DOI: 10.19884/j.1672-5220.202406001

Knowledge Graph Based Method for Tracing Quality of Aerospace Products

WANG Ning[1], CAO Lijun[1], DING Siyi[2], MENG Yan[2], LIU Huan[1], ZHENG Xiaohu[2], HUANG Wenbin[1], LIU Xiaojia[1*]

1. Shanghai Spaceflight Precision Machinery Institute, Shanghai 201600, China
2. College of Information Science and Technology, Donghua University, Shanghai 201620, China

Abstract: Nowadays, the internal structure of spacecraft has been increasingly complex. As its "lifeline", cables require extensive manpower and resources for manual testing, and it is challenging to quickly and accurately locate quality problems and find solutions. To address this problem, a knowledge graph based method is employed to extract multi-source heterogeneous cable knowledge entities. The method utilizes the bidirectional encoder representations from transformers (BERT) network to embed word vectors into the input text, then extracts the contextual features of the input sequence through the bidirectional long short-term memory (BiLSTM) network, and finally inputs them into the conditional random field (CRF) network to predict entity categories. Simultaneously, by using the entities extracted by this model as the data layer, a knowledge graph based method has been constructed. Compared to other traditional extraction methods, the entity extraction method used in this study demonstrates significant improvements in metrics such as precision, recall and an F1 score. Ultimately, employing cable test data from a particular aerospace precision machining company, the study has constructed the knowledge graph based method in the field to achieve visualized queries and the traceability and localization of quality problems.

Key words: knowledge graph; named entity recognition; quality control; aerospace product

CLC number: TP391 **Document code**: A

Article ID: 1672-5220(2024)05-0513-12

Open Science Identity (OSID)

0 Introduction

In recent years, science and technology improvements have greatly enhanced spacecraft performance. This has led to more complex onboard electronics with stricter requirements. Quality of spacecraft cable networks, crucial for communication and energy flow, significantly affects spacecraft quality[1-4]. Aerospace cables have complex systems and varied production sources. A thorough conduction test of the cable network is essential during the final spacecraft assembly. These tests generate extensive raw data that need thorough checking, leading to inefficiency and resource wastage. Quickly finding and fixing aerospace cable issues to ensure stable and reliable information and energy systems is now a pressing challenge.

The integration of artificial intelligence and big data with manufacturing has led to the development of smarter information systems. These systems enhance production by analyzing data and predicting outcomes, thereby achieving intelligent manufacturing. Knowledge engineering, a subset of artificial intelligence, leverages human knowledge encoded in computers to solve problems. The knowledge graph supports intelligent manufacturing through its vast data handling and analysis capabilities. Its applications in manufacturing have expanded, including data integration[5-8], product design[9-12], scheduling, quality control[13-20] and maintenance[21-24].

Knowledge extraction, the key technology for constructing knowledge graphs, provides foundational data support. Knowledge extraction methods can be broadly categorized into rule-based, machine learning-based and deep learning-based methods[25]. Early methods were predominantly based on rules manually written by experts, but suffered from poor maintainability and robustness due to their heavy reliance on expert experience. Subsequently, traditional machine learning methods were introduced into entity recognition tasks. For example, Yu et al.[26] improved a multi-layer hidden Markov model (HMM) to identify simple entities at lower levels. Li et al.[27] first introduced the maximum entropy model into Chinese text classification tasks. He et al.[28] combined the conditional random field (CRF) model with manually created rules for recognition. The emergence of deep learning subsequently made knowledge extraction techniques more accurate. Hammerton[29] utilized the sequential characteristics of long short-term memory (LSTM) networks and was the first to apply deep learning models to the task of named entity recognition. Pinheiro et al.[30] achieved good results on the CoNLL-2003 dataset with a convolutional neural

Received date: 2024-06-06

* Correspondence should be addressed to LIU Xiaojia, email: lxj9039@126.com

Citation: WANG N, CAO L J, DING S Y, et al. Knowledge graph based method for tracing quality of aerospace products[J]. *Journal of Donghua University (English Edition)*, 2024, 41(5): 513-524.

network (CNN)-CRF model. Lample et al.[31] combined bidirectional long short-term memory (BiLSTM) with CRF, achieving an F1 score of 90.94% on the CoNLL-2003 dataset. Chiu et al.[32] employed a convolutional network for preliminary feature extraction before the BiLSTM network, enabling better feature capture among words, and obtained an impressive F1 score of 86.28% on the OntoNotes dataset. However, the above-mentioned entity recognition networks still face an issue in the word embedding layer: dynamic word embedding representation cannot be achieved. The embedding representation of each word remains a fixed high-dimensional vector, lacking the ability to effectively extract contextual features.

Therefore, this paper utilizes bidirectional encoder representations from transformers (BERT) networks for word embedding representation, incorporating three types of embedding layers to achieve dynamic representation of words. Addressing the complexities of spacecraft cable network testing, the paper proposes a knowledge extraction model based on BERT-BiLSTM-CRF networks. This model performs entity recognition on data from cable testing processes and utilizes a Neo4j graph database to construct a corresponding knowledge graph. Through this database, rapid identification of cable quality issues is enabled, facilitating the formulation of solutions and achieving traceability and control over cable quality.

1 Aerospace Cable Network Knowledge Graph

1.1 Entity extraction model

By analyzing the application of the knowledge graph in the field of aerospace product quality early warning and control, the overall architecture of the knowledge graph was designed to support the subsequent engineering construction and the application of the aerospace product quality early warning and control. Based on the current conditions, the accumulated quality issue data from past cable network production tests were used as demonstration points for practical application, and the graph was constructed. The goal was entity identification within the knowledge graph for the quality early warning and control domain. A framework was designed for the extraction of entities related to quality early warning and control. The entity extraction framework for the knowledge graph is structured into three layers as shown in Fig. 1: a data pre-processing layer, an entity extraction model layer, and a knowledge representation layer.

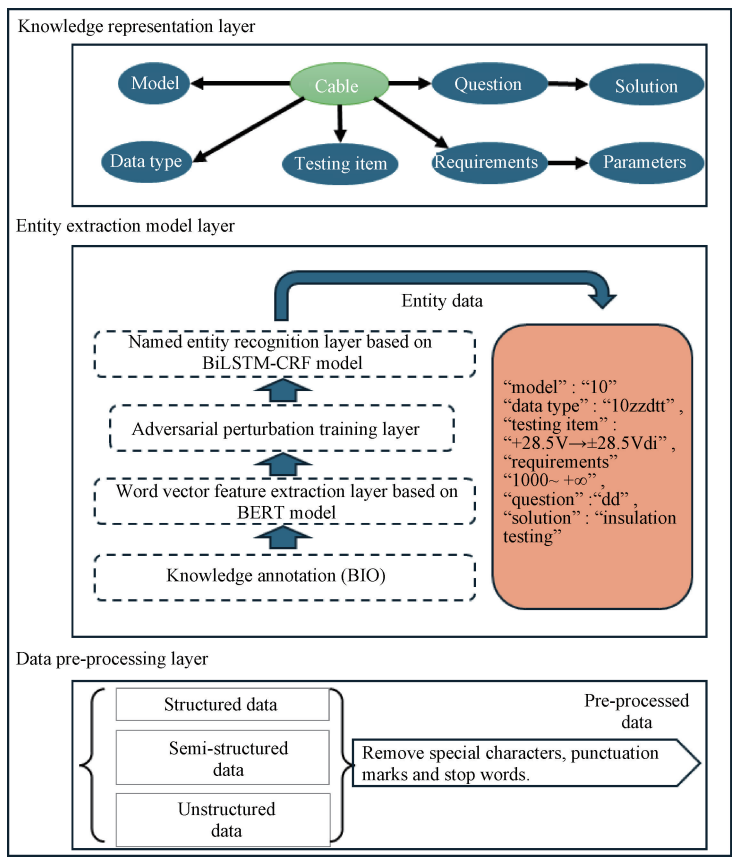
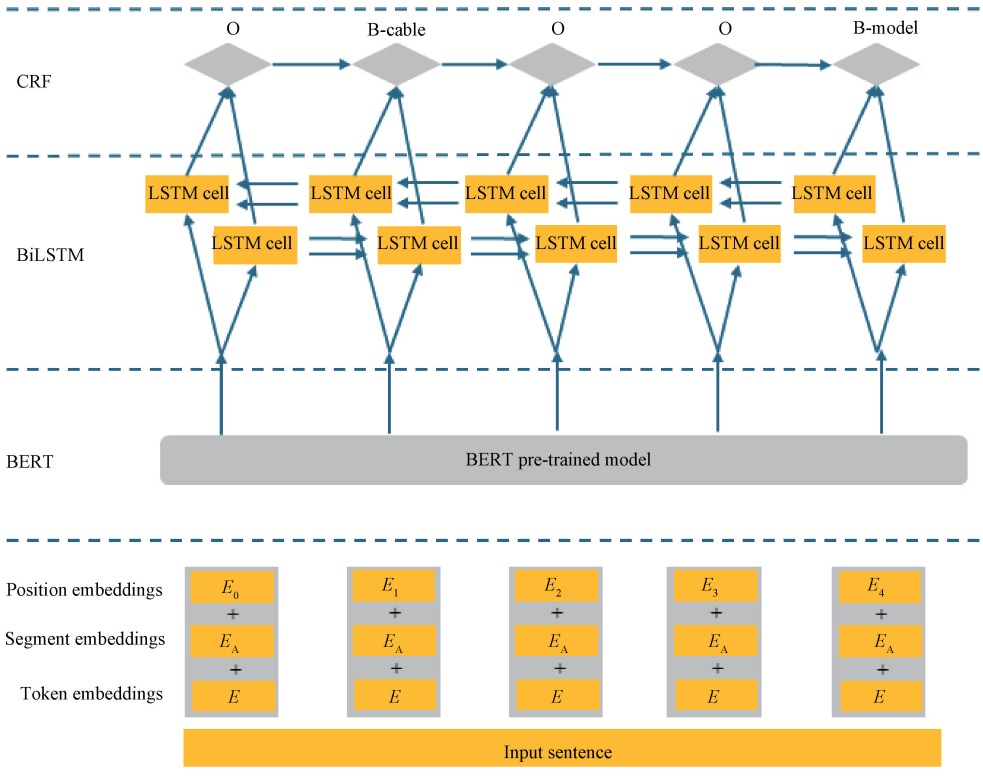


Fig. 1 Entity extraction framework

The data pre-processing layer focuses on data pre-processing, handling unstructured textual data, as well as structured and semi-structured tabular data. This task involves formatting the textural data, including its format, form and description methods. It also involves removing mixed noise from the textural data, such as stop words, numbering and other irrelevant information. Subsequently, structured tabular data are processed according to the actual needs of entity extraction from the knowledge graph, orienting the data processing towards specific entity requirements. The entity extraction model layer mainly handles the entire knowledge extraction process. It defines knowledge for problem types and completes the annotation of data using a combination of various annotation methods. Vector representation and feature extraction are carried out on the annotated data. Model learning and training are then conducted on vector-processed data to achieve the goal of automated entity extraction. The knowledge automatically extracted by the model is manually verified and optimized, integrating unstructured textual data to construct knowledge triples. By defining the relationships between entity triples, a quality knowledge network is formed. Based on this network connection, applications such as quality information inquiries can be conducted, and it can be embedded in current intelligent application systems as underlying data support for data management and application services[33].

1.2 BERT-BiLSTM-CRF named entity recognition network

The model firstly utilizes BERT to extract semantically informative word embeddings that encapsulate contextual relationships within the sequence text. These embeddings are then processed through a BiLSTM network that further captures the contextual features of the sequence text. Finally, the resulting features are input into a CRF to learn the constraints of the labels, thereby improving the recognition accuracy. The entire entity recognition process is presented in Fig. 2.



E—semantic meaning of words; E_i(i=0, 1, …, 4)—position of words within a sentence; E_A—differentiate between sentences in a sentence pair; B—beginning of an entity; O—word outside any entity.

Fig. 2 Entity recognition process

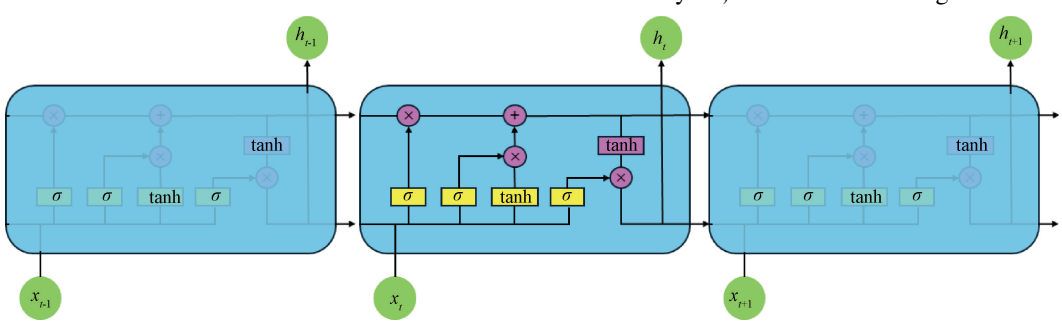
1.2.1 BERT word embedding layer

In the domain of natural language processing (NLP), words are commonly mapped into a continuous vector space to facilitate computers' understanding of natural language. This allows computers to use the distance and the similarity between vectors to capture semantic relationships. Moreover, because traditional text data are high-dimensional and sparse, word vector embeddings are used to transform them into low-dimensional and dense vectors, thereby decreasing the computational complexity and storage requirements. Traditional word vector models, such as the bag of words and methodologies based on the co-occurrence matrix, have the advantages of simplicity and intuitiveness for some straightforward NLP tasks. However, for complex tasks like entity recognition, these models are inefficient

due to the massive and cumbersome dimensions of the matrices that they require, which necessitates substantial storage space and computational resources. With the progress in artificial intelligence, neural network-based word embedding models such as word2vec and glove[34] have been widely adopted in the realm of word embeddings. Nonetheless, both the skip-gram and continuous bag of words (CBOW) algorithms of word2vec and the global word frequency statistics-based glove employ word-level embeddings, implying that each word has a unique vector. This means that they are not effective at handling the varying meanings of words in different contexts. While word2vec and glove excel at capturing local connections between words, they fall short when it comes to processing longer contextual texts, especially in capturing long-term dependencies. In contrast, the BERT model, which is based on the transformer architecture, introduces a self-attention mechanism. The representation of each word is determined collectively by the representations of itself and all other words in the context. Using a weighting mechanism, the model calculates the significance of other words in representing the current word and performs a weighted summation of their representations, thus capturing the contextual representation of the current word. This process fully harnesses contextual information, enabling the BERT model to offer different representations for polysemous words in various contexts and addressing the issue of words that have multiple meanings. Additionally, initial word vectors of the BERT model are constructed from three parts: token embeddings, segment embeddings and position embeddings. Token embeddings are used to convert input words into vector representations in a high-dimensional space. This vectorized representation captures the semantic information of each token and provides input for downstream attention mechanisms and neural network layers. Segment embeddings are used to differentiate between two separate sentences within the input sequence. The BERT model can accept either a single sentence or a pair of sentences as input, and each token in the sentences is appended with a segment embedding that indicates which sentence the token belongs to. This plays a pivotal role in guiding the model predictions about the relationships of tokens across different sentences. Because the transformer architecture in the BERT model does not intrinsically grasp the positional relationships of words in a sequence like what recurrent neural networks do, the BERT model requires position embeddings to capture the sequence order of words. This allows the BERT model to understand token positions within the sequence, helping to capture distances and dependencies, which is crucial for understanding language properties like the syntactic structure. By integrating these three types of embeddings, the BERT model can concretize each token into a vector rich in information. Together, this information provides the necessary input for self-attention layers of the BERT model, enabling the BERT model to perform efficient and precise language understanding and prediction[35-36].

1.2.2 BiLSTM feature extraction layer

Building upon the context-aware word representations provided by the BERT model, the BiLSTM model[37] further enhances the capture of sequential information and long-distance dependencies inherent in textual data. It comprises two LSTM layers, one for processing forward information (forward LSTM) and the other for handling backward information (backward LSTM). This arrangement allows for better contextual information learning and effectively avoids the problem of gradient vanishing associated with recurrent neural networks. The BiLSTM model replaces the hidden layer units of the bidirectional recurrent neural network with LSTM neural units. These units include four neural network layers, as illustrated in Fig. 3.



σ—activation function; h_{t-1}—hidden state at previous time step; h_t—hidden state at current time step; h_{t+1}—hidden state at next time step; x_{t-1}—input at previous time step; x_{t-1}—input at current time step; x_{t+1}—input at next time step.

Fig. 3 Specific structure of LSTM neural unit

Each LSTM unit receives the input at the current time step, and the hidden state and the cell state from the previous time step, and then outputs the hidden state and the cell state at the current time step. The input, output and update formulas are as follows.

1) Input gate. The input gate determines how much of the current information should be added to the memory cell.

$$i_t = \sigma(\boldsymbol{W}_i[h_{t-1}, x_t] + b_i), \quad (1)$$

where i_t is the the input of the input gate; W_i and b_i are the weight and bias of the input gate, respectively.

2) Forget gate. The forget gate determines how much of the information from the previous state should be retained in the memory cell.

$$f_t = \sigma(\boldsymbol{W}_f[h_{t-1}, x_t] + b_f), \tag{2}$$

where f_t is the output of the forget gate; $\boldsymbol{W}_f$ and b_f are the weight and bias of the forget gate, respectively.

3) Candidate memory cell. New candidate memory content is generated to be partially added to the current candidate memory cell.

$$\widetilde{C}_t = \tanh(\boldsymbol{W}_c[h_{t-1}, x_t] + b_c), \tag{3}$$

where $\boldsymbol{W}_c$ and b_c are the weight and bias, respectively; $\widetilde{C}_t$ is the output of this memory cell.

4) Cell state update. The current memory cell combines the filtered previous memory and the filtered candidate memory.

$$C_t = f_t C_{t-1} + i_t \widetilde{C}_t, \tag{4}$$

where C_t and C_{t-1} are the state of the current memory cell and the state of the previous memory cell, respectively.

5) Output gate. The output gate determines how much of the current memory cell state is used to produce the hidden state.

$$o_t = \sigma(\boldsymbol{W}_o[h_{t-1}, x_t] + b_o), \tag{5}$$

where o_t is the output gate result; $\boldsymbol{W}_o$ and b_o are the weight and bias, respectively.

6) Hidden state update. The hidden state at the current time step is the result of the current memory cell state processed by the tanh activation function and then filtered by the output gate.

$$h_t = o_t \tanh(C_t). \tag{6}$$

To further illustrate the working principles of BERT-BiLSTM model, Table 1 shows its specific workflows.

Table 1 Pseudocode flow of BERT-BiLSTM model for sequence labeling task

Step	Explanation
1	Tokenization: tokens = ["A", "type", "cable"]
2	Add special tokens: bert_input = ["[CLS]"] + tokens + ["[SEP]"]
3	BERT encoding: bert_output = BERT(bert_input) #BERT output: [CLS_representation, rep_A, rep_type, rep_cable, SEP_representation]
4	Remove special tokens: bert_output_processed = bert_output[1:-1] # Processed BERT output: [rep_A, rep_type, rep_cable]
5	BiLSTM: bilstm_output = BiLSTM(bert_output_processed) # BiLSTM output: [BiLSTM_rep_A, BiLSTM_rep_type, BiLSTM_rep_cable]
6	Fully connected layer: dense_output = DenseLayer(bilstm_output) # Fully connected layer output: [Dense_rep_A, Dense_rep_type, Dense_rep_cable]
7	Predicted labels: predicted_labels = [] for token_rep in dense_output: label = Classifier(token_rep) predicted_labels.append(label) # Predicted labels: [B-CABLE, I-CABLE, I-CABLE]

Take the phrase "A type cable" as an example. First, the BERT model adds special tokens [CLS] and [SEP] to the beginning and the end of the input sentence, respectively. Then, it encodes the entire sentence to generate a 768-dimensional vector for each word, representing its contextual information. The output from the BERT layer serves as the input of the BiLSTM layer, capturing the forward and backward dependencies of each word in the sequence. For instance, the output of the BiLSTM layer may be a 128-dimensional vector containing both forward and backward information. Finally, a fully connected layer is used to predict labels for each word.

1.2.3 CRF constraint layer

CRF is a probabilistic graphical model used for sequence labeling tasks. It predicts the optimal sequence of labels by considering contextual information across the entire sequence. The specific workflow of the BiLSTM-CRF model is outlined in Table 2.

Table 2 Pseudocode flow of BiLSTM-CRF model

Step	Explanation
1	input_sentence = ["A", "type", "cable"]
2	embedding_output = bert_layer(input_sentence)
3	bilstm_output = bilstm_layer(embedding_output)
4	Mapping the output of BiLSTM to scores for each label: dense_output = dense_layer(bilstm_output)
5	The CRF layer computes the score of label sequences and the optimal label sequence: crf_output = crf_layer(dense_output)
6	predicted_labels = crf_output

The CRF model first uses the BERT model to embed natural language input into word embeddings, converting them into embedding vectors. These vectors are processed through a BiLSTM layer to extract features, and then input into a fully connected layer for label prediction. For example, the score for the label corresponding to "A" might be [B-Cable: 2.5, I-Test: 1.0, B-STATUS: 0.5, ⋯]. In this case, "I" denotes the inside of an entity. The results are then fed into the CRF layer to compute the optimal path of the entire label sequence. The specific workflow of the CRF layer is as follows.

1) Define feature functions. State feature functions $f_i(y_i, X, i)$ capture the relationship between the ith element of the input sequence X and its corresponding label y_i. Transition feature functions $f_{i,j}(y_{i-1}, y_i, X, i)$ capture the relationship between adjacent labels y_{i-1} and y_i in the label sequence.

2) Score computation. For a given input sequence X and a given label sequence Y, the CRF model calculates the total score $S(X, Y)$ by using feature functions and weights.

$$S(X, Y) = \sum_i \sum_k w_k f_k(y_{i-1}, y_i, X, i), \quad (7)$$

where f_k represents the kth feature function; w_k denotes its weight parameter, indicating the impact of that feature on the overall model.

3) Partition function. It computes the exponential sum of scores for all possible label sequences, known as the normalization factor $Z(X)$.

$$Z(X) = \sum_Y \exp(S(X, Y)). \quad (8)$$

4) Conditional probability. It computes the conditional probability $P(Y \mid X)$ of the given input sequence X and the given label sequence Y based on the normalization factor.

$$P(Y \mid X) = \frac{\exp(S(X, Y))}{Z(X)}. \quad (9)$$

5) To find the optimal feature weights w, the log-likelihood function of the training data is maximized. Finally, use the Viterbi algorithm to find the optimal label sequence Y that maximizes the conditional probability $P(Y \mid X)$.

$$\log P(Y \mid X) = S(X, Y) - \log Z(X). \quad (10)$$

2 Case Analysis

2.1 Standard dataset

The dataset used in this experiment is OntoNotes Release 5.0 which is an integrated dataset created by BBN Technologies, the University of Colorado, the University of Pennsylvania and the USC Information Sciences Institute. This dataset contains annotated corpora of various text types covering three languages (Arabic, English and Chinese). It includes structural information (such as syntax and predicate argument structures) and shallow semantics (such as word senses linked to ontologies and coreference). The textual data consists of annotations for up to 2.9 million words, with specific quantities as shown in Table 3.

Table 3 Dataset content

Text type	Word number		
	Arabic	English	Chinese
News	300 000	625 000	250 000
Broadcast news	—	200 000	250 000
Broadcast conversation	—	200 000	150 000
Web text	—	300 000	150 000
Telephone conversation transcript	—	120 000	100 000
Pivot	—	—	300 000

2.2 Experimental results

To validate the efficiency of performance prediction, the proposed BERT-BiLSTM-CRF model was compared with state-of-the-art performance methods, such as BiLSTM-Softmax, BiLSTM-CRF and word2vec-BiLSTM-CRF. The parameter settings of the BERT-BiLSTM-CRF model presented in this paper are shown in Table 4, and the comparison results can be seen in Fig. 4. In Fig. 4, P stands for precision, R stands for recall, and F_1 stands for the F1 score:

$$P = \frac{N_{TP}}{N_{TP} + N_{FP}}, \quad (11)$$

$$R = \frac{N_{TP}}{N_{TP} + N_{FN}}, \quad (12)$$

$$F_1 = \frac{2PR}{P + R}, \quad (13)$$

where N_{TP} refers to the number of true positive instances correctly identified; N_{FP} refers to the number of false positive instances incorrectly identified as positive; N_{FN}

refers to the number of false negative instances incorrectly identified as negative.

Table 4 Main BERT-BiLSTM-CRF model parameters

Parameter	Description	Value
num_layers	Number of layers	15
num_units	Number of units	256
l2_rate	Fully connected layer weights	0.1
max_patience	Maximum patience	20
Clip	Gradient clipping	5
batch_size	Number of samples	64
learning_rate	Learning rate	0.002
dropout_rate	Prevention of model overfitting	0.5
nb_epoch	Maximum iteration count	200

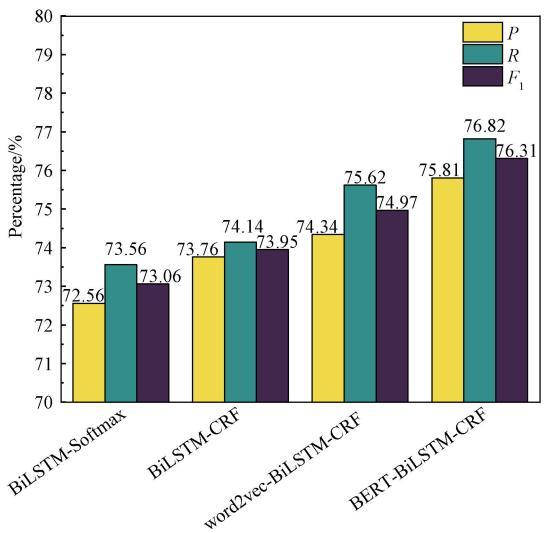


Fig. 4 Comparative experimental results

It is evident that the BERT-BiLSTM-CRF model achieves the highest scores across all metrics, including precision, recall, and the F1 score. Specifically, compared to the BiLSTM-Softmax model, the F1 score of the BiLSTM-CRF model increases from 73.06% to 73.95%. This improvement can be attributed to the fact that in the BiLSTM model, the output prediction results are completely independent of each other. Each step selects the label with the maximum probability, often resulting in consecutive identical labels like B-Person followed by another B-Person, which is incorrect. In this case, the CRF model imposes a constraint. Since the CRF model considers the sequential nature of labels, it effectively prevents illogical label sequences, resulting in better performance. Furthermore, when compared to the word2vec-BiLSTM-CRF model, the F1 score of the BERT-BiLSTM-CRF model increases from 74.97% to 76.31%. This is because, in the word2vec embedding model, each word is fixed to a unique vector and does not effectively consider the different meanings of the same word in varying contexts. However, the self-attention mechanism in the BERT model addresses this issue effectively. Additionally, when comparing the word2vec-BiLSTM-CRF and BERT-BiLSTM-CRF models to the BiLSTM-Softmax and BiLSTM-CRF models, it is clear that adding a word embedding layer before the BiLSTM layer significantly enhances the model's recognition accuracy. This is because both word2vec and BERT models can initially capture the semantic context of the text, enabling them to better handle polysemy, contextual information and the relationships between labels.

To further demonstrate the effectiveness of the BERT-BiLSTM-CRF model, the experiment compares the results after dissecting the model, with the results shown in Fig. 5.

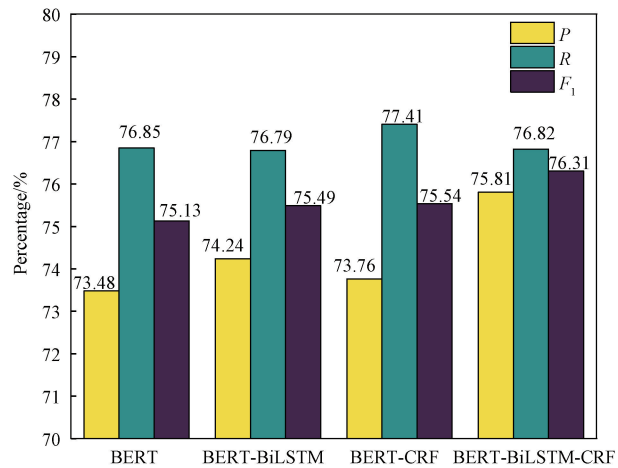


Fig. 5 Ablation experiment results

The BERT-BiLSTM model shows a minor improvement in the F1 score, but there is a decline in recall. This outcome occurs because the BERT model itself is capable of capturing contextual features and can adequately perform feature extraction for sequence tasks. Adding the BiLSTM network might provide redundant features in certain situations, increasing model complexity without contributing additional effective information, leading to over-interpretation of the text. The BERT-CRF model shows improvements across all three metrics compared to the BERT model. This is because the CRF model can be viewed as a safeguard that only eliminates incorrect answers and does not interfere with the BERT model. The comparison reveals that the BERT-BiLSTM model and the BERT-CRF model individually do not significantly enhance the BERT model and may even affect it, but the combination of BiLSTM with CRF shows a more noticeable improvement. This is because the CRF layer alone does not provide additional feature information, and it is simply another form of a decoder. BiLSTM refines features first, and then CRF performs conditional global optimal decoding, making the combination more effective than using each individually due to their complementary strengths.

2.3 Application validation

The constructed entity recognition and extraction model was applied to the aerospace cable production inspection task. The extracted entities were input as the data layer into the knowledge graph framework to build a knowledge graph model for the cable production testing domain. This allowed for the rapid identification of problem points during the production testing process, thereby improving the efficiency and the quality of cable production. With the aforementioned BERT-BiLSTM-CRF model as technical support, aerospace product fault tracking report data were selected as training and test samples. After data pre-processing to eliminate and clean irregular samples, a total of 866 fault data samples were selected. The method of corpus tagging used in the data pre-processing process was combined with the work experience of frontline workers to assist in data annotation, maximizing the quality of the tagged data. A total of nearly 8 976 sequences of data labeled with BIO were collected. Adhering to the Pareto principle in model training, 20% of the data was used as the test dataset, with approximately 1 795 BIO-labeled sequences, and the remaining 80% were used as the training dataset, with approximately 7 181 BIO-labeled sequences. The annotated entity data used in the experiments are shown in Table 5. The data are sourced from on-site inspection records of a specific cable model from a certain enterprise. The identified entities were imported into the Neo4j knowledge graph framework to produce the knowledge graph for the cable production testing domain. The display result of the knowledge graph for the cable continuity test section is shown in Fig. 6, with the specific parameters encrypted.

Table 5 Annotated entity data

Token	Lable	Token	Lable
A	B-CABLE	B	B-CABLE
type	I-CABLE	type	I-CABLE
cable	I-CABLE	cable	I-CABLE
during	O	during	O
length	B-TEST	length	B-TEST
measurement	O	measurement	O
process	O	process	O
result	B-RESULT	result	B-RESULT
overlength	B-STATUS	normal	B-STATUS
conductive	B-CONDUCTIVE	conductive	B-CONDUCTIVE
test	I-CONDUCTIVE	test	I-CONDUCTIVE
voltage withstand	B-TESTITEM	capacitance	B-TESTITEM
test	I-TESTITEM	test	I-TESTITEM

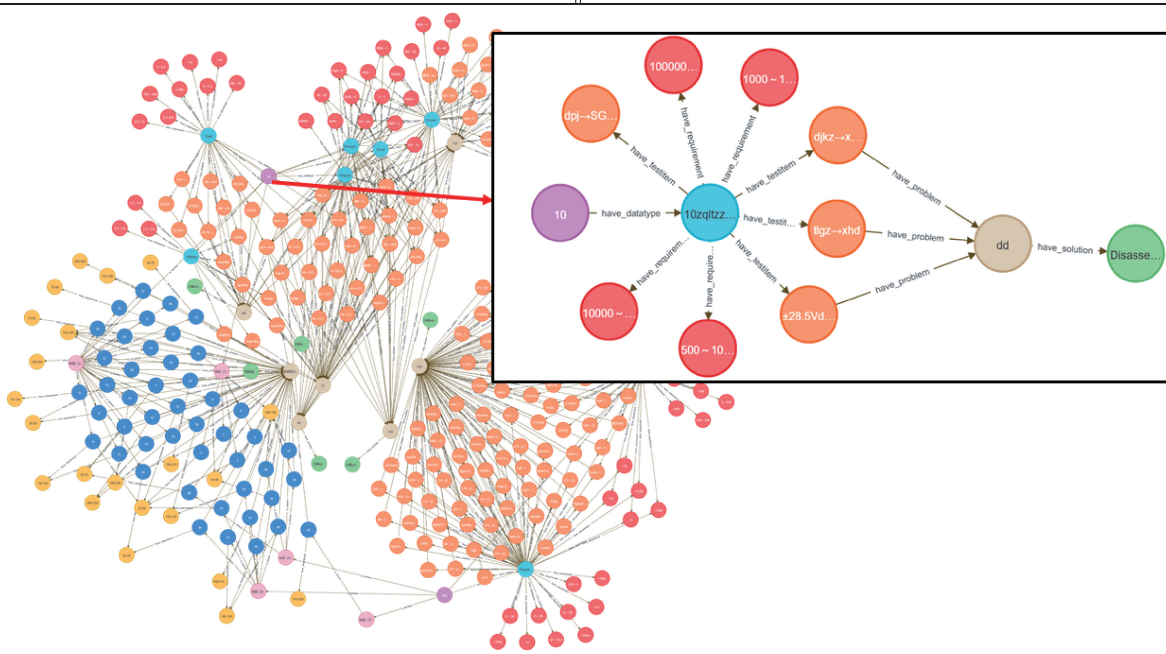


Fig. 6 Knowledge graph for cable continuity test section

Based on the established knowledge graph database, it is possible to visualize queries of data generated during the aerospace cable production testing process, allowing for rapid tracing and pinpointing of issues. Taking the cable continuity test scenario as an example, the specific implementation process is illustrated in Fig. 7.

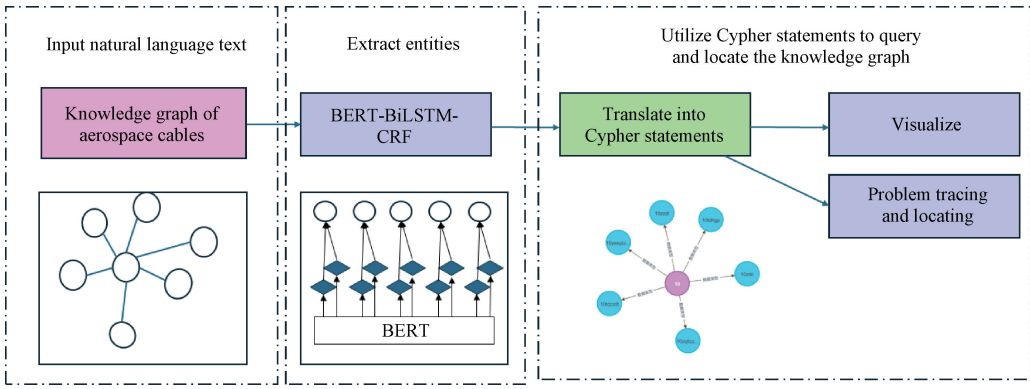


Fig. 7 Visualization and traceability process

Users can input natural language text into the system for querying. For example, by entering "What types of data are there for cable model 10?", the entered natural language is processed using the trained BERT-BiLSTM-CRF model discussed earlier to identify and extract entities. These entities are then translated into the Cypher query language for the Neo4j graph database, for example the above statement being converted to "MATCH (c: CableModel {name: '10'}) -[: DataType]->(q: DataType)". This query is then executed in the knowledge graph to return results. When it is necessary to trace issues related to the quality of aerospace cables, corresponding query statements can be directly input, such as "What issues could arise with cable model 16B during the +28.5V→±28.5Vdi test, and what are the solutions?" In this case, it would be converted to "MATCH (c: CableModel {name: '16B'}) -[: TestItem]->(q: TestItem) -[: CorrespondingIssue]->(p: Issue)-[: Solution]->(s: Solution)", and the results would be returned and outputted using Neo4j.

To validate the speed and the effectiveness of the knowledge graph-based quality traceability system proposed in this paper, the above method was compared with the traditional manual inspection method. Traditional manual inspection involves workers conducting continuity tests on the cables under test, followed by manually comparing the test results with specified values to identify quality issues. The results of 500 cable tests were selected and divided into 100 batches as experimental data. Accuracy and time for final quality issue localization were used as metrics for comparative experiments. The experimental results are shown in Fig. 8. It can be seen that in terms of accuracy in localizing quality issues, the proposed method is comparable to the traditional manual inspection method. However, the proposed method improves the detection speed by 75.4% compared to the traditional manual inspection method, demonstrating the reliability of the above method.

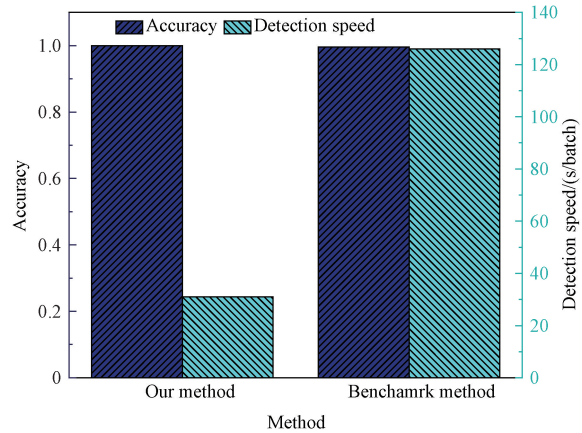


Fig. 8 Application performance comparison

3 Summary and Outlook

This paper addresses the challenge of managing complex and heterogeneous data during aerospace cable production testing which often necessitates extensive human effort to trace and resolve quality issues. Leveraging a BiLSTM-CRF network as the foundation, augmented with a BERT network for high-dimensional vector representation of input sentences, the research employs multiple embeddings to dynamically represent natural language inputs. This approach, compared to conventional static embedding networks, enhances the model ability to learn contextual features. This paper introduces an entity recognition model tailored for the cable testing domain, integrated into knowledge graph modeling. Initially, a Python program was developed to integrate and transform structured and semi-structured data. For non-textual data, a BERT-BiLSTM-CRF entity recognition model was applied to

identify and extract entities specific to aerospace cable testing. The information was stored in the Neo4j graph database, facilitating the construction of a knowledge graph based method for aerospace cable production testing scenarios.

In the future, this knowledge graph based method aims to support the development of a question-answering system for aerospace cable production testing, enabling enhanced problem tracing and resolution capabilities through direct interaction using natural language.

References

[1] WANG L W, WANG Y L, WANG W, et al. Experimental study on anti-disturbance technology of tail harness of aerospace cable network [J]. *Aerospace Manufacturing Technology*, 2021(4): 1-6. (in Chinese)

[2] ZHANG Y Y. Research on cables optimization and standard verification technology [D]. Beijing: Beijing University of Technology, 2018. (in Chinese)

[3] JING W T. The research of coupling effects of wire inside a missile to HEMP[D]. Changsha: National University of Defense Technology, 2005. (in Chinese)

[4] LIU C Z. Research of aircraft wiring networks fault location based on waveform matching[D]. Tianjin: Civil Aviation University of China, 2017. (in Chinese)

[5] HEDBERG T D Jr, BAJAJ M, CAMELIO J A. Using graphs to link data across the product lifecycle for enabling smart manufacturing digital threads [J]. *Journal of Computing and Information Science in Engineering*, 2020, 20(1): 011011.

[6] DOMBROWSKI U, REISWICH A, IMDAHL C. Knowledge graphs for an automated information provision in the factory planning [C]//Proceedings of 2019 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM). Washington, D. C., USA: IEEE, 2019.

[7] REN T. Research and application of OPC UA information model based on knowledge graph [D]. Hangzhou: Zhejiang University, 2021. (in Chinese)

[8] GRANGEL-GONZÁLEZ I, HALILAJ L, VIDAL M E, et al. Knowledge graphs for semantically integrating cyber-physical systems [M]//Lecture Notes in Computer Science. Cham: Springer International Publishing, 2018: 184-199.

[9] LIU X. Knowledge representation combining case-based reasoning with knowledge graphs for stamping die design [D]. Dalian: Dalian University of Technology, 2019. (in Chinese)

[10] LI X L, ZHANG S S, HUANG R, et al. Structural modeling of heterogeneous CAM model based on process knowledge graph[J]. *Journal of Computer-Aided Design & Computer Graphics*, 2018, 30(7): 1342-1355. (in Chinese)

[11] LIN C. Research and application of design intention reasoning method based on STEP knowledge graph [D]. Hangzhou: Zhejiang University of Technology, 2019. (in Chinese)

[12] HUET A, SEGONDS F, PINQUIE R, et al. Context-aware cognitive design assistant: implementation and study of design rules recommendations [J]. *Advanced Engineering Informatics*, 2021, 50: 101419.

[13] LIU Y Y. Design and development of CNC machining information recommendation system based on knowledge graph[D]. Beijing: Beijing University of Posts and Telecommunications, 2021. (in Chinese)

[14] DUAN Y. Building and application of a metal cutting process knowledge graph[D]. Chengdu: Sichuan University, 2021. (in Chinese)

[15] DUAN Y, HOU L, LENG S. A novel cutting tool selection approach based on a metal cutting process knowledge graph[J]. *The International Journal of Advanced Manufacturing Technology*, 2021, 112(11): 3201-3214.

[16] LI R Q, DAI W B, HE S, et al. A knowledge graph framework for software-defined industrial cyber-physical systems [C]//Proceedings of the 45th Annual Conference of the IEEE Industrial Electronics Society (IECON 2019). Washington, D. C., USA: IEEE, 2019.

[17] HAO X. Research on prediction and processing method of workshop production abnormality based on LSTM[D]. Harbin: Harbin Institute of Technology, 2020. (in Chinese)

[18] ING H. Knowledge graph construction for product quality analysis of machining lines[D]. Wuhan: Huazhong University of Science and Technology, 2020. (in Chinese)

[19] JI F B. Research on the construction method of knowledge graph for parameter optimization of manufacturing production process [D]. Jinan: Qilu University of Technology (Shandong Academy of Science), 2021. (in Chinese)

[20] LIU X, SHI Y Q, LUO X, et al. Expert knowledge-based apparel recommendation question and answer system [J]. *Journal of Donghua University (English Edition)*, 2022, 39(1): 55-64.

[21] YANG N. Research on key technologies of

turbine intelligent diagnosis and health management[D]. Beijing: North China Electric Power University, 2020. (in Chinese)

[22] LIU X. Research on knowledge graph construction technology for fault analysis[D]. Beijing: Beijing University of Posts and Telecommunications, 2019. (in Chinese)

[23] LIU H Y, MA R Z, LI D Y, et al. Machinery fault diagnosis based on deep learning for time series analysis and knowledge graphs [J]. *Journal of Sign Process System*, 2021, 16(93): 1433-1455.

[24] LIU B, LIU Y, ZHENG X H, et al. Exploring techniques for building language models targeted at sewing equipment operation and maintenance management [J]. *Journal of Donghua University (English Edition)*, 2024, 41(3): 315-322.

[25] ZHANG Q Q, CHEN M D, LIU L Z. A review on entity relation extraction[C]//2017 Second International Conference on Mechanical, Control and Computer Engineering (ICMCCE). New York, USA: IEEE, 2017.

[26] YU H K, ZHANG H P, LIU Q, et al. Chinese named entity identification using cascaded hidden Markov model[J]. *Journal on Communications*, 2006(2): 87-94. (in Chinese)

[27] LI R L, WANG J H, CHEN X Y, et al. Using maximum entropy model for Chinese text categorization[J]. *Journal of Computer Research and Development*, 2005 (1): 94-101. (in Chinese)

[28] HE Y X, LUO C W, HU B R. Geographic entity recognition method based on CRF model and rules combination [J]. *Computer Applications and Software*, 2015, 32(1): 179-185, 202.

[29] HAMMERTON J. Named entity recognition with long short-term memory[C]//Proceedings of the seventh conference on natural language learning at HLT-NAACL 2003. Morristown, NJ, USA: Association for Computational Linguistics, 2003.

[30] PINHEIRO P O, COLLOBERT R. Recurrent convolutional neural networks for scene parsing [C]//Proceedings of the 2014 European Conference on Computer Vision. Zurich, Switzerland: Springer, 2014.

[31] LAMPLE G, BALLESTEROS M, SUBRAMANIAN S, et al. Neural architectures for named entity recognition[C]//Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Stroudsburg, PA, USA: Association for Computational Linguistics, 2016.

[32] CHIU J P C, NICHOLS E. Named entity recognition with bidirectional LSTM-CNNs[J]. *Transactions of the Association for Computational Linguistics*, 2016, 4: 357-370.

[33] QIN Y, SHEN G W, ZHAO W B, et al. A network security entity recognition method based on feature template and CNN-BiLSTM-CRF[J]. *Frontiers of Information Technology & Electronic Engineering*, 2019, 20(6): 872-884.

[34] CHU D P, WAN B, LI H, et al. Geological entity recognition based on ELMO-CNN-BiLSTM-CRF model[J]. *Earth Science*, 2021, 46(8): 3039-3048.

[35] LUO L, YANG Z H, YANG P, et al. An attention-based BiLSTM-CRF approach to document-level chemical named entity recognition [J]. *Bioinformatics*, 2018, 34(8): 1381-1388.

[36] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach, CA, USA: Curran Associates Inc., 2017.

[37] DEVLIN J, CHANG M W, LEE K, et al. BERT: pre-training of deep bidirectional transformers for language understanding [C]// Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Stroudsburg, PA, USA: Association for Computational Linguistics, 2019.

基于知识图谱的航天产品质量追溯方法

王　宁[1], 曹立俊[1], 丁司懿[2], 孟　岩[2], 刘　欢[1], 郑小虎[2], 黄文斌[1], 刘骁佳[1*]
1. 上海航天精密机械研究所, 上海 201600
2. 东华大学 信息科学与技术学院, 上海 201620

摘　要: 如今航天器内部结构功能越来越复杂。电缆作为其“生命线”, 在人工检测过程中需要耗费大量人力物力, 如何快速定位质量问题并找出解决方案仍然具有挑战性。针对此问题, 该文采用基于知识图谱的方法来抽取多源异构的电缆知识实体, 该模型通过基于 transformer 的双向编码表征（bidirectional encoder representations from transformers, BERT）网络对输入文本进行词向量嵌入, 然后通过双向长短时记忆（bidirectional long short-term memory, BiLSTM）网络对输入序列的上下文特征进行提取, 最终输入随机条件场（conditional random field, CRF）网络中预测实体类别, 同时以此模型提取出的实体作为数据层, 构建知识图谱。所搭建的实体抽取模型的准确率、召回率、F1 值等指标均比常见模型有所提升。最终以某航天精密机械加工所的线缆测试数据构建了该领域的知识图谱模型, 实现了可视化查询及对质量问题的追溯定位。

关键词: 知识图谱; 命名实体识别; 质量控制; 航天产品