IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 29 October 2024, accepted 6 December 2024, date of publication 12 December 2024, date of current version 23 December 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3516390

RESEARCH ARTICLE

Named Entity Recognition in Aviation Products Domain Based on BERT

MINGYE YANG, BERNADIN NAMOANO, MARYAM FARSI, AND JOHN AHMET ERKOYUNCU
Centre for Digital Engineering and Manufacturing, Cranfield University, MK43 0AL Cranfield, U.K.

Corresponding author: Mingye Yang (mingye.yang.381@cranfield.ac.uk)

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/Z533221/1.

ABSTRACT The aviation products' manufacturing industry is undergoing a profound transformation towards intelligence, among which the construction of a knowledge graph specifically for the aviation field has become the core link in achieving cognitive intelligence. In the process of knowledge graph construction, named entity recognition (NER) is a key step and one of the main tasks of knowledge extraction. Given the high degree of specialisation of aviation product text data and the wide span of contextual information, existing models often perform poorly in entity extraction. This paper proposes a new Named Entity Recognition (NER) method specifically tailored for the aviation product field (BBC-Ap), introducing an innovative approach that leverages domain-specific ontologies and advanced deep learning algorithms to significantly enhance the accuracy and efficiency of entity extraction from complex technical documents. The first step of this method is to establish an ontology model of aviation products and annotate the relevant text data to form a dataset for training the named entity model. Next, it adopts a multi-level model structure based on BERT, in which BERT is used to generate word vector representations, a bidirectional long short-term memory network (BiLSTM) is used as an encoder to extract semantic features, and a conditional random field (CRF) is used as a decoder to achieve optimal label assignment. Through experiments on the constructed aviation product dataset, the model achieved a Precision value of 91.74%, a Recall value of 92.46%, and an F1 score of 92.1%, Compared with other baseline models, the F1-score is improved by 0.9% to 1.5%. At the same time, the model also performs well on standard datasets such as CoNLLpp, with a Precision value of 92.87%, a Recall value of 92.54%, and an F1-Score of 92.70%. Finally, the model was used to successfully construct a knowledge graph reflecting the relationships between aviation products in Neo4j, further demonstrating the effectiveness and practicality of the method.

INDEX TERMS Aviation, named entity recognition (NER), knowledge graph, bidirectional encoder representations from transformers (BERT), bidirectional long short-term memory network (Bi-LSTM).

I. INTRODUCTION

With the advent of the fourth industrial revolution – Industry 4.0, the aviation product manufacturing industry is undergoing a profound transformation towards intelligence, digitization, and efficiency. Knowledge graphs have received widespread attention in the field of intelligent manufacturing in recent years. They can autonomously build a huge ''knowledge'' network and have powerful semantic expression, knowledge storage, and reasoning capabilities. In the aviation product manufacturing industry, building a domain knowledge graph has become a key link in realizing cognitive intelligence [1], [2]. In the process of building a knowledge graph, the extraction of aviation product entities is crucial. Aviation products' manufacturing companies have accumulated a large amount of data in the design and manufacturing process, which contains rich product information. Due to the complexity of the data, manual extraction is difficult and prone to errors and cannot meet the needs of efficient extraction and integration of large-scale data. At the same time, the storage format of historical product parameter data is not uniform, and most of them are stored in the form of

The associate editor coordinating the review of this manuscript and approving it for publication was Rosario Pecora.

text and data tables. Therefore, how to effectively extract this information and convert it into usable knowledge has become a powerful challenge [3].

Knowledge extraction technology can extract structured ternary knowledge from a large amount of unstructured text data, construct a knowledge graph database by extracting entities and relationships between products, and represent the relationships between product entities with node sets and relationship sets between nodes to provide richer aviation product information [4]. Knowledge extraction mainly includes entity extraction, relationship extraction and attribute extraction. Entity extraction is also called Named Entity Recognition (NER), which is currently divided into three types: rule-based and dictionary-based methods, machine learning-based methods, and deep learning-based methods. Among these, the deep learning-based method has a more effective fitting effect on nonlinear problems and can extract more complex features than traditional machine learning. It has become a method of named entity recognition [5], [6], [7].

Early named entity recognition mainly used rule-based and dictionary-based methods, where experts manually constructed rule templates, and the selected features included statistical information, punctuation, keywords, etc. This method has a long system construction cycle and requires the establishment of knowledge bases in different fields as an auxiliary. At the same time, the process of formulating rules is time-consuming, and it is difficult to cover all languages [8], [9]. Machine learning-based methods mainly use models such as hidden Markov models, maximum entropy models, support vector machines, and conditional random fields. The named entity recognition task is regarded as a sequence labeling problem, and a large amount of corpus is provided to the labeling model for learning to label each position of the sentence [10]. Zhang et al. [10] summarized the advantages and disadvantages of seven machine learning methods and compared their performance on two standard biological corpora (GENIA and JNLPBA). The results showed that the CRF model outperformed other methods on both biological corpora. Zhou and Su [11] proposed a hidden Markov model (HMM) implemented as the PowerNE system. The data sparsity problem is solved by HMM and an effective constraint relaxation algorithm. PowerNE can effectively apply and integrate various internal and external evidence of entity names. Evaluation results show that using the formal training and test data of the MUC-6 and MUC-7 English-named entity tasks, PowerNE achieves F values of 96.6 and 94.1 respectively.

In recent years, with the widespread application of neural network models in various fields, more and more NER research has begun to adopt deep learning methods. The main advantage of such NER methods is that they can use the powerful learning ability of neural networks to model and solve complex natural language processing tasks [12]. At the same time, due to the high scalability of deep learning models, these methods can also be used to process large-scale text data [13]. Xu et al. [14] developed a knowledge extraction model based on advanced NLP technology, which is designed for the field of coal mine construction. The model uses a combination of BERT's bidirectional encoder representation, bidirectional long short-term memory network (BiLSTM) and conditional random field (CRF). The dynamic word vector is obtained through the BERT pre-trained language model, which enhances the processing ability of complex semantic information. Subsequently, using the BiLSTM-CRF model, the model can effectively identify and extract key entities in the text, showing good performance. Jiang et al. [15] developed a new named entity recognition model using the ALBERT-BiLSTM-CRF architecture, which is designed specifically for power equipment defect text data, effectively supporting the power industry to extract key information from unstructured text and optimize equipment management and maintenance decisions. The model uses lightweight ALBERT instead of traditional BERT, reducing complexity and training time while combining BiLSTM and CRF to improve the recognition accuracy and sequence labeling performance of entities in specific fields. Burke et al. [16] developed NukeLM, a language model customized for the nuclear energy field. The model was pre-trained on approximately 1.5 million nuclear energy-related abstracts from the U.S. Department of Energy's Office of Scientific and Technical Information database. By fine-tuning the model, NukeLM can effectively perform binary classification tasks related to the nuclear fuel cycle and multi-class classification tasks based on research topics. Using a BERT-like architecture, NukeLM shows excellent performance on both text classification tasks. Yang et al. [17] designed the Chinese-named entity recognition model BERT-BiLSTMIDCNN-ELU-CRF (BBIEC) to address the problems in the field of COVID-19 information epidemiological surveys. The model achieved good results on the COVID-19 dataset and epidemiological survey information at multiple entity levels. Wei et al. proposed an attention-based BiLSTM-CRF model which achieved reasonable performance on the JNLPBA corpus with an F1 score of 73.50. The model can enhance the ability of neural networks to extract important information and does not rely on any feature engineering, only requiring general pre-trained word vectors. This makes the model highly portable and scalable.

The current research on deep learning-based NER is mainly concentrated in the medical field and general field such as the social media field, and there are still many inadequacies in the study of aviation [18], [19], [20]. The main challenges faced by developing NER methods in the aviation field are: (1) Few publicly acceptable data sets: Currently, the aviation field suffers from a scarcity of datasets readily applicable for named entity recognition tasks, necessitating the manual collection and annotation of data to fulfill research requirements. (2) The vocabulary of entities in aviation is highly specialized, and many entities are specific nouns, such

as aircraft models, engine components, flight control systems, etc. These proper nouns are often composed of multiple words or abbreviations and are significantly different from the named entities of the generic domain. For example, recognition of terms such as ''Boeing 737 MAX'' and ''CFM56 engine'' does not perform well in existing general-purpose NER models because these models are often pre-trained on a wider range of textual data and lack the ability to effectively recognize unique terms in aviation. More importantly, these proper nouns require not only proper word segmentation, but also the ability to recognize the domain-specific semantic relationships they contain. In the face of these complex proprietary entities, the existing general model has low recognition accuracy, especially when dealing with long entities or multi-word entities, which is prone to misidentification or loss of important information. (3) Lack of standardized and complete ontology classification method: General entity classification such as ''place'', ''name'', ''country'', etc. is not applicable to aviation product texts. It is necessary to establish a reasonable ontology classification method based on the actual classification of aviation products. For example, in the field of aviation, a more detailed division of entities may be required, such as ''aircraft model'', ''part number'', ''aviation standard'', ''operating procedure'', etc. Without a reasonable and structured ontology classification method, the NER system may not accurately capture the nuances of entities when processing aerial text, resulting in difficulties for the model to generate useful structured data.

In response to the above challenges, this paper proposes a novel fine-tuned NER model to improve the extraction capability of complex aviation product entities. The main contributions of this study are as follows:

(1) Using open-source aviation product files, an aviation product entity dataset is constructed to provide a corpus for subsequent named entity recognition research in the aviation product field.

(2) A fine-tuned aviation product entity recognition model is proposed, which integrates a BERT layer, an optimized BiLSTM layer, and a CRF layer. The BERT layer captures contextualized word embeddings, while the optimized BiLSTM layer effectively encodes long-term dependencies and contextual relationships in the text. The CRF layer ensures optimal label prediction by considering dependencies between adjacent entities. By carefully adjusting the model parameters, this approach achieves the accuracy required for named entity recognition (NER) tasks in the aviation field. Experimental results demonstrate that this model significantly outperforms other baseline models, highlighting its performance in handling the complex and specialized language of aviation product texts.

(3) The extracted entities and relationships extracted by other methods are organized in the form of triples and imported into the Neo4j database to form a knowledge graph structure, providing strong support for the application of knowledge graph technology in the field of aviation products.

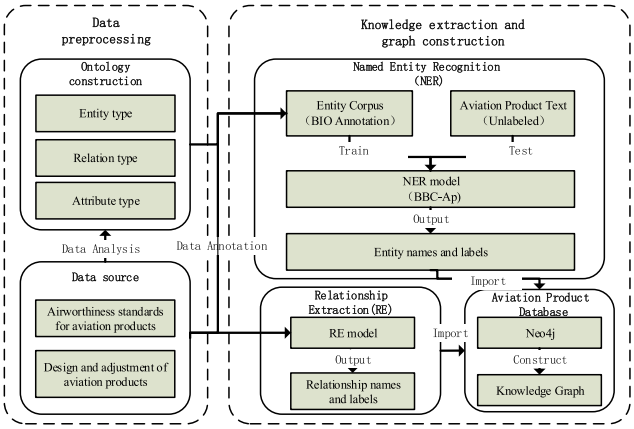

FIGURE 1. Knowledge extraction process.

The rest of this paper is structured as follows: Section II introduces the main methodology of this paper, Section III presents the experimental design and results, and Section IV highlights the concluding remarks of the paper outcomes.

II. METHODOLOGY

Figure 1 shows the steps of NER and relationship extraction for aviation product text. First, ontology modeling is performed based on the actual text and existing expert opinions to define entity and relationship types. Subsequently, the text file is annotated using the BIO scheme and divided into training set, validation set, and test set. Then, the BBC-Ap model is built to identify entities, and relationship extraction draws on methods that have been established by others [21]. Finally, the identified entities and relationships are imported into Neo4j to build a knowledge graph. In the subsequent sections, Section A will present the ontology model and text annotation methods, while Section B will provide a detailed explanation of the composition and principles of the BBC-Ap model.

A. ONTOLOGY MODEL

1) ENTITY AND RELATIONSHIP DEFINITIONS

Before annotating the aviation product text, it is necessary to first define the entity and relationship tags in the text. By defining these tags, it can be ensured that each entity and relationship can be accurately identified and classified in the subsequent annotation process, thereby improving the accuracy and consistency of the annotation [22], [23]. This detailed tag definition work is the basis and key step for achieving high-quality text annotation.

Aviation products can be roughly divided into four levels from the whole to the part: asset, system, module, and component, which are connected to each other through the 'Contains' relationship. At the same time, there are certain attributes describing the aviation product in the text, such as rotation speed, etc. These attributes are defined as 'parameter' and are also connected to the corresponding products through the 'Contains' relationship. Except for asset, the other three

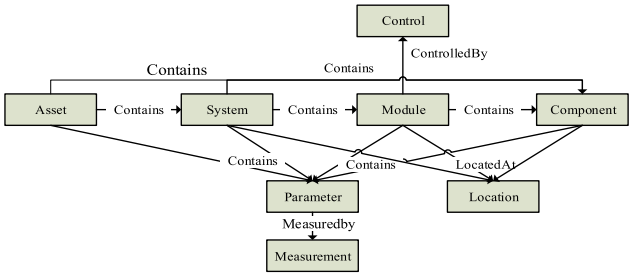

FIGURE 2. Ontology model structure.

levels all have location information to describe the location of the child level set on the parent level. The label of this location information is 'location' and is connected to the level through the 'LocatedAt' relationship. In the module of the aviation product, there is a control and operation interface, which is used to control the operation of the module and detect the operating status. These components are defined as 'Control' and are connected to the module through the 'ControlledBy' relationship. The units of measurement of aviation product parameters are also defined in the text to facilitate statistics and conversion between different units. The labels of these units of measurement are 'Measurement' and are connected to the parameters through the 'Measuredby' relationship. Figure 2 shows the detailed information and structure of the ontology model.

2) TEXT ANNOTATION STRATEGY

This paper chooses to use the BIO annotation strategy to annotate the text content, annotating each element as "B-X", "I-X" or "O" [24]. Among them, "B-X" means that the segment where this element is located belongs to type X and this element is at the beginning of this segment, "I-X" means that the segment where this element is located belongs to type X and this element is in the middle of this segment, and "O" means that it does not belong to any type. This paper finally sets 17 different entity tags, as shown in

Table 1, which is consistent with the ontology model presented in Figure 2.

TABLE 1. Entity TAGS.

Entity type	Labels
Asset	B-Asset, I-Asset
System	B- System, I- System
Module	B- Module, I- Module
Component	B- Component, I- Component
Parameter	B- Parameter, I- Parameter
Control	B- Control, I- Control
Measurement	B- Measurement, I- Measurement
Location	B- Location, I- Location
Non-Entity	O

B. NAMED ENTITY RECOGNITION MODEL

After completing the aviation product ontology modeling, it is necessary to extract aviation product entities from the corpus. Since there are many proper nouns in the aviation product text, the traditional NER model cannot meet the accuracy requirements. Therefore, this paper proposes a multi-layer named entity recognition model, named BBC-Ap, whose structure is shown in Figure 3. The model consists of a BERT layer, a BiLSTM layer, and a CRF layer. Specifically, the BERT layer is responsible for word vector embedding, the BiLSTM layer is responsible for encoding, and the CRF layer is responsible for decoding and optimal label prediction. The implementation of each layer in the model will be introduced in detail below.

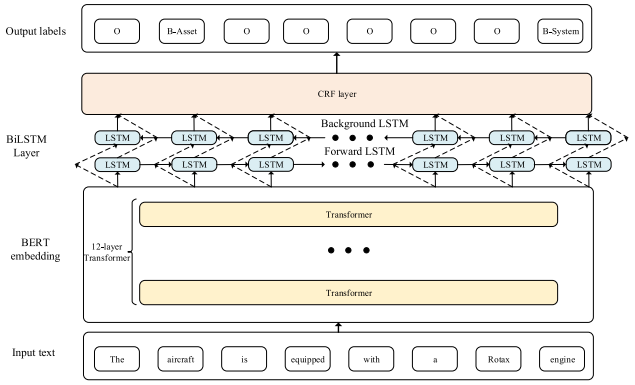

FIGURE 3. NER model architecture.

1) EMBEDDING LAYER

In the named entity recognition task, how to encode and reasonably represent text information is a key issue. The encoding and representation of text information directly affect the performance of the model and the accuracy of recognition. The BERT model adopts a context-based embedding method, which can dynamically generate word vectors according to the context of words in the sentence, thereby capturing richer semantic information [4], [25], [26]. The core of BERT is composed of multiple stacked Transformer encoder layers, each of which contains a self-attention mechanism and a feedforward neural network, allowing the model to capture complex dependencies in the input sequence.

The self-attention mechanism allows the model to focus on tokens at different positions when processing a sequence and calculate the attention weights between tokens to capture the dependencies in the input sequence [27], [28]. This mechanism determines the importance of each token in the current context by comparing it with other tokens in the sequence and assigning different weights. In this way, the self-attention mechanism can recognize and represent complex syntactic and semantic relationships. The calculation of attention weights is shown in the following formula, d_k is the dimension of the key vector K, QK^T denotes the dot product of the query vector (Q) with the key vector, which is divided by $\sqrt{d_k}$ is for scaling to avoid the dot product value being too large. Then the weights are calculated by softmax function and finally multiplied with the value vector (V) to get the final

attention value.

$$\text{Attention (Q, K, V)} = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

The feedforward neural network can further transform the output of the self-attention mechanism to extract higher-level features [29]. Each feedforward neural network layer contains two linear transformations and an activation function, as shown in the Equation (2), where W_1 and W_2 are weight matrices, b_1 and b_2 are bias terms, and $max(0, x)$ is the rectified linear unit (ReLU) activation function.

$$FFN\ (x) = max\ (0, xW_1 + b_1)\ W_2 + b_2 \tag{2}$$

2) BiLSTM LAYER

After obtaining the word vectors output by the BERT model, it is necessary to use the encoder to further encode and model these word vectors. LSTM (Long Short-Term Memory) is a type of RNN (Recurrent Neural Network). Due to its unique design, it is very suitable for modeling time series data, such as text data [30]. The LSTM model consists of the following components at each time step t: input word vector, cell state, temporary cell state, hidden state, forget gate (f_t), memory gate (input gate, i_t), output gate (o_t). Its structure is shown in Figure 4.

The calculation process of LSTM can be summarized as: by forgetting the information in the cell state and remembering the new information, useful information can be passed to the subsequent moment, while useless information is discarded [31], [32]. At each time step, LSTM outputs the hidden state, and this process is implemented by the forget gate, memory gate and output gate. The forget gate determines which information needs to be discarded. The calculation formula is as follows, where w_f is the weight matrix, h_{t-1} is the hidden state of the previous moment, x_t is the input word vector of the current moment, b_f is the bias term, and $\sigma()$ is the sigmoid activation function.

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{3}$$

The memory gate determines what new information needs to be stored. The calculation formula is as follows:

$$i_t = \sigma\ (W_i \cdot [h_{t-1}, x_t] + b_i) \tag{4}$$

The calculation formula of temporary cell state is:

$$\tilde{C}_t = tanh\ (W_C \cdot [h_{t-1}, x_t] + b_C) \tag{5}$$

W_i and W_C are weight matrices, b_i and b_C are bias terms, and *tanh* is the tanh activation function.

The update formula of cell state is:

$$C_t = f_t * C_{t-1} + i_t * C_t \tag{6}$$

The output gate determines which parts of the cell state will be used as output. The calculation formula is as follows:

$$o_t = \sigma\ (W_o \cdot [h_{t-1}, x_t] + b_o) \tag{7}$$

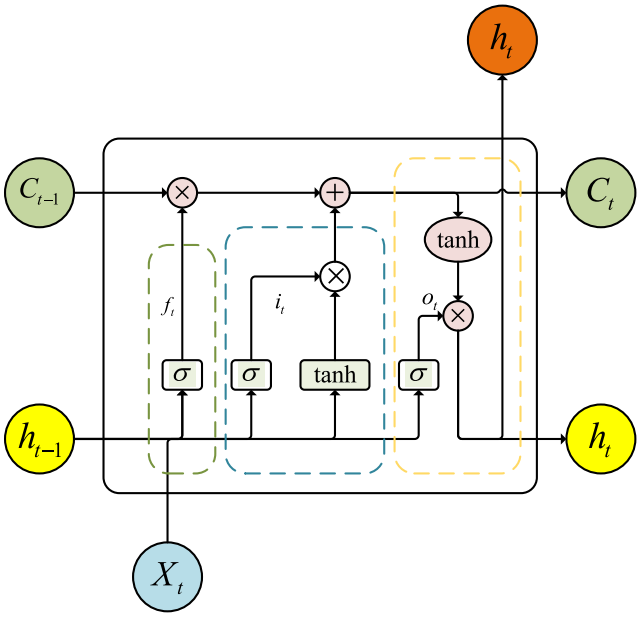

FIGURE 4. LSTM model structure.

The calculation formula of the hidden layer state is:

$$h_t = o_i * \tanh\ (C_t) \tag{8}$$

However, there is still a problem in using LSTM to model sentences: it is impossible to encode information from back to front. Bidirectional LSTM (BiLSTM) further enhances the model's ability to capture time series data by considering both forward and backward sequence information at the same time [33]. BiLSTM consists of two independent LSTM layers, one that processes the sequence from front to back and the other that processes the sequence from back to front, and finally concatenates the outputs of the two. Specifically, if $h_t^{forward}$ and $h_t^{backward}$ represent the hidden states of the forward and backward LSTMs, respectively, the final output of BiLSTM is:

$$h_t^{BiLSTM} = \left[h_t^{forward}, h_t^{backward}\right] \tag{9}$$

In summary, after obtaining the context word vector of the BERT model, using BiLSTM for encoding modeling can effectively capture the temporal dependencies in the text, thereby further improving the performance and accuracy of the named entity recognition task. The word vector enters the BiLSTM layer, which learns the context information and outputs the score probability of each word corresponding to each label. Based on the traditional BiLSTM model, this paper further optimizes the loss function and optimizer. Experiments show that the optimized model has significant improvements in performance and accuracy.

3) CRF LAYER

In the named entity recognition task, the labels output from the BiLSTM layer do not consider the relationship between labels. For example, there may be words starting with

label I-X, two consecutive words with label B-X, etc., which will reduce the effect of the model. In order to correct the output of the BiLSTM layer and ensure the rationality of the predicted labels, the conditional random field (CRF) layer can be used [34].

The CRF layer corrects the output of the BiLSTM layer by learning the transition probability between labels in the data set, thereby ensuring the rationality of the predicted labels. Specifically, the CRF layer can determine whether the transition between labels is reasonable based on contextual information. For example, the CRF layer can learn that label B-X can only appear after label 'O', but not after label I-X. During the training process of the model, the output of all BiLSTMs will be used as the input of the CRF layer, and the final prediction result will be obtained by learning the sequential dependency information between labels [35].

In the named entity recognition task, the CRF layer corrects the output of the BiLSTM layer and ensures the rationality of the predicted label through the following steps [36]. First, the input is the output of the BiLSTM layer, that is, the hidden state of each time step. Next, the transition probability between labels is learned through the transfer matrix, which represents the probability of transferring from one label to another. Then, the score function is used to calculate the score of a given sequence, combining the output of the BiLSTM layer and the transfer matrix for calculation. The calculation formula of the score function is:

$$s(X, y) = \sum_{t} A_{y_{t-1}, y_t} + h_t[y_t] \tag{10}$$

where y is the given sequence label, X is the observed sequence of the sentence, h_t is the output of each time step t, y_t is the label of time step t, and $h_t[y_t]$ represents the score of the BiLSTM output on the label y_t.

Finally, through the decoding process, the Viterbi algorithm is used to find the label sequence with the highest score as the final prediction result. Through these steps, the CRF layer can effectively capture the dependencies between labels and improve the performance and accuracy of the named entity recognition task.

The combination of the three layers—BERT for word embeddings, BiLSTM for sequence encoding, and CRF for label optimization—creates a highly effective NER model tailored specifically for the aviation domain. In this model, several key innovations are introduced. First, the use of BiLSTM enables the model to encode both forward and backward sequence information, improving its ability to capture contextual dependencies. Second, the loss function and optimizer have been fine-tuned, significantly enhancing the model's performance and accuracy. Additionally, the integration of BERT allows the model to extract deep contextual word embeddings, which BiLSTM further refines to capture long-distance dependencies within the text. This structure allows the BBC-Ap model to overcome the limitations of traditional NER models, providing superior performance in recognizing complex aviation entities such as parts, components, and technical systems. These innovations represent a significant advancement in applying NER techniques to highly specialized technical fields like aviation.

III. EXPERIMENTS RESULTS AND DISCUSSIONS

This section will provide a comprehensive explanation of the series of experiments designed to validate the performance of the BBC-Ap model. Section A details the composition of the ApNER dataset, the specifications of the hardware used in the experiments, the hyperparameter settings, and the evaluation metrics employed. Section B discusses the experimental design principles, and the outcomes achieved. Section C showcases the entities and relationships extracted, culminating in the formation of the knowledge graph structure.

A. EXPERIMENTAL PREPARATION

1) DATASET CONSTRUCTION

In order to verify the proposed model, this study used the aircraft design standard documents and airworthiness standard documents published by the European Aviation Safety Agency (EASA) and the US Electronic Federal Regulations (eCFR), as well as the open-source aircraft design dataset AircraftVerse [37] as the corpus to construct a named entity recognition dataset for aviation products (ApNER) [38]. The specific composition is shown in the Table 2.

TABLE 2. Dataset composition.

Data sources	Type of data	Number of files
EASA	aircraft design standard documents	7
eCFR	airworthiness standard	1
AircraftVerse	aircraft design dataset	8

The dataset contains the aviation product hierarchy and information related to aviation product parameters. The number of entities under each label is shown in the Table 3, The bold in the table is the total number of entities, a total of 8837 entities.

TABLE 3. Number of entities under each labels.

Label	Number
Asset	372
System	2516
Module	1938
Component	1789
Parameter	1735
Measurement	248
Location	67
Control	172
Total	**8837**

Since the data in the dataset comes from three different sources, to eliminate the possible impact of data from the

same source on the results, the files were randomly shuffled during the annotation process. Then, we split the dataset in a ratio of 8:1:1 to construct a training set, a validation set, and a test set. These split datasets were input into the model for training and validation to ensure the performance and reliability of the model.

2) EXPERIMENTAL PARAMETER SETTINGS AND EVALUATION METRICS

This study uses PyTorch to build a deep learning experimental environment, and the programming language is python3.8, uses the AdamW optimizer with an initial learning rate of 0.00005 to update the model parameters, and adds Dropout after the BiLSTM layer to prevent overfitting. The training environment of the model is shown in the Table 4.

TABLE 4. Experimental environment setup.

Hardware	Hardware Parameters
CPU	Intel(R) Xeon(R) Platinum 8260C CPU @ 2.30GHz
Number of CPUs	12
GPU	NVIDIA GeForce RTX 3090
Memory	24GB
RAM	86GB

The hyperparameter settings of the model are shown in the Table.

TABLE 5. Experimental hyperparameter settings.

Parameter	Value
Initial Learning Rate	0.00005
Epochs	20
batch_size	32
Embedding Size	768
Hidden Size	256
Dropout Rate	0.7

This study uses precision (P), recall (R), and F1 score to evaluate model performance. Precision refers to the ratio of the number of named entities correctly identified by the model to the total number of named entities identified by the model (both correct and incorrect). The higher the precision, the higher the accuracy of the model in the results it identifies as entities. The formula is as follows:

$$P = \frac{TP}{TP + FP} \tag{11}$$

where TP is the number of true positives (named entities correctly identified) and FP is the number of false positives (non-entities incorrectly identified as entities).

Recall (R) refers to the ratio of the number of named entities correctly identified by the model to the total number of named entities actually existing in the dataset. The higher the recall rate, the more real entities the model can find. The formula is:

$$R = \frac{TP}{TP + FN} \tag{12}$$

where FN is the number of false negatives (named entities that fail to be identified).

The F1 score is the harmonic mean of precision and recall and is used to measure the overall performance of the model. The F1 score provides a balance between precision and recall. When the gap between precision and recall is large, the F1 score will be relatively low. The higher the F1 score, the better the model performs in terms of both precision and recall. Its formula is:

$$F1 = 2 \times \frac{P \times R}{P + R} \tag{13}$$

B. EXPERIMENTAL RESULTS

1) NAMED ENTITY RECOGNITION RESULTS

The recognition results of the model proposed in this study in the aviation product database mentioned above are shown in Table 6, where the bold font is the entity category with the best recognition result, and the underlined font is the entity category with the second-best recognition result. The confusion matrix of the recognition results of each label is shown in Figure 5. The two most accurately recognized entity labels are system and module, which shows that our model can almost completely recognize all aviation systems and modules. However, in terms of parameter recognition, the model performs poorly. This may be because some parameters of aviation products have strong independence, belong to only a specific module or component, and appear less frequently in the corpus, resulting in poor recognition results.

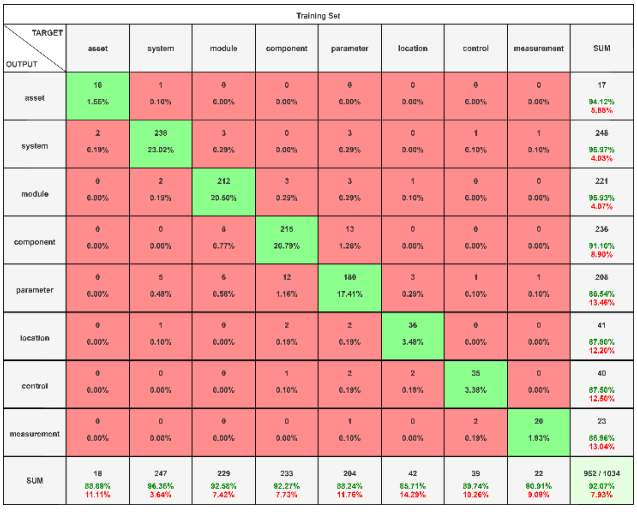

FIGURE 5. Confusion matrix of our model NER results.

2) VALIDATION ON STANDARD DATASETS

To fully verify the performance of the proposed model, this study not only tested it on a specialized aviation

TABLE 6. Label-level recognition results of our model.

Labels	Precision %	Recall %	F1-Score %
asset	94.12	88.89	91.43
system	**95.97**	**96.36**	**96.16**
module	95.93	92.58	94.22
component	91.10	92.27	91.68
parameter	86.54	88.24	87.38
location	87.80	85.71	86.75
control	87.50	89.74	88.61
measurement	86.96	90.91	88.89

TABLE 7. Verification results on standard datasets.

Dataset	Precision (%)	Recall (%)	F1-Score (%)
CoNLL2003[39]	91.51	92.06	91.79
CoNLLpp[40]	92.87	92.54	92.70
CrossNER[41]	62.42	59.11	60.72
ApNER[38]	91.74	92.46	92.10

product dataset, but also conducted the same verification on a recognized standard dataset. This dual verification method ensures the generalization ability and reliability of the model. The selected standard datasets are detailed as follows:

(1) CoNLL2003 [39]: The CoNLL-2003 named entity dataset is designed for the CoNLL-2003 shared task and includes two language versions: English and German. The dataset consists of eight files. The English part comes from Reuters Corpus and covers news reports published by Reuters from August 1996 to August 1997.

(2) CoNLLpp [40]: TheCoNLLpp dataset is an improved version of the Conll-2003 named entity recognition dataset. Its characteristic is that in the test set, about 5.38% of the sentence labels have been manually verified and corrected to improve the accuracy and consistency of the annotation. In addition, to ensure the coherence of the research and the integrity of the data, the CoNLLpp dataset contains the training set and validation set of the original Conll-2003 dataset. This makes the dataset an important resource for evaluating the performance of named entity recognition algorithms in practical applications.

(3) CrossNER [41]: The CrossNER dataset is an English named entity recognition dataset for multiple different fields (literature, politics, English, music, science, and artificial intelligence). This paper mainly uses the scientific field dataset for verification.

Table 7 presents the performance verification results of BBC-Ap model on different standard datasets. Specifically, on the CoNLL-2003 and CoNLLpp datasets, the model shows strong performance, reaching an F1 score of 92.7% on the CoNLLpp dataset and 91.79% on the CoNLL2003 dataset. It is worth noting that the performance of the model on the CoNLL++ dataset is improved compared to the CoNLL-2003 dataset, which reflects the effectiveness of manual verification and optimization of the CoNLL-2003 dataset. However, on the CrossNER dataset, the performance of the model is lower, which shows that although our model is suitable for named entity recognition tasks in the aviation product field and the general field, its application in the scientific field still needs further optimization and adjustment.

3) COMPARISON WITH OTHER BASELINE MODELS

Considering that there are few entity recognition models for the aviation field, this paper uses four baseline models for experiments. The detailed description of each model is as follows:

(1) Albert-BiLSTM-CRF [42]: The Albert model reduces the number of model parameters based on the Bert model and uses two strategies for optimization: parameter sharing and sentence-order prediction SOP (sentence-order prediction) to reduce computing resource consumption and improve training efficiency.

(2) Roberta-BiLSTM-CRF [43]: RoBERTa is an improvement on the BERT pre-training method. It removes the Next Sentence Prediction (NSP) task in BERT and modifies the size and batch of training data.

(3) Deberta-BiLSTM-CRF [44]: The Deberta model introduces a new mechanism in the decoding process, which enables the model to understand better and generate text. Each token in the model is represented by two-word vectors, which encode its content and position respectively. At the same time, the model divides the attention weight into two independent parts: content attention weight and position attention weight. This decomposition method enables the model to capture the dependency between words more accurately.

(4) BERT-BIGRU-CRF [45]: The GRU model is a variant of the LSTM model. Compared with the LSTM model, the GRU model has only two gates in the model: the update gate and the reset gate. The GRU model uses the same gating for forgetting and selective memory.

All baseline methods uniformly use the CRF layer as the final output to ensure the fairness of the analysis. The experimental results are shown in the Table 8. The underlined data is the second best. We can find that compared with other models, BERT as a word embedding layer has shown obvious advantages in processing aviation product texts, especially in the recognition of long proper nouns. This may be because the

changes in other models are not very meaningful for aviation product texts and are more suitable for texts in fields such as medicine. In addition, since aviation product texts contain many long proper nouns, the LSTM model as an encoder can more effectively remember long-term information than the GRU model, effectively solving the problem of long-term dependence.

The BBC-Ap model further considers the domain-specific adaptability during training, especially in the long entity recognition task in the aviation field. Although traditional BERT, ALBERT and other models are superior in natural language processing tasks, when faced with a large number of terminologies in the aviation field, the common features in the pre-trained model may not be enough to support the accurate recognition of complex entities. Through adaptive training, our model can capture subtle differences in the aviation field, significantly improving the model's ability to recognize complex entity structures.

TABLE 8. Model experiment results.

Baseline Model	Precision (%)	Recall (%)	F1-Score (%)
Albert-BiLSTM-CRF [42]	89.87	91.46	90.66
Roberta-BiLSTM-CRF [43]	90.60	92.01	91.30
Deberta-BiLSTM-CRF [44]	90.72	92.12	91.41
BERT-BIGRU-CRF [45]	90.58	91.68	91.13
BBC-Ap	**91.74**	**92.46**	**92.10**

4) ABLATION EXPERIMENT

This paper evaluates the proposed model, including the BERT model, the optimized LSTM model, and the impact of the CRF model on the model performance. The experiment uses the BERT model as the baseline model, and then performs 4 different experiments: the BERT model, the BERT model with the CRF layer, the Bert model with the BiLSTM, and the final proposed model.

To ensure the fairness of the experiment, the models used are trained in the same experimental environment and use the same hyperparameters in Table 5. The results are shown in Figure 7. From the experimental results, the recognition ability of the model is improved after adding the CRF layer and the BiLSTM layer. Specifically, after adding the CRF layer, the F1 value of the model is increased by 0.24%, which proves the effectiveness of the CRF layer in capturing label sequence information; after adding the BiLSTM layer, the F1 value of the model is increased by 0.31%, which shows that the BiLSTM layer increases the ability of the model to capture contextual information. The model proposed in this paper improves the Recall value by 1.33% and the F1 value by 0.92%, which proves the effectiveness of this paper's model optimization.

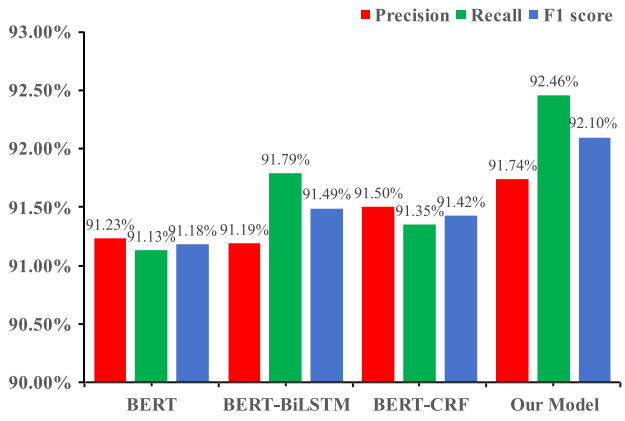

FIGURE 6. The impact of each component on the model.

5) THE IMPACT OF HYPERPARAMETERS ON THE MODEL

In this paper, empirical tests were conducted to determine the effects of various parameters and hyperparameters; however, the primary focus is on two key factors: dropout and learning rate. By exploring the impact of these techniques on our model, we aim to gain insights into their roles in optimizing performance and addressing challenges such as overfitting and convergence.

Dropout is a regularization technique used to prevent the model from overfitting by randomly ''dropping out'' a fraction of the neurons during training, forcing the model to rely on distributed representations. In our experiments, we evaluated the model's performance with different Dropout values ranging from 0.3 to 0.7 to observe how varying degrees of dropout impacted the model's accuracy and generalization ability.

As shown in Figure 7, it is evident that the model has the best performance when Dropout value is set to 0.5. This optimal value provided a balanced trade-off between regularization and retaining sufficient information flow through the network during training. For Dropout values less than 0.5 (e.g., 0.3), the model tended to overfit the training data, as it failed to effectively prevent co-adaptation of neurons. This overfitting manifested as a decrease in all evaluation metrics, such as precision, recall, and F1-score, on the test set. For example, with a Dropout value of 0.3, the F1-score dropped by 1%, indicating poor generalization.

On the other hand, when the Dropout value exceeded 0.5 (e.g., 0.7), the model's performance rapidly deteriorated. This decline was primarily due to the model losing too much semantic information during training, as a higher dropout rate means a larger fraction of neurons is randomly deactivated. As a result, the model was unable to fully capture the underlying patterns and features in the data, leading to underfitting. For instance, with a Dropout value of 0.7, the model's accuracy decreased by 2%, the performance metrics showed a consistent downward trend, indicating a failure to learn sufficient representations.

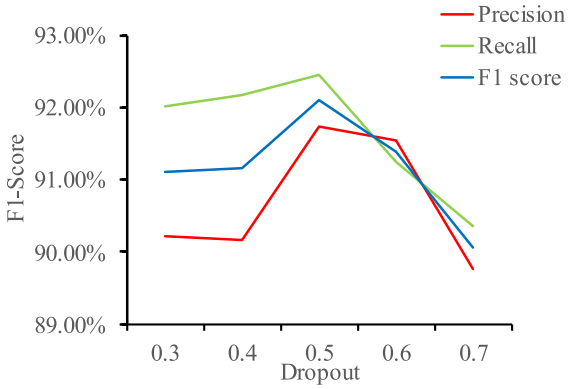

FIGURE 7. Impact of Dropout on model performance.

The learning rate is another critical hyperparameter that governs how quickly or slowly the model updates its weights during the optimization process. In our experiments, we tested different learning rates, including 0.00001, 0.00005, 0.0001, and 0.001, to evaluate their effect on convergence and model performance.

As shown in Figure 8, the model works best when the learning rate is 0.00005. At this rate, the model converged steadily and achieved the highest evaluation metrics, suggesting that the learning rate was neither too fast nor too slow, allowing for stable updates and better convergence to an optimal solution. However, when we increased the learning rate to values such as 0.0001 and 0.001, the model's performance metrics began to degrade significantly. A larger learning rate caused the model to overshoot the optimal weights, preventing it from fully converging. Conversely, when the learning rate was reduced to 0.00001, the model's performance also decreased, but at a slower pace. This was because a smaller learning rate led to slower convergence, requiring significantly more training epochs to reach an optimal solution. While this slower pace did not lead to immediate degradation, it prolonged the training process, and the model ultimately failed to capture the underlying patterns as effectively as it did with the learning rate of 0.00005.

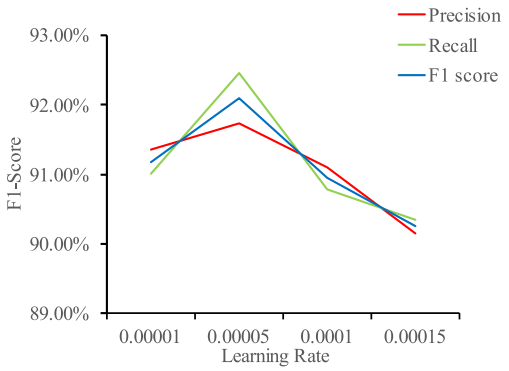

FIGURE 8. Impact of Learning Rate on model performance.

C. KNOWLEDGE GRAPH CONSTRUCTION

After the proposed model was successfully used for entity extraction, a fully supervised relationship extraction method

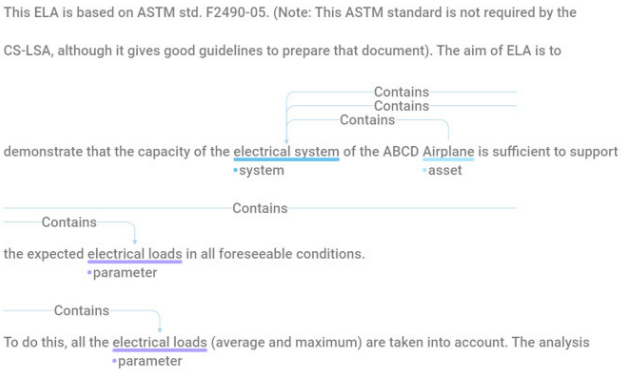

FIGURE 9. Entity and relation extraction example.

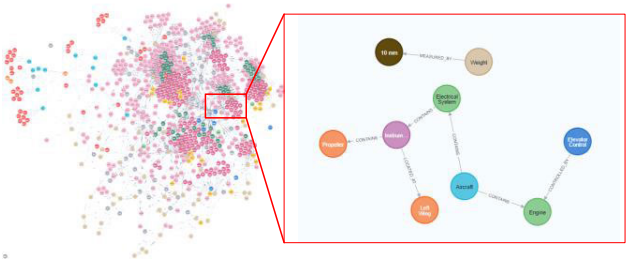

FIGURE 10. Aviation product knowledge graph structure.

was further implemented to identify specific relationships between entities. As the Figure 9 shows, after this process, the established database finally contained more than 8,000 nodes and more than 20,000 relationships.

In constructing the knowledge graph, we first defined entity types and relationship types based on domain-specific requirements. The initial step involved processing the extracted entities and relationships through pre-defined schemas to ensure consistency and correctness. Specifically, each entity was assigned a category (e.g., component, asset, module) to ensure the structure adhered to the semantic needs of our domain.

We used Neo4j as our graph database tool due to its intuitive and clear structure, which excels at representing nodes (entities) and edges (relationships). This choice allowed us to efficiently query the graph and analyze both direct and indirect relationships between entities. Neo4j's Cypher query language further enabled advanced queries that facilitated the exploration of complex network patterns within the data.

To build the graph, we first inputted the extracted entities and relationships into Neo4j, where nodes represented the entities, and directed edges represented the relationships. The relationships were annotated with attributes such as relationship type, timestamp, and confidence score derived from the extraction process. In this way, we successfully constructed a knowledge graph, the structure of which is shown in the Figure 10, which shows the complex network between entities and relationships in detail.

IV. CONCLUSION

This paper presents a novel tailored named entity recognition (NER) method for aviation products, addressing the

challenges posed by the vast amount of text information and the complexity of manual extraction. The method is developed through an in-depth analysis of aviation product text files. The aviation product ontology model is initially constructed to define the entity categories and relationship categories of aviation products. Then, an aviation product dataset containing more than 8,000 entities is constructed using appropriate annotation strategies. On this basis, this paper proposes an optimized BERT-BiLSTM-CRF model, which can fully learn and recognize proper nouns in aviation products and solve the long-distance dependency problem in text files. The experimental results show that the three entity recognition indexes of the model reach 91.74%, 92.46% and 92.10%, respectively, which has excellent performance in the entity recognition of complex aerospace products. Finally, the identified relationships and entities are imported into Neo4j to form a knowledge graph structure based on aviation products. The named entity recognition model proposed in this paper enables aerospace manufacturing enterprises to extract extensive and accurate product information from complex data sets, which greatly reduces the dependence on manual labor. By addressing a unique linguistic challenge in aviation and integrating the results into a structured knowledge graph, this research provides a powerful tool for improving data utilization and supporting intelligent decision making in aviation product development and management.

In future research, a more suitable relationship extraction method can be proposed based on the characteristics of the aviation product corpus. In addition, entity alignment methods and knowledge graph completion methods can be used to improve the accuracy of the information contained in the knowledge graph, thereby enhancing its auxiliary function in the aviation field. Future research will focus on automating the data collection and annotation process to significantly enhance the scalability of the model. This endeavor may necessitate the development of more sophisticated deep learning techniques, tailored to adapt to the unique nuances of specialized aviation data, thereby ensuring more flexible and precise applications.

Furthermore, continuous exploration of the model's robustness will be essential, particularly in handling rare or unseen entities and adapting to variations in terminology across different aviation subfields. Robustness tests will include stress-testing the model under diverse operational scenarios, such as noisy or incomplete data, to ensure its resilience. Investigating techniques like transfer learning and domain adaptation will also be crucial to enhance the model's ability to generalize effectively to new, unseen aviation datasets, further solidifying its reliability and versatility in real-world applications.

DATA STATEMENT

Data supporting this study are openly available from the CORD repository: https://doi.org/10.57996/cran.ceres-2614.

REFERENCES

[1] A. Rivas, I. Grangel-González, D. Collarana, J. Lehmann, and M.-E. Vidal, "Unveiling relations in the Industry 4.0 standards landscape based on knowledge graph embeddings," in *Proc. 31st Int. Conf., Database Expert Syst. Appl.*, Jun. 2020, pp. 179–194, doi: 10.1007/978-3-030-59051-2_12.

[2] M. Yahya, J. G. Breslin, and M. I. Ali, "Semantic Web and knowledge graphs for Industry 4.0," *Appl. Sci.*, vol. 11, no. 11, p. 5110, May 2021, doi: 10.3390/app11115110.

[3] N. Melluso, I. Grangel-González, and G. Fantoni, "Enhancing industry 4.0 standards interoperability via knowledge graphs with natural language processing," *Comput. Ind.*, vol. 140, Sep. 2022, Art. no. 103676, doi: 10.1016/j.compind.2022.103676.

[4] A. Harnoune, M. Rhanoui, M. Mikram, S. Yousfi, Z. Elkaimbillah, and B. El Asri, "BERT based clinical knowledge extraction for biomedical knowledge graph construction and analysis," *Comput. Methods Programs Biomed. Update*, vol. 1, Jan. 2021, Art. no. 100042, doi: 10.1016/j.cmpbup.2021.100042.

[5] T. Al-Moslmi, M. G. Ocaña, A. L. Opdahl, and C. Veres, "Named entity extraction for knowledge graphs: A literature overview," *IEEE Access*, vol. 8, pp. 32862–32881, 2020, doi: 10.1109/ACCESS.2020.2973928.

[6] B. Abu-Salih and S. Alotaibi, "A systematic literature review of knowledge graph construction and application in education," *Heliyon*, vol. 10, no. 3, 2024, Art. no. e25383, doi: 10.1016/j.heliyon.2024.e25383.

[7] M. Baigang and F. Yi, "A review: Development of named entity recognition (NER) technology for aeronautical information intelligence," *Artif. Intell. Rev.*, vol. 56, no. 2, pp. 1515–1542, Feb. 2023, doi: 10.1007/s10462-022-10197-2.

[8] D. R. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Lingvisticae Investigationes*, vol. 30, no. 1, pp. 3–26, Aug. 2007, doi: 10.1075/li.30.1.03nad.

[9] R. Sharnagat, "Named entity recognition: A literature survey," *Center Indian Lang. Technol.*, vol. 56, no. 1, pp. 1–27, Jun. 2014.

[10] L. Zhang, Y. Pan, and T. Zhang, "Focused named entity recognition using machine learning," in *Proc. 27th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2004, pp. 281–288, doi: 10.1145/1008992.1009042.

[11] G. Zhou and J. Su, "Machine learning-based named entity recognition via effective integration of various evidences," *Natural Lang. Eng.*, vol. 11, no. 2, pp. 189–206, Jun. 2005, doi: 10.1017/s1351324904003559.

[12] J. Sun, Y. Liu, J. Cui, and H. He, "Deep learning-based methods for natural hazard named entity recognition," *Sci. Rep.*, vol. 12, no. 1, pp. 1–15, Mar. 2022, doi: 10.1038/s41598-022-08667-2.

[13] J. Li, A. Sun, J. Han, and C. Li, "A survey on deep learning for named entity recognition," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 1, pp. 50–70, Jan. 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9039685/?casa_token=ZfEBsJbZMIEAAAAA:8LijNCGBsXGXrjH0gqwOzimj62hCAtDEQp7_IpwCwKlGfcA836NjtnIQIsTRLmOWbLTXT9W3

[14] N. Xu, Y. Liang, C. Guo, B. Meng, X. Zhou, Y. Hu, and B. Zhang, "Entity recognition in the field of coal mine construction safety based on a pre-training language model," *Eng., Construct. Architectural Manage.*, vol. 31, no. 1, pp. 1–24, Dec. 2023, doi: 10.1108/ecam-05-2023-0512.

[15] Y. Jiang, H. Jiang, J. Chen, and X. Miao, "A lightweight named entity recognition method for Chinese power equipment defect text," in *Proc. 9th Int. Forum Electr. Eng. Autom. (IFEEA)*, Nov. 2022, pp. 368–372, doi: 10.1109/ifeea57288.2022.10037787.

[16] L. Burke, K. Pazdernik, D. Fortin, B. Wilson, R. Goychayev, and J. Mattingly, "NukeLM: Pre-trained and fine-tuned language models for the nuclear and energy domains," 2021, *arXiv:2105.12192*.

[17] C. Yang, L. Sheng, Z. Wei, and W. Wang, "Chinese named entity recognition of epidemiological investigation of information on COVID-19 based on BERT," *IEEE Access*, vol. 10, pp. 104156–104168, 2022, doi: 10.1109/ACCESS.2022.3210119.

[18] H. Wei, M. Gao, A. Zhou, F. Chen, W. Qu, C. Wang, and M. Lu, "Named entity recognition from biomedical texts using a fusion attention-based BiLSTM-CRF," *IEEE Access*, vol. 7, pp. 73627–73636, 2019, doi: 10.1109/ACCESS.2019.2920734.

[19] A. T. Ray, O. J. P. Fischer, R. T. White, B. F. Cole, and D. N. Mavris, "Development of a language model for named-entity-recognition in aerospace requirements," *J. Aerosp. Inf. Syst.*, vol. 21, no. 6, pp. 489–499, Jun. 2024, doi: 10.2514/1.i011251.

[20] J. Chu, Y. Liu, Q. Yue, Z. Zheng, and X. Han, "Named entity recognition in aerospace based on multi-feature fusion transformer," *Sci. Rep.*, vol. 14, no. 1, pp. 1–14, Jan. 2024, doi: 10.1038/s41598-023-50705-0.
[21] N. Zhang, X. Xu, L. Tao, H. Yu, H. Ye, S. Qiao, X. Xie, X. Chen, Z. Li, and L. Li, "DeepKE: A deep learning based knowledge extraction toolkit for knowledge base population," in *Proc. Conf. Empirical Methods Natural Lang. Process., Syst. Demonstrations*, Jan. 2022, pp. 98–108, doi: 10.18653/v1/2022.emnlp-demos.10.
[22] E. Batbaatar and K. H. Ryu, "Ontology-based healthcare named entity recognition from Twitter messages using a recurrent neural network approach," *Int. J. Environ. Res. Public Health*, vol. 16, no. 19, p. 3628, Sep. 2019, doi: 10.3390/ijerph16193628.
[23] J. Santoso, E. I. Setiawan, C. N. Purwanto, E. M. Yuniarno, M. Hariadi, and M. H. Purnomo, "Named entity recognition for extracting concept in ontology building on Indonesian language using end-to-end bidirectional long short term memory," *Expert Syst. Appl.*, vol. 176, Aug. 2021, Art. no. 114856, doi: 10.1016/j.eswa.2021.114856.
[24] N. Alshammari and S. Alanazi, "The impact of using different annotation schemes on named entity recognition," *Egyptian Informat. J.*, vol. 22, no. 3, pp. 295–302, Sep. 2021, doi: 10.1016/j.eij.2020.10.004.
[25] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL*, vol. 1, 2019, pp. 4171–4186.
[26] M. Joshi, O. Levy, L. Zettlemoyer, and D. Weld, "BERT for coreference resolution: Baselines and analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process., 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 5803–5808, doi: 10.18653/v1/d19-1588.
[27] J. Deng, L. Cheng, and Z. Wang, "Self-attention-based BiGRU and capsule network for named entity recognition," 2020, *arXiv:2002.00735*.
[28] A. Zukov-Gregoric, Y. Bachrach, P. Minkovsky, S. Coope, and B. Maksak, "Neural named entity recognition using a self-attention mechanism," in *Proc. IEEE 29th Int. Conf. Tools Artif. Intell. (ICTAI)*, Nov. 2017, pp. 652–656, doi: 10.1109/ICTAI.2017.00104.
[29] J. Chiu and E. Nichols, "Named entity recognition with bidirectional LSTM-CNNs," *Trans. Assoc. Comput. Linguistics*, vol. 4, pp. 357–370, Dec. 2016, doi: 10.1162/tacl_a_00104.
[30] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Phys. D, Nonlinear Phenomena*, vol. 404, Mar. 2020, Art. no. 132306, doi: 10.1016/j.physd.2019.132306.
[31] K. Smagulova and A. P. James, "A survey on LSTM memristive neural network architectures and applications," *Eur. Phys. J. Special Topics*, vol. 228, no. 10, pp. 2313–2324, Oct. 2019, doi: 10.1140/epjst/e2019-900046-x.
[32] R. C. Staudemeyer and E. R. Morris, "Understanding LSTM—A tutorial into long short-term memory recurrent neural networks," 2019, *arXiv:1909.09586*.
[33] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "The performance of LSTM and BiLSTM in forecasting time series," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2019, pp. 3285–3292, doi: 10.1109/BigData47090.2019.9005997.
[34] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF models for sequence tagging," 2015, *arXiv:1508.01991*.
[35] C. Lee, "LSTM-CRF models for named entity recognition," *IEICE Trans. Inf. Syst.*, vol. E100.D, no. 4, pp. 882–887, 2017, doi: 10.1587/transinf.2016edp7179.
[36] R. Yan, X. Jiang, and D. Dang, "Named entity recognition by using XLNet-BiLSTM-CRF," *Neural Process. Lett.*, vol. 53, no. 5, pp. 3339–3356, Oct. 2021, doi: 10.1007/s11063-021-10547-1.
[37] A. D. Cobb, A. Roy, D. Elenius, F. Heim, B. Swenson, S. Whittington, J. Walker, T. Bapty, J. Hite, K. Ramani, and C. McComb, "AircraftVerse: A large-scale multimodal dataset of aerial vehicle designs," in *Proc. Adv. Neural Inf. Process. Syst.*, Jun. 2023, pp. 44524–44543.
[38] M. Yang, "Aviation product named entity recognition dataset (ApNER)," Cranfield Univ., U.K., Aug. 2024, doi: 10.57996/CRAN.CERES-2614.
[39] E. F. Tjong, K. Sang, and F. D. Meulder, "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition," Jun. 2003, *arXiv:cs/0306050*.
[40] Z. Wang, J. Shang, L. Liu, L. Lu, J. Liu, and J. Han, "CrossWeigh: Training named entity tagger from imperfect annotations," 2019, *arXiv:1909.01441*.
[41] Z. Liu, Y. Xu, T. Yu, W. Dai, Z. Ji, S. Cahyawijaya, A. Madotto, and P. Fung, "CrossNER: Evaluating cross-domain named entity recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 13452–13460. [Online]. Available: https://github.com/zliucr/CrossNER
[42] P. Zhao, W. Wang, H. Liu, and M. Han, "Recognition of the agricultural named entities with multifeature fusion based on Albert," *IEEE Access*, vol. 10, pp. 98936–98943, 2022, doi: 10.1109/ACCESS.2022.3206017.
[43] K. L. Tan, C. P. Lee, K. S. M. Anbananthen, and K. M. Lim, "RoBERTa-LSTM: A hybrid model for sentiment analysis with transformer and recurrent neural network," *IEEE Access*, vol. 10, pp. 21517–21525, 2022, doi: 10.1109/ACCESS.2022.3152828.
[44] Y. Jeong and E. Kim, "SciDeBERTa: Learning DeBERTa for science technology documents and fine-tuning information extraction tasks," *IEEE Access*, vol. 10, pp. 60805–60813, 2022, doi: 10.1109/ACCESS.2022.3180830.
[45] S. Nosouhian, F. Nosouhian, and A. K. Khoshouei, "A review of recurrent neural network architecture for sequence learning: Comparison between LSTM and GRU," *Preprints*, Jul. 2021, doi: 10.20944/PREPRINTS202107.0252.V1.

MINGYE YANG was born in Liaoning, China, in 2000. He received the double bachelor's degree in engineering and finance from Jilin University, in 2022. He is currently pursuing the double master's degree with Nanjing University of Aeronautics and Astronautics and Cranfield University. His research interests include knowledge graphs and natural language processing.

BERNADIN NAMOANO is currently a Lecturer of digital twinning and machine learning with Cranfield University. His work involves data modeling, analytics, and architectures to support digital ecosystem development in aerospace, transportation, and manufacturing.

MARYAM FARSI received the Ph.D. degree in non-linear structural mechanics from Imperial College London, in 2014. She is currently a Senior Lecturer of engineering optimization with Cranfield University. She leads complex systems and optimization research with the Centre for Digital Engineering and Manufacturing. Her research interests include complex systems engineering, data analysis, and optimization across different applications in aerospace, manufacturing, transport, and healthcare.

JOHN AHMET ERKOYUNCU received the Ph.D. degree in uncertainty modeling for maintenance from Cranfield University, in 2011. He is currently the Head of the Digital Engineering and Manufacturing Centre. He is active with Innovate U.K., and EPSRC-funded projects around research topics: digital twins, augmented reality, digitization (degradation assessment), and simulation of complex manufacturing and maintenance procedures. He is a Chartered Engineer, an Associate Member of CIRP, and the Co-Chair of the Through-life Engineering Services Council.

Named entity recognition in aviation products domain based on BERT

Yang, Mingye

2024-12-01

Yang M, Namoano B, Farsi M, Ahmet Erkoyuncu J. (2024) Named entity recognition in aviation products domain based on BERT. IEEE Access, Volume 12, December 2024, pp. 189710-189721
https://doi.org/10.1109/access.2024.3516390