TYPE Original Research
PUBLISHED 07 November 2024
DOI 10.3389/fpls.2024.1482275

OPEN ACCESS

EDITED BY
Xiangjun Zou,
South China Agricultural University, China

REVIEWED BY
Lanhui Fu,
Wuyi University, China
Santo Ángel Ortega Acosta,
Universidad Autónoma de Guerrero, Mexico
Gilbert Yong San Lim,
SingHealth, Singapore

*CORRESPONDENCE
Haiyan Song
yybbao@sxau.edu.cn

RECEIVED 18 August 2024
ACCEPTED 22 October 2024
PUBLISHED 07 November 2024

CITATION
Wang K, Miao Y, Wang X, Li Y, Li F and Song H (2024) Research on the construction of a knowledge graph for tomato leaf pests and diseases based on the named entity recognition model.
Front. Plant Sci. 15:1482275.
doi: 10.3389/fpls.2024.1482275

Research on the construction of a knowledge graph for tomato leaf pests and diseases based on the named entity recognition model

Kun Wang[1], Yuyuan Miao[1], Xu Wang[1], Yuze Li[1], Fuzhong Li[1] and Haiyan Song[2*]

[1]Software College, Shanxi Agricultural University, Jinzhong, Shanxi, China, [2]Agricultural Engineering College, Shanxi Agricultural University, Jinzhong, Shanxi, China

Introduction: Tomato leaf pests and diseases pose a significant threat to the yield and quality of Q6 tomatoes, highlighting the necessity for comprehensive studies on effective control methods.

Methods: Current control measures predominantly rely on experience and manual observation, hindering the integration of multi-source data. To address this, we integrated information resources related to tomato leaf pests and diseases from agricultural standards documents, knowledge websites, and relevant literature. Guided by domain experts, we preprocessed this data to construct a sample set.

Results: We utilized the Named Entity Recognition (NER) model ALBERT-BiLSTM-CRF to conduct end-to-end knowledge extraction experiments, which outperformed traditional models such as 1DCNN-CRF and BiLSTM-CRF, achieving a recall rate of 95.03%. The extracted knowledge was then stored in the Neo4j graph database, effectively visualizing the internal structure of the knowledge graph.

Discussion: We developed a digital diagnostic system for tomato leaf pests and diseases based on the knowledge graph, enabling graphical management and visualization of pest and disease knowledge. The constructed knowledge graph offers insights for controlling tomato leaf pests and diseases and provides new research directions for pest control in other crops.

KEYWORDS

tomato leaf pests and diseases, knowledge graph, ALBERT-BiLSTM-CRF, Neo4j graph database, digital diagnosis

1 Introduction

Agricultural production plays a crucial role in social development. In practical agricultural activities, crops are easily affected by various adverse factors. Among them, pests and diseases are one of the most significant factors affecting crop yield and income (Abdullah et al., 2023). Crop diseases and insect pests are characterized by their wide spread and propensity to cause disasters, which seriously affects the healthy growth of crops. Tomato, as a widely cultivated high-value vegetable globally, inevitably faces various factors such as improper cultivation practices, inadequate control measures, and environmental pollution during its cultivation process. This results in the occurrence of seventeen common types of pests and diseases, such as damping-off, viral diseases and leaf mold et al. These diseases not only severely impact the quality of tomatoes, but also significantly reduce yields, leading to substantial economic losses.

The traditional relational database knowledge management methods are unable to effectively represent and store such knowledge, facing challenges such as difficulties in integrating heterogeneous data, inefficiencies in expressing data relationships, and inadequacies in knowledge refinement. The term "knowledge graph" was used by Edgar W. Schneider in 1972 to describe data structures and control flows in educational modules (Schneider, 1973). In 2012, Google introduced the Google Knowledge Graph, effectively expressing relationships between data, marking the formal naming of knowledge graphs (Singhal, 2012). Research and construction of Knowledge Graphs have penetrated various domains of social life such as education (Xue et al., 2024), healthcare systems (Shi et al., 2017; Wu et al., 2023), and smart cities (Liu et al., 2021). They represent entities, attributes, and their relationships from the natural world in a graphical form, allowing for a better expression of domain-specific information closer to human cognitive understanding (Wu et al., 2021a). Based on the scope of knowledge coverage, Knowledge Graphs are mainly classified into two types: vertical Knowledge Graphs and general Knowledge Graphs (Ji et al., 2021). General Knowledge Graphs cover a wide range of knowledge across multiple domains, primarily addressing common-sense issues in the real world. Currently influential general Knowledge Graphs include YAGO2 (Hoffart et al., 2011), DBpedia (Lehmann et al., 2015), Freebase (Bollacker et al., 2008), as well as Chinese Knowledge Graph websites such as CN-DBpedia (Xu et al., 2017) and Baidu Zhixin (Niu et al., 2011). General Knowledge Graphs can be likened to "structured encyclopedic knowledge bases," emphasizing breadth of knowledge and commonly used in search engines, constructed mainly in a bottom-up manner. Vertical Knowledge Graphs, also known as domain-specific Knowledge Graphs, are typically tailored to specific domains and require higher levels of depth and accuracy in knowledge representation. They serve as specialized knowledge repositories based on industry needs. For instance, Xindong Wu and colleagues proposed the Huapu-KG model, integrating HAO intelligence (human intelligence + artificial intelligence + organizational intelligence) for constructing genealogy Knowledge Graphs, addressing challenges of heterogeneity, autonomy, complexity, and evolution in genealogy data (Wu et al., 2021b). Wang, Chengbin et al. utilized hybrid corpora, Chinese word segmentation rules, and statistical methods to build a Knowledge Graph, demonstrating the potential and practicality of natural language processing and Knowledge Graph technologies in Earth science research (Wang et al., 2018). Yu, Tong et al. integrated traditional Chinese medicine terms, documents, and databases into a large-scale Knowledge Graph for visualization, retrieval, and recommendation services, facilitating the sharing, interpretation, and utilization of knowledge in traditional Chinese medicine and health preservation. Furthermore, some researchers combine Knowledge Graphs with specific technologies to address particular issues (Yu et al., 2017). For example, Zhuotong Li et al. proposed a method using Knowledge Graphs and graph neural networks for fault localization, demonstrating excellent accuracy in analyzing network anomalies on the SDON platform, thereby supporting industrial-scale alarm analysis and fault localization (Li et al., 2021). Qi Chen et al. introduced a novel zero-shot learning method using Knowledge Graphs to classify a large amount of social text data, achieving superior performance over six state-of-the-art NLP deep learning models on COVID-19-related tweet datasets (Chen et al., 2021). This method resolves issues with deep learning models requiring extensive labeled data and enhances the processing and analysis capabilities of CPSS data. Yi Luo et al. proposed the DTKGIN model, combining Knowledge Graphs and intent graphs to predict drug-target interactions, effectively addressing issues of high sparsity and cold start, demonstrating its effectiveness in predicting potential drug-target interactions through case studies (Luo et al., 2024).

Traditional relational database knowledge management methods cannot effectively represent and store such knowledge, facing issues such as inability to integrate heterogeneous data, inefficiency in expressing data relationships, and inability to refine knowledge. Knowledge Graph is a method proposed by Google in 2012 to effectively express relationships between data through a semantic network (Singhal, 2012). Research and construction of Knowledge Graphs have penetrated various domains of social life such as education (Xue et al., 2024), healthcare systems (Shi et al., 2017; Wu et al., 2023), and smart cities (Liu et al., 2021). They represent entities, attributes, and their relationships from the natural world in a graphical form, allowing for a better expression of domain-specific information closer to human cognitive understanding (Wu et al., 2021a).Based on the scope of knowledge coverage, Knowledge Graphs are mainly classified into two types: vertical Knowledge Graphs and general Knowledge Graphs (Ji et al., 2021). General Knowledge Graphs cover a wide range of knowledge across multiple domains, primarily addressing common-sense issues in the real world. Currently influential general Knowledge Graphs include YAGO2 (Hoffart et al., 2011), DBpedia (Lehmann et al., 2015), Freebase (Bollacker et al., 2008), as well as Chinese Knowledge Graph websites such as CN-DBpedia (Xu et al., 2017) and Baidu Zhixin (Niu et al., 2011). General Knowledge Graphs can be likened to "structured encyclopedic knowledge bases," emphasizing breadth of knowledge and commonly used in search engines, constructed mainly in a bottom-up manner. Vertical Knowledge Graphs, also known as domain-specific Knowledge Graphs, are typically tailored to specific domains and require higher levels of depth and accuracy in knowledge representation. They serve as specialized knowledge repositories based on industry needs. For instance, Xindong Wu and colleagues proposed the Huapu-KG model, integrating HAO intelligence (human intelligence + artificial intelligence + organizational intelligence) for constructing genealogy Knowledge Graphs, addressing challenges of heterogeneity, autonomy, complexity, and evolution in genealogy data (Wu et al., 2021b). Wang, Chengbin et al. utilized hybrid corpora, Chinese word segmentation rules, and statistical methods to build a Knowledge Graph, demonstrating the potential and practicality of natural language processing and Knowledge Graph technologies in Earth science research (Wang et al., 2018). Yu, Tong et al. integrated traditional Chinese medicine terms, documents, and databases into a large-scale Knowledge Graph for visualization, retrieval, and recommendation services, facilitating the sharing, interpretation, and utilization of knowledge in traditional Chinese medicine and health preservation. Furthermore, some researchers combine Knowledge Graphs with specific technologies to address particular issues (Yu et al., 2017). For example, Zhuotong Li et al. proposed a method using Knowledge Graphs and graph neural networks for fault localization, demonstrating excellent accuracy in analyzing network anomalies on the SDON platform, thereby supporting industrial-scale alarm analysis and fault localization (Li et al., 2021). Qi Chen et al. introduced a novel zero-shot learning method using Knowledge Graphs to classify a large amount of social text data, achieving superior performance over six state-of-the-art NLP deep learning models on COVID-19-related tweet datasets (Chen et al., 2021). This method resolves issues with deep learning models requiring extensive labeled data and enhances the processing and analysis capabilities of CPSS data. Yi Luo et al. proposed the DTKGIN model, combining Knowledge Graphs and intent graphs to predict drug-target interactions, effectively addressing issues of high sparsity and cold start, demonstrating its effectiveness in predicting potential drug-target interactions through case studies (Luo et al., 2024).

Building a Knowledge Graph primarily involves four steps: data acquisition, knowledge extraction, knowledge fusion, and knowledge processing. Knowledge extraction, as a core step in the construction of a knowledge graph, is responsible for accurately obtaining useful knowledge from vast and complex data related to crop pests and diseases. This includes extracting information on symptoms, control measures, disease categories, distribution ranges, causes of disease, and locations of occurrence. Traditional methods of knowledge extraction typically rely on manually defined rules or statistical models, which excel in adaptability to specific domain languages and rules. However, they face limitations in performance and generalization capability when dealing with the complex contexts and diverse textual data of the current information explosion era. Deep learning effectively addresses many complex problems that traditional methods struggle with. Deep learning-based knowledge extraction significantly enhances the efficiency and accuracy of information processing. It automates the learning and extraction of abstract features from data, eliminating the need for manually designed feature engineering and greatly simplifying the data processing workflow. Moreover, deep learning models leverage robust language models and sequence modeling capabilities to better understand the grammar structure, semantic relationships, and contextual information within texts. Currently, many scholars extensively explore and apply deep learning models in the knowledge extraction process. For example, Bihui Yu et al. proposed a new entity recognition model based on character embedding, Iterated Dilated Convolutional Neural Networks (IDCNN), and Conditional Random Fields(CRF), demonstrating excellent performance in corpus experiments in the military equipment domain (Yu and Wei, 2020). Lample et al. introduced the Bi-LSTM-CRF model, which achieved high entity extraction evaluation metrics by incorporating a CRF module on top of a BiLSTM network in corpus datasets (Lample et al., 2016). In 2018, Luo et al. enhanced the Bi-LSTM-CRF model with an Attention mechanism, achieving an F1 score of over 90% on public document-level datasets (Luo et al., 2018). In the same year, Devlin et al. introduced the BERT pre-training model, generating deep bidirectional semantic representations based on input corpus to fully represent contextual information, thereby aiding multiple natural language processing tasks (Devlin et al., 2019). BERT's advent brought significant breakthroughs to entity recognition technology. In 2020, Google proposed the ALBERT model (Lan et al., 2019), reducing computational complexity compared to BERT and ensuring stable model training. Xiao Zhang et al. tackled challenges in the terahertz domain's QA system with long and short entity issues, proposing an entity recognition method based on ALBERT-BiLSTM-CRF (Zhang et al., 2020). They leveraged pre-trained deep bidirectional models to comprehensively understand sentence semantics, effectively improving the accuracy of long entity recognition.

After comparing existing entity recognition methods, we selected a fine-tuned ALBERT model with a BiLSTM-CRF layer as the final entity recognition model. We developed an efficient and accurate method for extracting knowledge about tomato leaf pests and diseases. The extracted triplet data was stored in a graph database, enabling the visualization and knowledge inference of the tomato leaf pest and disease knowledge graph. This allows complex agricultural knowledge to be clearly and intuitively presented to agricultural professionals. The main contributions of this study are as follows:

1. Addressing issues such as diverse data types, complex attributes and relationships, challenging deep-level association mining between data, and weak reasoning interpretability in traditional expert systems in tomato leaf disease and pest data. We utilized the ALBERT-BiLSTM-CRF model to extract knowledge about tomato diseases and pests, including symptoms, control measures, disease categories, distribution ranges, causes, and locations. By analyzing commonalities and characteristics among different diseases and pests, we organically linked information about various tomato diseases and pests, reducing redundant work and enhancing the efficiency and comprehensiveness of tomato disease and pest control.
2. To address the lack of open-source specialized corpora for tomato leaf disease and pest issues, we extracted various

data about tomato diseases and pests from national or local standard documents. We employed web scraping techniques to gather data from professional agricultural knowledge websites and scanned texts from specialized books and literature on tomato leaf disease and pest control. With assistance from domain experts, we completed data preprocessing to form the domain-specific corpora required for constructing the tomato leaf disease and pest Knowledge Graph. The ALBERT-BiLSTM-CRF model was used to extract knowledge from sample data stored in a Neo4j database, establishing a rich, logically strong, and widely applicable Knowledge Base for tomato leaf disease and pest.

3. Based on the above solutions, we designed a digital diagnostic system for tomato leaf disease and pest integrated with advanced software development frameworks. This system combines deep learning algorithms and graph databases to process, analyze, and store information about tomato leaf disease and pest, providing users with precise prevention and control recommendations.

2 Materials and methods

2.1 Knowledge graph construction process

2.1.1 Knowledge graph schema construction

A well-designed model layer facilitates the reuse of domain knowledge, creating a shared understanding of domain information and knowledge between humans and machines. A well-structured hierarchical ontology ensures the quality of knowledge extraction during the construction of the knowledge graph data layer, playing an irreplaceable role throughout the entire knowledge graph construction process (Guo et al., 2024). Referring to the ontology construction method from Stanford, we mainly followed six steps to construct the ontology for the knowledge graph in this study: determining the domain and scope, listing the main elements, defining the terms and the relationships between terms, defining classes and the class hierarchy, defining properties and relationship constraints, and instantiating the ontology.

The knowledge graph we constructed is focused on the agricultural domain, specifically on the ontology construction related to tomato leaf pests and diseases. We have tried to summarize as many elements as possible that should appear in the knowledge graph. Table 1 lists the main elements included in the knowledge graph for the domain of tomato leaf pests and diseases.

Next, we organize the domain-specific terms and the relationships and attributes between them. Table 2 lists some professional terms and their relationships for two sub-domain knowledge graphs. Based on this, we define constraints and regulations for class relationships, intrinsic properties, and extrinsic properties. Finally, the defined properties, relationships, and constraints are used to instantiate the ontology. The hierarchical structure diagram of the tomato leaf pest and disease model layer, constructed using the Protégé tool, is shown in Figure 1.

TABLE 1 Some elements included in the knowledge graph for the domain of tomato leaf pests and diseases.

Subdomain Knowledge Graph	Main Elements
Tomato Leaf Diseases	Disease name, alias, symptoms, physical control, chemical control, biological control, disease category, distribution range, causes, affected parts, pathogen, etc.
Tomato Leaf Pests	Pest name, alias, phylum, class, order, family, genus, morphological characteristics, habits, symptoms, physical control, chemical control, biological control, occurrence area, affected parts, foreign name, etc.

2.1.2 Construction of the knowledge graph data layer

We used natural language processing methods and techniques to construct the data layer for tomato leaf pests and diseases. The sequence labeling strategy we selected is based on the BIO method, where B represents the beginning part of a tomato leaf pest or disease entity, I represents a non-beginning part of the entity, and O represents a non-entity, as shown in Table 3.

The quality of data sample annotation determines the effectiveness of the model's entity extraction. Considering that manual annotation is time-consuming, labor-intensive, and often lacks accuracy, we used the entity annotation software Colabeler to assist in annotating preprocessed data samples. During the annotation process, controversial areas were discussed with domain experts to ensure the accuracy of the results.

The annotated sample data is saved with files in the ".ann" format, as shown in Figure 2. In the ".ann" file, T1 represents the unique identifier or label for the annotation. "DiseaseName" indicates that the annotation marks a disease name in the text, with the number "0" representing the starting position of the annotation (i.e., starting from the 0th character), and the number "2" representing the ending position of the annotation (i.e., ending

TABLE 2 Sub-domain knowledge graph terms and relationships between terms.

Subdomain knowledge graph	Technical terms	Relationships
Tomato leaf diseases	Alias, Chinese name, Foreign name, Stem, Fruit, Flower, Ventilation and drainage, Proper spacing	Aliases, Affected parts, Harmful periods, Control methods, etc.
Tomato Leaf Pests	Alias, Leaves, Lepidoptera, Phylum Mollusca, Class Gastropoda, Order Stylommatophora, Family Limacidae, Genus Liriomyza Coleoptera, Natural enemies, Traps, Manual trapping	Aliases, Affected parts, Classification, Control methods, etc.

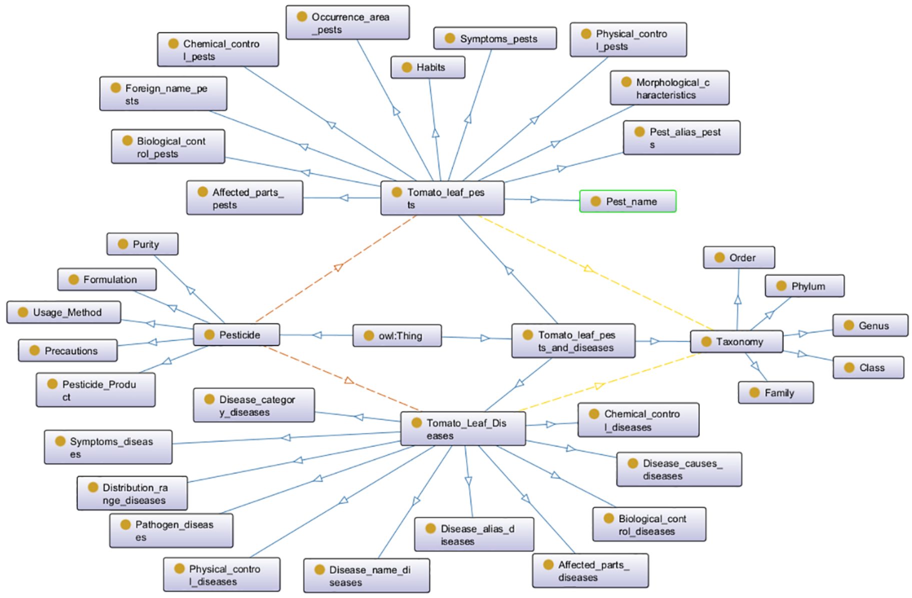


FIGURE 1
Ontology hierarchy diagram for the tomato leaf pests and diseases knowledge graph.

at the 2nd character). In the “10001.ann” file, T2 to T12 are similar to T1.

Algorithm 1 outlines the process for generating BIO format data required for the NER task. The “Generate_BIO” algorithm traverses all annotated “.ann” files and corresponding “.txt” files in the specified directory, converting the data in each annotation file into BIO format (training data files).

```
Input:”.ann” files and “.txt” files
Output:BIO Format Data
DEFINE root_dir、stream_path
file_list = glob.glob(root_dir + ‘/*.ann’)
FOR
  ann_path = normalize_path(ann_path)
  txt_path = convert_to_txt_path(ann_path)
  .strip() Remove whitespace characters from the ends
  of text or label strings.
  IF ann!= ‘ ’
    Assign the label “O” to all text.
    FOR
      TRY:
        T, typ, word = split_tabs_and_strip(line)
        t, s, e = typ.split()
        s, e = convert_positions_to_int(s, e)
        label[s] = concatenate(‘B-’, t)
        while s< subtract(e, 1):
        s = add(s, 1)
        label[s] = concatenate(‘I-’, t)
      EXCEPT:
        continue
      Connecting Text and BIO Labeling Using Spaces
    END
  print (ann_path, e)
END
```

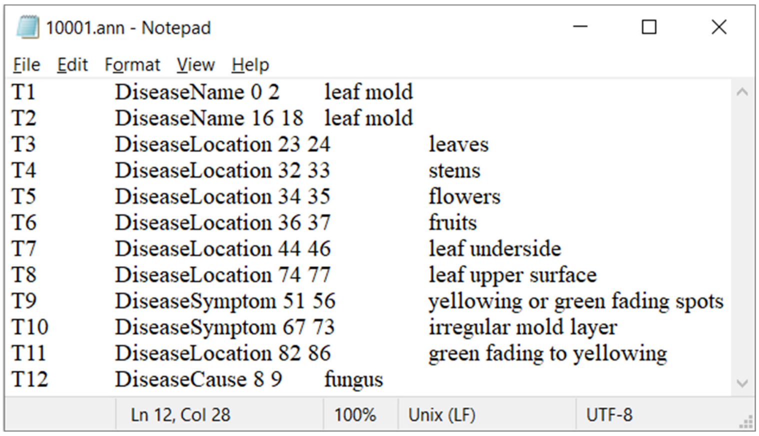
Algorithm 1. Generate_BIO.

TABLE 3 BIO data annotation example.

Character	BIO annotation
powdery	B-DiseaseName
mildew	I-DiseaseName
mainly	O
affects	O
leaves	B-DiseaseLocation

2.2 ALBERT-BiLSTM-CRF Model

We use the ALBERT-BiLSTM-CRF model to perform entity extraction tasks on tomato leaf pest and disease data samples. The model consists of three parts: ALBERT(A Lite Bidirectional Encoder Representations from Transformers), BiLSTM(Bi-directional Long Short-Term Memory), and CRF(Conditional Random Field), with the model structure illustrated in Figure 3. ALBERT is a pre-trained language model capable of understanding and processing natural language text; BiLSTM is a type of recurrent neural network designed for handling sequential data; and CRF is a conditional random field used to identify structural patterns in sequences. For example, in the sample “Leaf mold mainly affects leaves”, we use ALBERT as the pre-trained model to capture the semantic information of each word in the given context. First, the model tokenizes the sentence and converts it into word



```
T1      DiseaseName 0 2       leaf mold
T2      DiseaseName 16 18     leaf mold
T3      DiseaseLocation 23 24         leaves
T4      DiseaseLocation 32 33         stems
T5      DiseaseLocation 34 35         flowers
T6      DiseaseLocation 36 37         fruits
T7      DiseaseLocation 44 46         leaf underside
T8      DiseaseLocation 74 77         leaf upper surface
T9      DiseaseSymptom 51 56          yellowing or green fading spots
T10     DiseaseSymptom 67 73          irregular mold layer
T11     DiseaseLocation 82 86         green fading to yellowing
T12     DiseaseCause 8 9      fungus
```

FIGURE 2
'10001.ann' file.

embeddings, including ['*leaf mold*', '*mainly*', ' *affects*', '*leaves*']. ALBERT processes these word embeddings through its self-attention mechanism, considering the contextual information of each word in the sentence, and generates contextually relevant representations, such as ['*leaf mold_emb*', '*mainly_emb*', '*affects_emb*', '*leaves_emb*']. These representations are then used as input to the BiLSTM model. BiLSTM models the sequence of each word's contextual representation to capture the semantic information of words in the sentence. The output of the ALBERT layer, ['*leaf mold_emb*', '*mainly_emb*', '*affects_emb*', '*leaves_emb*'], is fed directly into the BiLSTM. The BiLSTM outputs a sequence of hidden states for each word, such as ['*leaf mold_hidden*', '*mainly_hidden*', '*affects_hidden*', '*leaves_hidden*'], which reflect each word's semantic representation in its context. The BiLSTM output sequence is typically used as input to the CRF layer. The CRF layer combines the tag probability sequences output by the BiLSTM with a predefined transition matrix and uses the Viterbi algorithm to find the optimal tag sequence. For the entity recognition task of leaf mold, the optimal tag sequence might be ['*B-DiseaseName*', '*O*', '*O*', '*B-DiseaseLocation*'].

2.2.1 ALBERT

ALBERT, which stands for A Lite BERT, is a lightweight version of the BERT model. In the field of deep learning, the ALBERT model has gained significant attention for its outstanding performance (Lan et al., 2020). The ALBERT model consists of two parts: MASK preprocessing and a bidirectional Transformer, with the model structure illustrated in Figure 4.

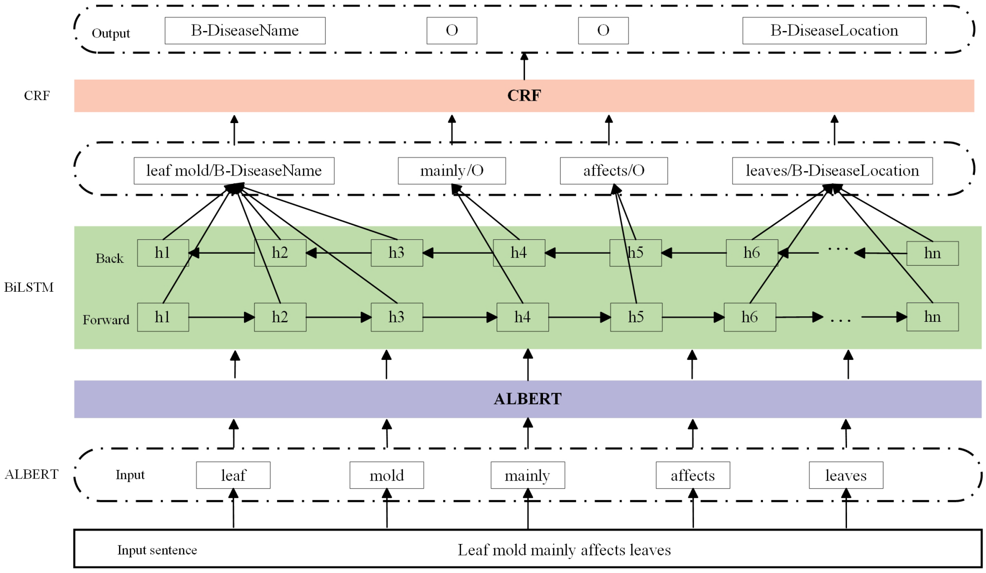


FIGURE 3
ALBERT BiLSTM CRF model structure.

The input representation for ALBERT consists of three embeddings: token embeddings (word embeddings), sentence embeddings, and position embeddings, which are concatenated to clearly represent a text sentence within a token sequence, as shown in Figure 5. ALBERT captures both word-level and sentence-level representations through two tasks: Masked Language Model (MLM) and Next Sentence Prediction (NSP), and it is jointly trained on these tasks. In the MLM task, during preprocessing, the target text is first segmented at the character level, and then random MASK operations are applied to the segmented characters. By randomly masking characters in the training text, the Transformer model is forced to build associations within the context of each input character unit, enhancing the ability to capture text features. Additionally, special tokens [CLS] and [SEP] are added at the beginning of the text and between sentences, respectively, to form the final sequence vector for further processing by the Transformer model. For example, in the sentence "*Leaf mold mainly affects leaves*", if '*leaves*' is randomly masked, the resulting sequence vector might be "[CLS] *Leaf mold mainly affects leaves [MASK]*[SEP]". In the NSP task, the model receives two sentences as input and predicts whether the second sentence is the subsequent sentence of the first one. This task helps the model understand the logical relationship and coherence between sentences.

The Transformer part is the key structure of the ALBERT model, mainly consisting of input, output, and the Encoder section. The Encoder section includes one Self-Attention layer, one Feed Forward layer, and two Add & Norm layers. The Self-Attention layer is the core of the Encoder, receiving the word vector sequence transformed by the embedding layer as input. It generates contextually relevant representations for each word, with the computation method described in Formula 1. In the sentence "*Leaf mold mainly affects leaves*", the Self-Attention layer balances and integrates the semantic information of each word, producing updated word vector representations such as "'*leaf mold*', '*mainly*', '*affects*', '*leaves*'".

$$\boldsymbol{Attention}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \boldsymbol{softmax}(\frac{\boldsymbol{QK^T}}{\sqrt{\boldsymbol{d_K}}})\boldsymbol{V} \quad (1)$$

In Formula 1, Q, K, and V are the input word vector matrices, and d_K is the dimension of the K matrix. The Feed Forward layer performs nonlinear transformations and mappings on the context-aware vectors of each word, such as "leaf mold", "mainly", "affects", and "leaves", to enhance their feature

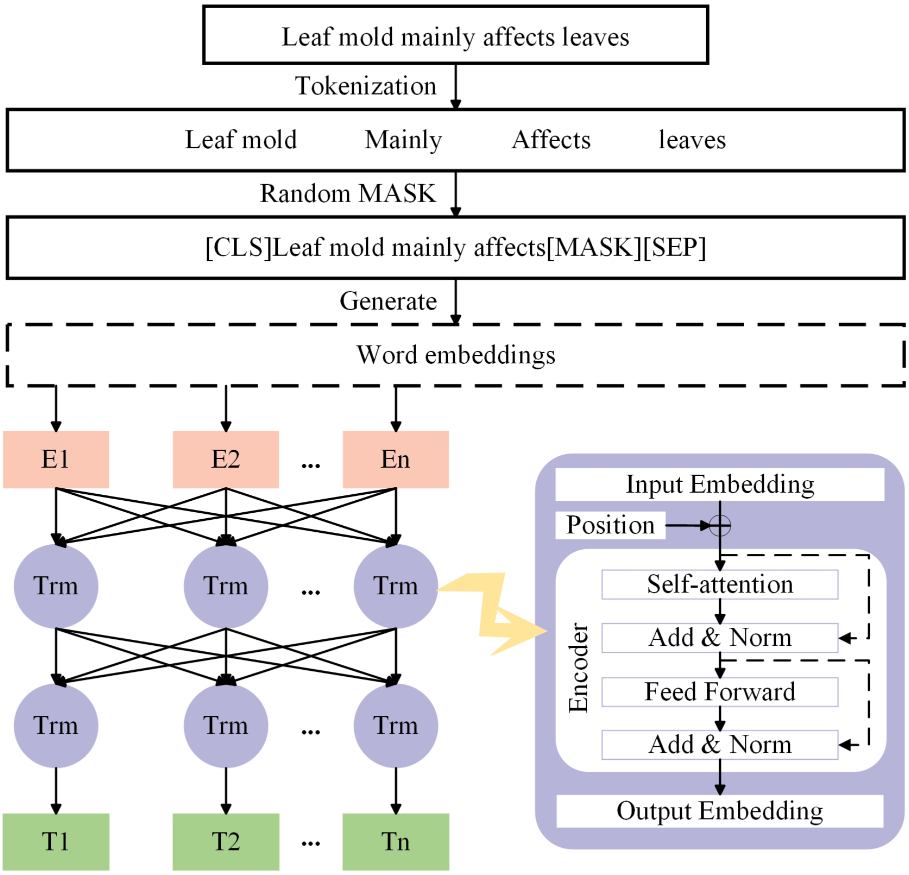


FIGURE 4
ALBERT model structure.

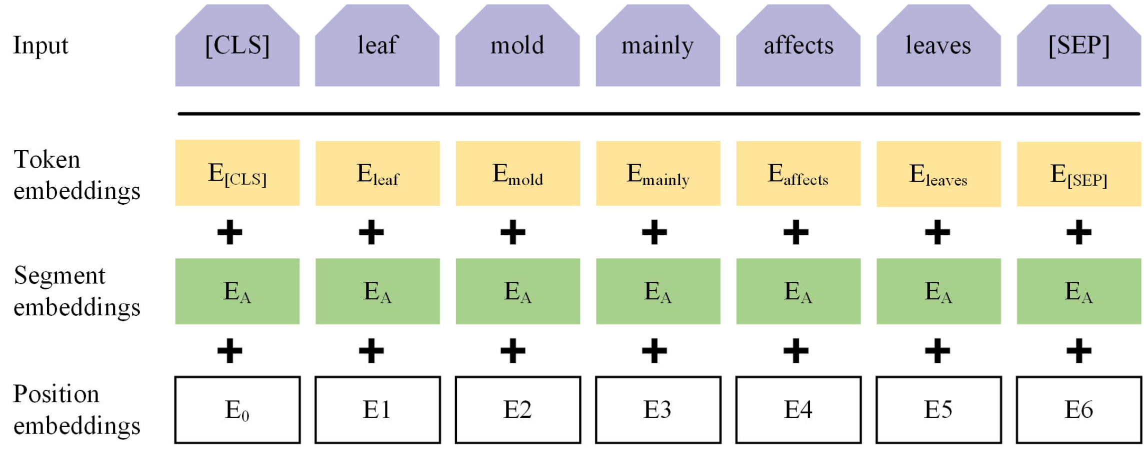


FIGURE 5
ALBERT input vector representation.

representation. The Add & Norm layer combines the original input with the processed vectors through residual connections and performs layer normalization to ensure model training stability and accelerate convergence.

2.2.2 BiLSTM module

LSTM, which stands for Long Short-Term Memory, is essentially a type of recurrent neural network (RNN). It is primarily composed of four parts: the input gate, output gate, forget gate, and memory cell (Ma et al., 2021).

The forget gate determines how much of the previous cell state c_{t-1} should be retained in the current cell state c_t at time t. The calculation method is given by Formula 2:

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

In Formula 2, f_t represents the operation of the forget gate at time t, σ is the sigmoid function, W_f is the parameter matrix, h_{t-1} is the previous state vector, x_t is the input vector at time t, and $[h_{t-1}, x_t]$ denotes the concatenation of the two vectors. b_f is the bias term. The calculation methods for the input gate are given by Formulas 3–5:

$$\boldsymbol{i_t = \sigma(W_i \bullet [h_{t-1}, x_t] + b_i)} \tag{3}$$

$$\boldsymbol{\widetilde{c_t} = tanh(W_c \bullet [h_{t-1}, x_t] + b_c)} \tag{4}$$

$$\boldsymbol{c_t = f_t \otimes c_{t-1} + i_t \otimes \widetilde{c_t}} \tag{5}$$

In Formula 3, i_t represents the operation of the input gate at time t, which calculates how much of the current input x_t should be retained in the cell state c_t. Here, σ is the sigmoid function, W_i is the parameter matrix, and b_i is the bias term. The cell state c_t is computed using Formulas 4, 5, where the previous cell state c_{t-1} is element-wise multiplied by the forget gate f_t, and the current input cell state $\widetilde{c_t}$ is element-wise multiplied by the input gate i_t, with the two products then summed. The output gate controls the cell state c_t, and its computation is given by Formulas 6, 7:

$$\boldsymbol{o_t = \sigma(W_o \bullet [h_{t-1}, x_t] + b_o)} \tag{6}$$

$$\boldsymbol{h_t = o_t \otimes tanhc_t} \tag{7}$$

In the formulas, o_t represents the operation of the output gate at time t, and h_t denotes the final cell output value.

Given that unidirectional LSTM networks can only process information in one direction and cannot account for context from both directions, researchers later introduced BiLSTM, or Bidirectional Long Short-Term Memory networks. BiLSTM employs one forward LSTM and one backward LSTM, and merges the outputs from the same time step. This allows the model to capture dependencies between words in the text sequence and the contextual information, providing a more accurate and comprehensive feature representation for tasks such as entity extraction and classification (Lu et al., 2023). The specific structure is shown in Figure 6.

2.2.3 CRF module

In the entity extraction process, the CRF module is responsible for optimizing the annotation sequences output by the BiLSTM. It can label disease-related terms in the text data and perform joint modeling of these labels. By constructing a probabilistic model based on a designed score function, the CRF computes probabilities to select the optimal path among possible paths, resulting in accurate entity extraction.

3 Results and discussion

3.1 Data collection and preprocessing

Regarding tomato leaf pests and diseases, there is currently no complete and reliable dataset available for knowledge graph construction. This lack of data can impact the accuracy and reliability of analysis models, as the size and diversity of data samples are crucial factors determining model performance. To

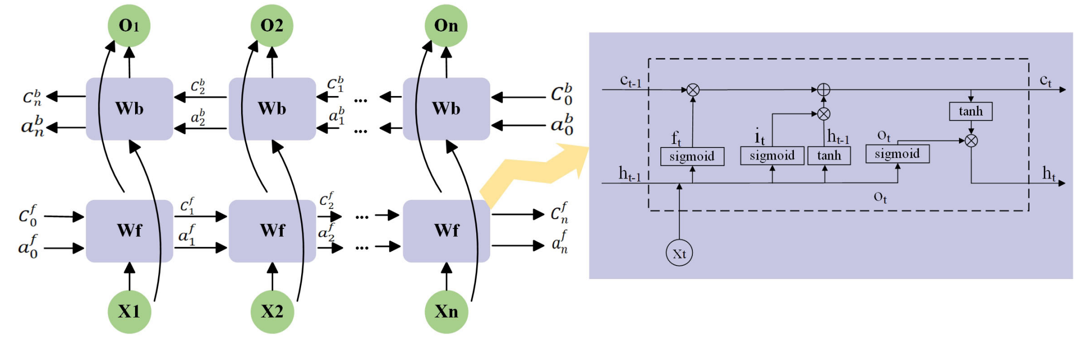


FIGURE 6
BiLSTM model structure.

build a reliable dataset for tomato leaf diseases, we used web scraping techniques to gather a large amount of semi-structured and unstructured text data on tomato pests and diseases from various websites, including: China Agricultural Information Technology Network (https://cast.caas.cn/index.html), National Agricultural Science Data Center (https://www.agridata.cn/#/home), China Pesticide Information Network (http://www.chinapesticide.org.cn/), Baidu Encyclopedia and National Standard Information Public Service Platform (https://std.samr.gov.cn/gb/). Additionally, we scanned electronic books such as Illustrated Guide to Tomato Pest and Disease Diagnosis and Control, Tomato Cultivation and Pest Control Technology Research and New Technologies for Tomato Pest and Disease Control to obtain text data on tomato pests and diseases. The data samples obtained, as shown in Figure 7, indicate clear relationships between different entities. Taking speckle blight disease as an example, speckle blight serves as the disease name entity, disease location the entities representing infection sites such as leaves, stems, and fruits. The disease symptoms, such as nearly circular grayish-white spots with slightly darkened edges, grayish-white centers, and brown edges. These symptom entities describe the external characteristics of speckle blight disease. Additionally, the disease cause entity indicates that the disease overwinters through mycelium and conidia on crop residues and perennial solanaceous weeds. In terms of control, there are multiple methods and related entities. Agricultural control entities include the use of disease-resistant varieties, practicing crop rotation with non-Solanaceae for 3-4 years, and applying phosphorus and potassium fertilizers. Physical control entities involve sun-drying seeds and soaking them in warm water. Chemical control entities consist of various pesticides, such as 75% chlorothalonil wettable powder at 1200 times dilution and 64% oxadixyl wettable powder at 500 times dilution, which are used for chemical control of the disease.

Before extracting entities related to tomato leaf pests and diseases, it is necessary to preprocess the dataset. A good preprocessing process ensures the high quality and accuracy of the dataset, facilitating subsequent tokenization and annotation work. With the assistance of domain experts, the preprocessing of the raw sample data includes normalization, elimination of irrelevant information and noise data, and correction of possible language errors to ensure the consistency and reliability of the data. Through systematic preprocessing, we constructed a NER dataset for tomato leaf diseases and pests—the Tomato Leaf Pest and Disease Dataset—providing reliable data support for the construction of a knowledge graph. The dataset includes a training set (1160) and a test set (290), split in an 8:2 ratio.

3.2 Evaluation metrics

We set up a training environment suitable for six models: BiLSTM-CRF, 1DCNN-CRF, 1DCNN-CRF-2, BiLSTM-Attention-CRF, BERT-BiLSTM-CRF, and ALBERT-BiLSTM-CRF. To evaluate the performance of the models, we used the same three evaluation metrics commonly found in existing NER papers: Precision (P), Recall (R), and the F1-score as the evaluation standards. The formulas are defined as Equations 8–10, where TEN represents the number of correctly identified entities, FEN is the number of falsely identified non-entities, and NC is the number of correctly identified entities that were not recognized.

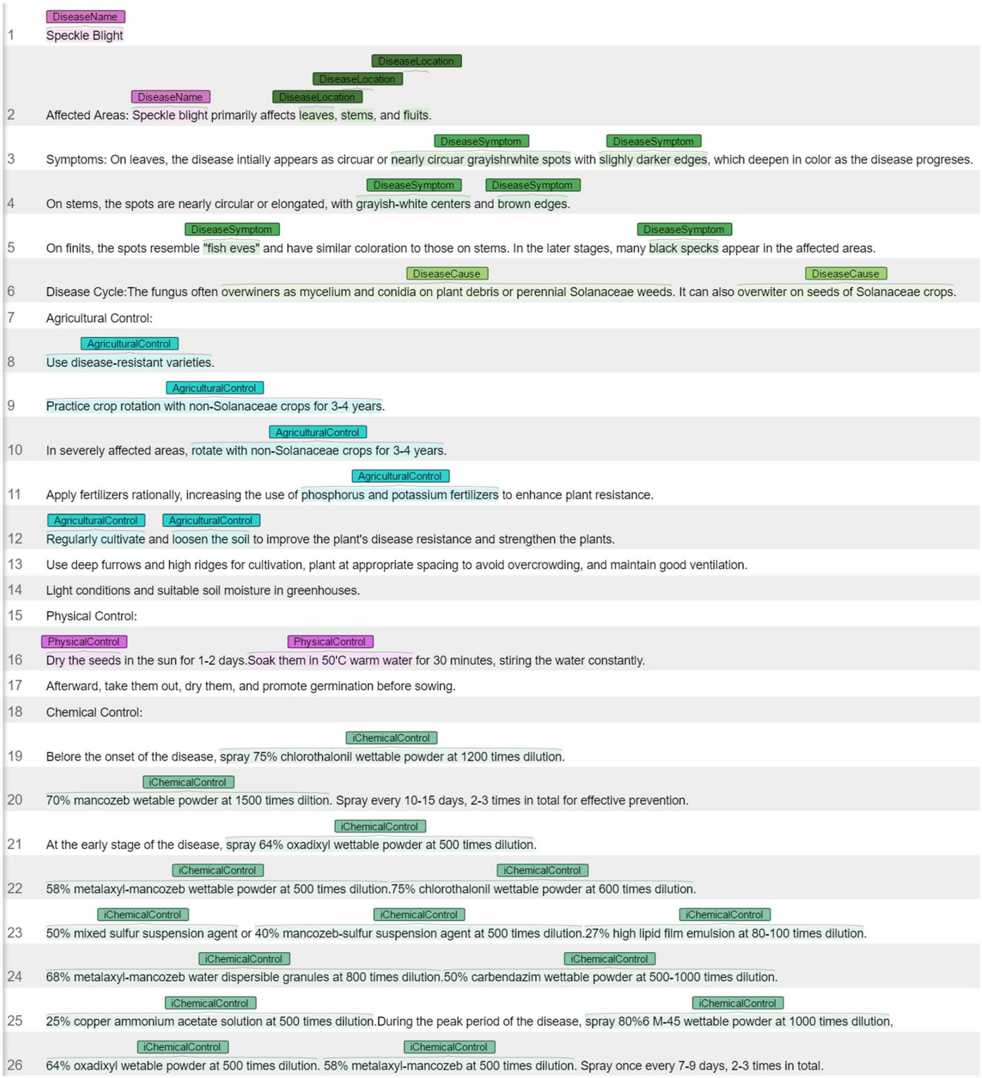
$$P = \frac{TEN}{TEN + FEN} \times 100\,\% \tag{8}$$

$$R = \frac{TEN}{TEN + NC} \times 100\,\% \tag{9}$$

$$F1 = \frac{2 \times P \times R}{P + R} \times 100\,\% \tag{10}$$

3.3 Parameter settings

To ensure the reliability of the experimental results, each model was initialized with the same parameters before the



Speckle Blight

Affected Areas: Speckle blight primarily affects leaves, stems, and fiuits.

Symptoms: On leaves, the disease intially appears as circuar or nearly circuar grayishrwhite spots with slighly darker edges, which deepen in color as the disease progreses.

On stems, the spots are nearly circular or elongated, with grayish-white centers and brown edges.

On finits, the spots resemble "fish eves" and have similar coloration to those on stems. In the later stages, many black specks appear in the affected areas.

Disease Cycle:The fungus often overwiners as mycelium and conidia on plant debris or perennial Solanaceae weeds. It can also overwiter on seeds of Solanaceae crops.

Agricultural Control:

Use disease-resistant varieties.

Practice crop rotation with non-Solanaceae crops for 3-4 years.

In severely affected areas, rotate with non-Solanaceae crops for 3-4 years.

Apply fertilizers rationally, increasing the use of phosphorus and potassium fertilizers to enhance plant resistance.

Regularly cultivate and loosen the soil to improve the plant's disease resistance and strengthen the plants.

Use deep furrows and high ridges for cultivation, plant at appropriate spacing to avoid overcrowding, and maintain good ventilation.

Light conditions and suitable soil moisture in greenhouses.

Physical Control:

Dry the seeds in the sun for 1-2 days.Soak them in 50'C warm water for 30 minutes, stiring the water constantly.

Afterward, take them out, dry them, and promote germination before sowing.

Chemical Control:

Belore the onset of the disease, spray 75% chlorothalonil wettable powder at 1200 times dilution.

70% mancozeb wetable powder at 1500 times diltion. Spray every 10-15 days, 2-3 times in total for effective prevention.

At the early stage of the disease, spray 64% oxadixyl wettable powder at 500 times dilution.

58% metalaxyl-mancozeb wettable powder at 500 times dilution.75% chlorothalonil wettable powder at 600 times dilution.

50% mixed sulfur suspension agent or 40% mancozeb-sulfur suspension agent at 500 times dilution.27% high lipid film emulsion at 80-100 times dilution.

68% metalaxyl-mancozeb water dispersible granules at 800 times dilution.50% carbendazim wettable powder at 500-1000 times dilution.

25% copper ammonium acetate solution at 500 times dilution.During the peak period of the disease, spray 80%6 M-45 wettable powder at 1000 times dilution,

64% oxadixyl wetable powder at 500 times dilution. 58% metalaxyl-mancozeb at 500 times dilution. Spray once every 7-9 days, 2-3 times in total.

FIGURE 7
Raw text data on tomato leaf pests and diseases.

TABLE 4 Model parameter settings.

Parameter Name	Model Name	Parameter Value
Optimizer	All	Adam
Loss_Function	All	crf.sparse_loss
Learning_Rate	All	1e-5
Lstm_Units	All	128
Batch_Size	ALBERT-BiLSTM-CRF	16
	Others	64
Drop_Rate	ALBERT-BiLSTM-CRF	0.1
	Others	0.5
Max_Len	All	200
Epoch	All	30

study, as shown in Table 4. In this table, "Optimizer"refers to the optimization algorithm, "Loss_Function" refers to the loss function, "Learning_Rate"refers to the learning rate, "Lstm_Units" refers to the dimensionality of the output space of the LSTM network, "Batch_Size" refers to the amount of data processed by the model at one time, "Drop_Rate" refers to the dropout rate, "Max_Len" refers to the maximum length of the data processed, and "Epoch" refers to the training cycles of the model.

3.4 Experimental results analysis

The performance of the six models—BiLSTM-CRF, 1DCNN-CRF, 1DCNN-CRF-2, BiLSTM-Attention-CRF, BERT-BiLSTM-CRF, and ALBERT-BiLSTM-CRF—in entity recognition on the Tomato Leaf Disease Dataset is shown in Table 5.

TABLE 5 Comparison of entity recognition performance for tomato leaf pests and diseases.

Model Name	Precision/%	Recall/%	F1-score/%
BiLSTM-CRF	87.73	86.31	87.02
1DCNN-CRF	85.39	87.14	86.05
1DCNN-CRF-2	84.92	89.74	87.26
BiLSTM-Attention-CRF	85.15	84.83	85.01
BERT-BiLSTM-CRF	93.10	90.84	91.96
ALBERT-BiLSTM-CRF	95.10	95.03	95.48

As shown in Figure 8, the traditional BiLSTM-CRF and 1DCNN-CRF models have relatively slow convergence speeds, approaching convergence only around the 12th epoch, with a recall rate of approximately 0.85. The BiLSTM-Attention-CRF model converges the fastest but has a lower final recall rate of about 0.80. The ALBERT-BiLSTM-CRF model achieves both fast convergence and the highest recall rate, approximately 0.95. The BERT-BiLSTM-CRF model follows, with a recall rate of around 0.90, while the improved 1DCNN-CRF-2 also has a high recall rate of about 0.89. This indicates that pre-trained models (BERT or ALBERT) significantly enhance model performance and convergence speed compared to the base BiLSTM-CRF model. Additionally, incorporating attention mechanisms or improving CNN structures also contributes to performance improvements. Considering model performance, convergence speed, and complexity, we use the ALBERT-BiLSTM-CRF model for the knowledge extraction task in the subsequent system implementation.

3.5 System design and implementation

The tomato leaf disease and pest knowledge graph system we designed is developed based on the Windows 10 64-bit operating system, using VSCode as the integrated development tool, Vue3 as the frontend development framework, Neo4j as the graph database, and ECharts for knowledge graph visualization. The system supports cross-platform usage (both PC and mobile), ensuring a good user experience across various devices. Users can select a specific pest or disease name from the list and then view the related symptoms visually. As shown in Figures 9A, B, when the system receives the query for "*yellow leaf curl disease*", it first executes the corresponding Cypher query to retrieve information about nodes related to the "*yellow leaf curl disease*" entity.

ECharts is then used to render the entity and node information, and the results are finally displayed to the user in Figures 10A, B.

4 Dicussion

For the tomato leaf pest and disease domain, we collected and organized relevant data to construct a dataset, TLP2D, containing approximately 10,400 entities and 1,450 samples. Compared to other corpora in the same field, our dataset not only includes a broader range of categories, but also offers more detailed descriptions, providing timely decision analysis for tomato leaf pest and disease control. By comparing different entity extraction models, we found that the ALBERT-BiLSTM-CRF model performs well on TLP2D. This model leverages the strengths of ALBERT, BiLSTM, and CRF models to effectively extract entities from

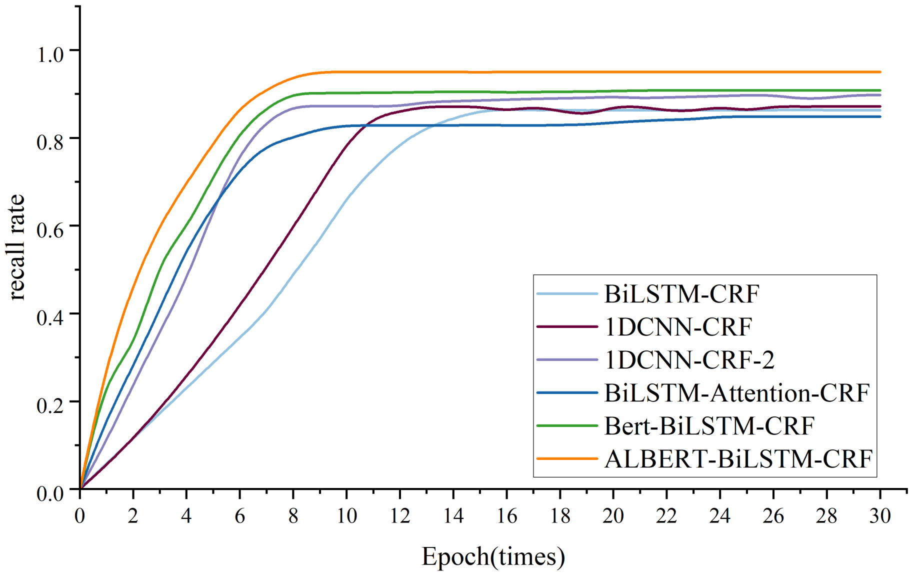


FIGURE 8
Comparison of model recall rates.

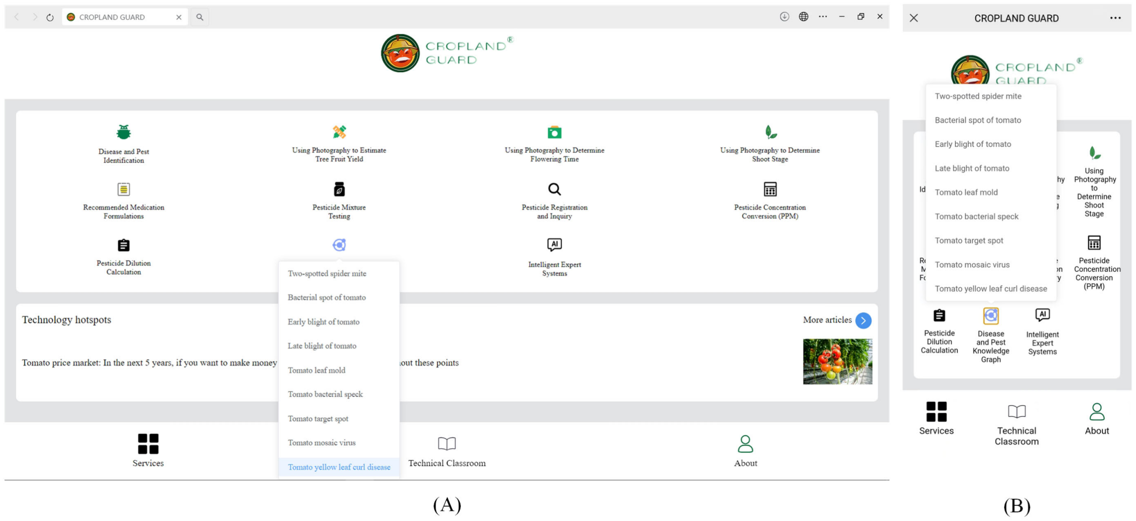

CROPLAND GUARD
Disease and Pest Identification
Using Photography to Estimate Tree Fruit Yield
Using Photography to Determine Flowering Time
Using Photography to Determine Shoot Stage
Recommended Medication Formulations
Pesticide Mixture Testing
Pesticide Registration and Inquiry
Pesticide Concentration Conversion (PPM)
Pesticide Dilution Calculation
Intelligent Expert Systems
Two-spotted spider mite
Bacterial spot of tomato
Early blight of tomato
Late blight of tomato
Tomato leaf mold
Tomato bacterial speck
Tomato target spot
Tomato mosaic virus
Tomato yellow leaf curl disease
Technology hotspots
More articles
Tomato price market: In the next 5 years, if you want to make money ... out these points
Services
Technical Classroom
About
Disease and Pest Knowledge Graph
(A)
(B)

FIGURE 9
Selecting a specific pest or disease: **(A)** PC, **(B)** Mobile.

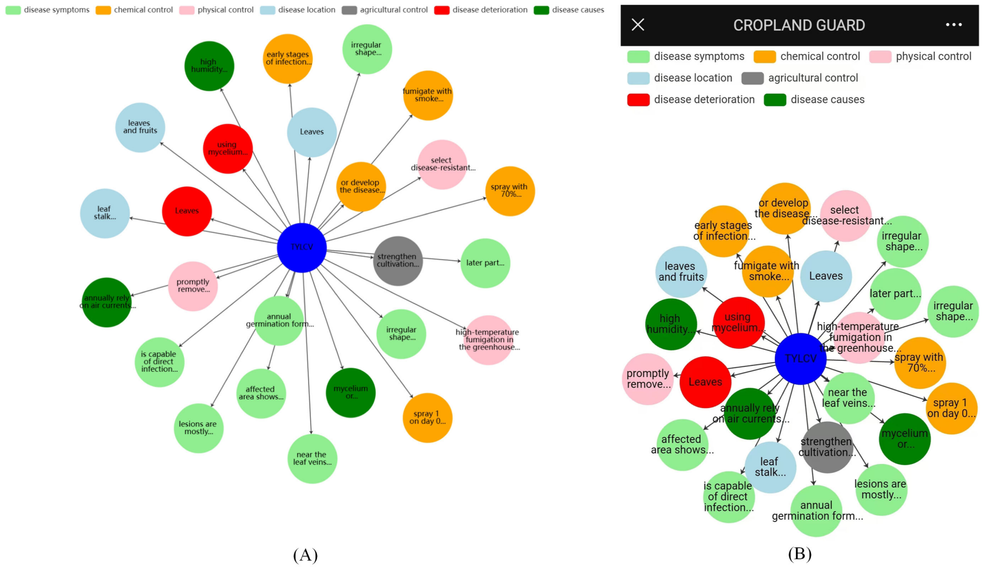


FIGURE 10
Pest and disease entity query display: **(A)** PC, **(B)** Mobile.

TLP2D, supporting the construction of a tomato leaf pest and disease knowledge graph and providing a reference for knowledge graph construction in other fields. The development of the tomato leaf pest and disease digital diagnostic system offers precise pest and disease control recommendations for farmers, lowers the barrier to professional knowledge sharing, and introduces new ideas for pest and disease management. However, we also recognize that there is room for improvement in this work. Future efforts will focus on expanding the domain scope, optimizing data annotation methods, and continuously fine-tuning the NER model to better serve agriculture.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

KW: Conceptualization, Data curation, Formal Analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. YM: Data curation, Software, Writing – review & editing. XW: Formal Analysis, Investigation, Methodology, Writing – review & editing. YL: Data curation, Software, Writing – review & editing. FL: Supervision, Writing – review & editing. HS: Funding acquisition, Project administration, Resources, Writing – review & editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This research was supported by the Shanxi Provincial Basic Research Program (202103021224173, Key Technology Research on Tomato Picking Robot Platform Based on 5G and Artificial Intelligence Technology) and the Shanxi Provincial Key R&D Project (20220214060102, Research on Intelligent Decision-Making System for Precision Agriculture).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Abdullah, H. M., Mohana, N. T., Khan, B. M., Ahmed, S. M., Hossain, M., Islam, K. S., et al. (2023). Present and future scopes and challenges of plant pest and disease (P&D) monitoring: Remote sensing, image processing, and artificial intelligence perspectives. *Remote Sens. Applications: Soc. Environ.* 32, 100996. doi: 10.1016/j.rsase.2023.100996

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In: *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, Vancouver, Canada, 9-12. New York: Association for Computing Machinery, 1247–1250.

Chen, Q., Wang, W., Huang, K., and Coenen, F. (2021). Zero-shot text classification via knowledge graph embedding for social media data. *IEEE Internet Things J.* 9, 9205–9213. doi: 10.1109/JIOT.2021.3093065

Devlin, J., Chang, M. W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 1, 4171–4186. doi: 10.18653/v1/N19-1423

Guo, Y., Wang, L., Zhang, Z., Cao, J., Xia, X., and Liu, Y. (2024). Integrated modeling for retired mechanical product genes in remanufacturing: A knowledge graph-based approach. *Advanced Eng. Inf.* 59, 102254. doi: 10.1016/j.aei.2023.102254

Hoffart, J., Suchanek, F. M., Berberich, K., Lewis-Kelham, E., De Melo, G., and Weikum, G. (2011). YAGO2: exploring and querying world knowledge in time, space, context, and many languages. In *Proceedings of the 20th international conference companion on World wide web*, 229–232.

Ji, S., Pan, S., Cambria, E., Marttinen, P., and Philip, S. Y. (2021). A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Trans. Neural Networks Learn. Syst.* 33, 494–514. doi: 10.1109/TNNLS.2021.3070843

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named entity recognition. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 260–270. doi: 10.18653/v1/N16-1030

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. (2020). ALBERT: A lite BERT for self-supervised learning of language representations. *arxiv preprint* arxiv:1909.11942.

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., et al. (2015). Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web* 6, 167–195. doi: 10.3233/SW-140134

Li, Z., Zhao, Y., Li, Y., Rahman, S., Wang, F., Xin, X., et al. (2021). Fault localization based on knowledge graph in software-defined optical networks. *J. Lightwave Technol.* 39, 4236–4246. doi: 10.1109/JLT.2021.3071868

Liu, J., Li, T., Ji, S., Xie, P., Du, S., Teng, F., et al. (2021). Urban flow pattern mining based on multi-source heterogeneous data fusion and knowledge graph embedding. *IEEE Trans. Knowledge Data Eng.* 35, 2133–2146. doi: 10.1109/TKDE.2021.3098612

Lu, Y., Tang, L., Chen, C., Zhou, L., Liu, Z., Liu, Y., et al. (2023). Reconstruction of structural long-term acceleration response based on BiLSTM networks. *Eng. structures* 285, 1.1–1.14. doi: 10.1016/j.engstruct.2023.116000

Luo, Y., Duan, G., Zhao, Q., Bi, X., and Wang, J. (2024). DTKGIN: Predicting drug-target interactions based on knowledge graph and intent graph. *Methods* 226, 21–27. doi: 10.1016/j.ymeth.2024.04.010

Luo, L., Yang, Z., Yang, P., Zhang, Y., Wang, L., Lin, H., et al. (2018). An attention-based BiLSTM-CRF approach to document-level chemical named entity recognition. *Bioinformatics* 34, 1381–1388. doi: 10.1093/bioinformatics/btx761

Ma, C., Dai, G., and Zhou, J. (2021). Short-term traffic flow prediction for urban road sections based on time series analysis and LSTM_BILSTM method. *IEEE Trans. Intelligent Transportation Syst.* 23, 5615–5624. doi: 10.1109/TITS.2021.3055258

Niu, X., Sun, X., Wang, H., Rong, S., Qi, G., and Yu, Y. (2011). Zhishi.me - Weaving Chinese Linking Open Data. *The Semantic Web – ISWC 2011. ISWC 2011. Lecture Notes in Computer Science*, vol 7032. Springer, Berlin, Heidelberg. doi: 10.1007/978-3-642-25093-4_14

Schneider, E. W. (1973). Course modularization applied: the interface system and its implications for sequence control and data analysis. *Behav. Objectives* (21).

Shi, L., Li, S., Yang, X., Qi, J., Pan, G., and Zhou, B. (2017). Semantic health knowledge graph: semantic integration of heterogeneous medical knowledge and services. *BioMed. Res. Int.* 2017, 2858423. doi: 10.1155/2017/2858423

Singhal, (2012). *Official Google Blog: Introducing the Knowledge Graph: things, not strings*. Available online at: http://googleblog.blogspot.pt/2012/05/introducing-knowledge-graph-things-not.html (Accessed January 2, 2015).

Wang, C., Ma, X., Chen, J., and Chen, J. (2018). Information extraction and knowledge graph construction from geoscience literature. *Comput. geosciences* 112, 112–120. doi: 10.1016/j.cageo.2017.12.007

Wu, X., Duan, J., Pan, Y., and Li, M. (2023). Medical knowledge graph: Data sources, construction, reasoning, and applications. *Big Data Min. Analytics* 6, 201–217. doi: 10.26599/BDMA.2022.9020021

Wu, Y., Feng Fu, X., Xu, L., and Feng Jiang, Z. (2021a). A summary of the latest research on knowledge graph technology. *International Conference on Computer Engineering and Artificial Intelligence (ICCEAI)*, Shanghai, China, 2021, 38–42. doi: 10.1109/ICCEAI52939.2021.00007

Wu, X., Jiang, T., Zhu, Y., and Bu, C. (2021b). Knowledge Graph for China's Genealogy 1 1. A shorter version of this paper won the Best Paper Award at IEEE ICKG 2020 (the 11th IEEE International Conference on Knowledge Graph, ickg 2020. bigke. org). *IEEE Trans. Knowledge Data Eng.* 35, 634–646. doi: 10.1109/TKDE.2021.3073745

Xu, B., Xu, Y., Liang, J., Xie, C., Liang, B., Cui, W., et al. (2017). CN-DBpedia: A never-ending Chinese knowledge extraction system. *Advances in Artificial Intelligence: From Theory to Practice. Lecture Notes in Computer Science*, vol 10351. (Cham: Springer). doi: 10.1007/978-3-319-60045-1_44

Xue, Y., Mu, H., and Xue, L. (2024). Exploration and Practice of Blended Teaching in the Course of Engineering Drawing Empowered by Knowledge Graph, *2024 13th*

International Conference on Educational and Information Technology (ICEIT), Chengdu, China, 2024, 345–349. doi: 10.1109/ICEIT61397.2024.10540772

Yu, T., Li, J., Yu, Q., Tian, Y., Shun, X., Xu, L., et al. (2017). Knowledge graph for TCM health preservation: Design, construction, and applications. *Artif. Intell. Med.* 77, 48–52. doi: 10.1016/j.artmed.2017.04.001

Yu, B., and Wei, J. (2020). IDCNN-CRF-based domain named entity recognition method, *2020 IEEE 2nd International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*, Weihai, China, 2020, 542–546. doi: 10.1109/ICCASIT50869.2020.9368795

Zhang, X., Li, C., and Du, H. (2020). Named entity recognition for terahertz domain knowledge graph based on albert-bilstm-CRF. In *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*. Chongqing, China, Vol. 1, 2602–2606. doi: 10.1109/ITNEC48623.2020.9084840