

ORIGINAL RESEARCH

Open access

A Knowledge Graph-Enhanced Large Language Model Framework for Garden Pest and Disease Question Answering

Peter Nilsson^{1*}, Eva Johansson¹, Lars Andersson²

Abstract

As urban landscaping advances, the struggle with garden pests and diseases grows more critical. Substantial interest has been drawn to large language models for their skill in interpreting what users mean and producing suitable responses. The addition of knowledge graphs has offered these models a well-organized, high-quality foundation of facts. This investigation merges knowledge graphs (KGs), large language models (LLMs), and similar tools to engineer a smart question-answering (Q&A) framework for garden pest and disease contexts. Forge a knowledge graph dedicated to garden pests and diseases by amassing dependable data through web scraping and scholarly review. Pinpoint central entities and their links to shape a conceptual schema. Employ the ERNIE-BiLSTM-CRF architecture to pull facts from unstructured text. Testing reveals the fact-extraction system advanced in this work scores above 92% on precision, recall, and F1 metrics, beating competing designs. Put forward a Q&A strategy that fuses the garden pest and disease KG with the ERNIE-Bot-turbo engine. Knowledge is turned into vectors, and the closest matches are located via similarity search; these top hits are then bundled with the query into cues fed to the language engine for natural language output. Trials stacking our technique against ERNIE-Bot-turbo and ChatGLM-6B showed strong results across easy, mid-level, and tough problems, sidestepping off-base replies to out-of-scope questions. It exceeded both baselines in correctness, hitting a 90% exact-answer mark for straightforward queries.

Keywords Large language models, Knowledge graph, ERNIE, Garden diseases and pests

*Correspondence:

Peter Nilsson

peter.nilsson@slu.se

¹ Department of Agricultural Knowledge Graphs and LLMs, Faculty of Agriculture, Swedish University of Agricultural Sciences, Uppsala, Sweden

² Department of Pest and Disease Question Answering Systems, Faculty of Agriculture, Lund University, Lund, Sweden

Introduction

Old-style question-answering setups can partially parse user inputs and return information [1], yet the inherent suppleness and imprecision of everyday speech throw up steep barriers to grasping the deeper meaning of convoluted questions [2].

The neatly arranged facts inside knowledge graphs (KGs) [3, 4] have pushed forward the evolution of Q&A platforms [5]. These platforms dissect inquiries and hunt through the graph for solutions, tapping ties among entities to carry out multi-step logic [6]. Ge *et al.* [7] rolled out a mechanized model for rice nutrient advice leaning on KGs and precedent-based logic. Beyond lifting hit rates, the model brings clear traceability and meaningful real-world worth. Liu *et al.* [8] crafted a Chinese mineral Q&A tool anchored in a knowledge graph. The effort harnessed the BERT [9] model to field mineral queries, securing a 91.2% accuracy level on the evaluation batch. Liu *et al.* [10] unveiled a wireless networking Q&A mechanism that unites prompt-driven learning with knowledge graphs.

Breakthroughs in large language models (LLMs) [11–13] now let Q&A platforms weigh full-text queries end-to-end and give back fluid, human-style replies, a leap toward genuine machine intelligence in the field [14, 15]. Kim *et al.* [16] delved into mixing causal logic methods with KGs and LLMs, outlining a random-walk reasoning tactic rooted in graphs to lift LLM output via cause-and-effect chains. Jiang *et al.* [17] detailed a path to hone LLM answers and extraction exactness via delicate tuning [18] and recurrent improvement loops. They spotlighted gains from folding in retrieval-augmented generation [19] to raise system capability. Also, Singhal *et al.* [20] gauged how well LLMs internalize medical insight and put forth a fresh yardstick for appraisal. Their runs spotlighted the strength of PaLM and Flan-PaLM against the MultiMedQA collection. Instruction-based tuning tricks [21] were likewise adopted to lift domain performance in medicine.

KG-powered Q&A rests on rigidly ordered bases of facts to give pinpoint answers, yet it stumbles when facing layered queries [22, 23], above all those demanding multi-hop reasoning across tangled entity webs. On the flip side, large language models (LLMs) show sharper skill at catching user drift and spinning coherent prose, though they still trip on everyday logic and knotty deduction [24, 25].

To clear these hurdles, the present work weds the spot-on data muscle of KGs with the nuanced linguistic grasp of LLMs [26–28], laying out a Q&A route that leans on both KGs and LLMs, shaped for landscape pest and disease oversight. By knitting the crisp factual web of KGs together with the learned parameter memory of LLMs, this scheme can dig into and make sense of user questions on landscape pests and ailments, turning out authoritative, adaptable, natural-language guidance. The approach remedies the stiff back-and-forth of KG-based Q&A [22, 23] and the slip-ups of pure LLM Q&A [29], donating a workable pattern to the Chinese open-source landscape.

Knowledge Graph Construction

The present effort weaves together bottom-up detail gathering and top-down design to assemble a knowledge graph for garden pest and disease administration [30]. The step-by-step assembly workflow is displayed in **Figure 1**.

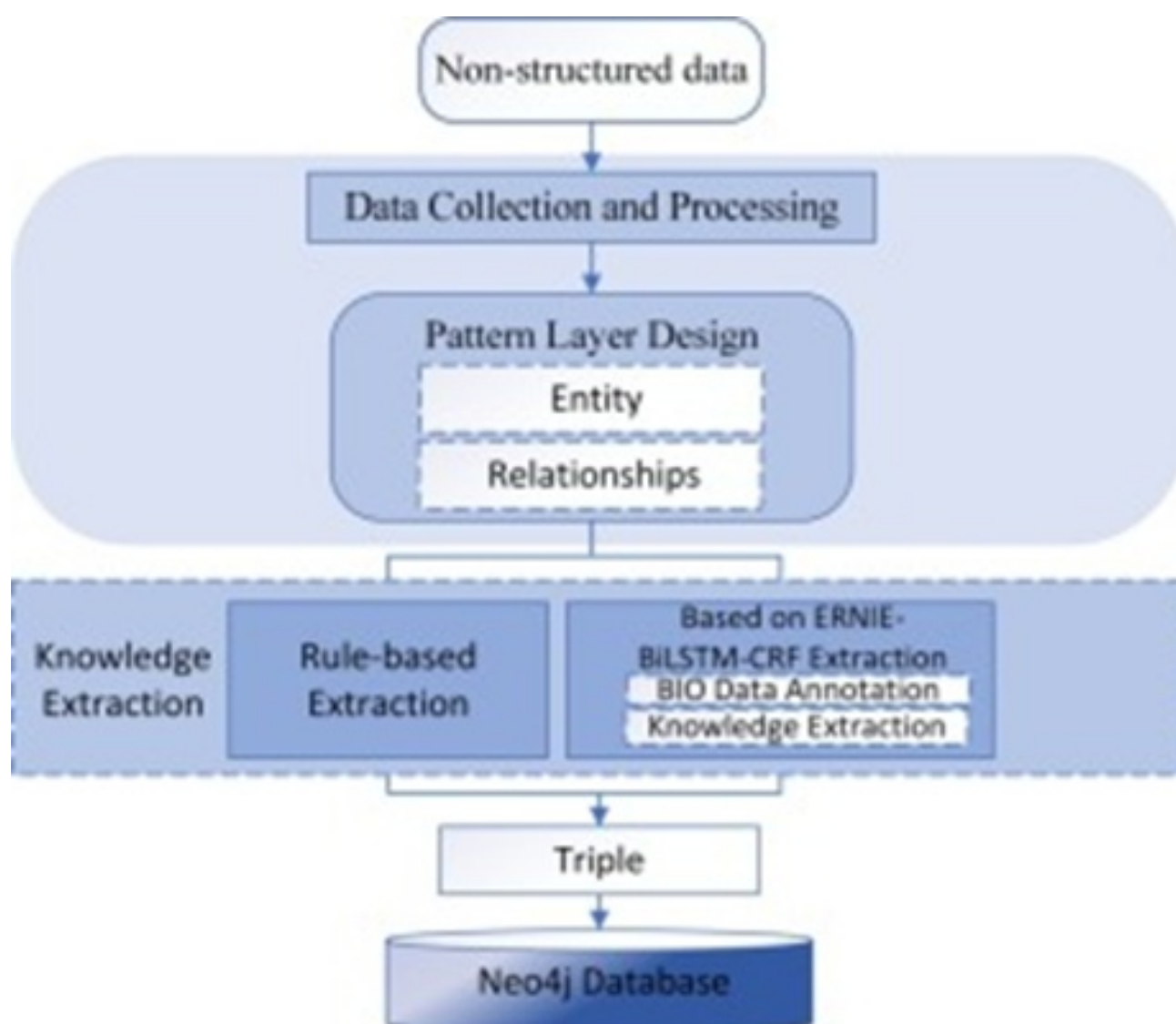


Figure 1. Construction process of knowledge graph.

Data gathering and refinement

The information tied to landscape pests and diseases was independently sourced and compiled for this work [31]. The central data pool was the “Landscape Plant Protection Network” (<http://www.ylzb98.com/tpmore.asp>). The project employed the Python programming language, weaving together the requests library [32] for issuing web calls, lxml [33] for interpreting page markup, and Xpath [34] for pinpointing elements, to enable accurate capture of semi-structured material. While crawling the previously mentioned landscape pest and disease HTML documents, stray symbols like “ n”, “ r”, “ xa0”, and blank areas were purged. Specific fields underwent tailored text replacement and tidying to uphold consistency and quality across the dataset. The cleaned content was then organized around pest and disease names, with sanitized information for each pest or disease placed in its own folder. In all, 2,859 pest-specific folders and 1,501 disease-specific folders were assembled.

Knowledge graph schema design

A hybrid top-down and bottom-up design tactic was taken to chart out the classes and their hierarchical arrangement [22]. In addition, the properties of the entity classes and the rules constraining them were nailed down. Finally, by grounding

the ontology in examples, it became serviceable for concrete situations featuring landscape pests and diseases. Mirroring the nature of information in this specialty, the study bifurcated the domain into two overarching types—diseases and pests—each encompassing related conceptual detail. Facts were encoded as triples (entity, relationship, entity). The conceptual blueprint of the knowledge graph is provided in **Figure 2**.

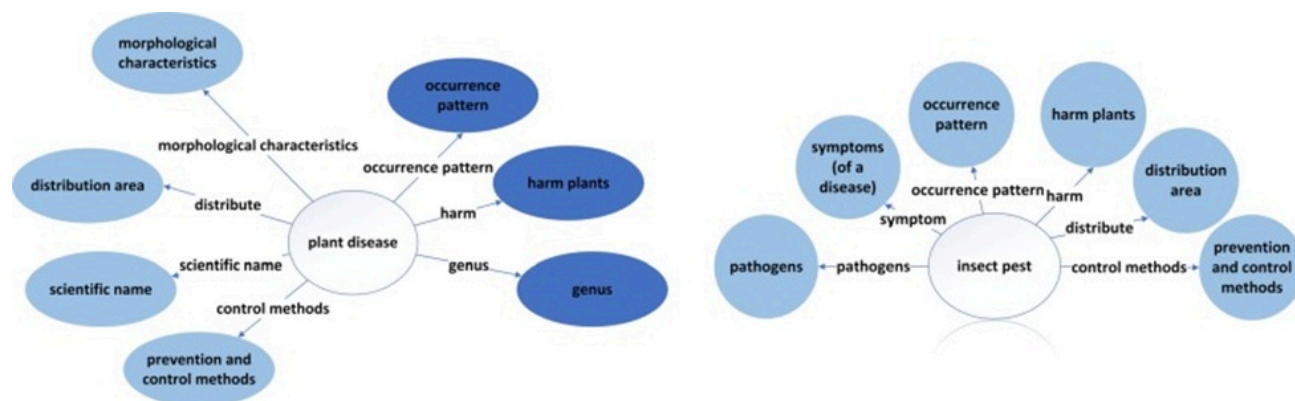


Figure 2. Knowledge graph concept model diagram.

While constructing, the entity3 groups needed for tackling landscape pest and disease challenges are first scoped out and depicted as nodes across the knowledge graph. Pulling from the research topic, the entities singled out include “disease,” “pest,” “region,” “morphological characteristics,” “plant,” “occurrence pattern,” and “control methods.”

Relationships4 bridging entities, equivalent to edges inside the knowledge graph, encapsulate the bonds among separate nodes. In line with the conceptual layering of landscape pests and diseases, a “scientific name” link binds the “pest” entity to the “scientific name” entity, whereas a “pathogen” link joins the “disease” entity with the “pathogen” entity. Based on these links, the “scientific name” and “pathogen” relationships are generalized. After examining the associations spanning all concepts, every relation populating the landscape pest and disease knowledge graph was documented.

Rule-based extraction from semi-structured knowledge

To manage the semi-structured content on garden pests and diseases, this effort devises specialized functions and turns to XPath for fetching pertinent entities out of the schema tier of the knowledge graph. Once the data is cleaned, regular expressions along with string manipulation techniques are called upon to detect and retrieve salient pieces—disease names, pest names, scientific names, and categories are examples—as entities lifted from the sanitized text. These entities are bucketed by the traits of the harvested information. Sticking to the schema layer blueprint outlined earlier, logical rules are crafted to recast the semi-structured text into triple format, with each triple following the (entity, relation, entity) template. One such example is (*Salvia splendens* stem borer, morphological characteristics, adult body length is approximately 15-18 mm). The harvested triple data can be seen in **Figure 3**.

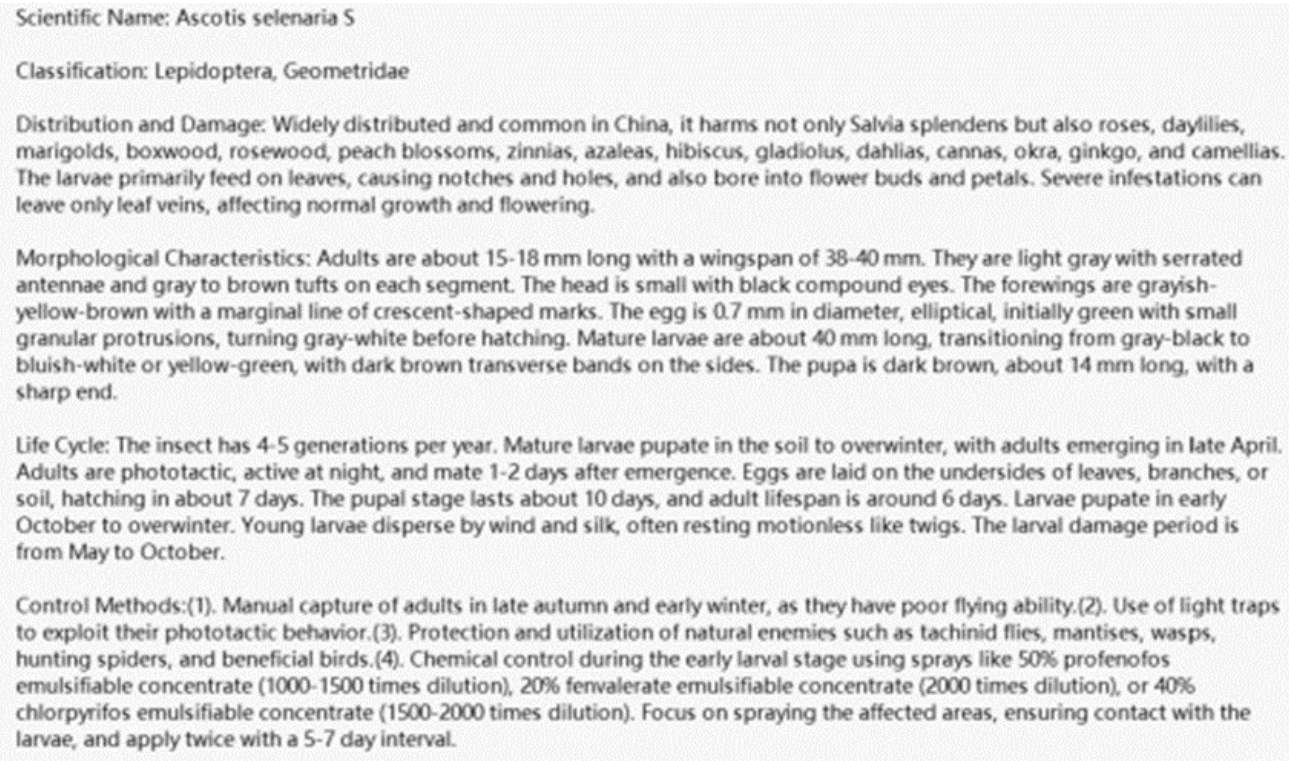


Figure 3. Partial semi-structured knowledge extraction results.

Unstructured Knowledge Extraction

This investigation merges the schema layer’s predefined entities, their attributes, and inter-entity links, leaning on the brat annotation toolkit and BIO sequence labeling for exact tagging of nested entity passages. For feature-level text analysis, the ERNIE-BiLSTM-CRF named entity recognition architecture is deployed, allowing streamlined harvesting of unstructured content [35]. What follows offers a granular walk-through of the model’s rationale, its composition, and the job of each sub-module. In the end, head-to-head experiments, ablation trials, and extraction effectiveness tests serve to corroborate the model’s fitness and viability for this study.

BIO data annotation

Within nested entities, this work labels diseases, pests, regions, and plants by tacking on the suffixes “disease,” “insect,” “city,” and “plant” to mark each category apart. The BIO [36] tagging protocol is applied to dispense distinct markers to every entity class. The accompanying data annotation specifics appear in Table 1.

Table 1. BIO tags for annotated data.

Name	Beginning tag	Intermediate tag	End tag	Other tag
Disease	B-disease	I-disease	I-disease	O
Insect damage	B-insect	I-insect	I-insect	O
Region	B-city	I-city	I-city	O
Plant	B-plan	I-plant	I-plant	O

Following the entity classes outlined earlier, the data points “Yangsanzhou moth,” “Henan,” and “poplar tree” are first slotted into the entity pools for “pest,” “region,” and “plant,” respectively. Then, the ties connecting these three entities are weighed. Per the predefined landscaping pest and disease model layer, a “distribution” bond connects “Yangsanzhou moth” and “Henan,” giving the structured triple (Yangsanzhou moth, distribution, Henan). In parallel, a “harm” bond exists between “Yangsanzhou moth” and “poplar tree,” yielding the structured triple (Yangsanzhou moth, harm, poplar tree). After BIO annotation is carried out, adding further entities merely demands observing the entity type information and settling the exact relationship type from the associations codified in this study’s conceptual model. For cases where one data entry houses multiple triples, entity category markers alone can be used to spot inter-entity relationships and pull triple data. Consequently, the entity relationship extraction puzzle in this study can be reframed purely as a named entity recognition challenge.

Named entity recognition (NER) model structure

Anchored in the BIO tagging scheme, an end-to-end pipeline combining ERNIE [37, 38] embedding with BiLSTM-CRF [39] is harnessed for both training and label prediction. The inner arrangement of each tier in the ERNIE-BiLSTM-CRF model is rendered in **Figure 4**.

- **Input layer:** The input layer accepts the annotated landscaping pest and disease material, reshaping it into a list of single characters accompanied by a parallel label list, with every character and label linked to an index.
- **ERNIE layer:** Through an embedding step, the string of landscaping pest and disease characters gets mapped into a computation-ready vector representation.
- **BiLSTM layer:** This layer activates the BiLSTM model: vectors emerging from the ERNIE layer are initially passed into the forward-directed LSTM arm, which walks the input vector chain in standard temporal flow. The same input vector chain is then fed through the backward-directed LSTM arm, which traverses it in reverse temporal flow. The two resulting chains are then merged to yield the layer’s final output vector.
- **CRF layer:** The CRF layer steps in to adjust and decode the label chain coming out of the BiLSTM layer. Candidate label sequences are graded, and the decoding stage emits the definitive entity label chain.
- **Output layer:** Following the highest-probability chain sourced from the CRF layer, the annotated chain is translated back into matching entities.

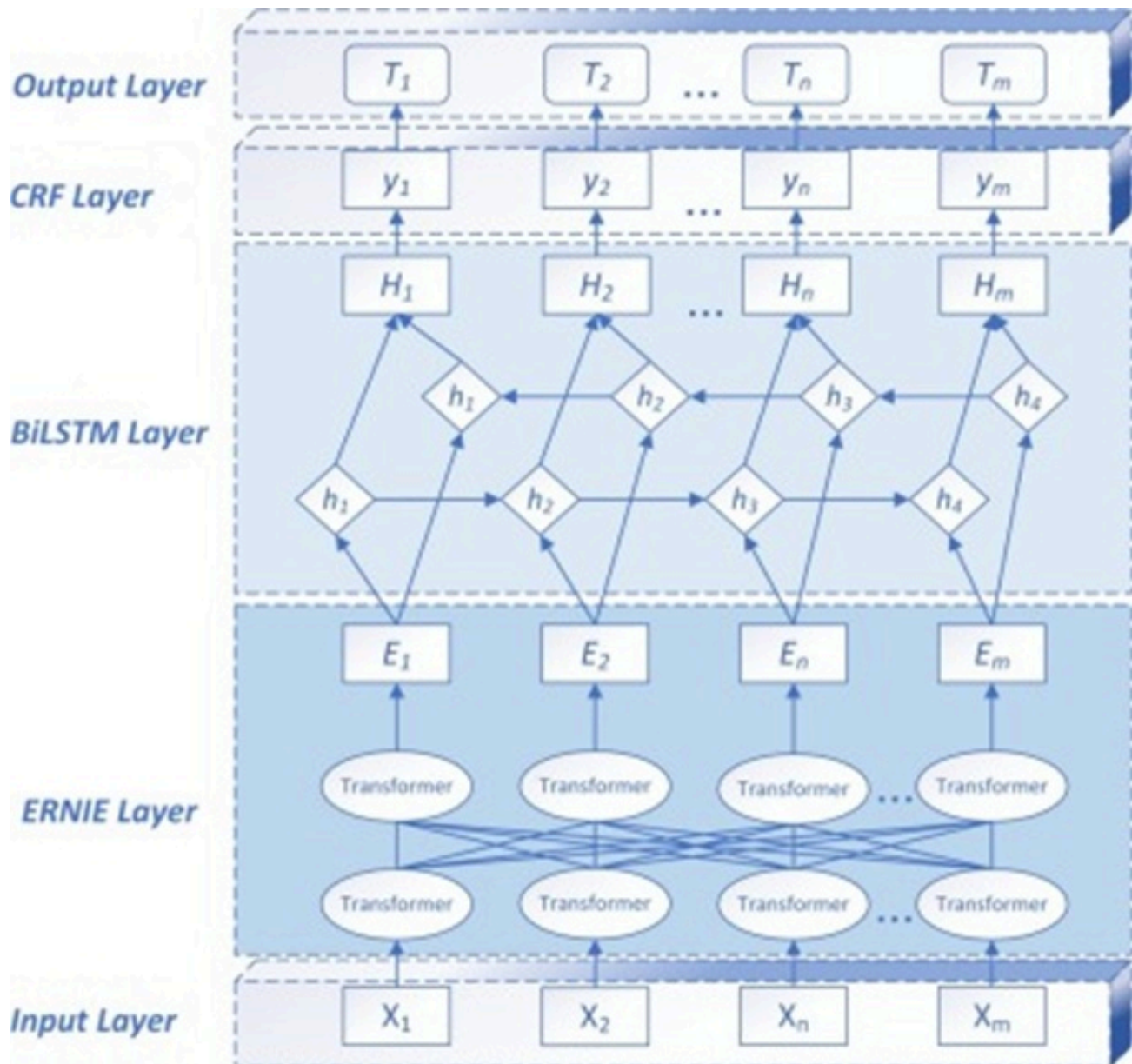


Figure 4. Structure of the ERNIE-BiLSTM-CRF model.

ERNIE pre-trained model layer

Within named entity recognition pipelines, ERNIE [40, 41] has the capacity to absorb semantic patterns shaped by surrounding text and to learn the associated feature representations during its pre-training cycle. It constitutes an optimized successor to BERT, built with further refinements. The ERNIE pre-trained architecture turns raw text into machine-digestible vector forms, enabling performant computation. The embedding stage weaves together tokenization, word-level embeddings, positional encoding, segment embeddings, and injected knowledge, transforming input utterances into high-dimensional vectors deeply imbued with semantic properties. Its backbone relies on a stacked bidirectional Transformer encoder and adopts masking schemes on three tiers: individual character masking, chunk-based phrase masking, and whole-entity masking. Character-level masking randomly blanks out single glyphs, thereby modelling local semantic bonds; phrase-level masking hides entire consecutive expressions, sharpening broader contextual awareness; and entity-level masking zeroes in on named-entity spans.

BiLSTM layer

In the present setup, BiLSTM [42] ingests the vector outputs released by ERNIE and distills contextual signals from them, constructing a fuller semantic picture. By incorporating knowledge flowing from both the earlier and later parts of a sequence, BiLSTM stitches together the directional feature representations, strengthening its stability. The update rules governing the cell state and the hidden state at every time tick are detailed across Eqs. 1 to 6.

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

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$

$$c'_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (3)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot c'_t \quad (4)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = o_t \odot \tanh(c_t) \quad (6)$$

Here, f_t , i_t , o_t , and c'_t stand for the forget gate activation, input gate activation, output gate activation, and the candidate cell state associated with the latest input, in that order. The backward-directed LSTM scans the token series from the end toward the start along the temporal axis. Following the same set of formulas, it works out the backward hidden representation h_t^{backward} , which is then merged with the forward hidden representation h_t^{forward} . The final combined result of BiLSTM is described by Eq. 7:

$$h_t = [h_t^{\text{forward}}, h_t^{\text{backward}}] \quad (7)$$

This output vector encodes each word in conjunction with its context and becomes the feed for the subsequent CRF stage.

CRF layer

Incorporating a CRF [43] module into the network injects inter-label restrictions during the prediction step, which steers the final tag sequence toward validity. At the point of decoding, designate y_t as the tag for the t -th token, P as the matrix of emission potentials, and T as the transition score table between tags. The overall sequence score follows the form in Eq. 8:

$$\text{Score}(\mathbf{y}) = \sum_{i=1}^N S_{y_i, i} + \sum_{i=1}^{N-1} P_{y_i, y_{i+1}} \quad (8)$$

The CRF specification expresses the joint probability through Eq. 9:

$$P(y|X) = \frac{\exp(\text{Score}(y))}{\sum_{y'} \exp(\text{Score}(y'))} \tag{9}$$

Model parameters are learned by driving this joint probability upward, which amounts to performing maximum likelihood estimation. The CRF block in this architecture grasps the dependencies among output labels, thereby enforcing that the emitted tag chain remains coherent.

Storage of Knowledge

The Neo4j graph database is enlisted here to store domain knowledge on pest and disease administration in landscaping, and CSV-formatted files serve as the bulk ingestion vehicle. The corresponding import instructions are given in [Algorithm 1](#). Loading structured content into Neo4j involves two principal phases: first, instantiating entity nodes, and second, wiring the relationships that connect those entities. By scripting relationship-creation commands, links are erected on the basis of observed interactions among entities. Once every triple has been pulled in from the CSV sources, Cypher query syntax is applied to persist the entities alongside their interconnections within Neo4j.

Input: CSV files: insects_nodes.csv, disease_name.csv, relationships.csv

```
//Import entity nodes
1:  LOAD CSV WITH HEADERS FROM 'file:///insects_nodes.csv' AS row
2:      CREATE (:Entity {title: row.title});
3:  LOAD CSV WITH HEADERS FROM 'file:///disease_name.csv' AS row
4:      CREATE (:Entity {title: row.title});
// Import relationships and create links
5:  LOAD CSV WITH HEADERS FROM 'file:///relationships.csv' AS row
6:      MATCH (Entity1:Entity {title: row.Entity1})
7:      MATCH (Entity2:Entity {title: row.Entity2})
8:      CREATE (Entity1)-[:RELATION {type: row.relation}]->(Entity2);
```

Output: Graph with nodes and relationships

Algorithm 1. Import CSV statement.

The atomic entities of landscape diseases and pests are visualized in [Figure 5](#), together with the web of relations knitting them inside the knowledge graph. Looking at entity links, the graph records a “distribution” association between “Liaoning” and “Dendrolimus spectabilis,” along with a “damage” association between “Pinus densiflora” and “Dendrolimus spectabilis.”



Figure 5. “Red Pine Caterpillar” entity relationship.

Construction of Question-Answering Methods

Merging the orderly factual framework of knowledge graphs with the parametric recollection of large language models allows Large Language Models to carry out thorough interpretation and sense-making of user questions centred on pest and disease handling in landscaping. The outcome is the production of malleable, professionally toned natural language answers, overcoming both the narrow back-and-forth typical of knowledge graph retrieval and the unreliability commonly seen in purely LLM-driven Q&A.

Question-answer generation process

The Q&A workflow that leans on knowledge graphs and large language models is laid out in [Figure 6](#). Taking the previously assembled knowledge graph covering pest and disease handling in landscaping as the foundational reference

base, a Q&A knowledge pool is created. When a user supplies an inquiry through the interface, the pre-trained ERNIE engine builds embedding vector snapshots of both the question and the reference data, lodging the resulting knowledge vectors inside a vector-oriented store. A similarity scan is next run between the pool of knowledge vectors and the question vector to retrieve the top-matching entities housed in the vectorized reference base. From there, Cypher [44] statements are issued to fetch every relational pathway linked to those entities. The retrieved knowledge graph segments are then blended with the user's initial query following a predefined prompt5 [10] scaffolding, assembling a final prompt. At the last stage, the assembled prompt is passed into the pre-trained ERNIE-Bot-Turbo conversational large language model, which crafts a polished natural language response rich in domain expertise. We exploit the structure inside the knowledge graph to carve out subgraphs, shorten reasoning chains, and select relevant entities, lessening both sparsity and noise. Additionally, we adopt the LangChain toolkit to regulate the retrieval-augmented generation (RAG) sequence, supporting iterative retrieval followed by generation. We also tap the ERNIE-Bot-Turbo large language model, whose strength at handling extended context and fusing external knowledge helps the model track complex inference threads while keeping semantic flow intact.

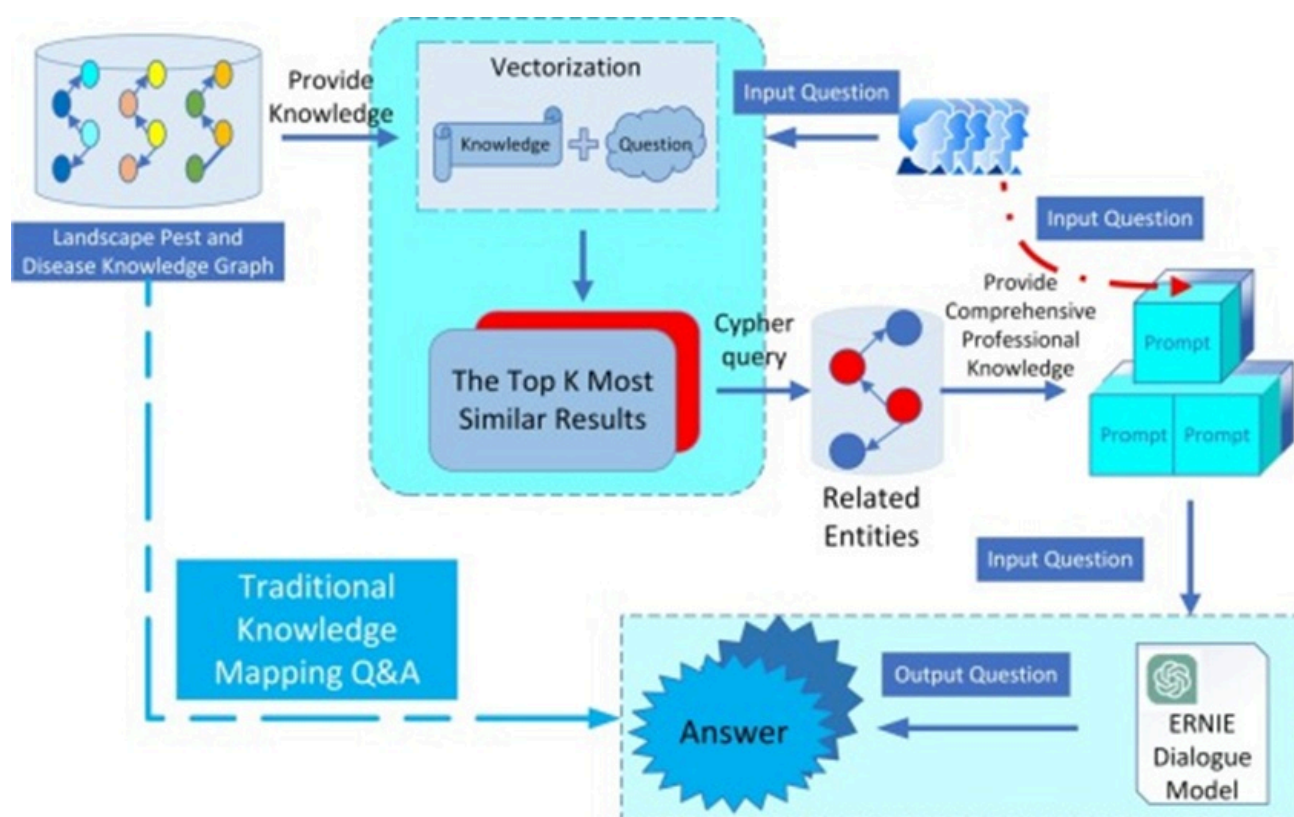


Figure 6. Q&A process.

Data vectorization processing and storage

Before applying text vectorization [45] to bulky documents living in the knowledge graph storage, the present study first applied the text splitter available through the LangChain suite to break the knowledge-bearing text into chunks [30]. The splitting configuration specified the boundary marker, the extent of each resulting text block, and the overlap kept between consecutive blocks. Taking cues from a close reading of the dataset, the block length was pegged at 384 characters to reconcile the desire for intact information against the demands of computational efficiency. To keep context from being severed, an overlap of 20 characters was introduced between neighbouring blocks. Once the chunk-splitting step wrapped up, the following task was to translate those blocks into vector embeddings. The ERNIE-bge-large-zh embedding model, offered by ERNIE, was used in this study to convert chunks into vectors spanning 512 dimensions. A representative excerpt of the vectorized results is shown in Figure 7.

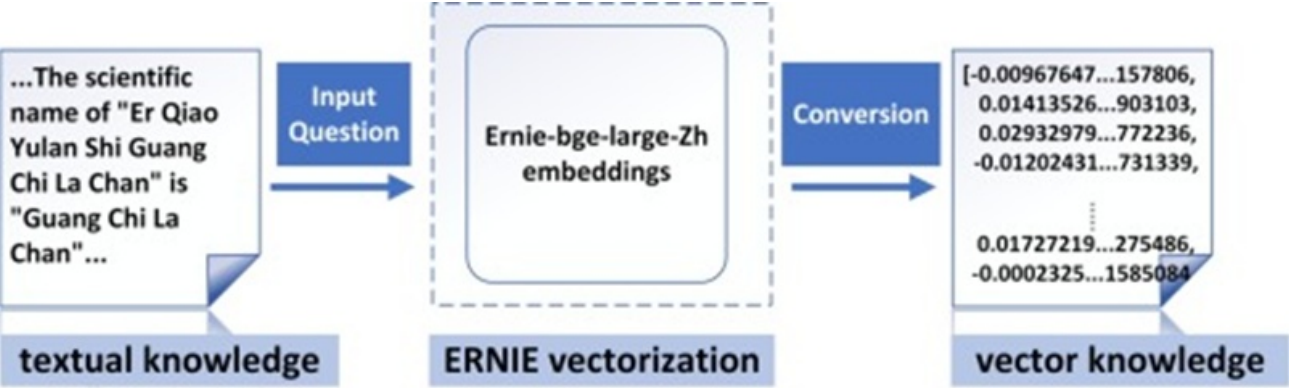


Figure 7. The vectorization process.

The Chroma [46] vector database was selected to store the vectorized landscape of the plant pest and disease knowledge graph, thereby setting up a vector-grounded knowledge base. At the outset, a routine was outlined to call the ERNIE embedding model, turning knowledge text into vector forms. The Chroma vector database was employed to establish and curate the vectorized knowledge base, with the previously outlined ERNIE embedding routine supplied as the collection’s embedding function. Owing to this design, whenever fresh text is incorporated into the knowledge base, Chroma autonomously computes its vector stand-in via the ERNIE model and then places it into the designated vector store. Throughout the data intake step, every processed vector chunk received a unique identifier (ID) to ease subsequent lookup and administration. The update procedure was then invoked to refresh the base, ensuring that every datum was accurately indexed and persisted.

Through the above workflow, the produced knowledge vectors were housed inside the Chroma vector database, supplying a sturdy footing for later retrieval and querying steered by vector similarity.

Vector similarity matching

Whenever an end-user submits a question, the platform initially projects that question into a vector form via embedding techniques, situating it as a single coordinate within the vector space to yield the query vector. Following this, the platform probes the pre-built plant pest and disease vector store to uncover the knowledge-bearing vectors exhibiting the strongest alignment with the query vector [30]. The overall procedure is sketched in Figure 8.

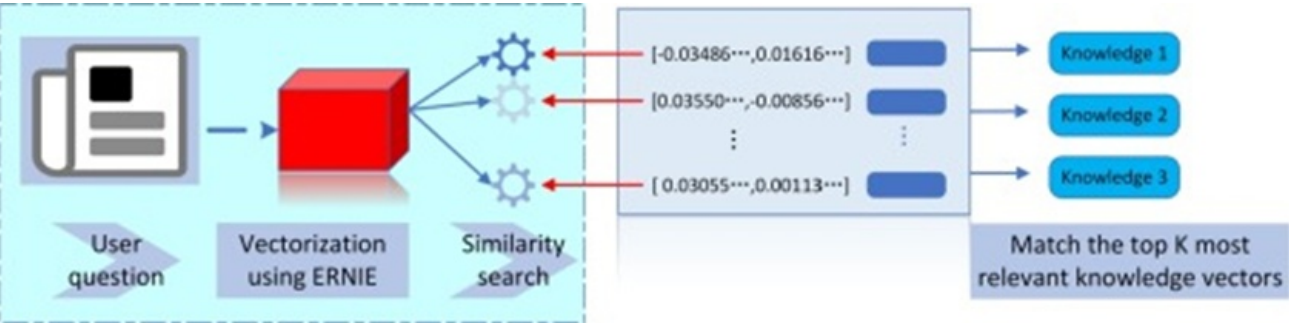


Figure 8. Similarity matching process.

Within the Chroma vector store, the standard similarity-based lookup mechanism operating over questions and knowledge vectors relies on Euclidean distance [47], as portrayed in Figure 9. Its mathematical expression is laid out in Eq. 10:

$$d(p, q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2} \quad (10)$$

Where $d(p, q)$ captures the Euclidean gap between p and q .

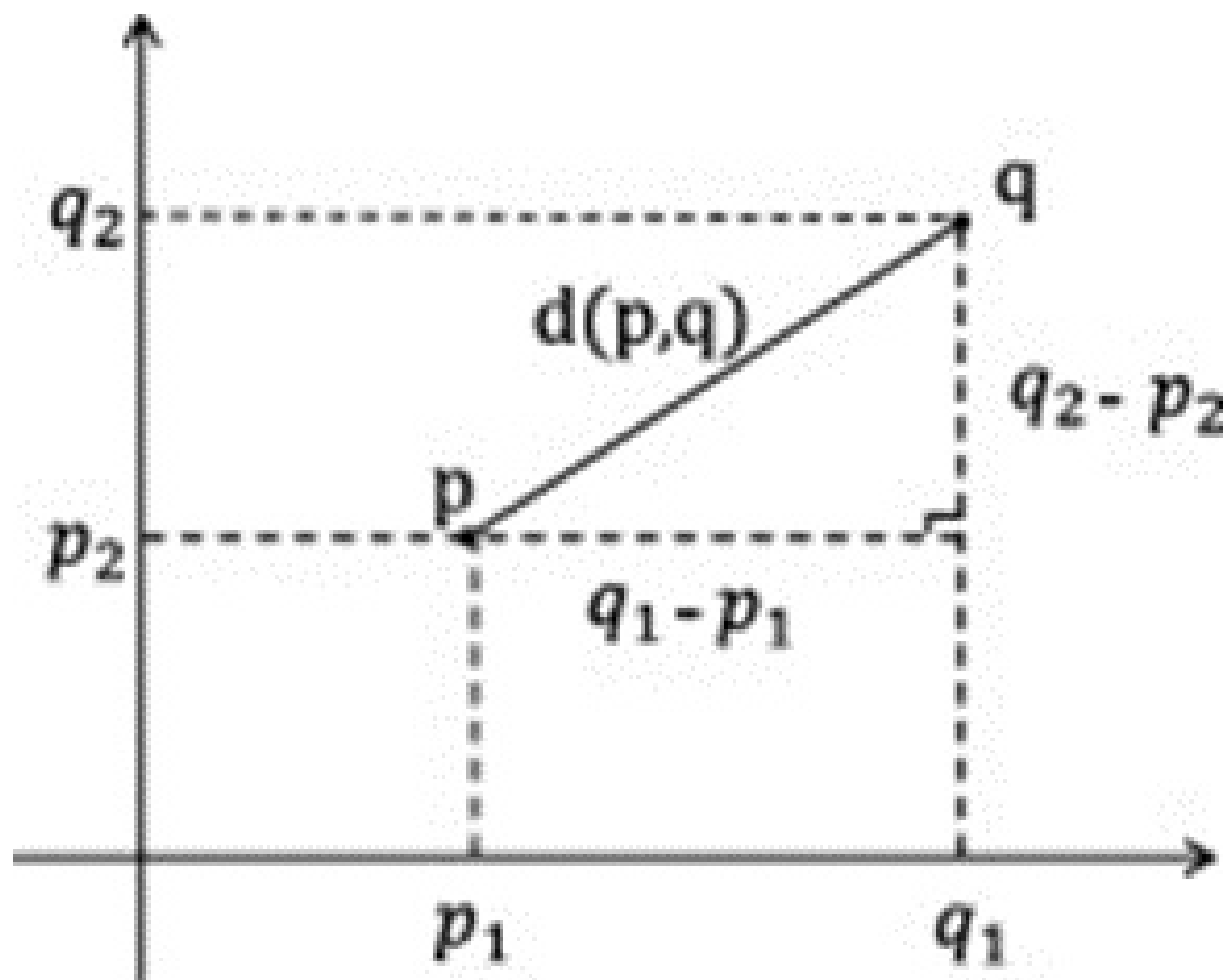


Figure 9. 2D coordinate system.

When operating inside a three-dimensional coordinate setting, each individual vector holds the coordinates x , y , and z . **Figure 10** offers a view of Euclidean distance inside such a three-dimensional coordinate setting.

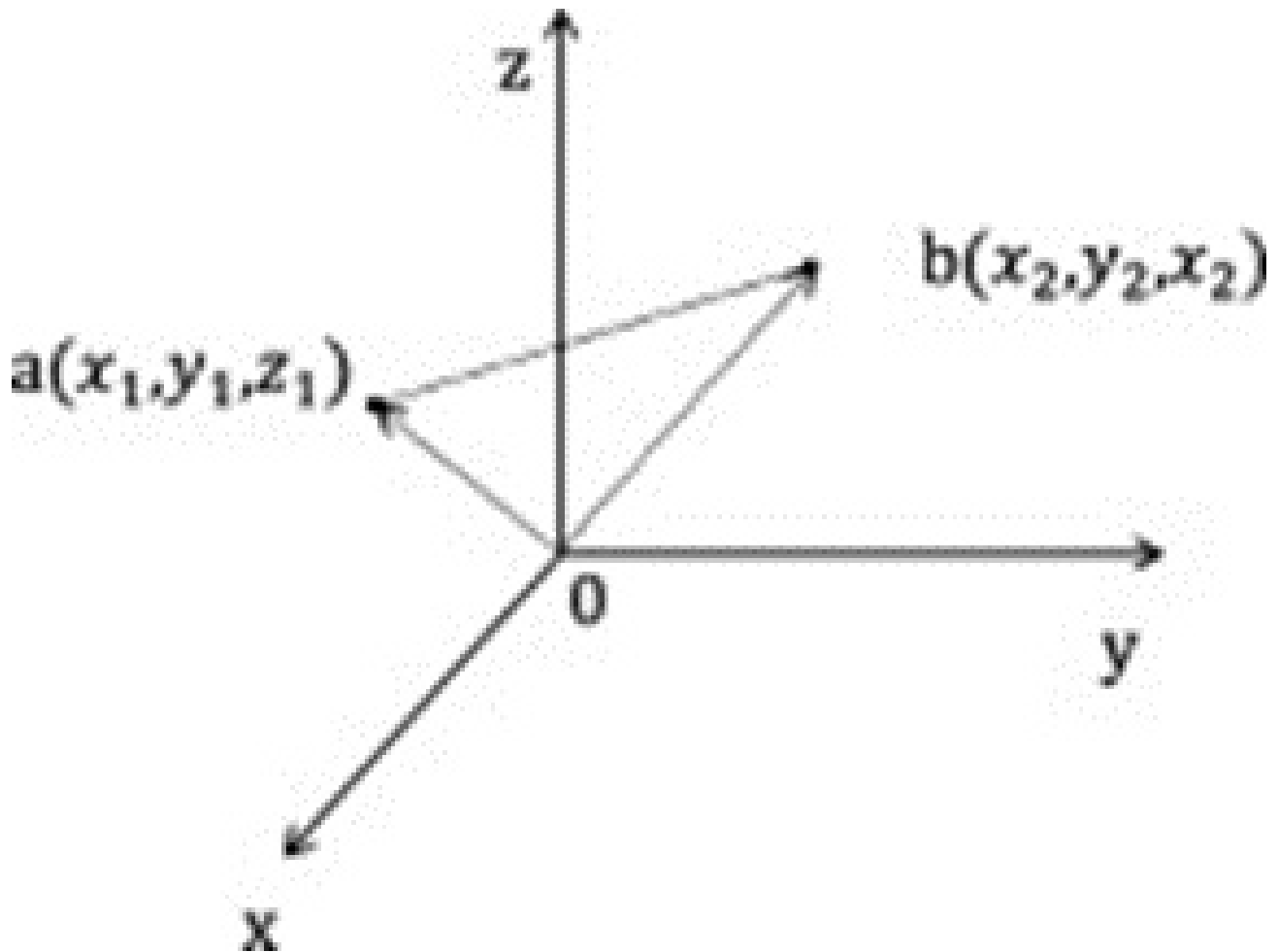


Figure 10. 3D coordinate system.

The corresponding formula for computing it is supplied by [Eq. 11](#):

$$d(p, q) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (11)$$

Where $d(p, q)$ captures the Euclidean distance separating points p and q .

For any arbitrary pair of points situated within a three-dimensional coordinate frame, the Euclidean distance separating them can be determined through [Eq. 12](#):

$$\text{dist}(X, Y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (12)$$

In this expression, $\text{dist}(X, Y)$ conveys the Euclidean distance between points X and Y .

Earlier, a knowledge repository of corpus fragments dealing with forest pests was put together. This repository comprises a collection of document groupings, represented as $F=\{f_1, f_2, \dots, f_n\}$, with n standing for the total document count. Every document element d_{ij} (referring to the j -th document inside the i -th grouping) undergoes preprocessing and vector transformation. For each document grouping, a knowledge lookup structure $V = \{v_{ij}\}$ is generated, wherein v_{ij} matches the vector associated with document d_{ij} . A similarity matching scheme [48] is deployed to identify the knowledge vector that most closely mirrors the query vector. The computational flow can be untangled into these successive stages:

(1) The question passage q gets converted into a query vector Q via vectorization, formalized in Eq. 13.

$$Q = \text{embed}(q) \quad (13)$$

Here, $\text{embed}(q)$ designates the embedding operation that takes the question passage q and maps it to the query vector Q .

(2) The similarity ratings between the query vector Q and all knowledge vectors stored in the repository are computed, after which the top k best-matching knowledge vectors $v_{k1}, v_{k2}, \dots, v_{kk}$ are extracted. This step is formalized in Eq. 14.

$$\text{score}(Q, V) = \{v_{k1}, v_{k2}, \dots, v_{kk}\} \quad (14)$$

Here, $\text{score}(Q, V)$ indicates the similarity rating tying query vector Q to the collection of knowledge vectors V , while v_k designates the set of the top k best-aligned vectors.

(3) The top k best-aligned knowledge vectors are turned into response segments T_k by way of a knowledge transformation mapping, as captured by Eq. 15.

$$T_k = \text{trans}(d_{v1}, d_{v2}, \dots, d_{vk}) \quad (15)$$

Here, $\text{trans}(d_{v1}, d_{v2}, d_{v3}, \dots, d_{vk})$ describes the mechanism that renders the best-aligned knowledge vectors into natural language answer passages. These passages function as the system's response output and can be augmented with extra contextual data—pest identifiers, for example—where the situation calls for it.

Knowledge graph retrieval

After the K knowledge vectors displaying the tightest match with the user's question have been drawn from the vector store via vector similarity scanning, those vectors are reverse-mapped back to their original plain-text format. In the next step, the raw text material is pushed through a knowledge graph lookup procedure aimed at pulling out comprehensive, domain-accurate pest-centric data. With the source text linked to the knowledge vectors that correspond to the question vector now in hand, Cypher queries are deployed to probe the entities inside the knowledge graph that align with those vectors. All entities plus relationship edges connected to the matching entities are fetched; the harvested content is subsequently merged and transcribed into coherent text. As illustrated in Figure 11, these retrieval products become the knowledge payload of the prompt, which is then blended with the user's original query to formulate the complete prompt.



Figure 11. Knowledge graph retrieval.

Professional Q&A

To boost the domain authority of the Q&A pipeline, this subsection embraces a prompt-centric learning paradigm [49], injecting specialist-level facts drawn from knowledge graphs straight into the large language model, thereby lifting the precision of its generated answers. By sourcing material from the landscape pest and disease knowledge graph, the model gains the power to address specialised questions that go beyond its intrinsic training coverage, expanding the horizon of queries it can meaningfully handle.

Through merging user-submitted questions with matched knowledge graph snippets, prompt strings get assembled for consumption by the ERNIE-Bot-turbo conversational large language model. For this research, the prompts were manually curated and architected. Some sample prompt formulations are shown in Table 2.

Table 2. Prompt examples.

Prompt ID	Rewritten prompt template
P1	You are a specialist in landscape pests and plant diseases. If the requested information is available within the knowledge graph, provide an appropriate response; otherwise, return the statement: "Unable to answer questions unrelated to landscape pests and diseases."
P2	Based on the following information, generate an answer to the query.

To begin with, prompt P1 is built to ensure that the origin of any knowledge drawn upon is rigorously confined to the landscape pest and disease knowledge graph assembled in this investigation. When fielding expert-level questions, a user's query is fused with the matching graph-sourced knowledge, aligned following the procedure outlined earlier. to fashion prompt P2, which subsequently drives the generation of a domain-grounded natural language response.

Throughout the answer generation phase handled by the ERNIE-Bot-turbo conversational engine, the assembled prompts are forwarded into the model, which proceeds to generate authoritative natural language answers for pertinent queries. The ERNIE large language model exploits its robust linguistic comprehension to internalise the knowledge bundled within the prompts and output accurate, fluid replies. This dynamic is pictured in Figure 12.

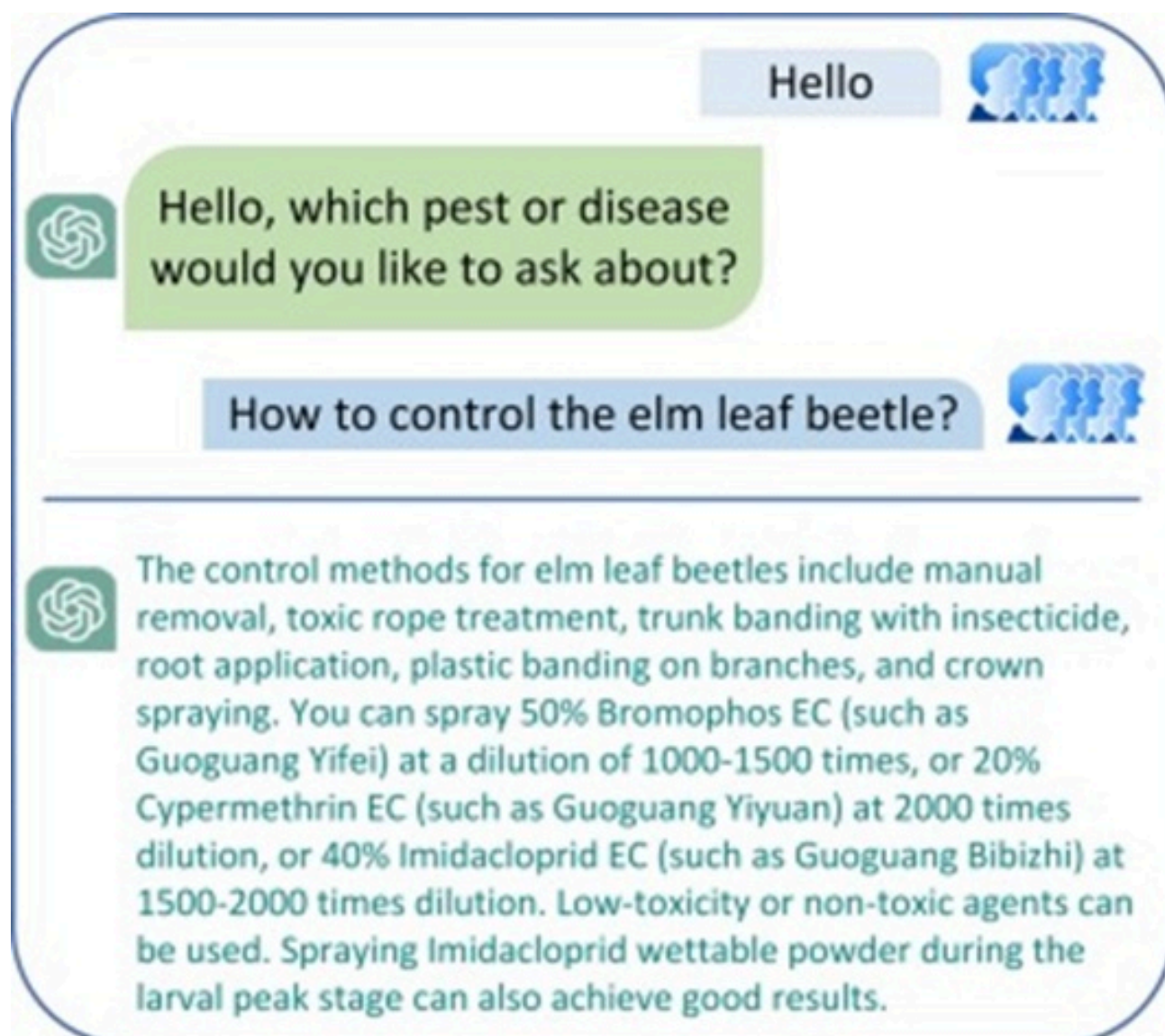


Figure 12. Dialogue example.

Results and Discussion

The experiment on named entity recognition

Experimental data

The investigation relied upon expert-level textual material covering landscape pests and diseases, harvested through web crawling. Once the raw content had been cleaned and filtered, a dataset conforming to the BIO annotation specification was built. The dataset principally covers diseases, pests, plants, and geographic distribution territories. It contains 2,631 sentences, over 170,000 characters, and a cumulative count of 8,925 tagged entities. To prevent overfitting during the experimental runs, the data was portioned into training, validation, and test splits following a 7:2:1 ratio.

Experimental environment and evaluation metrics

Model training was carried out on an RTX 2080 Ti (11G), and three standard criteria from the named entity recognition (NER) domain—precision (P), recall (R), and F1-score—were used as performance gauges. Their definitions are as

follows:

- Precision (P): The proportion of correctly flagged entities in relation to every entity the model predicted, given by Eq. 16.

$$P = \frac{T_p}{T_p + F_p} \times 100\% \quad (16)$$

- Recall (R): The proportion of correctly flagged entities relative to the full set of genuine entities appearing in the data, given by Eq. 17.

$$R = \frac{T_p}{T_p + F_N} \times 100\% \quad (17)$$

- F1-score: The harmonic mean that balances precision and recall, given by Eq. 18.

$$F1 = \frac{2PR}{P+R} \times 100\% \quad (18)$$

In the above, TP denotes true positives, FP false positives, and FN false negatives. After numerous adjustment iterations, the finalised core hyperparameter configuration for the NER experiments targeting landscape pests and diseases is assembled in Table 3.

Table 3. Configuration of core parameters for some models.

Model parameters	Parameter value
batch_size	32
seq_max_length	64
dropout	0.4
learning_rate	0.01

Comparative experiment

In this subsection, side-by-side experimental runs are performed, pitting the traditional HMM model and the BERT-BiLSTM-CRF pipeline against the ERNIE-BiLSTM-CRF architecture put forward in this paper, so as to confirm the effectiveness of the selected modelling approach. The precision, recall, and F1 figures attained by each model are laid out in Table 4 and Figure 13. Examining the metrics across all contenders reveals that the ERNIE-BiLSTM-CRF configuration achieves the optimal results on the task of identifying named entities within landscape pest and disease texts.

Table 4. Recognition results.

Model	F1	R	P
HMM	76.90%	72.93%	81.34%
BERT-BiLSTM-CRF	89.58%	90.01%	89.17%

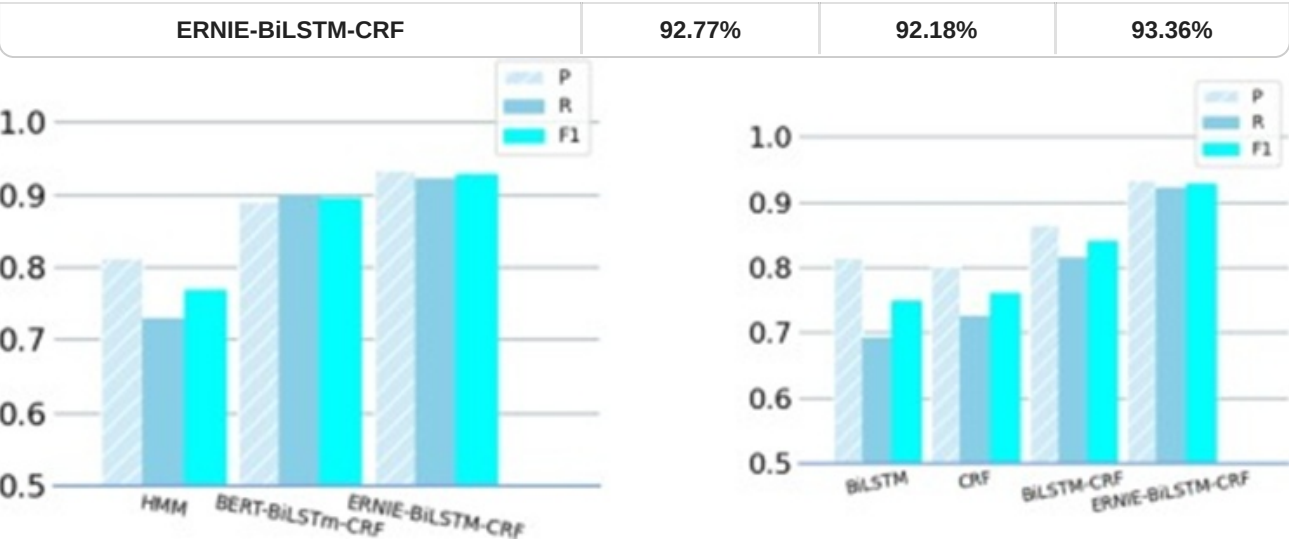


Figure 13. Experimental results.

Ablation experiment

This subsection carries out ablation analysis employing the BiLSTM model alone, the CRF model alone, the BiLSTM-CRF pairing, and the full ERNIE-BiLSTM-CRF architecture, with the goal of verifying the contribution brought by each submodule. The candidate models, together with their empirical outcomes, are collected in Table 5.

Table 5. Recognition results.

Model	F1	R	P
BiLSTM	74.96%	69.34%	81.56%
CRF	76.19%	72.61%	80.14%
BiLSTM-CRF	84.04%	81.53%	86.72%
ERNIE-BiLSTM-CRF	92.77%	92.18%	93.36%

In order to judge the worth of different entity recognition architectures, a comparison was conducted across four configurations: BiLSTM, CRF, BiLSTM-CRF, and ERNIE-BiLSTM-CRF. The resulting figures, plotted in Figure 13, make it evident that the ERNIE-BiLSTM-CRF variant outstrips the field substantially across precision, recall, and F1-score. Concretely, the ERNIE-BiLSTM-CRF setup reached a precision of 93.36%, a recall of 92.18%, and an F1-score of 92.77%, revealing a markedly stronger knack for precisely and stably flagging domain-relevant entities. On the opposite end, foundational schemes like BiLSTM and CRF produced considerably weaker outcomes, with F1-scores settling at 74.96% and 76.19%, respectively. The BiLSTM-CRF hybrid offered a noticeable but intermediate uptick (F1-score: 84.04%), bearing out the benefit that comes from coupling sequential representation learning with structured output prediction. These empirical outcomes verify the efficacy of fusing a knowledge-enriched pre-trained model (ERNIE) with the BiLSTM-CRF backbone for the fine-grained extraction of entities in the landscape pest and disease management sphere.

What this indicates is that bringing the ERNIE pre-trained model into the fold bolsters the system's ability to encode textual meaning, which translates into more accurate detection of landscape pest and disease entities.

Evaluation experiment of triplet extraction effectiveness

This subsection dissects the capability of the ERNIE-BiLSTM-CRF model when it comes to spotting various entity categories and the relationships linking them, delivering a structured appraisal of the compiled triplet dataset. The analysis concentrates on the identification of disease designations, pest designations, plant references, and region entities within the landscape pest and disease corpus, along with the triples that these entities collectively form. These four groupings were chosen on the grounds that they constitute the most prototypical, most frequently recurring classes throughout the data. Together, they represent the large majority of labelled cases and serve as the direct source for building the knowledge triples leveraged in downstream QA reasoning workflows. **Table 6** presents the recognition scores measured for nested entities. The data makes clear that disease designations and pest designations enjoy the strongest identification performance, with precision, recall, and F1 all surpassing the 94% mark. Recognition precision settles at 89.17% for regions and at 88.94% for plant entities. A deeper look reveals that disease and pest designations tend to exhibit more sharply defined boundaries and follow somewhat predictable surface patterns—disease labels, for instance, routinely terminate with the suffix “disease”—which translates into better learned representations. The relatively sparse supply of regional data and the limited number of training instances constrain what the model can absorb, whereas the extensive variability seen in plant naming conventions makes boundary delineation a tougher problem, leading to marginally weaker performance figures. Taken as a whole, the entity recognition behaviour exhibited by the model is solid.

Table 6. Recognition performance of the model for different entities.

Entity Category	F1	R	P
Disease Name	97.89%	96.36%	97.01%
Pest Names	95.49%	94.87%	96.12%
Plant	87.86%	86.81%	88.94%
Area	89.92%	90.68%	89.17%

In the context of this study, the landscape pest and disease knowledge graph is structured as collections of (entity, relation, entity) triples. The type of each entity governs the relations it may enter into, and any given entity can be involved in several relations simultaneously. To take a concrete case, “Elm Leaf Beetle” belongs to the pest class, while “Elm” falls under plants. Consistent with the schema formalised earlier, this pairing produces the triple (Elm Leaf Beetle, damages, Elm). What is more, a “distribution” association connects the “Elm” entity with the region entity “Liaoning,” giving rise to the triple (Elm, distributed in, Liaoning). Validation conducted via random spot-checking indicates that the precision of the structured landscape pest and disease triple collection reaches 90.12%, a figure that safeguards the overall soundness of the knowledge graph.

Answering experiments

Experimental data

For validation purposes, a question set numbering 200 items was assembled in this work, evenly split into 50 straightforward questions, 50 mid-level questions, 50 challenging questions, and 50 off-topic questions.

Comparison experiment of different question answering models

To confirm the utility of the specialised Q&A framework devised here, a head-to-head evaluation of answering quality was staged among the introduced model, the ERNIE-Bot turbo model, and the ChatGLM-6B model, spanning straightforward, mid-level, challenging, and off-topic question categories.

Drawing on the collected outcomes, one can see that for straightforward queries, all three approaches manage to return fairly on-target replies. That said, when the task shifts to supplying concrete management measures, the technique

advanced in this paper delivers richer remedies—covering explicit suggestions and pesticide details aligned with distinct management tactics—relative to the alternative approaches.

For mid-level queries, the introduced method surpasses the competing approaches when it comes to recognising pests such as *Galeruca daurica* and furnishing capsule descriptions of those pests. In the realm of suggested management actions, this technique supplies customised solutions matched to separate management strategies, whereas the other two approaches are marred by factual drift and a shortage of fine-grained detail in their outputs.

Turning to challenging queries, the outputs produced by the comparison approaches lean toward the vague and generic, failing to productively steer the user toward follow-up inquiries, a shortcoming that can breed inaccurate replies. By contrast, the introduced technique prunes the pool of candidate answers, returns more precise information, and boosts interactivity by nudging users to volunteer extra specifics, thereby lifting the overall dependability of the responses.

For off-topic queries, the two alternative approaches frequently conjure hallucinated replies because they lack an internal mechanism for judging topical relevance. Since such queries reside outside the landscape pest and disease management sphere, the underlying knowledge repository holds no pertinent material. The introduced technique, on the other hand, is able to detect that a question is out of scope and issues a refusal to answer in place of a fabricated response, thus steering clear of hallucinations.

Accuracy assessment

During the retrieval phase, a set of 200 benchmark Q&A pairs tied to landscape pest and disease management was chosen, and two plant protection specialists were enlisted to gauge the RAG performance through Mean Reciprocal Rank (MRR) and Hit Rate. Hit Rate: Should the data pulled back from ChromaDB contain the identifier (the correct-answer identifier was hand-labelled by domain experts to assess model behaviour) that matches the query, it registers as a hit. MRR: Computed according to Eq. 19. $MRR = 1/(rank + 1)$, where rank captures the slot occupied by the correct identifier within the ordered retrieval list.

As reported in Table 7, the proposed QA model either matched or edged past the upper-middle tier of published sector benchmarks [50] on both measures, signalling sturdy retrieval capability.

Table 7. Retrieval performance comparison.

Model	Hit Rate@5	MRR@5
This paper's QA model	0.87	0.72
ERNIE-Bot-turbo	0.73	0.61
ChatGLM-6B	0.69	0.54

For the generation stage, the straightforward, mid-level, and challenging question collections are brought to bear, with the Q&A routine executed via the models characterised earlier. The effectiveness of each model across varying question classes is gauged with accuracy as the chief yardstick, thereby appraising the soundness of the introduced Q&A methodology. The procedure for deriving accuracy is spelled out in Eq. 20.

$$\text{Accuracy} = \frac{\text{Number of Correct Answers}}{\text{Total Number of Questions}} \times 100\%$$

(20)

As unfolded in Table 8, the correctness of model answers trails off as question difficulty climbs. Nonetheless, on straightforward, mid-level, and challenging questions, the introduced technique scored accuracy figures of 0.90, 0.70, and

0.60, respectively. Stacked beside the two sizeable language models that operate without domain-specialised knowledge, the introduced technique notched higher accuracy on all three question tiers, substantiating its capacity to amplify the expert-level reliability of large models within narrow vertical domains.

Table 8. Scores of different models on various questions.

Accuracy	Model		
	ChatGLM-6B	ERNIE-Bot-turbo	This paper's QA model
Simple question	0.82	0.80	0.90
Moderate problem	0.62	0.60	0.70
Complex issue	0.44	0.40	0.60
Average	0.63	0.60	0.73

Through weaving a structured knowledge graph into the RAG pipeline, we not only sharpened the semantic precision of information retrieval but also meaningfully lifted the overall computational economy. In practical terms, the system trims away superfluous retrieval invocations and cuts down the rate at which off-target content is generated, thereby yielding quicker turnaround times and lighter resource demands.

Such efficiency gains become especially pivotal in deployment contexts that call for prompt, trustworthy delivery of specialist insight—landscape pest and disease diagnosis being a prime example. Our evaluative trials indicate that folding in the knowledge graph upgrades retrieval performance and simultaneously raises the faithfulness of the generated content. This operational optimisation feeds straight into improved usability, broader scalability, and easier deployment of the platform, rendering it particularly well-suited to fields such as smart agriculture, where high-efficiency knowledge services are a must.

Thus, the uplift in system efficiency amounts to more than a simple technical refinement; it stands as a determining element of the platform's real-world applicability.

Expert satisfaction evaluation

Two domain specialists in landscape architecture were engaged to appraise the responses emitted by the different models, to corroborate the strength of the proposed method. A pool of 150 questions—drawn from the 50 straightforward, 50 moderate, and 50 complex items—was fed into three separate models to solicit answers. The outputs from each model were then assessed by the specialists along several axes, among them domain authority, correctness, and lucidity. The specialists picked the most satisfactory response from the model-generated options. As displayed in **Figure 14**, the horizontal axis marks the question categories, and the vertical axis captures the specialists' satisfaction ratios.

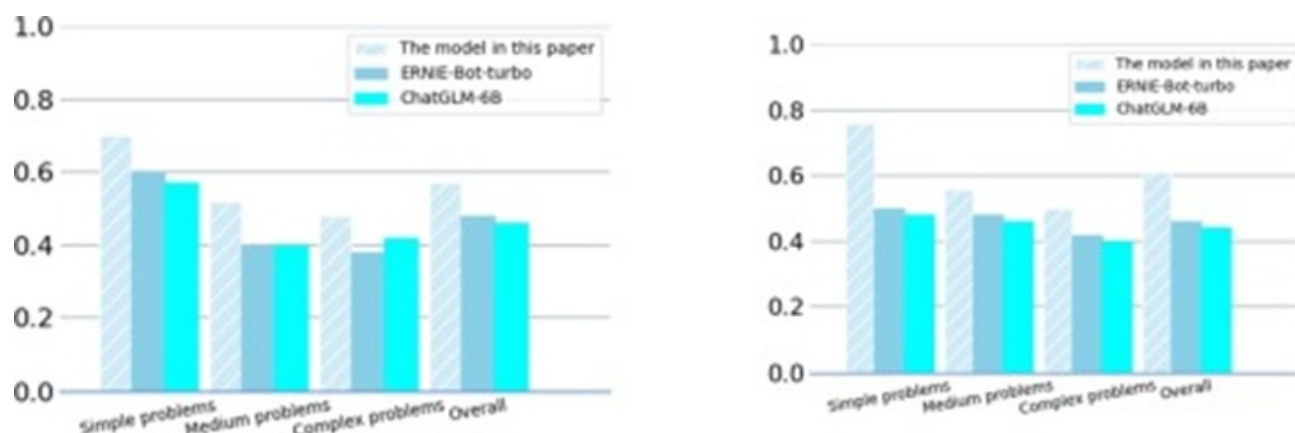


Figure 14. Satisfaction rate.

Throughout the trial, the two specialists were tasked with judging the Q&A outputs of the three models on straightforward, moderate, and complex questions, in addition to an overall performance rating. Concerning straightforward questions, both specialists voiced the strongest approval for the Q&A technique put forward in this study, registering satisfaction rates of 0.70 and 0.76, respectively. On questions of moderate difficulty, the specialists again awarded peak satisfaction marks to the introduced technique, logging rates of 0.52 and 0.56. For complex questions, the introduced technique likewise garnered the highest satisfaction scores of 0.48 and 0.50. In the aggregate assessment, satisfaction with the introduced technique came to 0.57 and 0.61, both figures outstripping those of the two competing models. These patterns underscore the edge held by the proposed Q&A approach in terms of both domain authority and factual reliability.

A question-answering framework is introduced that marries landscape pest and disease knowledge graphs with large language models, yielding marked gains in both the domain authority and factual precision of the generated answers. That said, fusing knowledge graphs with large language models carries nontrivial technical hurdles [26]. All trials reported here were performed inside a tightly managed investigative setting; additional confirmation on live, production-grade knowledge service deployments remains necessary. The sheer intricacy of integration stands as a major obstacle. Beyond that, guaranteeing real-time delivery and answer consistency when the knowledge graph is called upon and refreshed on the fly calls for deeper study. Future lines of refinement may centre on the following avenues: (1) crafting streamlined, highly automated pipelines for assembling and refreshing knowledge graphs, to lighten the data curation workload [35, 51]; (2) advancing tightly fused model designs that embed knowledge graphs directly into the training and inference loops [52]; (3) trialling and stress-testing such integrated setups across diverse subject areas to map out their behavioural patterns and derive context-adaptive tuning recipes [53]; (4) sharpening prompt design to compress the required context footprint while safeguarding response quality [54]. For all of these reasons, we stress that the present contribution should be viewed as a preliminary exploratory endeavour; deeper development, wide-ranging testing, and grounding in real-world validation are all still required.

Conclusion

The present investigation delivered a multi-turn conversational question-answering engine purpose-built for the landscape pest and disease management domain. The Q&A strategy, which intertwines knowledge graphs with large language models, supplies answers that are both more exact and more professionally grounded within specialised verticals. To lock in domain fit and extraction fidelity, we adopted the trained ERNIE-BiLSTM-CRF architecture for pulling out entities and relations, as it permits sharper pinpointing of domain-relevant entities and relational links in the landscape pest and disease management arena. LangChain, for its part, is pressed into service for lookup and answer synthesis that rests on the assembled knowledge graph, effectively linking structured factual repositories with natural language outputs to power downstream Q&A workloads. By furnishing sturdy informational backing, it equips landscape practitioners to arrive at better-informed, more judicious choices. Carrying this work into practice carries deep importance for advancing the long-

term viability of China's landscaping sector, upholding ecological equilibrium, and enriching the everyday lived experience of city dwellers.

Acknowledgements

The authors would like to thank the National Key Research and Development Program of China (grant number [2022YFD1301104](#)) and the earmarked fund for [CARS36](#).

Conflict of interest

None

Financial support

This study is supported by the National Key Research and Development Program of China (grant number [2022YFD1301104](#)) and the earmarked fund for [CARS36](#).

Ethics statement

None

Received: 12 Oct 2022

Revised: 23 Nov 2022

Accepted: 05 Jan 2023

Published online: 15 March 2023

Rights and permissions

Open Access The author(s) retain copyright. This article is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. It may be shared and adapted for non-commercial purposes with appropriate attribution, an indication of changes, and distribution of adaptations under the same license. Third-party material may be subject to separate terms identified in its credit line. View the license at <https://creativecommons.org/licenses/by-nc-sa/4.0/>.

References

Zaib M, Zhang WE, Sheng QZ, Mahmood A, Zhang Y. Conversational question answering: a survey. *Knowl. Inf. Syst.* 2022;64(12):3151-95.
<https://doi.org/10.1007/s10115-022-01735-x>.

Guo T, Peng M, Mulan YI. Advances in question answering in natural language processing. *J. Wuhan Univ. Natur. Sci. Ed.* 2019;65(05):417-26.
<https://doi.org/10.14188/j.1671-8836.2019.05.001>.

Hogan A, Blomqvist E, Cochez M, d'Amato C, De Melo G, Gutierrez C, et al. Knowledge graphs. *ACM Comput. Surv.* 2021;54(4):1-37.
<https://doi.org/10.1145/3447772>.

Zhu X, Li Z, Wang X, Jiang X, Sun P, Wang X, et al. Multi-modal knowledge graph construction and application: a survey. *IEEE Trans. Knowl. Data Eng.* 2022;36(2):715-35.
<https://doi.org/10.1109/TKDE.2022.3221228>.

Huang X, Zhang J, Li D, Li P. Knowledge graph embedding based question answering. In: *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. New York: ACM; 2019. p. 105-13.
<https://doi.org/10.1145/3289600.3290956>.

Berant J, Liang P. Semantic parsing via paraphrasing. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Stroudsburg: ACL; 2014. p. 1415-25.

Ge W, Zhou J, Zheng P, Yuan L, Rottok LT. A recommendation model of rice fertilization using knowledge graph and case-based reasoning. *Comput. Electron. Agric.* 2024;219:108751.
<https://doi.org/10.1016/j.compag.2024.108751>.

Liu C, Ji X, Dong Y, He M, Yang M, Wang Y. Chinese mineral question and answering system based on knowledge graph. *Expert Syst. Appl.* 2023;231:120841.
<https://doi.org/10.1016/j.eswa.2023.120841>.

Devlin J, Chang MW, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. *arXiv [Preprint]*. 2018. arXiv:1810.04805. Available from: <https://arxiv.org/abs/1810.04805>

Liu P, Qian B, Sun Q, Zhao L. Prompt-wnqa: a prompt-based complex question answering for wireless network over knowledge graph. *Comput. Netw.* 2023;236:110014.
<https://doi.org/10.1016/j.comnet.2023.110014>.

Naveed H, Khan AU, Qiu S, Saqib M, Anwar S, Usman M, et al. A comprehensive overview of large language models. *arXiv [Preprint]*. 2023. arXiv:2307.06435. Available from: <https://arxiv.org/abs/2307.06435>

Liu H, Ren Y, Li X, Deng Y, Wang Y, Cao Q, et al. Research status and application of artificial intelligence large models in the oil and gas industry. *Pet. Explor. Dev.* 2024;51(4):1049-65.
[https://doi.org/10.1016/S1876-3804\(24\)60510-0](https://doi.org/10.1016/S1876-3804(24)60510-0).

Zhao WX, Zhou K, Li J, Tang T, Wang X, Hou Y, et al. A survey of large language models. *arXiv [Preprint]*. 2023. arXiv:2303.18223. Available from: <https://arxiv.org/abs/2303.18223>

Gu X, Hu Y, Hao X. Is AGI approaching: rethinking the development of AI and future education in the context of the ChatGPT craze. *J. East China Norm. Univ., Educ. Sci.* 2023;41(7):117.
<https://doi.org/10.16382/j.cnki.1000-5560.2023.07.011>.

Wan F, Huang X, Cai D, Quan X, Bi W, Shi S. Knowledge fusion of large language models. *arXiv [Preprint]*. 2024. arXiv:2401.10491. Available from: <https://arxiv.org/abs/2401.10491>

Kim Y, Kang E, Kim J, Huang HH. Causal reasoning in large language models: a knowledge graph approach. *arXiv [Preprint]*. 2024. arXiv:2410.11588. Available from: <https://arxiv.org/abs/2410.11588>

Jiang F, Qin C, Yao K, Fang C, Zhuang F, Zhu H, et al. Enhancing question answering for enterprise knowledge bases using large language models. In: *International Conference on Database Systems for Advanced Applications*. Cham: Springer; 2024. p. 273-90.
https://doi.org/10.1007/978-3-031-55756-9_16.

Krajewski J, Ludziejewski J, Adamczewski K, Pióro M, Krutul M, Antoniuk S, et al. Scaling laws for fine-grained mixture of experts. *arXiv [Preprint]*. 2024. arXiv:2402.07871. Available from: <https://arxiv.org/abs/2402.07871>

Zhao P, Zhang H, Yu Q, Wang Z, Geng Y, Fu F, et al. Retrieval-augmented generation for AI-generated content: a survey. *arXiv [Preprint]*. 2024. arXiv:2402.19473. Available from: <https://arxiv.org/abs/2402.19473>

Singhal K, Azizi S, Tu T, Mahdavi SS, Wei J, Chung HW, et al. Large language models encode clinical knowledge. *Nature*. 2023;620(7972):172-80.
<https://doi.org/10.1038/s41586-023-06291-2>.

Wang J, Zhang B, Du Q, Zhang J, Chu D. A survey on data selection for LLM instruction tuning. *arXiv [Preprint]*. 2024. arXiv:2402.05123. Available from: <https://arxiv.org/abs/2402.05123>

Abu-Salih B. Domain-specific knowledge graphs: a survey. *J. Netw. Comput. Appl.* 2021;185:103076.
<https://doi.org/10.1016/j.jnca.2021.103076>.

Chen X, Jia S, Xiang Y. A review: knowledge reasoning over knowledge graph. *Expert Syst. Appl.* 2020;141:112948.
<https://doi.org/10.1016/j.eswa.2019.112948>.

Shen S, Logeswaran L, Lee M, Lee H, Poria S, Mihalcea R. Understanding the capabilities and limitations of large language models for cultural commonsense. *arXiv [Preprint]*. 2024. arXiv:2405.04655. Available from: <https://arxiv.org/abs/2405.04655>

Zhou L, Schellaert W, Martínez-Plumed F, Moros-Daval Y, Ferri C, Hernández-Orallo J. Larger and more instructable language models become less reliable. *Nature*. 2024;1-8.
<https://doi.org/10.1038/s41586-024-07930-y>.

Pan S, Luo L, Wang Y, Chen C, Wang J, Wu X. Unifying large language models and knowledge graphs: a roadmap. *IEEE Trans. Knowl. Data Eng.* 2024.
<https://doi.org/10.1109/TKDE.2024.3388635>.

Kau A, He X, Nambissan A, Astudillo A, Yin H, Aryani A. Combining knowledge graphs and large language models. *arXiv [Preprint]*. 2024. arXiv:2407.06564. Available from: <https://arxiv.org/abs/2407.06564>

Zhu Y, Wang X, Chen J, Qiao S, Ou Y, Yao Y, et al. LLMs for knowledge graph construction and reasoning: recent capabilities and future opportunities. *World Wide Web*. 2024;27(5):58.
<https://doi.org/10.1007/s11280-024-01297-6>.

Shanahan M. Talking about large language models. *Commun. ACM*. 2024;67(2):68-79.
<https://doi.org/10.1145/3624724>.

Abu-Salih B, Alotaibi S. A systematic literature review of knowledge graph construction and application in education. *Heliyon*. 2024.
<https://doi.org/10.1016/j.heliyon.2024.e29477>.

Ferrara E, De Meo P, Fiumara G, Baumgartner R. Web data extraction, applications and techniques: a survey. *Knowl.-Based Syst.* 2014;70:301-23.
<https://doi.org/10.1016/j.knosys.2014.07.007>.

Bale AS, Ghorpade N, Rohith S, Kamalesh S, Rohith R, Rohan BS. Web scraping approaches and their performance on modern websites. In: 2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC). IEEE; 2022. p. 956-9.
<https://doi.org/10.1109/ICESC54411.2022.9885462>.

Thivaharan S, Srivatsun G, Sarathambekai S. A survey on Python libraries used for social media content scraping. In: 2020 International Conference on Smart Electronics and Communication (ICOSEC). IEEE; 2020. p. 361-6.
<https://doi.org/10.1109/ICOSEC49089.2020.9215386>.

Li Y, Wang B, Luan X. Xpath agent: an efficient XPath programming agent based on LLM for web crawler. arXiv [Preprint]. 2024. arXiv:2502.15688. Available from: <https://arxiv.org/abs/2502.15688>

Dagdelen J, Dunn A, Lee S, Walker N, Rosen AS, Ceder G, et al. Structured information extraction from scientific text with large language models. *Nat. Commun.* 2024;15(1):1418.
<https://doi.org/10.1038/s41467-024-45512-8>.

Bhushan RC, Donthi RK, Chilukuri Y, Srinivasarao U, Swetha P. Biomedical named entity recognition using improved green anaconda-assisted Bi-GRU-based hierarchical ResNet model. *BMC Bioinform.* 2025;26(1):34.
<https://doi.org/10.1186/s12859-025-06043-7>.

Sun Y, Wang S, Li Y, Feng S, Chen X, Zhang H, et al. ERNIE: enhanced representation through knowledge integration. arXiv [Preprint]. 2019. arXiv:1904.09223. Available from: <https://arxiv.org/abs/1904.09223>

Zhang Z, Han X, Liu Z, Jiang X, Sun M, Liu Q. ERNIE: enhanced language representation with informative entities. arXiv [Preprint]. 2019. arXiv:1905.07129. Available from: <https://arxiv.org/abs/1905.07129>

Huang Z, Xu W, Yu K. Bidirectional LSTM-CRF models for sequence tagging. arXiv [Preprint]. 2015. arXiv:1508.01991. Available from: <https://arxiv.org/abs/1508.01991>

Sun Y, Wang S, Li Y, Feng S, Tian H, Wu H, et al. ERNIE 2.0: a continual pre-training framework for language understanding. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2020;34:8968-75.
<https://doi.org/10.1609/aaai.v34i05.6428>.

Sun Y, Wang S, Feng S, Ding S, Pang C, Shang J, et al. ERNIE 3.0: large-scale knowledge enhanced pre-training for language understanding and generation. arXiv [Preprint]. 2021. arXiv:2107.02137. Available from: <https://arxiv.org/abs/2107.02137>

Siami-Namini S, Tavakoli N, Siami Namin A. The performance of LSTM and BiLSTM in forecasting time series. In: *2019 IEEE International Conference on Big Data (Big Data)*. IEEE; 2019. p. 3285-92.
<https://doi.org/10.1109/BigData47090.2019.9005997>.

Bale TL, Vale WW. CRF and CRF receptors: role in stress responsivity and other behaviors. *Annu. Rev. Pharmacol. Toxicol.* 2004;44(1):525-57.
<https://doi.org/10.1146/annurev.pharmtox.44.101802.121410>.

Green A, Junghanns M, Kießling M, Lindaaker T, Plantikow S, Selmer P. openCypher: new directions in property graph querying. In: *EDBT*. 2018. p. 520-3.

Ali D, Asaad A, Jimenez MJ, Nanda V, Paluzo-Hidalgo E, Soriano-Trigueros M. A survey of vectorization methods in topological data analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* 2023;45(12):14069-80.
<https://doi.org/10.1109/TPAMI.2023.3297584>.

Edwards RG, Joo B. The Chroma software system for lattice QCD. arXiv [Preprint]. 2004. arXiv:hep-lat/0409003. Available from: <https://arxiv.org/abs/hep-lat/0409003>

Li Y, Gu C, Dullien T, Vinyals O, Kohli P. Graph matching networks for learning the similarity of graph structured objects. In: *International Conference on Machine Learning*. PMLR; 2019. p. 3835-45. Available from: <http://proceedings.mlr.press/v97/li19d.html>

Rodríguez-Serrano JA, Perronnin F, Lladós J, Sánchez G. A similarity measure between vector sequences with application to handwritten word image retrieval. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE; 2009. p. 1722-9.
<https://doi.org/10.1109/CVPR.2009.5206769>.

White J, Fu Q, Hays S, Sandborn M, Olea C, Gilbert H, et al. A prompt pattern catalog to enhance prompt engineering with ChatGPT. arXiv [Preprint]. 2023. arXiv:2302.11382. Available from: <https://arxiv.org/abs/2302.11382>

Guu K, Lee K, Tung Z, Pasupat P, Chang M. Retrieval augmented language model pre-training. In: International Conference on Machine Learning. PMLR; 2020. p. 3929-38. Available from: <http://proceedings.mlr.press/v119/guu20a.html>

Ji S, Pan S, Cambria E, Marttinen P, Philip SY. A survey on knowledge graphs: representation, acquisition, and applications. *IEEE Trans. Neural Netw. Learn. Syst.* 2021;33(2):494-514.
<https://doi.org/10.1109/TNNLS.2021.3070843>.

Tian S, Luo Y, Xu T, Yuan C, Jiang H, Wei C, et al. KG-Adapter: enabling knowledge graph integration in large language models through parameter-efficient fine-tuning. In: Findings of the Association for Computational Linguistics ACL 2024. 2024. p. 3813-28.

Chen Y, Wen Z, Fan G, Chen Z, Wu W, Liu D, et al. MAPO: boosting large language model performance with model-adaptive prompt optimization. *arXiv [Preprint]*. 2024. arXiv:2407.04118. Available from: <https://arxiv.org/abs/2407.04118>

Marvin G, Hellen N, Jjingo D, Nakatumba-Nabende J. Prompt engineering in large language models. In: International Conference on Data Intelligence and Cognitive Informatics. Cham: Springer; 2023. p. 387-402.
https://doi.org/10.1007/978-981-99-7960-0_28.