship types or peer entities: identified as relationship modification.

Relationship layer comparison explores changes in business correlations, process sequences and module affiliations. Such differences are mostly implicit business discrepancies that are difficult to fully identify via traditional methods.

3) The Third Layer: Attribute Layer Comparison

For entities and relationships with complete matching, attribute values are compared item by item. If the values of the same attribute differ between the two graphs, it is identified as an attribute modification. Attribute layer comparison mainly detects detailed changes such as field parameters, rule thresholds, standard contents and permission settings.

4) Summary and Classification of Differences

Based on the results of three-layer comparison, all differences are divided into three primary categories and nine secondary categories (Table 4). Meanwhile, all differences are classified into three levels (high, medium and low) according to their impacts on metrology business: high-level differences refer to modifications affecting metrology judgment, data structures and core processes; medium-level differences involve changes to ordinary business processes and minor rules; low-level differences include adjustments to page texts, remarks and non-core descriptions.

TABLE 4. Complete Classification of Version Differences

Primary Difference Category	Secondary Difference Category	Description
Newly Added Differences	Entity Addition	New modules, fields, rules and other entities in the new version
	Relationship Addition	New business correlations between entities
	Attribute Addition	New descriptive attributes of entities
Deleted Differences	Entity Deletion	Entities existing in the old version are removed in the new version
	Relationship Deletion	Disappearance of original correlations between entities
	Attribute Deletion	Removal of original attributes of entities
Modified Differences	Entity Modification	Changes to entity names or identifiers
	Relationship Modification	Adjustments to correlations between entities
	Attribute Modification	Changes to values or contents of entity attributes

D. COMPLETE DETECTION WORKFLOW

A standardized and implementable detection workflow consisting of six steps is summarized as follows:
1.Data Collection: Export full-volume business data, configurations, documents and process information of the old and new versions of the metrology base platform.
2.Graph Construction: Preprocess the two sets of data, extract entities, relationships and attributes, and build knowledge graphs for both versions.
3.Three-layer Comparison: Conduct matching of entities, relationships and attributes layer by layer, and mark all difference points.
4.Difference Classification and Grading: Classify differences by type and assess their impact levels according to preset rules.
5.Correlation Tracing: Track other associated entities of each difference relying on the graph network, and mark cascading impacts.
Report Generation: Integrate all difference information, classification results and impact explanations to output visualized detection reports.

V. EXPERIMENT AND RESULT ANALYSIS

A. EXPERIMENTAL DATA

Two consecutive versions (V1: old version, V2: new version) of an in-service metrology base platform in a certain region are selected as experimental objects. This is a medium-scale update involving modifications to multiple modules, rules and data fields. Basic information of the two versions is shown in Table 5.

TABLE 5. Basic Information of Experimental Data

Indicator	Version V1 (Old Version)	Version V2 (New Version)
Number of original data files	126	128
Total number of extracted entities	892	917
Total number of relationships between entities	2156	2243
Total number of entity attribute entries	4685	4721

B. EXPERIMENTAL SCHEME

Three experimental groups are set up for comparison:
1.Group 1: Manual item-by-item comparison (mainstream traditional method)
2.Group 2: General text comparison (common auxiliary method)
3.Group 3: The proposed knowledge graph-based detection method

Three evaluation indicators are adopted:
1.Detection Time: Total time consumed from the start of detection to the output of complete results (unit: minute).
2.Detection Accuracy: (Number of correctly identified differences ÷ Total actual differences) × 100%.
3.Omission Rate: (Number of undetected differences ÷ Total actual differences) × 100%.

First, three senior metrology operation and maintenance staff conduct comprehensive manual verification to count all actual differences between V1 and V2 as the standard reference data. Then experiments are carried out for the three groups respectively, and relevant indicators are recorded.

C. EXPERIMENTAL RESULTS AND ANALYSIS

1) Statistics of Total Differences

Comprehensive manual verification confirms that there are 187 differences in total between V1 and V2, including 42 newly added differences, 23 deleted differences and 122 modified differences. In terms of impact level, there are 31 high-level differences, 86 medium-level differences and 70 low-level differences.

2) Comparison of Indicators Among Three Methods

1.Manual item-by-item comparison: Total time = 426 minutes, detection accuracy = 78.6%, omission rate = 21.4%. This method consumes the longest time. Long-duration manual work leads to numerous omissions, mainly in associated relationships and implicit rule modifications.

2.General text comparison: Total time = 89 minutes, detection accuracy = 59.3%, omission rate = 40.7%. Although it is faster than manual comparison, it can only identify literal differences and fails to detect changes in fields, rules, processes and relationships, resulting in the lowest accuracy and highest omission rate. It can only be used as an auxiliary tool.

3.Proposed knowledge graph-based method: Total time = 41 minutes, detection accuracy = 96.8%, omission rate = 3.2%. Benefiting from automatic processing, this method has the shortest detection time. Supported by the three-layer graph comparison logic, it can identify both superficial modifications and implicit changes in correlations and rules, with accuracy far exceeding traditional methods. A small number of omissions only occur in individual low-level attribute modifications of remarks, which have no impact on core businesses.

3) Analysis of Detection Effect by Difference Type

1.Newly added differences: The proposed method identifies all 42 items with the best performance; the manual method misses 6 items, and the text comparison method misses 15 items.

2.Deleted differences: The proposed method identifies 22 items, with only one low-level entity deletion missed; the manual method misses 5 items, and the text comparison method misses 11 items.

3.Modified differences: This category has the largest quantity and most complex forms. The proposed method identifies 119 items with 3 omissions; the manual method misses 28 items, and the text comparison method misses 51 items.

In terms of impact level, the proposed method achieves nearly 100% accuracy for high and medium-level core business differences, fully meeting the core detection requirements of metrology business. A small number of omissions exist for low-level non-core differences, which do not affect the normal operation and maintenance of new versions.

D. EXPERIMENTAL CONCLUSIONS

Three conclusions are drawn from the comparative experiments:

1.In terms of detection efficiency, the knowledge graph-based method is far superior to manual comparison and general text comparison, which greatly shortens the version verification cycle.

2.In terms of detection quality, the proposed method achieves a significant increase in accuracy and a sharp drop in omission rate. It is particularly effective in identifying associated differences and implicit rule changes that are hard to detect via traditional methods.

3.The method has strong adaptability and can cover all types of contents including modules, fields, rules, processes and documents, satisfying the full-module version detection demands of metrology base platforms.

The experiments verify that applying knowledge graphs to version difference detection of metrology base platforms is highly feasible and practical.

VI. CONCLUSIONS

A. RESEARCH SUMMARY

Aiming at the problems of low efficiency, high omission rate and inability to analyze correlation relationships of traditional difference detection methods during version iterations of metrology base platforms, this paper conducts research on version difference detection based on knowledge graphs.

First, this paper analyzes the structural composition and iteration characteristics of metrology base platforms, and thoroughly discusses the advantages and defects of three traditional detection methods (manual comparison, text comparison and configuration file comparison), clarifying the design objectives of the new method.

Second, combined with metrology business rules, the overall framework of the knowledge graph for metrology base platforms is constructed. The classification of entities, relationships and attributes is defined, and a complete graph construction process is formulated.

Third, a three-layer graph comparison model is designed to identify differences by comparing entities, relationships and attributes layer by layer. Standardized rules for difference classification and grading are established, forming a complete automatic detection workflow.

Finally, an experimental environment is built, and comparative experiments are carried out using real version data of metrology base platforms.

The experimental results verify the effectiveness of the proposed method. It realizes automatic and multi-dimensional detection of version differences with superior efficiency and accuracy, and effectively supports version operation and maintenance, compliance verification and business transition of metrology base platforms. The model and workflow designed in this paper also provide references for version detection of similar industry-specific platforms.

B. LIMITATIONS

This study still has three limitations:

1.The data extraction module mainly adopts rule matching, which leads to a slight decline in extraction accuracy when processing highly irregular unstructured long documents.

2.The experiments only adopt data of medium-scale version updates, without large-scale tests for major versions and minimal minor versions. The generalization ability of the method needs further verification.

The current difference grading rules are fixed and cannot be flexibly adjusted according to different institutions and business scenarios.

C. FUTURE RESEARCH PROSPECTS

Combined with the existing limitations, three future research directions are proposed:

1.Introduce lightweight natural language processing technology to optimize entity, relationship and attribute extraction algorithms, and improve extraction accuracy for non-standard documents.

2.Expand experimental datasets to cover major, medium and minor version updates, continuously optimize model parameters and enhance the generalization capability of the method.

3.Design configurable difference grading rules to allow users to adjust judgment criteria according to business demands, so as to make the method adaptable to more grassroots metrology institutions.

REFERENCES

[1] R. J. C. Brown, "Measuring measurement – What is metrology and why does it matter?," Measurement, vol. 168, p. 108408, 2021, doi: 10.1016/j.measurement.2020.108408.

[2] A. Archenti et al., "Integrated metrology for advanced manufacturing," CIRP Ann., vol. 73, no. 2, pp. 639–665, 2024, doi: 10.1016/j.cirp.2024.05.003.

[3] F. Grasso Toro and H. Lehmann, "Brief overview of the future of metrology," Meas. Sens., vol. 18, p. 100306, 2021, doi: 10.1016/j.measen.2021.100306.

[4] D.-J. Lim et al., "Development of a virtual metrology system for smart manufacturing: A case study of spandex fiber production," Comput. Ind., vol. 145, p. 103825, 2023, doi: 10.1016/j.compind.2022.103825.

[5] G. E. Coelho et al., "Machine-to-machine systems for data acquisition and measurement with self-validation for the digital metrology transition," Measurement, vol. 258, p. 119149, 2026, doi: 10.1016/j.measurement.2025.119149.

[6] O. Velychko, I. Kuzmenko, and T. Gordiyenko, "Practical aspects of ensuring of the metrological traceability," Meas. Sens., vol. 38, p. 101529, 2025, doi: 10.1016/j.measen.2024.101529.

[7] L. Zeng et al., "The Research on a Knowledge Graph-Based Method for Detecting Version Differences in Metering Foundations," IEEE Access, vol. 13, pp. 161003–161015, 2025, doi: 10.1109/ACCESS.2025.3609451.

[8] R. H. Schmitt et al., "Advances in Large-Scale Metrology – Review and future trends," CIRP Ann., vol. 65, no. 2, pp. 643–665, 2016, doi: 10.1016/j.cirp.2016.05.002.

[9] C. Shaib, V. S. Govindarajan, J. Barrow, J. Sun, A. F. Siu, B. C. Wallace, and A. Nenkova, "Standardizing the Measurement of Text Diversity: A Tool and a Comparative Analysis of Scores," Comput. Lang., 2025, doi: 10.48550/arXiv.2403.00553.

[10] D. Sellami et al., "Knowledge graph representation learning: A comprehensive and experimental overview," Comput. Sci. Rev., vol. 56, p. 100716, 2025, doi: 10.1016/j.cosrev.2024.100716.

[11] Y. Wan et al., "Making knowledge graphs work for smart manufacturing: Research topics, applications and prospects," J. Manuf. Syst., vol. 76, pp. 103–132, 2024, doi: 10.1016/j.jmsy.2024.07.009.

[12] S. L. Allen, A. Byrne, and A. K. Coskun, "Version Detection for Software Discovery in the Cloud," in Proc. 21st Int. Middleware Conf. Demos Posters, Delft, Netherlands: ACM, 2020, pp. 5–6, doi: 10.1145/3429358.3429372.

[13] L. Nahar, M. Awrangjeb, and M. S. Islam, "AI-enabled defect detection in industrial products: A comprehensive survey, key insights and future research challenges," Adv. Eng. Inform., vol. 69, p. 104067, 2026, doi: 10.1016/j.aei.2025.104067.

[14] Y. Chen et al., "An Overview of Knowledge Graph Reasoning: Key Technologies and Applications," J. Sens. Actuator Netw., vol. 11, no. 4, 2022, doi: 10.3390/jsan11040078.

[15] C. Peng et al., "Knowledge Graphs: Opportunities and Challenges," Artif. Intell. Rev., vol. 56, no. 11, pp. 13071–13102, 2023, doi: 10.1007/s10462-023-10465-9.

[16] R. Kang et al., "Knowledge graphs for operational decision-making in industrial maintenance: A systematic review," Expert Syst. Appl., vol. 299, p. 130172, 2026, doi: 10.1016/j.eswa.2025.130172.