Engineering Science and Technology, an International Journal 52 (2024) 101675

Contents lists available at ScienceDirect

Engineering Science and Technology, an International Journal

journal homepage: www.elsevier.com/locate/jestch

Full Length Article

# Innovative agricultural ontology construction using NLP methodologies and graph neural network

Krithikha Sanju Saravanan *, Velammal Bhagavathiappan

*Department of Computer Science and Engineering, College of Engineering, Guindy, Anna University, Chennai, India*

ARTICLE INFO

*Keywords:*
Term extraction
Relation extraction
Natural language processing
Regular expressions
Graph neural network

ABSTRACT

Advancements in technology brought various innovations to agricultural practices. As a part of the development, establishing an agricultural ontology would unleash the growth of cross-domain agriculture and Natural Language Processing (NLP). For constructing such domain-based ontology, semantic and syntactic understanding of the domain data is needed. In agriculture, the availability of pre-determined domain-based data is not sufficient hence, a standard methodology with syntactic and general semantic features are required for processing the data. In this research work, Agricultural Domain based Ontology Construction (ADOC) is proposed and the overall framework has three approaches for establishing the agriculture domain based ontologies. The input text documents undergo anaphora resolution phase utilizing the semantic-based method. In the first method of ADOC the ontology is developed using the terms and relationships that are extracted from the NLP techniques. The second method of ADOC uses pretrained BERT model and Hearst patterns while the third model of ADOC is based on pretrained BERT with regular expressions and unsupervised Graph Neural Network (GNN) for creating the agricultural ontology. The efficacy of the proposed ADOC utilizing BERT with regular expressions and GNN method shows an outstanding result when compared to other proposed and prevailing systems, with a precision and recall of 96.67% and 98.31%.

## 1. Introduction

Agriculture is an essential process for the survival and progress of the human race. It also plays a powerful role in supporting the livelihoods around the world by encompassing many activities related to raising livestock, cultivating crops etc., [1]. A country's Gross Domestic Product (GDP) is significantly influenced by the agriculture [2] because the agricultural sector, natural resources, government policies with overall economic structure are the major factors for the variations in GDP of the country. Advancements across various sectors, elevate the process of the agriculture department for increasing the production. In this data era, information serves as the backbone for the upliftment of farming [3] by furnishing farmers to gain invaluable knowledge and resource management. Hence accurate information on market trends, innovative farming techniques with sustainable practices enable the farmers to take enlightened decisions for enhancing their profitability.

Web resources plays a crucial role in the generation and dissemination of the domain specific data through information sharing, data crowdsourcing, artificial intelligence systems, research and innovations. Multi-modal data with respect to agriculture is constantly increasing through online media, newspapers, articles, and other internet websites. Considering the text data for the agriculture domain from various internet sources, knowledge extraction with high accuracy is a huge research problem [4]. Extracting useful information from domain-specific data needs a complete understanding of the domain syntactically and semantically. One such knowledge extraction process is the construction of ontology. Developing a domain specific ontology, keywords for that domain and semantic understanding of the sentences are important [5,6]. Extracting those domain-specific terms from the input text is a challenging task [7,8]. An efficient information extraction system should also withdraw the compound terms of the domain effectively.

Creating an ontology for the agriculture domain has a huge scope of research in the field of information retrieval [9]. Ontology studies the relationship between words i.e., entities and often those relationships are represented using directed graphs connecting the entities. Establishment of the relationships between the agriculture terms from text documents unveils the opportunities for automatic responsive systems

* Corresponding author.
*E-mail address:* krithikhasanju3008@gmail.com (K. Sanju Saravanan).

https://doi.org/10.1016/j.jestch.2024.101675
Received 6 July 2023; Received in revised form 21 February 2024; Accepted 15 March 2024
Available online 19 March 2024

like Interactive Voice Response (IVR) and Domain-specific chatbots [10–12]. While constructing the effective ontology from text documents of agriculture the following points must be considered,

1. Identify the entities from document using the semantic understanding of the sentences.
2. Extract the relevancy between the entities using both semantic and syntactic features of the text.
3. Recognize the standard relationships between the entities for creating an ontology.

The ADOC system consists of following contributions regarding the domain specific terms and relationships extraction,

1. Nominal anaphora resolution is done for the collected dataset so that no information is lost while extracting the entities.
2. In the first method, the agriculture terms are extracted automatically using NLP techniques and the relationship between the terms are diagnosed, then the ontology is created using standard relationships of domain using Hearst patterns with rule based approach.
3. In the second method, pre-trained BERT model with NLP techniques is employed to extract the entities using embeddings, then Hearst pattern with regular expressions and positional features are used for recognizing relationships.
4. In the third method, Pre-trained BERT model with NLP techniques is utilized for generating the word embeddings, then the unsupervised GNN with regular expressions and positional features is used for analysing the embeddings to detect the relationships.
5. Hyperparameter tuning has been done for optimizing the GNN model.

Various relationships for the agricultural domain based ontology are extracted and tabulated in the later sections. Our proposed ADOC system using GNN achieves encouraging results on comparing with the contemporary systems. The ontology graph constructed contains words on its node and relation at the edges.

In this research article, section 2 describes about the detailed literature survey that has been done for the proposed ADOC, section 3 gives the detailed design and explanation about the overall methodologies, section 4 discusses the step by step implementation results obtained for the proposed ADOC system, section 5 is the conclusion and section 6 provides the references for the ADOC research work.

## 2. Literature survey

In this section, literature survey for the ADOC work has been carried out and it is classified under three subtitles based on the functionalities, namely, anaphora resolution, automatic term extraction and relationship extraction for ontology.

### 2.1. Anaphora resolution

Anaphora Resolution is one of the major processes in the field of NLP that helps for better information retrieval. There are several nominal anaphora resolution methods in which anaphors are mapped against their corresponding antecedents. The presence of anaphors affects the understanding of the domain specific texts while extracting the entities and relationships. Semantically oriented rule-based approach is used by [13] for resolving the sortal anaphora in the biomedical literature. [14] proposed a mention ranking model for mapping the abstract anaphors with their antecedent using Bidirectional Long Short Term Memory (BiLSTM). Word embeddings based deterministic algorithms are utilized for semantically choosing the antecedents to replace the anaphors [15]. A multilingual based anaphora resolution was done with pair wise ranking method from the extracted words and their features [16]. Anaphors are resolved using rules-based structure with NLP based pre-processing where linguistic filters are also used [17].

[18] proposed a statistical model that automatically captures semantic and syntactic features for resolving the anaphors by encoding the input texts and decoding the anaphora in chain form with attention mechanism. [19] proposed a hierarchical neural network consisting of a hierarchically stacked BiLSTM layer with a Maxpool layer for mapping the anaphors with its antecedents in the conversational systems. Further to improve the gradient flow, skip connections were introduced in the existing neural network. For modelling the anaphors with antecedents, semantic and syntactic with lexical constraints using Markov Logical Network (MLN) and relatedness embeddings are used [20]. Language specific rules and the set of features from ensemble classifiers for identifying and replacing the pronominal type anaphors are constructed [19]. Combination of machine learning techniques with NLP methods handles the anaphora resolution effectively [21]. An ensemble model of Convolutional Neural Network with a source aware n-gram model is used for resolving the anaphors [22].

On surveying the above-mentioned frameworks, the following should be addressed in the proposed system. Identification of nominal anaphors and while matching the identified anaphors with its antecedents should not be done only with word embeddings, positional arguments should also be considered with word embeddings to avoid ambiguity in the further processing.

### 2.2. Automatic agriculture term extraction

A new algorithm which improves the keyphrase assignment algorithm named as KEA++ was proposed by [23] for automatically extracting the agricultural terms from the document involving candidate identification and filtering. AGROVOC is also employed as a knowledge base for semantic matching and controlled vocabulary in the KEA++ algorithm. [24] proposed a method for extracting the terms automatically using regular expressions that are obtained from the parts of speech (POS) information, domain specific patterns and statistical properties of the entities. The regular expression based term extraction method is also compared with Termine software for term extraction which uses c-value method. A basic Named Entity Recognition (NER) technique with regular expression and NLP methodologies named as Regular Expression and NLP based Term Extraction scheme (RENT) is utilized for extracting the agriculture terms automatically [25]. The RENT method also uses AGROVOC, Wordnet and NAL thesaurus for extracting more terms effectively. [26] designed a term extraction method using tokenization and POS patterns of NLP. [27] recommended a custom NER model from python's Spacy library for identification and extraction of agriculture domain specific terms from unstructured text documents. Table 1 summarizes the existing works in the agriculture term extraction.

Apart from the above methods, there are few more methods to extract the domain specific terms from the documents. C-Value is one of the parameters that detects the terms using frequency estimation and linguistic filters [28]. NC-Value is another parameter which recognizes the multi-word term using linguistic information and statistical information [28,29]. Term Recognition Using Combined Knowledge Sources (TRUCKS) [30] is the hybrid technique for term extraction which incorporates shallow linguistics information and statistical features using NC-Value. Adding to that, information weight based on the contextual information is also added to NC-Value to select the context-rich terms whose value is named as SNC-value. SNC-Value is used for the final ranking and selection of the terms.

Terms can also be extracted from the given text using syntactic and semantic graphs. Usually, the vertices of the graphs are occupied with words/terms and edges are filled with contexts/relationships. A bidirectional dependency graph is used to represent the given text and terms are extracted using the combination of Conditional Random Field (CRF) and Long-Short Term Memory (LSTM) [31]. In [32], four different classification techniques (Naive Bayes Classifier, Logistic Regression, Support Vector Machine, Random Forest Algorithm) are utilized for

**Table 1**
Summary of the existing works for extracting the agriculture terms.

| Title | Objective | Methodology | Remarks |
|---|---|---|---|
| Thesaurus-based index term extraction for agricultural documents. [23] | To extract agriculture terms automatically from the text document. | KEA++ algorithm with AGROVOC. | It eliminates the incorrect and meaningless phrases by using the controlled vocabulary. Achieves 52.9 % precision, 47.9 % recall and f-measure of 46.8 %. |
| A practical approach for term and relationship extraction for automatic ontology creation from agricultural text. [24] | Automatic extraction of agriculture terms from the text document. | Regular expression based on POS information, domain specific patterns and statistical properties. | It outperforms the Termine software for term extraction. Precision of 85.47 % is obtained. |
| RENT: Regular expression and NLP-based term extraction scheme for agricultural domain. [25] | Automatic agriculture terms including composite terms extraction are addressed. | RENT algorithm with AGROVOC, Wordnet and NAL thesaurus. | It outperforms the Termine software for term extraction. Secured precision more than 80 %, recall more than 60 % and f-measure more than 68 % for random samples. |
| An effective automated ontology construction based on the agriculture domain. [26] | Term extraction using NLP for agriculture text document. | Tokenization and POS patterns. | All words in the sentences are extracted with POS tagging and the agriculture words are extracted using POS tags patterns. |
| Developing a Model for the Automated Identification and Extraction of Agricultural Terms from Unstructured Text. [27] | Automatic agriculture term extraction from unstructured text documents. | Customized NER model using Spacy library in python. | The model possesses the precision of 50.73 %, recall of 54.52 % and f-measure of 51.81 %. Hence for further improving the efficiency of the model, AGROVOC and other vocabularies can be used. |

ensemble methods to find the keywords from the given text documents.

On surveying the above-mentioned models, the following issues should be addressed to provide an efficient Agricultural-based Automatic Term Extraction system. Effective identification and extraction of compound and ambiguous terms must be addressed by syntactic, semantic and positional features. Hence, there should be significant improvement in the scores of the evaluation metrics.

### 2.3. Relationship extraction for agricultural ontology

Relationships for the ontology construction can be extracted from the texts using a hierarchy of linguistic filters [24]. Four types of relationships for creating the agricultural ontology are designed using position vectors, patterns and wordnet similarity measures. This relationship extraction technique is evaluated by 10 fold cross validation method. [33] proposed a two steps framework for developing agriculture ontology. In the first step, domain specific regular expressions with NLP techniques are employed for automatic term extraction and in the second step, identification of semantic relationships between the extracted terms using the proposed RelExOnt method. [34] recommended the ontology model for the diseases and pest management of the grape crop in India. It also uses current weather data that can stored and used for forecasting, then using the internet the knowledge base for pest and disease is generated and stored as the OWL document. Here the TF-IDF method with AGROVOC verification process is used to extract terms and then GrapeOntoGenerator constructs the ontology automatically using Prodege API. [35] demonstrated the purpose agriculture ontology in the knowledge management system. The ontology is constructed using AGROVOC and Agriculture Ontology Service (AOS) used by the Food and Agriculture Organization of the United Nations (FAO). It also uses OWL for easily formatting the text to ontology. [36] developed OWL based ontology for vertical farm based on the relationships for controlling and monitoring services. The novel fertilization, nutrition imbalance and irrigation based ontology for hilly citrus crops was established by [37] using the data stored in RDF triplet formats. [38] established a Flora Phenotype Ontology (FLOPO) based on the NLP techniques and OWL. A rule based approach for the creation of agricultural ontology was formulated by [39]. [26] proposed a framework for constructing ontology using Formal Concept Analysis (FCA) and Jaccard similarity for extracting the entities with relationships. [40] established a pest-control ontology encompassing both theoretical and practical foundations. Subsequently, an ontology-driven web application was devised to effectively assist farmers in their pest-control decision-making. Lastly, a method for ontology development and evaluation was introduced and thoroughly demonstrated using the example of a pest-control ontology. [41] recommended a lattice structure for the knowledge representation of data gathered from IoT devices for smart agriculture. This lattice structure is constructed from spatiotemporal data and subsequently, rules are generated based on the properties of the lattice to facilitate reasoning. [42] developed a model for knowledge mapping in digital agriculture based on ontology to represent extracted knowledge from data mining tasks.

A framework for evaluating agricultural ontology is introduced in [43] blending concepts from key existing evaluation methods and aligning specific evaluation techniques with the intended purpose of the agricultural ontology. The application and user-friendliness of the framework are subsequently illustrated using the example of a pest-control ontology.

[44] centered on constructing a knowledge base ontology specifically for the domain of Climate Smart Agriculture (CSA). Data for this ontology was collected from secondary sources including websites, published research articles, reports and relevant ontologies. The formalization of the OntoCSA ontology was accomplished using Description Logics. The ontology was developed in the OWL using Protégé. Additionally, successful validation of the OntoCSA ontology was achieved using the HermiT reasoner within the Protégé platform. [45] constructed an agriculture domain ontology taking into account factors such as geographical information, soil characteristics, diseases, organic fertilizers, the availability of fertilizers and crop demand.

[46] conducts a comprehensive review, discussion, comparison and critique of various approaches and systems for constructing ontologies from text. Through the examination of existing ontology construction systems and approaches highlighted a consensus on specific challenges in automatic ontology construction that demand additional efforts. Challenges such as axioms learning, relation discovery, transformation of small/large-scale input data, reduction of human intervention and the establishment of a standard platform for evaluating ontology construction systems are identified as crucial. Subsequently, the paper advocates for a shift towards Deep Learning (DL) over traditional methods for Ontology Learning (OL) and provides the rationale behind this

recommendation. Table 2 summarizes the existing works of agriculture domain based ontology construction from relationship extraction.

On surveying the above-mentioned models, the following issues should be addressed to extract the relationships. Enhanced similarity measures should be created and relationships should be found with the help of similarity measures and pattern recognition. Recognizing the relationships with semantic understanding helps to understand the complex relations without ambiguity.

### 2.4. Challenges

The following are the challenges faced in the construction of the Ontology.

1. Anaphora resolution for domain-specific documents is a challenging task due to lack of domain information. The information lag is the key issue in solving NLP problems.
2. Extracting the agricultural entities is the most important step for constructing nodes in the ontology.
3. The domain knowledge is essential for processing with NLP techniques. The lack of domain knowledge has an effect even in identifying the basic text entities such as compound terms and also in distinguishing the domain-related text from the other domain text.
4. Establishing the standard relationship between the entities is a complex process because of ambiguity in the text between the entities and also, choosing the relationship for constructing a structured and defined ontology is difficult because of its convergence for the practical use.

Compared to the literature survey, the proposed ADOC system is unique in the following ways,

1. In the initial stages of ontology construction, the domain knowledge is manually given as input. The collection of domain words is used to implement the elementary understanding of the agricultural domain in the proposed ADOC system. And, also for addressing the lack of domain knowledge, syntactic features are used efficiently to balance it.
2. Semantic and syntactic features with positional features in the text are used for resolving the anaphora efficiently.
3. The compound terms, and other agricultural entities are extracted effectively using the trained N-gram model along with linguistic filters.
4. Using semantic similarity along with the syntactic features and positional features, the relationships between the extracted domain-specific terms are established.

In the ADOC framework, the resolved anaphors were integrated into the ontology construction process ensuring that relationships and associations between entities are accurately represented, contributing to a more cohesive and meaningful ontology. The challenges addressed have been successfully surmounted through the application of NLP techniques, DL methods and the active participation of domain experts.

## 3. Materials and methods

In this section, the ADOC system is fully explained in detail. The proposed ADOC work aims to construct an ontology for the agricultural domain from the input documents and the overall architecture diagram containing the methodologies involved is shown in Fig. 1. For creating an efficient ontology, the agricultural entities and relationships between the entities should be identified and extracted properly. While handling the text documents for processing, the existence of anaphors is very common and it has to be rectified in order to avoid the information loss. So, the input text documents have to undergo anaphora resolution phase using semantic based approach.

**Table 2**
Summary of the existing works for agriculture domain based ontology construction from relationship extraction.

| Title | Methodology | Extracted relationships | Remarks |
|---|---|---|---|
| A practical approach for term and relationship extraction for automatic ontology creation from agricultural text. [24] | position vectors, patterns and wordnet similarity measures. | Synonym, is_a, is_type_of, intercrop. | Relationship extraction techniques is evaluated by 10 fold cross validation method. This method is successful with average precision of 88 % on training data and 87 % on testing data. |
| Automatic relationship extraction from agricultural text for ontology construction. [33] | RelExOnt – it is the rule based framework for extracting the semantic relationships. | has_synonym, is_a, is_type_of, is_intercrop. | This method is successful with average precision of 86.89 %. |
| Ontology based system for pests and disease management of grapes in India. [34] | TF-IDF followed by AGROVOC based words verification and then constructs the ontology automatically using Prodege API. | Relationships based on pest and diseases in grapes. | Further the system uses fuzzy inference system with rule based expert system for providing a precise forecasting information of pests and disease. |
| Construction of the ontology-based agricultural knowledge management system. [35] | AGROVOC with AOS, OWL. | According to the domain prototype the concepts and relations are extracted. | Further it is used for the various agriculture related knowledge management systems. |
| An OWL-based ontology model for intelligent service in vertical farm. [36] | OWL based model. | Relationships for both controlling and monitoring services in the vertical farm. | No need to start from the base, it can be reused according to the domain of interests. |
| An ontology-based approach to integration of hilly citrus production knowledge. [37] | RDF triplet based model. | Fertilization, nutrition imbalance and irrigation based relationships. | The ontology constructed is used as a knowledge base for the hilly region citrus crops. |
| The flora phenotype ontology (FLOPO): tool for integrating morphological traits and phenotypes of vascular plants. [38] | NLP techniques using NLP tools and OWL. | Relationships of flora phenotype. | Helpful for the identification of more vocabularies related to the flora phenotype and for constructing other domain specific ontologies related to flowering plants. |
| Rule-based approach for automatic ontology population of agriculture domain. [39] | Rule based approach. | growsIn, hasOrigin, hasKnownAs, hasArilColor, hasArilTaste, hasSpineShape, hasSpineColor, hasFruitShape, hasFruitSize. | Effectively created the model which is fully automated. |
| An effective automated ontology | FCA and Jaccard similarity. | Soil based, weather based and pest based. | Precision score of 94.40 %, recall score of 89.21 % |

(continued on next page)

**Table 2** (*continued*)

| Title | Methodology | Extracted relationships | Remarks |
|---|---|---|---|
| construction based on the agriculture domain. [26] | | | and f-measure of 90.04 % are obtained |
| Addressing the 'Tower of Babel' of pesticide regulations: an ontology for supporting pest-control decisions. [40] | OWL based model | Pest control based. | The constructed ontology is utilized in web application to assist farmers for pest control decision making. Clarity, coherence, minimal encoding bias, conciseness, completeness metrics are used to evaluate the ontology. |
| An Ontological Knowledge Representation for Smart Agriculture [41] | Lattice structure for the knowledge representation | Smart agriculture based | Lattice structure is constructed from spatiotemporal data and rules are generated based on the properties of the lattice to facilitate reasoning. |

After resolving the anaphors in the text documents, the workflow for establishing the agriculture domain based ontology is divided into three sections, where NLP based techniques are used in the first section for developing the ontology. In the second section, the pretrained BERT model with NLP methodologies and regular expressions are utilized for building the ontology. The third section is based on the pretrained BERT model with NLP methodologies, regular expressions and GNN for fabricating the ontology. The designed agriculture ontology possesses extracted agricultural terms at nodes and relationships between the terms at the edges. For each method in ADOC, the block diagrams are shown in Figs. 2, 3 and 4.

### 3.1. Nominal anaphora resolution

In anaphora resolution phase of the agriculture domain based text documents, anaphors are mapped to the their corresponding antecedents. Anaphora resolution is an important process in removing the ambiguities present in the text. So that by resolving the anaphors, exact and more accurate information can be retrieved from the input documents. The pseudocode for the proposed nominal anaphora resolution algorithm is given in algorithm 1.

**Algorithm 1:** Semantic-based Anaphora Resolution for Agricultural Documents

Input: Input Text Document (D)
Output: Anaphora Resolved Text (D-resolved)
Procedure: Sentence Accumulator(D_entities, D)
Sent acc = {}
for element ∈ D_entities do

(*continued on next page*)

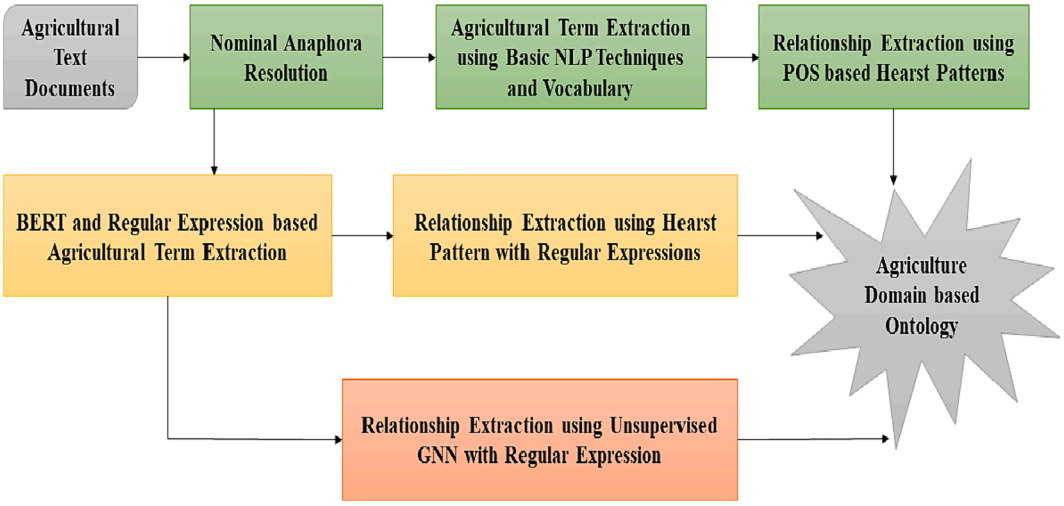


**Fig. 1.** Overall Architecture Diagram for the Proposed ADOC Framework

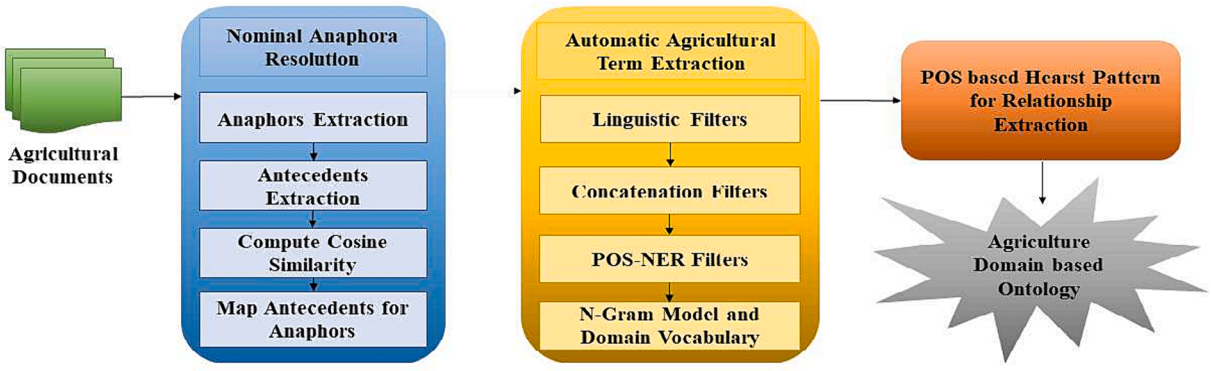


**Fig. 2.** Block Diagram of Ontology Construction using NLP techniques.

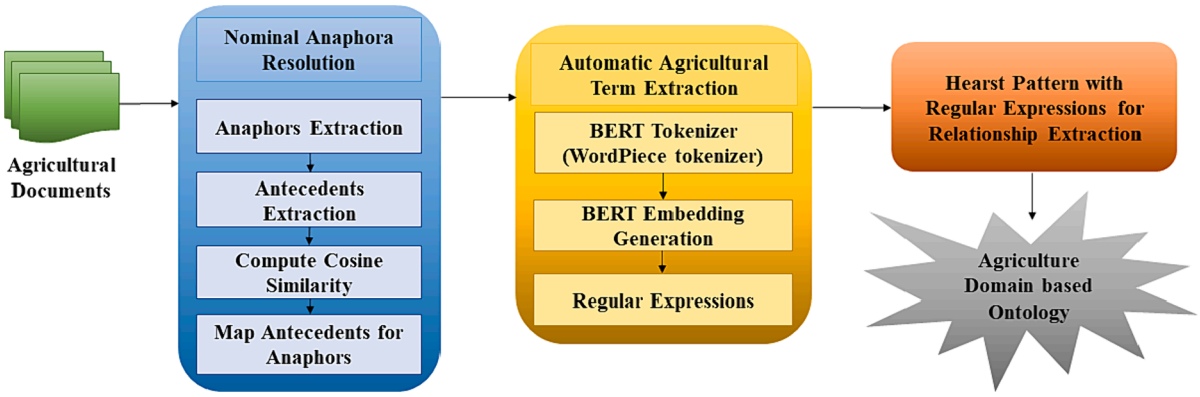


**Fig. 3.** Block Diagram of Ontology Construction using BERT model with Regular Expressions and NLP techniques.

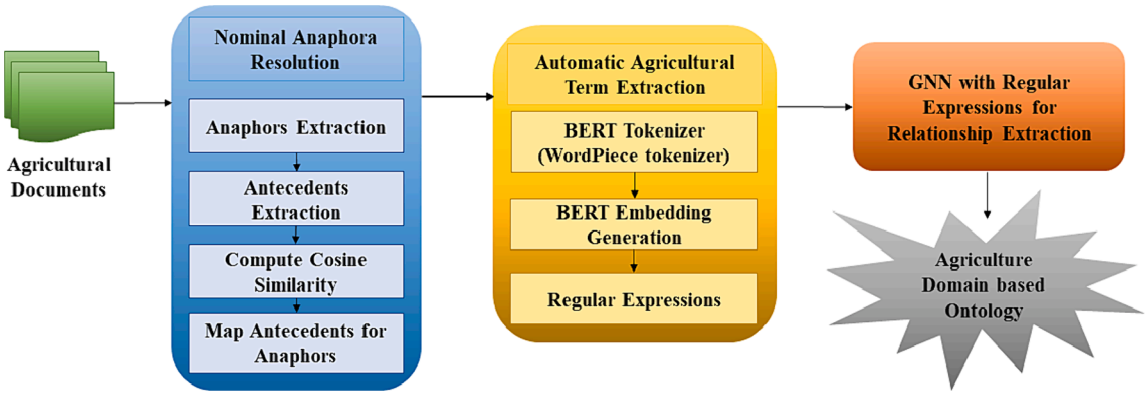


**Fig. 4.** Block Diagram of Ontology Construction using BERT model and GNN model with Regular Expressions.

(*continued*)

**Algorithm 1:** Semantic-based Anaphora Resolution for Agricultural Documents

```
    key = element.value
    value = D.split(".")[element.sent no]
    sent acc.append({ key: value })
endreturn sent acc
Procedure: Mapper(DAP, DAC, D)
Anaphora sent = Sentence Accumulator(DAP)
Antecedent sent = Sentence Accumulator(DAC)
SS Matrix = Sentence Similarity Matrix(DAP, DAC, measure = Cosine similarity)
AP-AC-MAP = {}
for entity ∈ DAP do
    key = DAP
    value = index(Max(SS Matrix[DAP.value]))
    AP-AC-MAP.append({key: value})
endD-resolved = Sentence substitution(D, AP-AC-MAP)
return D-resolved
```

The semantic and syntactic level features for the words in the documents are extracted using the NLP annotators like prodigy annotator [47], CoreNLP annotator [48] and AllenNLP annotator [49]. The annotated text words of the input agricultural document consist of POS tagging, word number, sentence number and semantical dependency of words. The NP identifier function in the algorithm extracts the noun phrases present in the annotated sentence. The Anaphoric Filter and Candidate Selector extract all the possible anaphors and antecedents from the annotated text. The Mapper function returns the resolved text by having the list of possible anaphors (DAP), the list of antecedents (DAC) and the text (D). The corresponding sentences for the words in the available list of antecedents and anaphors are extracted, then a similarity matrix is computed with the help of cosine similarity function. The cosine similarity [50–52] denotes the cosine of the angle in between the given two vectors. Since the given documents are related to single-domain i.e., agriculture, the direction of vectors should not deviate. So, cosine similarity measure gives the best result compared to the other measures. Therefore, semantic similarity is calculated from the computed cosine similarity measure, where the row represents the sentences from the list of anaphors and the column represents the sentences from the list of antecedents. The row-wise maximum is taken into account where the sentence for the antecedent is matched with the sentence for the anaphora. Then, the anaphora is substituted with the antecedent in the input documents. The result of this phase is anaphora resolved agricultural document.

### 3.2. *Agricultural term extraction*

In this section, the input is the anaphora resolved agricultural document and the output is the retrieved agricultural domain terms from the document. The domain based term extraction is a crucial and essential process for creating the domain specific ontologies. The important and required domain oriented terms from the input documents should be retrieved and those terms serves as the nodes in the ontology. By considering many use cases covering the vast scope of the specified agriculture domain can lead to standard ontology formation.

In the first method, the proposed NLP based algorithm consists of two divisions, namely, extraction of all possible candidate domain terms and term selection using agriculture vocabulary created from different sources such as AGROVOC [53], DSpace dictionary of agriculture, ChatGPT [54]. The pseudocode for the NLP based term extraction algorithm is given in algorithm 2. Candidate agriculture terms in the document are extracted using linguistic filters and rule-based mechanisms. Term selection is the process of choosing only the terms from the candidate terms that are most relevant to the domain with the use of customized vocabulary of domain specific words. By utilizing the NLP

based term extraction algorithm, the candidate term extraction phase consists of various NLP based filters. The input document is given to the character filter, where the words with a range of minimal and maximal lengths are chosen. So, the resulting candidates of the character filter are of a chosen range of length. Then, the tokenizer is used for the tokenization process where the sentences are broken down into smaller meaningful chunks of text.

ASCII filter [55,56] is applied on the extracted tokens and with the ASCII Equivalent, the non-ASCII characters are replaced or diacritics are removed. Next the concatenation filter is used for concatenating the concepts relating to the domain and it is designed especially for handling the compound and ambiguous agriculture terms effectively. Now the POS and NER tags are embedded for the filtered tokens and important terms are extracted with respect to the tags associated with tokens using rule based method. The resulting set of tokens of the POS and NER based filter is called partial terms since the majority of the extracted terms would be important terms that are present in the given text. A lowercase filter is applied to make all the characters in all the tokens to lowercase letters. Lemmatization [57,58] is the process of shortening the word to its root form. Lemmatization is applied on partial terms and the Natural Language Tool Kit's (NLTK) stopword removal python package is applied to remove the unimportant words from the partial terms. Hence by using the n-gram, the available multi-word terms belonging to agricultural domain are considered as a single term which can lead the domain term extraction model to be more efficient. The list of candidate terms is extracted using the N-gram model [59,60] which is trained only for retrieving the agricultural terms. The resulting list of words from the stop word removal and n-gram model is called as candidate list which is formed using another concatenation filter. Finally, the terms extracted are verified using the vocabulary organized using multiple sources.

**Algorithm 2:** Agriculture Term Extraction

```
Input: Anaphora Resolved Agricultural Document (Dresolved), Agriculture
  Vocabulary (AgVoc)
Output: Extracted Agricultural Terms (DTerms)
Dchar = Char Filter(Dresolved) //Applying character filter on D
DTokens = Tokenize(DChar) //Tokenization of Text
DASCII = ASCII Filter(DTokens) //ASCII Equivalent Replacement and Diacritic
  Elimination Filter
DConcat = Concat Filter (DASCII) //Concatenation Filter for Agriculture Domain
DPartTerms = POS_NER Filter(DConcat) //POS and NER Tagging and term filtering
DPartTermsLower = LowerCase Filter(DPartTerms) //Lowercase conversion filter
DPartTermsRoot = Lemmatization(DPartTermsLower)//Term Lemmatization
DPurePartTerms = Stopword Filter(DPartTermsRoot) //Stop word removal
DN-GTerms = N-Gram Term Extraction(DPurePartTerms) // N-Gram term extraction
DConcat2 = Concat Filter (DPurePartTerms, DN-GTerms) //Concatenation Filter for
  Agriculture Domain
DTerms = AgVoc(DConcat2)
return DTerms
```

In the second and third method, pretrained BERT model (BERT-Base-Uncased model) is used along with the regular expressions for retrieving the terms from the document. The stopwords and punctuations are removed from the document using NLTK in python. The uncased BERT-Base model has 12 layers of encoder stack with768 hidden units, 12 attention heads and 110 M parameters [61–63]. The multi-head self-attention layer and a feed forward layer are the two sub-layers of each layer, where both the sub-layers are followed by layer normalization with residual connections. For a domain based dataset, BERT employs the concept of pretraining the model in an unsupervised method for language modelling. Hence, BERT models are the deep contextual language representation models that belongs to the family of architectures called as transformers. Also, the Bert model is designed in a unique way to assist machines for understanding the meanings of the ambiguous languages present in the text by producing the word embeddings for the text. These word embeddings obtain the contextual information in the text. The BERT model has been pretrained with two important goals, i.e., Masked Language Modelling (MLM) and for predicting the next sentences. Hence, the pretrained BERT model is selected for extracting the agricultural terms from the anaphora resolved documents.

The BERT model used in this work is designed for pretraining deep bidirectional representation of words in the unlabelled text with joint conditioning on both sides of the contexts. The model has an embedding layer at the input end and a softmax layer at the output end. The embedding layer is used for converting the information in the input text to its numerical vector form, which is very suitable for further processing. The BERT model generates embeddings for the agricultural document which is the summation of three types of embeddings namely, token embeddings, segment embeddings and position embeddings. The token embeddings are for converting the words to its pretrained vector representations. The segment encoding helps by forming a vector through encoding the sentence number. The positional embeddings encode the position of all tokens in the sequence and it ranges from 0 to (512–1). The positional embeddings are used in BERT because it does not have any structure for sequences like Recurrent Neural Network, so the positional embeddings are for capturing the context with ordering of the text. The BERT model has two special tokens, [CLS] and [SEP] for proper understanding of the input texts. At the end of each input sentence, [SEP] token has to be inserted because it is used to decide the end of one input sentence and the start of the other in the sequence of texts. [CLS] is a special classification token and it is utilized in the last hidden units of the model. The overall architecture of BERT model is shown in Fig. 5.

The steps involved in the uncased BERT base model for obtaining the contextual embeddings for each token in the agricultural document are.

1. The input text paragraph is divided into tokens by applying the WordPiece tokenizer. Based on the vocabulary of 30,000 tokens, each token is mapped to a unique id. The input ids are truncated or padded to a fixed length of 512.
2. The segment ids are assigned to each token to indicate whether it belongs to the first or the second sentence in the input paragraph. The segment ids are either 0 or 1. If the input paragraph consists of only one sentence, then all segment ids are 0.
3. The attention masks are assigned to each token to indicate whether it is a real token or a padding token. The attention masks are either 1 or 0. The attention masks are used to avoid attending to padding tokens in the self-attention layers.
4. The input ids, segment ids, and attention masks are fed into the BERT model, which consists of an embedding layer and a stack of 12 encoder layers. Each encoder layer consists of a multi-head self-attention sublayer and a feed-forward sublayer, both followed by layer normalization and residual connections.
5. The embedding layer converts the input ids, segment ids, and position ids into embeddings and sums them to obtain the input embeddings. The position ids are learned up to a maximum position of 512. The input embeddings have a dimension of 768.
6. The input embeddings are passed through the encoder stack to obtain the contextual embeddings for each token. The contextual embeddings also have a dimension of 768. The contextual embeddings capture the semantic and syntactic information of each token in relation to the whole input paragraph.

After extracting the contextual embeddings, supervised regular expressions are used to extract the required domain specific terms from the input document.

### *3.3. Relationship extraction and ontology construction for agriculture domain*

In this section, the relationships between the agricultural terms extracted are identified and retrieved. The first method of ADOC after extracting the domain terms, the relationships between the terms are identified using the Hearst patterns [64,65]. Hence by utilizing the

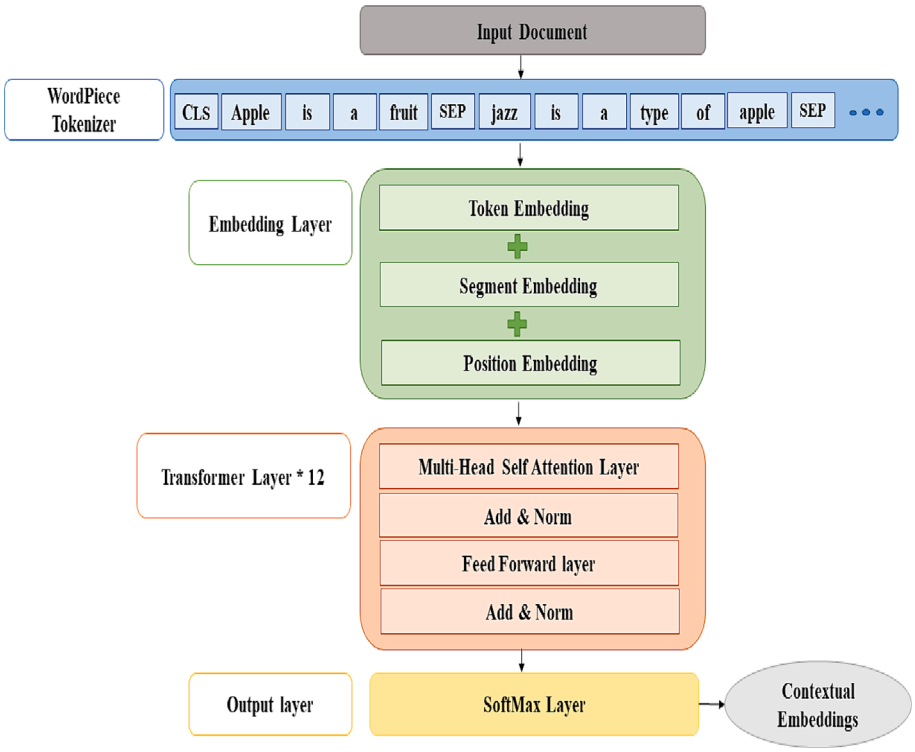


**Fig. 5.** Architecture Diagram of BERT model.

Hearst relation patterns, the relationships between the entities can be retrieved easily with semantic understanding. There are three categories of Hearst relations, namely, Hearst pattern 1, Hearst pattern 2 and Hearst pattern 3. The Hearst pattern 1 based relationship identification is a method for identifying hyponyms, which are words that are more specific types of a more general word. Hyponyms are a useful for understanding the meaning of words and for creating more specific relation patterns. The Hearst pattern 1 method uses a set of patterns to determine hyponyms that are found in the text document. For example, the pattern "is a" can be used to identify hyponyms. For example, in the sentence "Paddy is a type of food grain", the word "paddy" is a hyponym of the word "food grain". This is because the pattern "is a" is found in the sentence. It is a simple and effective method for identifying hyponyms. It has been used in a variety of applications, including natural language processing, information retrieval, and ontology learning.

Hearst pattern 1 technique uses a set of hand-crafted rules for identifying the meronymy and co-hyponymy relations. These rules are based on the document observations and these relations are often expressed using certain patterns of words. For example, the rule "A is a type of B" can be used to identify meronymy relations, such as "Alphonso is a type of mango". Hearst pattern 1 is a simple and easy to understand method. However, it is not very accurate because the rules used are based on a limited number of examples. As a result, Hearst pattern 1 is not able to identify all possible meronymy and co-hyponymy relations.

Hearst pattern 1 is the most basic pattern and it is most likely to be found in text. However, it is also the least specific pattern because it is difficult to use for extracting accurate hypernym relations. Hearst pattern 2 is more specific than Hearst pattern 1 and it is less likely to be found in text. However, it is also more likely to extract accurate hypernym relations. Hearst pattern 3 is the most specific pattern, and it is the least likely to be found in text. However, it is also the most likely to extract accurate hypernym relations. Hence, the Hearst pattern 2 is a good choice for most applications. However, for working with specific domain that is known to be difficult to extract hypernym relations then using Hearst pattern 3 yields better results and effectively recognizes the hypernymy relations. The lexical-syntactic pattern "X is a Y" can be used to identify the hypernymy relation in the sentence "orange is a fruit" between "orange" and "fruit". In this case, "orange" is the hyponym and "fruit" is the hypernym. So, Hearst pattern 3 is a powerful tool for extracting hypernymy relations from text.

The GNN is a type of neural network that handles the graph-structured data and can also be used for constructing the graph [66,67]. A graph consists of nodes (or terms) and edges (or relationships) that connect the nodes. Each node and edge can have some attributes or features associated with them. A GNN discovers a function that maps the nodes and edges with its features to the output, such as node labels, node embeddings, graph embeddings or graph labels. A GNN consists of two main components, a message passing layer and a readout layer. The message passing layer's work is to update the node features by aggregating the features of the neighboring nodes and edges. The process is repeated for multiple iterations by allowing the node features to capture both local and global information for creating the graph. The readout layer combines the node features to produce the output. A GNN model with PyTorch and PyTorch Geometric python packages are used with regular expressions for extracting the necessary relationships between the entities from text document. The input dimension is equal to the number of entities in the graph. The output dimension is also equal to the input dimension because the model is trying to reconstruct the input features as an autoencoder. The hidden dimension is arbitrarily chosen as 16. The constructed GNN models consists of two linear layers that is used for transforming the node features from input dimension to hidden dimension and from hidden dimension to output dimension. The forward method takes the node features and the edge indices as inputs and returns the updated node features after applying the linear layers and activation function. The GNN model is trained with the Mean Squared Error (MSE) loss function and optimizer. The required relationships with entities are extracted with the help of regular expressions. The flow diagram of the proposed GNN model is shown in Fig. 6. The proposed unsupervised GNN model generates its performance loss in the form of MSE. In order to model the GNN with less MSE, hyperparameter tuning has been done. In

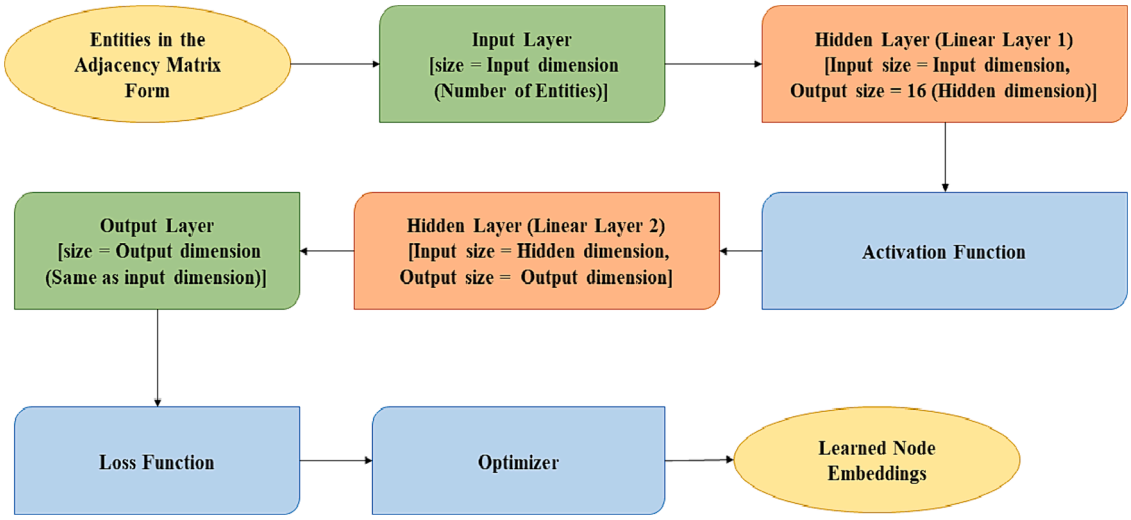


**Fig. 6.** Flow diagram of the proposed GNN model.

hyperparameter tuning of GNN, different learning rates, optimizers and activation functions are tried and their MSE is compared. After implementing with many combinations of the hyperparameters, the learning rate of 0.01shows the better performance with Mish activation function and Adam optimizer. The Table 3 shows the MSE losses obtained during the hyperparameter tuning of GNN model with learning rate of 0.01.

In this proposed ADOC work, mainly seven relationships (is a, is also, is type of/ are types of, is intercrop of, is cultivated in, disease in/ diseases in, fertilizer for/ fertilizers for) are considered and it is shown Table 4. In the first method, POS based Hearst patterns with positional feature are used to extract the relationships between the entities. In the second method, regular expressions are used with Hearst pattern and positional feature to retrieve the relationships between the entities. Finally, the third method uses a simple GNN with regular expressions for detecting the relationships between the entities. From the extracted terms and relationships, the constructed ontology graph is visualized using the Networkx and Matplotlib libraries in python.

The overall steps involved in the proposed BERT-GNN with regular expression method for extracting the entities and relationship are given below in a generalized form,

1. Imported the necessary libraries and modules. Torch is the main PyTorch library, torch.nn and torch.optim are modules for neural networks and optimization, torch_geometric.data is a module for graph data manipulation, re is a module for regular expressions, transformers is a library for natural language processing, and networkx is a library for graph analysis.
2. Loaded a pre-trained BERT model (Bert-base-uncased) and tokenizer from the transformers library.
3. Defined the input paragraphs as a string variable called text or given as a text document. This is the text that will be used to create the graph. It contains sentences that describe different entities and relations of the domain.
4. Tokenized the text using the BERT tokenizer. This means splitting the text into smaller units called tokens, which are the basic inputs for the BERT model. The tokens variable is a list of strings that represent the tokens. Then extract the embeddings for the document.
5. Defined a dictionary called entity patterns that contains regular expressions for different types of entities in the text. Regular expressions are patterns that can be used to match and extract specific strings from a larger text. For example, the pattern r"\b([A-Za-z] + )\s + is\s + a\s + fruit\b" matches any word that is followed by "is a fruit", such as "Apple is a fruit". The entity patterns dictionary maps each entity type to its corresponding pattern.

**Table 3**
The MSE losses obtained during the hyperparameter tuning of GNN model with learning rate of 0.01.

| Activation functionOptimizer | RELU | GELU | Swish | Mish |
|---|---|---|---|---|
| Adam | 0.01114 | 0.01073 | 0.01111 | 0.01071 |
| AdaGrad | 0.01124 | 0.01148 | 0.01282 | 0.01119 |
| SGD | 0.03706 | 0.03360 | 0.08594 | 0.04002 |
| RMSProp | 0.01152 | 0.01074 | 0.01122 | 0.01072 |
| AdaDelta | 0.03494 | 0.03501 | 0.09330 | 0.03617 |
| SGD with momentum | 0.02598 | 0.02855 | 0.01898 | 0.02845 |

**Table 4**
Relationships extracted in the proposed ADOC work with example.

| Relation Pattern | Example Text | Relation Extracted |
|---|---|---|
| Is a (e1, e2) | Paddy is a cereal grain.<br>Orange is a citrus fruit. | Is a (paddy, cereal grain)<br>Is a (Orange, citrus fruit) |
| Is also (e1, e2) | Paddy is also known as rice paddy. | Is also (paddy, rice paddy) |
| Intercrop of (e1, e2) | Cowpea is an intercrop of paddy. | Intercrop of (cowpea, paddy) |
| Type of (e1, e2) / types of (e1, e2) | Narendra Osar Paddy-3 is a type of paddy.<br>Narendra Osar Paddy-3, Narendra Paddy-5050, Narendra Osar Paddy-2008, Narendra Osar Paddy-2009 are the types of paddy. | Type of (Narendra Osar Paddy-3, paddy)<br>Type of (Narendra Osar Paddy-3, paddy), Type of (Narendra Paddy-5050, paddy), Type of (Narendra Osar Paddy-2008, paddy), Type of (Narendra Osar Paddy-2009, paddy) |
| Cultivated in (e1, e2) | Paddy is cultivated in October.<br>Paddy is cultivated in rainy season. | Cultivated in (paddy, October)<br>Cultivated in (paddy, rainy season) |
| Disease in (e1, e2) / diseases in (e1, e2) | Blast is a disease in paddy.<br>Blast, bacterial blight, sheath rot are the diseases in paddy. | Disease in (blast, paddy)<br>Disease in (blast, paddy), Disease in (bacterial blight, paddy), Disease in (sheath rot, paddy) |
| Fertilizer for (e1, e2) / fertilizers for (e1, e2) | Triazole and strobilurin are the fertilizers for paddy.<br>Tricyclazole is the fertilizer for blast. | Fertilizer for (triazole, paddy), fertilizer for (strobilium, paddy)<br>Fertilizer for (tricyclazole, blast) |

6. Created an empty dictionary called entities that will store the extracted entities and their types from the text based on the embeddings from BERT and regular expression patterns. To extract the required domain terms, loop over each entity type and pattern in the entity patterns dictionary and uses the function to find all matches in the text. For each match, it adds the matched string and its entity type to the entities dictionary. For example, it will add "apple" and "fruit" as a key-value pair to the entities dictionary.
7. Created a directed graph using networkx library. A directed graph is a data structure that consists of nodes and edges, where each edge has a direction from one node to another. Nodes can represent entities and edges can represent relations between them. A directed graph is used to model the ontology graph.
8. Added nodes to the graph using graph.add_node(). It loops over each entity and entity type in the entities dictionary and adds them as nodes to the graph. It also assigns an attribute called type to each node that stores its entity type. For example, it will add "apple" as a node with type "fruit" to the graph.
9. Added edges between related entities using graph.add_edge(). It defines another dictionary called related patterns that contains regular expressions for different types of relations in the text. For example, the pattern r"\b([A-Za-z_] + )\s + is\s + a\s + type\s + of\s+([A-Za-z_] + )\b" matches any word that is followed by "is a type of" another word, such as "Fuji is a type of apple". The related patterns dictionary maps each relation type to its corresponding pattern. It then loops over each relation type and pattern in the related patterns dictionary and uses the function to find all matches in the text. For each match, it adds an edge from the first matched word to the second matched word with an attribute called relation that stores its relation type. For example, it will add an edge from "fuji" to "apple" with relation "is a type of" to the graph.
10. Created a list of unique entities and the list will be used to map each entity to an index that represents its position in the list. This index will be used to create node features and edge indices for PyTorch Geometric.
11. Created a mapping from entities to indices and it is a dictionary comprehension that assigns each entity in the entity list to its corresponding index using the enumerate function. For example, it will assign "apple" to 0, "fuji" to 1, and so on.
12. Constructed an adjacency matrix and it is a matrix that represents a graph by storing 1 s for connected nodes and 0 s for unconnected nodes. The shape of the matrix is (num_entities, num_entities), where num_entities is the length of the entity list. It then loops over each entity and entity type in the entities dictionary and assigns 1 s to the diagonal elements of the matrix. This creates bidirectional edges between nodes in the graph.
13. Convert the adjacency matrix to edge indices. Where the edge indices are two lists of integers that represent source nodes and target nodes of edges in a graph. They are used by PyTorch Geometric to create data objects from graphs.
14. Created a Graph Data object and it is a container class that stores all information related to a graph in PyTorch Geometric format. It takes edge index as an argument that specifies how nodes are connected in the graph.
15. Defined a GNN model using nn.Module. A GNN model is a neural network that operates on graphs by propagating information along edges and updating node features based on their neighbors. It inherits from nn.Module which is a base class for all neural network modules in PyTorch. It defines two linear layers using nn.Linear(input_dim, hidden_dim) and nn.Linear(hidden_dim, output_dim) that transform node features from input dimension to hidden dimension and from hidden dimension to output dimension respectively. It also defines a forward function that takes x (node features) and edge index (edge indices) as inputs and returns x (updated node features) as output. It applies the first linear layer followed by the activation function on x and then applies the second linear layer on x without any activation function.
16. The dimensions for input, hidden, and output features are given using input_dim = num_entities, hidden_dim = 16, output_dim = input_dim. The input dimension is equal to num_entities because one-hot encoding is used for node features. The hidden dimension is an arbitrary choice that can be tuned based on performance. The output dimension is equal to input dimension and it is for reconstructing node features as output.
17. Created the GNN model using model = GNN(input_dim, hidden_dim, output_dim). It simply initializes an instance of the GNN class defined above with the specified dimensions.
18. Defined the loss function and optimizer using criterion = nn.MSELoss() and optimizer = optim.Adam(model.parameters(), lr = 0.01). The loss function is MSE which measures the difference between predicted node features and actual node features. The optimizer is Adam which updates the model parameters based on the gradient descent algorithm with adaptive learning rates. The GNN model takes the model parameters and learning rate as arguments. At the end of different trails, Adam optimizer and learning rate of 0.01 achieves lower MSE.
19. Generated the input features for entities (one-hot encoding) using input_features = torch.eye(num_entities). One-hot encoding is a way of representing categorical variables as binary vectors where only one element is 1 and rest are 0 s. For example, the one-hot encoding of "apple" with index 0 would be [1, 0, 0, …]. The use of torch.eye(num_entities) function is to create a diagonal matrix where each row corresponds to one-hot encoding of an entity based on its index.
20. Defined a training loop using num_epochs = 100 and the training loop iterates over a fixed number of epochs (100 epochs) where each epoch consists of a forward pass, a backward pass, and an optimization step. The forward pass computes the output node features by passing the input node features and edge indices to the model.forward() function. The backward pass computes the loss value by passing the output node features and input node features to the criterion() function. The optimization step updates the model parameters by calling optimizer.zero_grad() to clear any previous gradients, loss.backward() to compute new gradients based on loss value, and optimizer.step() to apply gradient descent algorithm. The training loop also prints the epoch number and loss value every 10 epochs.
21. Got the learned node embeddings from the output features of the model. Node embeddings are low-dimensional vector representations of nodes that capture their structural and semantic properties. To get them from the output features, detach it from the computation graph using torch.Tensor.detach method.
22. Printed each entity and its corresponding embedding vector. It iterates over a dictionary that maps each entity to its index in the graph. Then it accesses its embedding vector from node embeddings using that index.
23. Finally, printed the nodes and edges of the graph. Then using that visual representation of the created graph is viewed using Networkx library of python.

## 4. Results and discussion

The proposed ADOC framework is implemented and the results are obtained. The experimental setup, dataset description and experimental outcomes are discussed in this section.

### 4.1. Experimental setup

The proposed ADOC research work is implemented in Python using

NLTK, pandas, matplotlib, scipy, scikit-learn, spacy, PyTorch, Networkx, transformers and re packages. NLTK [68,69] is a language modelling toolkit that can be utilized for various language processing techniques. The Networkx [70] and matplotlib [71] packages are applied for creating graph and data visualization. The scipy [72] and scikit-learn [73] packages are employed for computations and implementation of mathematical functions such as cosine similarity formulation. The spacy [74] package consists of prodigy annotator which is used for semantic and syntactic annotations. The package transformers [75] is utilized for load and processing with BERT model. PyTorch library [76] is for designing and implementing GNN model. The pandas [77] library is for data manipulation and computation with visualization. The re [78] package is applied for creating and handling regular expressions.

### 4.2. Dataset description

The agricultural documents are extracted from various government websites and blogs. The data collected from different sources are not uniform, so the agriculture domain experts are employed for selecting and making the contents of the documents. The documents are mainly collected from Tamil Nadu Agricultural University Agritech portal (Agritech.tnau.ac.in) [79], Indian Council of Agricultural Research (dogr.icar.gov.in) [80,81], National Horticulture Research and Development Foundation (nhrdf.org), Food and Agriculture Organization of the United States (FAO.org) [25,26], Farmer portal (farmer.gov.in) [25,26], Department of Agriculture & Farmers Welfare (agricoop.nic.in) [25,26] and agricultural blogs. The dataset comprises of 541 pages and the language of the document is English.

### 4.3. Experimental outcomes

The proposed ADOC work constructed the agriculture domain based ontology in three different ways. Among the ADOC methods and other contemporary systems, the BERT based GNN with regular expressions shows better results. The step-by-step results obtained for the ADOC methods are discussed in this section and finally all the models are compared with each other and with the existing frameworks. The sample data of the research work is shown in Fig. 7.

The first step of processing the data is to perform anaphora resolution for the text document. Hence by the end of the anaphora resolution technique, all the anaphors in the text are replaced by its antecedents. The result of nominal anaphora resolution for the sample data is shown in Fig. 8.

The anaphora resolved text data is now used for extracting the agriculture terms using the methods discussed in ADOC framework. Among the two methods, the BERT based regular expression method performs more efficiently in extracting the terms than the NLP based techniques because all the domain based terms are not available in the vocabulary for agriculture and few irrelevant words are also available in the vocabulary. Hence this issue affects the term extraction using the proposed NLP based method. In both the methods, the terms are extracted in two steps, in the NLP based method all the possible agriculture terms are extracted and then using the domain based vocabulary, only the candidate terms are retrieved. Whereas, the BERT model extracts all the possible domain terms but in this research work, BERT is used for embedding generation and then the embeddings are processed with regular expressions are used to retrieve only the candidate terms. The sample output for term extraction for the above sample data is shown in Figs. 9 and 10.

The next step is to extract the relationships between the entities. The relationships are extracted using three methodologies, namely, POS based Hearst patterns, Hearst pattern with regular expressions and GNN with regular expressions. Among these three methods, GNN based regular expression method outperforms well compared to other methods. The Table 5 shows the relationships extracted for the sample data.

After extracting the relationships between the entities, the agricultural domain based ontology graph is constructed. The entities are the nodes of the graph and the relationships are the edges of the graph. Fig. 11 shows the agricultural domain based ontology graph for the sample data. The partial view of the constructed agriculture ontology is shown in Fig. 12.

### 4.4. Performance analysis

The agriculture domain does not have the standard dataset for evaluating the proposed ADOC models. In order to evaluate the designed research work, agriculture domain expects pertaining to Tamil Nadu Agriculture Department are engaged to analyze and evaluate the systems. The domain experts initially annotate terms in the documents with their knowledge and then the evaluation involves calculating the true positive (TP), true negative (TN), false positive (FP) and false negative (FN) for all the modules. The evaluation work has been divided into three as evaluation for anaphora resolution, evaluation for term extraction and evaluation for relationship extraction with ontology

**Apple is a fruit. It is cultivated in spring. Fuji, Gala, Granny Smith, pink lady, Envy, Honeycrisp, Pazazz, Jazz, Red delicious, Goldern delicious are the types of apple. Apple scab is a disease in apple. It is a fungal disease. Potash magnesium, Sulphate of potash, Boron are the fertilizers for apple. Strawberries, Raspberries, Blackberries are an intercrop of apple. Cotton is an intercrop of Beetroot. It is cultivated in March. Anthracnose is a disease in cotton. Paddy is also called as rice paddy. Blast is a disease in it. Calcium silicate is a fertilizer for blast.**

**Fig. 7.** Sample data of the research work.

**Apple is a fruit. Apple is cultivated in spring. Fuji, Gala, Granny Smith, pink lady, Envy, Honeycrisp, Pazazz, Jazz, Red delicious, Goldern delicious are the types of apple. Apple scab is a disease in apple. Apple scab is a fungal disease. Potash magnesium, Sulphate of potash Boron are the fertilizer for apple. Strawberries, Raspberries, Blackberries are an intercrop of apple. Cotton is an intercrop of Beetroot. Cotton is cultivated in March. Anthracnose is a disease in cotton. Paddy is also called as rice paddy. Blast is a disease in paddy. Calcium silicate is a fertilizer for blast.**

**Fig. 8.** The anaphora resolved text of the sample data.

**['Apple', 'fruit', 'Apple', 'cultivated', 'spring', 'Fuji', 'Gala', 'Granny Smith', 'pink lady', 'Envy', 'Honeycrisp', 'Pazazz', 'Jazz', 'Red delicious', 'Goldern delicious', 'types', 'apple', 'Apple scab', 'disease', 'apple', 'Apple scab', 'fungal disease', 'Potash magnesium', 'Sulphate of potash', 'Boron', 'fertilizers', 'apple', 'Strawberries', 'Raspberries', 'Blackberries', 'intercrop', 'apple', 'Cotton', 'intercrop', 'Beetroot', 'Cotton', 'cultivated', 'March', 'Anthracnose', 'disease', 'cotton', 'Paddy', 'rice paddy', 'Blast', 'disease', 'paddy', 'Calcium silicate', 'fertilizer', 'blast']**

**Fig. 9.** The result of all possible terms extracted from text of the sample data.

['Apple', 'fruit', 'Apple', 'spring', 'Fuji', 'Gala', 'Granny Smith', 'pink lady', 'Envy', 'Honeycrisp', 'Pazazz', 'Jazz', 'Red delicious', 'Goldern delicious', 'apple', 'Apple scab', 'apple', 'Apple scab', 'fungal disease', 'Potash magnesium', 'Sulphate of potash', 'Boron', 'apple', 'Strawberries', 'Raspberries', 'Blackberries', 'apple', 'Cotton', 'Beetroot', 'Cotton', 'March', 'Anthracnose', 'cotton', 'Paddy', 'rice paddy', 'Blast', 'paddy', 'Calcium silicate', 'blast']

**Fig. 10.** The result of the candidate agriculture domain terms extracted.

**Table 5**
Extracted relationships between the entities for the sample data.

| Entity 1 | Relationship | Entity 2 |
|---|---|---|
| Apple | Is a | Fruit |
| Apple | Cultivated in | Spring |
| Fuji | Type of | Apple |
| Gala | Type of | Apple |
| Granny Smith | Type of | Apple |
| Pink lady | Type of | Apple |
| Envy | Type of | Apple |
| Honeycrisp | Type of | Apple |
| Pazazz | Type of | Apple |
| Jazz | Type of | Apple |
| Red delicious | Type of | Apple |
| Goldern delicious | Type of | Apple |
| Apple scab | Disease in | Apple |
| Apple scab | Is a | Fungal disease |
| Potash magnesium | Fertilizer for | Apple |
| Sulphate of potash | Fertilizer for | Apple |
| Boron | Fertilizer for | Apple |
| Strawberries | Intercrop of | Apple |
| Raspberries | Intercrop of | Apple |
| Blackberries | Intercrop of | Apple |
| Cotton | Intercrop of | Beetroot |
| Cotton | Cultivated in | March |
| Anthracnose | Disease in | Cotton |
| Paddy | Is also | Rice paddy |
| Blast | Disease in | Paddy |
| Calcium silicate | Fertilizer for | Blast |

construction.

#### 4.4.1. Evaluation for anaphora resolution

For evaluating the anaphora resolution, the True Positive of Anaphora Resolution (TPAR), True Negative of Anaphora Resolution (TNAR), False Positive of Anaphora Resolution (FPAR), False Negative of Anaphora Resolution (FNAR) have been calculated with respect to the anaphora resolved text document and input document. TPAR represents the total count of having both the actual and predicted words as an anaphor and it is resolved. TNAR represents the total count of not having both the actual and predicted words as an anaphor. FPAR denotes the count of the words predicted as an anaphor but it is not an anaphor. FNAR denotes the count of the words which are actually anaphor but it is not predicted as anaphor or it has been mapped to the incorrect antecedent. From the above obtained values, 10 evaluation measures have been calculated. Sensitivity is also known as True Positive Rate or Hit Rate or Recall. It is estimated by dividing the number of TPAR by the summation of number of TPAR and FNAR. Mathematically, sensitivity is represented as TPAR/(TPAR + FNAR). Specificity is also known as True Negative Rate and it is calculated by dividing the number of TNAR by the summation of number of FPAR and TNAR. Mathematically, specificity is represented as TNAR/(FPAR + TNAR). Precision is also known as Positive Predictive value and it is defined as the number of TPAR divided by the summation of number of TPAR and FPAR. Mathematically, precision is denoted as TPAR/(TPAR + FPAR). Negative Predictive Value is the TNAR divided by the summation of number of TNAR and FNAR. Mathematically, it is denoted as TNAR/(TNAR + FNAR). False Positive Rate is the number of FPAR divided by the summation of number of FPAR and TNAR. Mathematically, it is denoted as FPAR/(FPAR + TNAR). False Discovery Rate is the number of FPAR divided by the summation of number of FPAR and TPAR. Mathematically, it is denoted as FPAR/(FPAR + TPAR). False Negative Rate is the number of FNAR divided by the summation of number of FNAR and TPAR. Mathematically, it is denoted as FNAR/(FNAR + TPAR). Accuracy is defined as the measure that finds how close the actual mapping of antecedents with the predicted mapping of antecedents for anaphors. Mathematically, it is represented as (TPAR + TNAR)/(TPAR + TNAR + FNAP + FPAR). F1-Score is calculated using precision and recall for evaluating the anaphora resolving system. Mathematically, it is represented as (2*TPAR)/(2*TPAR + FPAR + FNAR). Matthews Correlation Coefficient measures the quality of anaphora resolution. It is mathematically expressed as, (TPAR*TNAR – FPAR*FNAR)/[sqrt[(TPAR + FPAR)*(TPAR + FNAR)*(TNAR + FPAR)*(TNAR + FNAR)]]. For the semantic anaphora resolution for agricultural documents system, the above mentioned evaluation metrics have been calculated and the results are shown in the Table 6.

#### 4.4.2. Evaluation for term extraction

For evaluating the extracted agriculture terms, the True Positive of Agriculture Term Extraction (TPATE), True Negative of Agriculture Term Extraction (TNATE), False Positive of Agriculture Term Extraction (FPATE), False Negative of Agriculture Term Extraction (FNATE) have been calculated with respect to the anaphora resolved text document. TPATE represents the total count of having both the actual and predicted terms are agriculture domain based terms. TNATE represents the total count of not having both the actual and predicted terms of agriculture domain. FPATE denotes the count of the terms predicted as the domain terms but those terms are not the domain terms. FNATE denotes the count of the terms which are actually domain terms but it is not predicted as the domain terms. Sensitivity is estimated by dividing the number of TPATE by the summation of number of TPATE and FNATE. Specificity is calculated by dividing the number of TNATE by the summation of number of FPATE and TNATE. Precision is defined as the number of TPATE divided by the summation of number of TPATE and FPATE. Negative Predictive Value is the TNATE divided by the summation of number of TNATE and FNATE. False Positive Rate is the number of FPATE divided by the summation of number of FPATE and TNATE. False Discovery Rate is the number of FPATE divided by the summation of number of FPATE and TPATE. False Negative Rate is the number of FNATE divided by the summation of number of FNATE and TPATE. Accuracy is defined as the measure that finds how close the actual and predicted terms are similar. F1-Score is calculated using precision and recall for evaluating the terms extracted. Matthews Correlation Coefficient measures the quality of the domain term extraction technique. The two methods used in this research work have been compared using these evaluation metrics and the results are shown in Table 7. The NLP based model has a greater number of FNATE and FPATE than BERT based regular expression model. Hence, the BERT model with regular expressions achieves best values compared to the NLP technique based term extraction method. Also, both the systems are compared with the existing systems and it is shown in Fig. 13.

The evaluation metrics in Table 7 provide a detailed comparison between two methods employed for domain term extraction in agriculture. Method 2 which leverages BERT embeddings in conjunction with regular expressions, exhibits notable advantages over Method 1 employing general NLP techniques. Method 2 demonstrates superior sensitivity (0.9633), indicating its effectiveness in capturing a substantial portion of true positive instances crucial for comprehensive term extraction in agriculture. Additionally, Method 2 excels in precision (0.9906) and accuracy (0.9782) emphasizing its ability to provide accurate positive predictions and overall correctness in term extraction

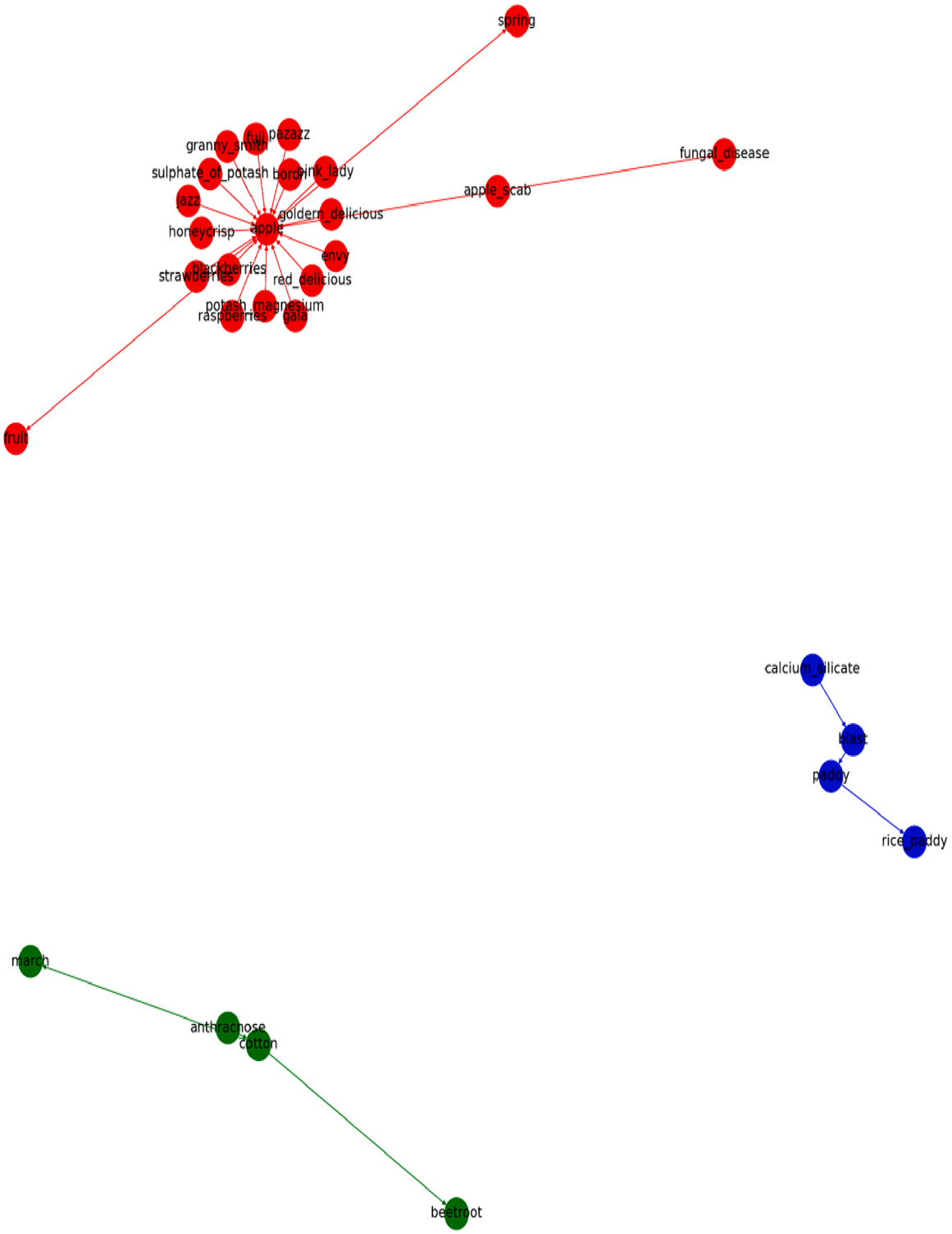


**Fig. 11.** Agricultural domain based ontology constructed for the sample data.

across both positive and negative instances. The higher specificity (0.9917), negative predictive value (0.9675) and lower false positive rate (0.0083) further underscore Method 2′s capability to filter out false positives and enhance the reliability of identified domain terms. The low false discovery rate (0.0094) and false negative rate (0.0367) reinforce Method 2′s precision and effectiveness in minimizing omissions of true positive domain terms. The robust F1 score (0.9767) and Matthews correlation coefficient (0.9565) in Method 2 affirm its balanced performance between precision and recall. In conclusion, Method 2, integrating BERT embeddings with regular expressions emerges as a superior choice for domain term extraction in agriculture, offering high accuracy, precision and sensitivity in identifying relevant terms within the agricultural domain.

The comparison of evaluation metrics presented in the Fig. 13 offer insights into the performance of various methods for agricultural term extraction. KEA++ demonstrates moderate precision, recall and F1 score, suggesting room for improvement, particularly in recall. The Regular Expressions method relying on POS and domain-specific

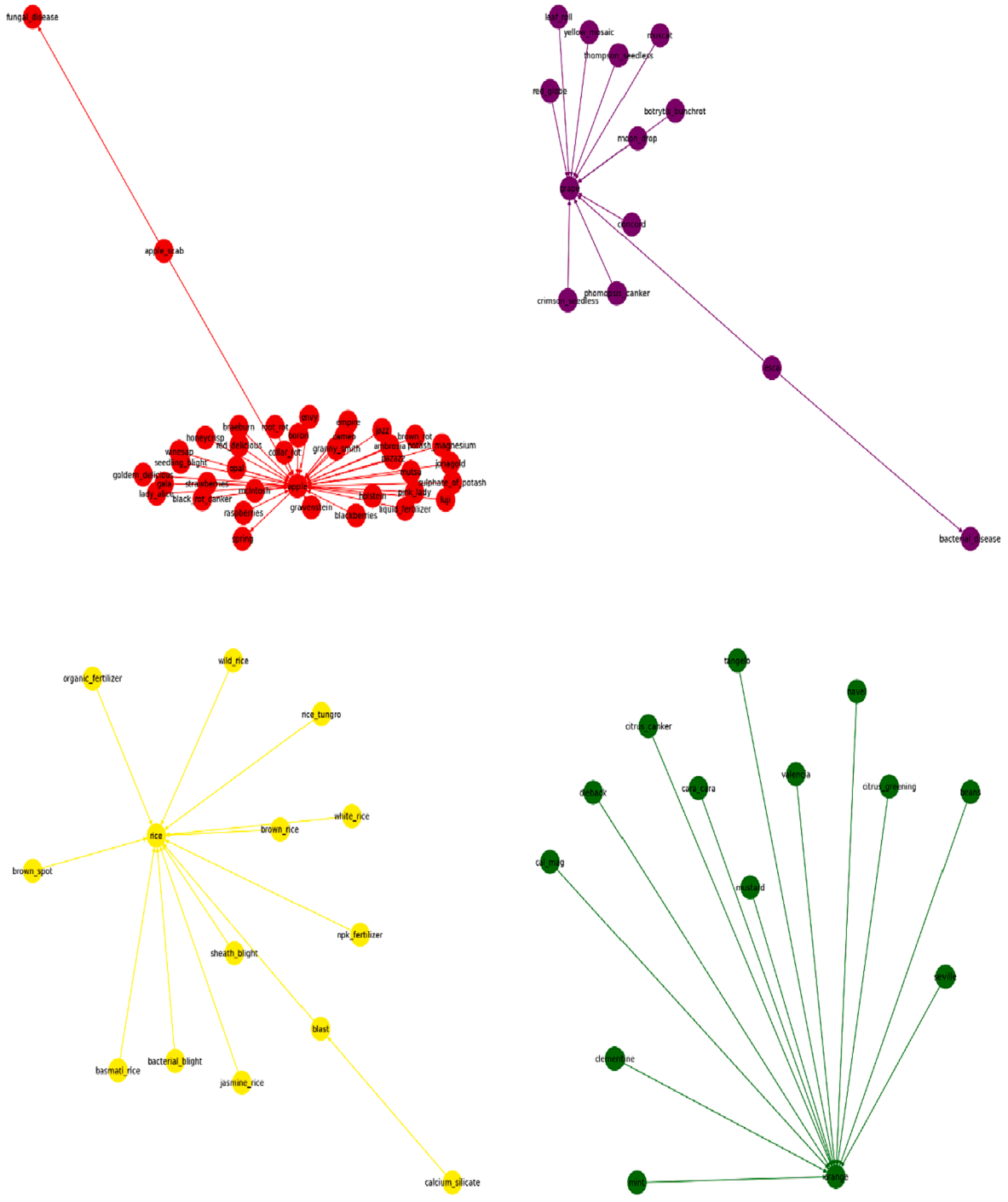


**Fig. 12.** The partial view of the constructed agriculture ontology.

patterns shows a high precision of 85.47 %, but the absence of recall and F1 score details limits a comprehensive assessment. RENT strikes a balance between precision and recall, presenting a robust approach. However, there is a need for improvement particularly in terms of recall. Customized NER using the Spacy model exhibits moderate performance indicating potential for refinement. The Proposed NLP method displays high precision, recall and F1 score demonstrating effectiveness in agricultural term extraction. However, the standout performer is the Proposed BERT + Regular Expressions method achieving exceptional precision, recall and F1 score showcasing the power of combining BERT embeddings with regular expressions for accurate term extraction in the agricultural domain. Overall, the Proposed BERT + Regular Expressions method emerges as the most robust approach for agricultural domain term extraction process.

### 4.4.3. *Evaluation for relationship extraction and ontology*

For evaluating the relationships extracted and constructing ontologies, the True Positive of Relationship Extraction and Ontology (TPREO), True Negative of Relationship Extraction and Ontology (TNREO), False Positive of Relationship Extraction and Ontology (FPREO), False Negative of Relationship Extraction and Ontology (FNREO) have been calculated with respect to the anaphora resolved text document and extracted domain terms. TPREO represents the total count of having both the actual and predicted relationships as the same

**Table 6**
Evaluation for anaphora resolution.

| Evaluation metric | Values obtained |
|---|---|
| Sensitivity or Recall | 0.9091 |
| Specificity | 0.9979 |
| Precision | 0.9375 |
| Negative Predictive Value | 0.9969 |
| False Positive Rate | 0.0021 |
| False Discovery Rate | 0.0625 |
| False Negative rate | 0.0909 |
| Accuracy | 0.9950 |
| F1 Score | 0.9231 |
| Matthews Correlation Coefficient | 0.9206 |

**Table 7**
Evaluation for Agriculture term extraction.

| Evaluation metric | Method 1 (NLP techniques) | Method 2 (BERT + Regular Expressions) |
|---|---|---|
| Sensitivity or Recall | 0.8706 | 0.9633 |
| Specificity | 0.9582 | 0.9917 |
| Precision | 0.9528 | 0.9906 |
| Negative Predictive Value | 0.8842 | 0.9675 |
| False Positive Rate | 0.0418 | 0.0083 |
| False Discovery Rate | 0.0472 | 0.0094 |
| False Negative rate | 0.1294 | 0.0367 |
| Accuracy | 0.9150 | 0.9782 |
| F1 Score | 0.9098 | 0.9767 |
| Matthews Correlation Coefficient | 0.8328 | 0.9565 |

and correctly identified in the document between the terms, considering both relationship extraction from term extraction and ontology construction. TNREO represents the total count of not having domain relationships between the terms both in actual and prediction, accounting for both relationship extraction and ontology construction. FPREO denotes the count of relationships predicted as wrong or where no actual relation is available between entities in the context of both relationship extraction and ontology construction. FNREO denotes the count of relationships actually present between terms but not predicted, encompassing both relationship extraction and ontology construction. Sensitivity is estimated by dividing the number of TPREO by the summation of number of TPREO and FNREO. Specificity is calculated by dividing the number of TNREO by the summation of number of FPREO and TNREO. Precision is defined as the number of TPREO divided by the summation of number of TPREO and FPREO. Negative Predictive Value is the TNREO divided by the summation of number of TNREO and FNREO. False Positive RREO is the number of FPREO divided by the summation of number of FPREO and TNREO. False Discovery RREO is the number of FPREO divided by the summation of number of FPREO and TPREO. False Negative RREO is the number of FNREO divided by the summation of number of FNREO and TPREO. Accuracy is defined as the measure that finds how close the actual and predicted relationships are similar in relationship extraction and ontology construction. F1-Score is calculated using precision and recall for evaluating the relationships extracted with domain terms and ontology construction. Matthews Correlation Coefficient measures the quality of the domain relationships extraction methodology with ontology construction. By evaluating the proposed relationship extraction models, GNN is found to have better outcomes and it is shown in Table 6. Next, the modelled systems are compared with the existing systems and shown in Fig. 14.

The evaluation metrics in Table 8 provide insights into the comparative strengths of three distinct methods for constructing an ontology focused on domain relationships in agriculture. Method 3, employing unsupervised GNN in conjunction with Regular Expressions emerges as particularly noteworthy in several aspects. With the highest sensitivity (0.9831), it excels in capturing a substantial portion of true positive instances, a crucial aspect for a comprehensive agricultural ontology. Moreover, Method 3 leads in specificity (0.9231), emphasizing its effectiveness in filtering out false positives and thereby enhancing the reliability of identified domain relationships. Its high precision (0.9667) underscores accuracy in positive predictions, indicating a low rate of false positives. Method 3 further achieves the lowest false positive rate (0.0769), showcasing its ability to minimize occurrences of false positives. Additionally, it demonstrates the lowest false negative rate (0.0169), showcasing its effectiveness in minimizing omissions of true positive domain relationships. Overall, Method 3 maintains high accuracy (0.9647), the highest F1 score (0.9748), and a notable Matthews correlation coefficient (0.9163), collectively signifying its robust performance in ontology construction with a focus on domain relationships in agriculture.

The evaluation of various relationship extraction and ontology construction methods reveals distinct strengths among the approaches. Position vector, patterns and WordNet achieves a high precision of 87.5 % but recall and F1 score are not mentioned. Precision indicates the accuracy of positive predictions and the reported value suggests a relatively low rate of false positives. RelExOnt demonstrates a precision of 86.89 % but recall and F1 score are not specified. Similar to the first method, a high precision implies accurate positive predictions with a

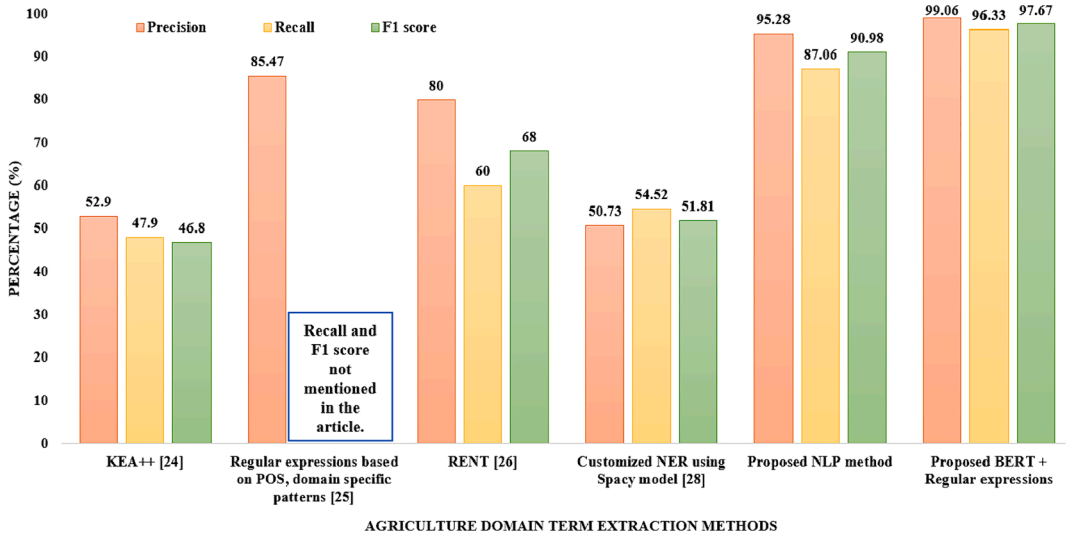


**Fig. 13.** Comparison of proposed ADOC and existing models of agriculture term extraction using Precision, Recall and F1 score evaluation metrics.

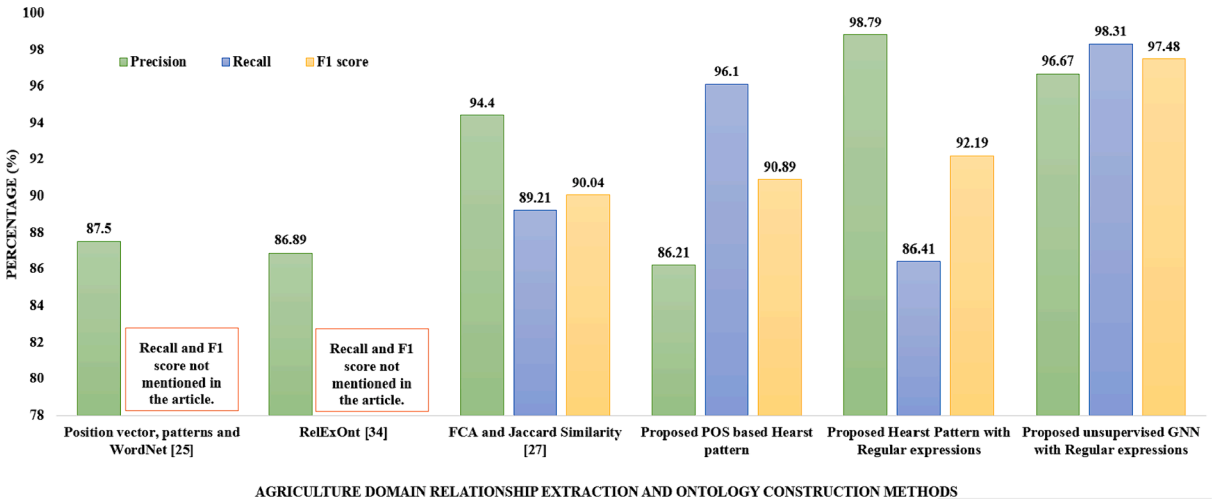


**Fig. 14.** Comparison of proposed ADOC and existing models of agriculture domain based relationship extraction and ontology construction using Precision, Recall and F1 scores.

**Table 8**
Evaluation for relationship extraction and ontology construction

| Evaluation metric | Method 1 (POS based Hearst pattern) | Method 2 (Hearst pattern + Regular Expressions) | Method 3 (unsupervised GNN + Regular Expressions) |
|---|---|---|---|
| Sensitivity or Recall | 0.9610 | 0.8641 | 0.9831 |
| Specificity | 0.3942 | 0.8864 | 0.9231 |
| Precision | 0.8621 | 0.9879 | 0.9667 |
| Negative Predictive Value | 0.7193 | 0.3786 | 0.9600 |
| False Positive Rate | 0.6058 | 0.1136 | 0.0769 |
| False Discovery Rate | 0.1379 | 0.0121 | 0.0333 |
| False Negative rate | 0.0390 | 0.1359 | 0.0169 |
| Accuracy | 0.8463 | 0.8660 | 0.9647 |
| F1 Score | 0.9089 | 0.9219 | 0.9748 |
| Matthews Correlation Coefficient | 0.4545 | 0.5245 | 0.9163 |

reduced likelihood of false positives. FCA and Jaccard Similarity excels with a precision of 94.4 %, recall of 89.21 % and an impressive F1 score of 90.04. High precision, recall, and F1 score collectively indicate a method capable of accurate positive predictions and a comprehensive representation of relationships in the ontology. Proposed POS based Hearst pattern achieves a precision of 86.21 %, recall of 96.1 % and an F1 score of 90.89. Demonstrates a balanced performance with high recall, indicating the method's ability to capture a significant proportion of true positive relationships. Proposed Hearst Pattern with Regular expressions exhibits a precision of 98.79 %, recall of 86.41 % and an F1 score of 92.19. The high precision suggests accurate positive predictions, while the balanced F1 score reflects a trade-off between precision and recall. Proposed unsupervised GNN with Regular expressions impresses with a precision of 96.67 %, recall of 98.31 % and an outstanding F1 score of 97.48. Achieves a remarkable balance between precision and recall, indicating accurate predictions and comprehensive coverage of relationships.

The ontology constructed is further evaluated using the below performance measures using domain experts,

1. Clarity: Assess the model's output for clear and easily understandable language. Clarity is satisfied in the proposed framework and the cumulative score given by the experts is found to be about 99.3 %.
2. Coherence: Evaluate how well the generated content flows logically and cohesively. Coherence is satisfied in the proposed framework and the cumulative score given by the experts is found to be about 99.9 %.
3. Minimal Encoding Bias: Examine the generated content for bias, ensuring that it avoids encoding or promoting stereotypes, discrimination, or unfairness. Minimal Encoding Bias is satisfied in the proposed framework and the cumulative score given by the experts is found to be about 99.7 %.
4. Conciseness: Measure the efficiency of expression and avoid unnecessary details in the generated text. Conciseness is satisfied in the proposed framework and the cumulative score given by the experts is found to be about 99.4 %.
5. Completeness: Assess whether the model provides comprehensive information in its outputs. Depending on the task, completeness can be measured by comparing the generated content against reference data. Completeness is satisfied in the proposed framework and the cumulative score given by the experts is found to be about 99.5 %.

## 5. Conclusion

The ontology is created for agriculture domain using the proposed ADOC framework in three different ways. The domain terms are extracted efficiently by utilizing the term extraction methods on the nominal anaphora resolved text documents. The research work also extracted the required relationships between the entities by applying three distinct techniques. Among the AODC methods, the BERT with regular expressions and the GNN with regular expressions produces better results with accuracy of 96.47 %. Also, all the three works are compared with the existing works. The GNN model with regular expressions has very minimal number of false positives and false negatives, so the performance is improved much compared to the other models. From the extracted domain terms and the relationships between the terms, the ontology is developed. The ontology constructed has the domain entities in the nodes and the directed edges between the entities hold the corresponding relationships. Employing anaphora resolution in this research work guarantees a thorough understanding of context, minimizing the risk of overlooking domain terms. The active involvement of domain experts in both document preparation and evaluation ensures the practical applicability and accuracy of the system. Despite its merits, the research exhibits limitations such as potential biases

introduced by reliance on domain experts and predefined relationships due to the absence of benchmark dataset. Future research work can be concentrated on refining ontology construction by automatically integrating additional types of domain relationships to achieve a more exhaustive representation. Then investigating how the ADOC framework adapts to various agricultural domains or languages thereby enhancing its applicability and analyzing user feedback for usability aspects to refine the system based on practical requirements. The developed ontology has a wide range of applications offering advantages to farmers by enhancing their knowledge, providing researchers with accurate data and assisting policymakers in optimizing resource allocation. It plays a supportive role in agricultural education, streamlines extension services and enables cross-domain integration, fostering a comprehensive understanding of market trends, weather patterns and environmental impact. The accessibility provided by user-friendly interfaces renders it a valuable tool for the advancement of agriculture in information retrieval.

## CRediT authorship contribution statement

**Krithikha Sanju Saravanan:** Conceptualization, Methodology, Resources, Data curation, Writing - original draft preparation, Writing—review and editing, Investigation, Validation. **Velammal Bhagavathiappan:** Methodology, Resources, Supervision and Validation.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

*Ethical Approval.*

Since this research work deals with text data across the web, ethical approval is not applicable.

*Funding*

No funding has been claimed for this research work.

## References

[1] H. Ahmadzai, S. Tutundjian, I. Elouafi, Policies for sustainable agriculture and livelihood in marginal lands: a review, Sustainability. 13 (16) (2021) 86–92, https://doi.org/10.3390/su13168692.
[2] M.M. Anwar, S. Farooqi, G.Y. Khan, Agriculture sector performance: an analysis through the role of agriculture sector share in GDP, J. Agric. Econ. Extens. Rural Dev. 3 (3) (2015) 270–275.
[3] Q.M. Zhang, A. Zeng, M.S. Shang, Extracting the information backbone in online system, PLoS One 8 (5) (2013) e62624.
[4] W.R. Padilla, J. García, J.M. Molina, Knowledge extraction and improved data fusion for sales prediction in local agricultural markets, Sensors 19 (2) (2019) 286, https://doi.org/10.3390/s19020286.
[5] A.S. Patel, G. Merlino, A. Puliafito, R. Vyas, O.P. Vyas, M. Ojha, V. Tiwari, An NLP-guided ontology development and refinement approach to represent and query visual information, Expert Syst. Appl. 213 (2023) 118998, https://doi.org/10.1016/j.eswa.2022.118998.
[6] R. Rawat, Logical concept mapping and social media analytics relating to cyber criminal activities for ontology creation, Int. J. Inf. Technol. 15 (2) (2023) 893–903, https://doi.org/10.1007/s41870-022-00934-9.
[7] Martínez-Cruz, R., Mahata, D., López-López, A. J., & Portela, J.: Enhancing Keyphrase Extraction from Long Scientific Documents using Graph Embeddings. arXiv preprint arXiv:2305.09316 (2023). https://doi.org/10.48550/arXiv.2305.09316.
[8] Martínez-Cruz, R., López-López, A. J., & Portela, J.: ChatGPT vs State-of-the-Art Models: A Benchmarking Study in Keyphrase Generation Task. arXiv preprint arXiv:2304.14177 (2023). https://doi.org/10.48550/arXiv.2304.14177.
[9] S. Mishra, S.K. Sharma, Advanced contribution of IoT in agricultural production for the development of smart livestock environments, Internet of Things. 22 (2023) 100724, https://doi.org/10.1016/j.iot.2023.100724.
[10] S. Jain, S. Basu, Y.K. Dwivedi, S. Kaur, Interactive voice assistants–does brand credibility assuage privacy risks? J. Bus. Res. 139 (2021) 701–717, https://doi.org/10.1016/j.jbusres.2021.10.007.
[11] A.R.D.B. Landim, A.M. Pereira, T. Vieira, de B.E. Costa, J.A.B. Moura, V. Wanick, E. Bazaki, Chatbot design approaches for fashion E-commerce: an interdisciplinary review, Internat. J. Fashion Design Technol. Educat. 15 (2) (2022) 200–210, https://doi.org/10.1080/17543266.2021.1990417.
[12] V. Kumari, C. Gosavi, Y. Sharma, L. Goel, Domain-specific chatbot development using the deep learning-based RASA framework, in: Communication and Intelligent Systems: Proceedings of ICCIS 2021, Springer Nature, Singapore, 2022, pp. 883–896.
[13] I. Tyagin, A. Kulshrestha, J. Sybrandt, K. Matta, M. Shtutman, I. Safro, Accelerating COVID-19 research with graph mining and transformer-based learning, Vol. 36, No. 11, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2022, pp. 12673–12679.
[14] Marasović, A., Born, L., Opitz, J., & Frank, A.: A mention-ranking model for abstract anaphora resolution. arXiv preprint arXiv:1706.02256 (2017). https://doi.org/10.48550/arXiv.1706.02256.
[15] Hou, Y.: A deterministic algorithm for bridging anaphora resolution. arXiv preprint arXiv:1811.05721 (2018). https://doi.org/10.48550/arXiv.1811.05721.
[16] M. Phadke, S. Devane, Pronoun resolution task for multilingual machine translation, in: In: 5th International Conference on next Generation Computing Technologies, 2020, p. NGCT-2019.
[17] K. Khandale, C.N. Mahender, in: Rule-Based Design for Anaphora Resolution of Marathi Sentence, IEEE, 2019, pp. 1–7.
[18] C. Lee, S. Jung, C.E. Park, Anaphora resolution with pointer networks, Pattern Recogn. Lett. 95 (2017) 1–7, https://doi.org/10.1016/j.patrec.2017.05.015.
[19] P. Khurana, P. Agarwal, G. Shroff, L. Vig, Resolving abstract anaphora implicitly in conversational assistants using a hierarchically stacked rnn, in: In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 433–442.
[20] Hou, Y. : Enhanced word representations for bridging anaphora resolution. arXiv preprint arXiv:1803.04790 (2018). https://doi.org/10.48550/arXiv.1803.04790.
[21] Hardmeier, C.: Pronoun prediction with latent anaphora resolution. In: Proceedings of the First Conference on Machine Translation: Shared Task Papers, vol. 2, pp. 576-580 (2016).
[22] C. Hardmeier, Predicting pronouns with a convolutional network and an n-gram model, in: Proceedings of the Third Workshop on Discourse in Machine Translation, 2017, pp. 58–62.
[23] Medelyan, O., & Witten, I. H.: Thesaurus-based index term extraction for agricultural documents. 1122-1129 (2005).
[24] N. Kaushik, N. Chatterjee, A practical approach for term and relationship extraction for automatic ontology creation from agricultural text, in: 2016 International Conference on Information Technology (ICIT), IEEE, 2016, pp. 241–247.
[25] N. Chatterjee, N. Kaushik, RENT: Regular expression and NLP-based term extraction scheme for agricultural domain, in: Proceedings of the International Conference on Data Engineering and Communication Technology: ICDECT 2016, Springer, Singapore, 2017, pp. 511–522.
[26] R. Deepa, S. Vigneshwari, An effective automated ontology construction based on the agriculture domain, ETRI J. 44 (4) (2022) 573–587, https://doi.org/10.4218/etrij.2020-0439.
[27] H. Panoutsopoulos, C. Brewster, B. Espejo-Garcia, Developing a model for the automated identification and Extraction of agricultural terms from unstructured text, Chem. Proc. 10 (1) (2022) 94, https://doi.org/10.3390/IOCAG2022-12264.
[28] K. Frantzi, S. Ananiadou, H. Mima, Automatic recognition of multi-word terms: the c-value/nc-value method, Int. J. Digit. Libr. 3 (2000) 115–130, https://doi.org/10.1007/s007999900023.
[29] E. Milios, Y. Zhang, B. He, L. Dong, Automatic term extraction and document similarity in special text corpora, in: Proceedings of the Sixth Conference of the Pacific Association for Computational Linguistics, 2003, pp. 275–284.
[30] Maynard, D., & Ananiadou, S. Identifying terms by their family and friends. In: The 18th International Conference on Computational Linguistics, vol. 1(2000).
[31] H. Luo, T. Li, B. Liu, B. Wang, H. Unger, Improving aspect term extraction with bidirectional dependency tree representation, IEEE/ACM Trans. Audio Speech Lang. Process. 27 (7) (2019) 1201–1212, https://doi.org/10.1109/TASLP.2019.2913094.
[32] A. Onan, S. Korukoğlu, H. Bulut, Ensemble of keyword extraction methods and classifiers in text classification, Expert Syst. Appl. 57 (2016) 232–247, https://doi.org/10.1016/j.eswa.2016.03.045.
[33] N. Kaushik, N. Chatterjee, Automatic relationship extraction from agricultural text for ontology construction, Information Processing in Agriculture. 5 (1) (2018) 60–73, https://doi.org/10.1016/j.inpa.2017.11.003.
[34] A. Chougule, V.K. Jha, D. Mukhopadhyay, in: Ontology Based System for Pests and Disease Management of Grapes in India, IEEE, 2016, pp. 133–138.
[35] Y.L. Zheng, Q.Y. He, Q.I.A.N. Ping, L.I. Ze, Construction of the ontology-based agricultural knowledge management system, J. Integr. Agric. 11 (5) (2012) 700–709, https://doi.org/10.1016/S2095-3119(12)60059-8.
[36] Sivamani, S., Bae, N. J., Shin, C. S., Park, J. W., & Cho, Y. Y.: An OWL-based ontology model for intelligent service in vertical farm. In: Advances in Computer Science and its Applications: CSA 2013, pp. 327-332. Springer, Berlin Heidelberg (2014).
[37] Y. Wang, Y. Wang, J. Wang, Y. Yuan, Z. Zhang, An ontology-based approach to integration of hilly citrus production knowledge, Comput. Electron. Agric. 113 (2015) 24–43, https://doi.org/10.1016/j.compag.2015.01.009.

[38] R. Hoehndorf, M. Alshahrani, G.V. Gkoutos, G. Gosline, Q. Groom, T. Hamann, C. Weiland, The flora phenotype ontology (FLOPO): tool for integrating morphological traits and phenotypes of vascular plants, J. Biomed. Semant. 7 (1) (2016) 1–11, https://doi.org/10.1186/s13326-016-0107-8.
[39] N.I.Y. Saat, S.A.M. Noah, Rule-based approach for automatic ontology population of agriculture domain, Inf. Technol. J. 15 (2) (2016) 46–51, https://doi.org/10.3923/itj.2016.46.51.
[40] A. Goldstein, L. Fink, O. Raphaeli, A. Hetzroni, G. Ravid, Addressing the 'Tower Of Babel'of pesticide regulations: an ontology for supporting pest-control decisions, J. Agric. Sci. 157 (6) (2019) 493–503, https://doi.org/10.1017/S0021859619000820.
[41] B.P. Bhuyan, R. Tomar, M. Gupta, A. Ramdane-Cherif, An ontological knowledge representation for smart agriculture, in: 2021 IEEE International Conference on Big Data (Big Data), IEEE, 2021, pp. 3400–3406, https://doi.org/10.1109/BigData52589.2021.9672020.
[42] Q.H. Ngo, T. Kechadi, N.A. Le-Khac, OAK: ontology-based knowledge map model for digital agriculture, in: Future Data and Security Engineering: 7th International Conference, FDSE 2020, Quy Nhon, Vietnam, November 25–27, 2020, Proceedings 7, Springer International Publishing, 2020, pp. 245–259, https://doi.org/10.1007/978-3-030-63924-2_14.
[43] A. Goldstein, L. Fink, G. Ravid, A framework for evaluating agricultural ontologies, Sustainability. 13 (11) (2021) 6387, https://doi.org/10.3390/su13116387.
[44] J.V. Fonou-Dombeu, N. Naidoo, M. Ramnanan, R. Gowda, S.R. Lawton, OntoCSA: a climate-Smart agriculture ontology, Internat. J. Agric. Environ. Inform. Syst. (IJAEIS). 12 (4) (2021) 1–20, https://doi.org/10.4018/IJAEIS.292476.
[45] V.M. Kushala, M.C. Supriya, H.R. Divakar, Construction of domain ontology considering organic fertilizers for a sustainable agriculture, in: 2022 Fourth International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT), IEEE, 2022, pp. 1–5, https://doi.org/10.1109/ICERECT56837.2022.10060598.
[46] F.N. Al-Aswadi, H.Y. Chan, K.H. Gan, Automatic ontology construction from text: a review from shallow to deep learning trend, Artif. Intell. Rev. 53 (2020) 3901–3928, https://doi.org/10.1007/s10462-019-09782-9.
[47] M. Neves, J. Ševa, An extensive review of tools for manual annotation of documents, Brief. Bioinform. 22 (1) (2021) 146–163, https://doi.org/10.1093/bib/bbz130.
[48] C.D. Manning, M. Surdeanu, J. Bauer, J.R. Finkel, S. Bethard, D. McClosky, The Stanford CoreNLP natural language processing toolkit, in: In: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2014, pp. 55–60.
[49] T. Anikina, A. Koller, M. Roth, Predicting coreference in abstract meaning representations, in: Proceedings of the Third Workshop on Computational Models of Reference, Anaphora and Coreference, 2020, pp. 33–38.
[50] J. Steinberger, M. Poesio, M.A. Kabadjov, K. Ježek, Two uses of anaphora resolution in summarization, Inf. Process. Manag. 43 (6) (2007) 1663–1680, https://doi.org/10.1016/j.ipm.2007.01.010.
[51] H. Lee, M. Recasens, A. Chang, M. Surdeanu, D. Jurafsky, Joint entity and event coreference resolution across documents, in: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, 2012, pp. 489–500.
[52] L. De Langhe, O. De Clercq, V. Hoste, Towards fine (r)-grained identification of event coreference resolution types, Computat. Linguist. Netherlands J. 12 (2022) 183–205.
[53] C. Caracciolo, A. Stellato, A. Morshed, G. Johannsen, S. Rajbhandari, Y. Jaques, J. Keizer, The AGROVOC linked dataset, Semantic Web. 4 (3) (2013) 341–348.
[54] S.S. Biswas, Potential use of chat gpt in global warming, Ann. Biomed. Eng. 51 (6) (2023) 1126–1127, https://doi.org/10.1007/s10439-023-03171-8.
[55] J. Choi, T. Lee, K. Kim, M. Seo, J. Cui, S. Shin, Discovering message templates on large scale bitcoin abuse reports using a two-fold NLP-based clustering method, IEICE Trans. Inf. Syst. 105 (4) (2022) 824–827, https://doi.org/10.1587/transinf.2021EDL8092.
[56] J. Peng, M. Zhao, J. Havrilla, C. Liu, C. Weng, W. Guthrie, R. Schultz, K. Wang, Y. Zhou, Natural language processing (NLP) tools in extracting biomedical concepts from research articles: a case study on autism spectrum disorder, BMC Med. Inf. Decis. Making 20 (11) (2020) 1–9, https://doi.org/10.1186/s12911-020-01352-2.
[57] J. Plisson, N. Lavrac, D. Mladenic, A rule based approach to word lemmatization, In: Proceedings of IS 3 (2004) 83–86.
[58] Balakrishnan, V., & Lloyd-Yemoh, E.: Stemming and lemmatization: A comparison of retrieval performances (2014).
[59] Y. Liu, L. Wang, T. Shi, J. Li, Detection of spam reviews through a hierarchical attention architecture with N-gram CNN and bi-LSTM, Inf. Syst. 103 (2022) 101865, https://doi.org/10.1016/j.is.2021.101865.
[60] Y. Doval, C. Gómez-Rodríguez, Comparing neural-and N-gram-based language models for word segmentation, J. Assoc. Inf. Sci. Technol. 70 (2) (2019) 187–197, https://doi.org/10.1002/asi.24082.
[61] M.P. Geetha, D.K. Renuka, Improving the performance of aspect based sentiment analysis using fine-tuned Bert Base uncased model, International Journal of Intelligent Networks. 2 (2021) 64–69, https://doi.org/10.1016/j.ijin.2021.06.005.
[62] Tida, V. S., & Hsu, S.: Universal spam detection using transfer learning of BERT model. arXiv preprint arXiv:2202.03480 (2022). https://doi.org/10.48550/arXiv.2202.03480.
[63] Ghavidel, H. A., Zouaq, A., & Desmarais, M. C.: Using BERT and XLNET for the Automatic Short Answer Grading Task. In: CSEDU, vol.1, pp. 58-67 (2020).
[64] Roller, S., & Erk, K.: Relations such as hypernymy: Identifying and exploiting hearst patterns in distributional vectors for lexical entailment. arXiv preprint arXiv:1605.05433 (2016). https://doi.org/10.48550/arXiv.1605.05433.
[65] Roller, S., Kiela, D., & Nickel, M.: Hearst patterns revisited: Automatic hypernym detection from large text corpora. arXiv preprint arXiv:1806.03191 (2018). https://doi.org/10.48550/arXiv.1806.03191.
[66] C. Wu, X. Li, R. Jiang, Y. Guo, J. Wang, Z. Yang, Graph-based deep learning model for knowledge base completion in constraint management of construction projects, Comput. Aided Civ. Inf. Eng. 38 (6) (2023) 702–719, https://doi.org/10.1111/mice.12904.
[67] Ali, S. J., Guizzardi, G., & Bork, D.: Enabling Representation Learning in Ontology-Driven Conceptual Modeling using Graph Neural Networks. In: 35th Intl. Conf. on Advanced Information Systems Engineering, (2023).
[68] Loper, E., & Bird, S.: Nltk: The natural language toolkit. arXiv preprint cs/0205028, (2002).
[69] Bird, S.: NLTK: the natural language toolkit. In: Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions, pp. 69-72 (2006).
[70] Hagberg, A., Swart, P., & S Chult, D.: Exploring network structure, dynamics, and function using NetworkX (No. LA-UR-08-05495; LA-UR-08-5495). Los Alamos National Lab.(LANL), Los Alamos, NM United States (2008).
[71] P. Barrett, J. Hunter, J.T. Miller, J.C. Hsu, P. Greenfield, matplotlib–a portable python plotting package, in: Astronomical Data Analysis Software and Systems XIV, 2005, p. 91.
[72] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, P. Van Mulbregt, SciPy 1.0: fundamental algorithms for scientific computing in python, Nat. Methods 17 (3) (2020) 261–272, https://doi.org/10.1038/s41592-019-0686-2.
[73] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, E. Duchesnay, Scikit-learn: machine learning in python, J. Machine Learn. Res. 12 (2011) 2825–2830.
[74] S. Jugran, A. Kumar, B.S. Tyagi, V. Anand, Extractive automatic text summarization using SpaCy in python & NLP, in: International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), IEEE, 2021, pp. 582–585.
[75] D. Rothman, A. Gulli, Transformers for natural language processing: build, train, and fine-tune deep neural network architectures for NLP with python, PyTorch, TensorFlow, BERT, and GPT-3, Packt Publishing Ltd, 2022.
[76] Imambi, S., Prakash, K. B., & Kanagachidambaresan, G. R.: PyTorch. Programming with TensorFlow: Solution for Edge Computing Applications, 87-104 (2021).
[77] McKinney, W.: pandas: a foundational Python library for data analysis and statistics. Python for high performance and scientific computing. 14(9), 1-9 (2011).
[78] Cox, R.: Regular expression matching can be simple and fast (but is slow in java, perl, php, python, ruby,...). URL: http://swtch. com/rsc/regexp/regexp1. html, 94 (2007).
[79] S. Kumar, S.K. Gupta, Structural and functional insight of knowledge management models in agriculture, Agric. Internat. 6 (2) (2019) 9–15, https://doi.org/10.5958/2454-8634.2019.00016.0.
[80] Gadge, S.S., Benke, A., Salunkhe, V., Soumia, P.S. and Singh, M.: ICAR-DOGR Annual Report 2016-17 (2017).
[81] A. Gupta, V. Mahajan, S. Anandhan, J. Gopal, DOGR-1549-agg (IC0616539; INGR16006), an onion (Allium cepa var. aggregatum) germplasm with unique Early multiplier; suitable for both rabi and kharif seasons; Early maturing with six uniform bulblets per bulb. indian journal of plant genetic, Resources 31 (1) (2017) 107–108.