Article

Research on Knowledge Graphs with Concept Lattice Constraints

Ning Lan, Shuqun Yang *, Ling Yin and Yongbin Gao

School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China; nlan@sues.edu.cn (N.L.); yl@sues.edu.cn (L.Y.); gaoyongbin@sues.edu.cn (Y.G.)
* Correspondence: shqyang@sues.edu.cn

Abstract: The application of knowledge graphs has been restricted in some domains, especially the industrial and academic domains. One of the reasons is that they require a high reliability of knowledge, which cannot be satisfied by the existing knowledge graph research. By comparison, traditional knowledge engineering has a high correctness, but low efficiency is an inevitable drawback. Therefore, it is meaningful to organically connect traditional knowledge engineering and knowledge graphs. Therefore, we propose a theory from Attribute Implications to Knowledge Graphs, named AIs-KG, which can construct knowledge graphs based on implications. The theory connects formal concept analysis and knowledge graphs. We firstly analyze the mutual transformation based on the ideas of symmetry with a strict proof among the attribute implication, the formal context and the concept lattice, which forms the closed cycle between the three. Particularly, we propose an Augment algorithm (IFC-A) to generate the Implication Formal Context through the attribute implication, which can make knowledge more complete. Furthermore, we regard ontology as a bridge to realize the transformation from the concept lattice to the knowledge graph through some mapping methods. We conduct our experiments on the attribute implication from the rule base of an animal recognition expert system to prove the feasibility of our algorithms.

Keywords: attribute implication; formal context; concept lattice; ontology; knowledge graph

1. Introduction

In the era of big data, knowledge graphs have been one of the important means of knowledge representation. They show the knowledge base in the form of a graph, making the knowledge interpretable and infertile, and enabling the cognitive machine. A knowledge graph is essentially a large-scale semantic network, which contains various entities, concepts, and semantic relations between entities. It has become a research hotspot in intelligent search, in-depth question answering, social networks and other fields. A knowledge graph is generally divided into a commonsense knowledge graph and a domain knowledge graph. The former is mainly studied by the major search engine companies to improve the search accuracy and strive to give the target answers directly; The latter can provide multiple target applications depending on the domain. In particular, domain knowledge graphs, such as subject and industry, have high requirements on the reliability of knowledge, and the existing knowledge graphs cannot meet this requirement. There are still some gaps in the application of domain knowledge graphs.

Many scholars have paid extensive attention to the research of knowledge verification and credibility calculation in the commonsense knowledge graph, so as to improve the reliability of knowledge. Ref. [1] proposed to verify the new knowledge extracted from a Nell system based on the MLNS model, mainly through constructing weighted first-order logic rules based on the entity constraints of the new knowledge and the credibility of the knowledge source. Ref. [2] proposed to verify candidate knowledge sets based on entity link prediction and set classification. Ref. [3] put forward the probabilistic soft logic model, which can use multi-source ontology and reasoning technology to infer the
missing relations. Ref. [4] proposed a knowledge verification method based on the Bayesian method, which mainly solves the problem of selecting the most correct knowledge for multi-data source knowledge fusion. Ref. [5] proposed a Bayesian probability graph model to calculate the real reliability. These studies focus mainly on the research of commonsense knowledge graphs. However, facing the more complex knowledge and structure of domain knowledge graphs, how can the requirements of high reliability of knowledge be satisfied?

As a typical representative of symbolism artificial intelligence, traditional knowledge engineering can solve many problems with clear rules and boundaries, which can efficiently satisfy the high requirements of knowledge reliability. The idea of solving problems in knowledge engineering is very forward-looking, but low efficiency is an inevitable drawback, which makes it difficult to fulfill the needs of large-scale open applications in the Internet era. As a new generation of knowledge engineering technology, with the support of big data technology, knowledge graphs can achieve data-driven large-scale automatic knowledge acquisition. This will be a better way of organically connecting traditional knowledge engineering and knowledge graphs, and making use of their respective advantages to make up for each other’s shortcomings.

As a powerful mathematical tool in traditional knowledge engineering, formal concept analysis (FCA) is a mathematization of the conceptual philosophical understanding that was put forward by German mathematician Professor Wille [6] in 1982. It is used to discover, rank and display concepts. In 1999, Ganter [7] summarized the early achievements of formal concept analysis theory. FCA formally describes concepts from two aspects, extent and intent, and uses a binary relationship to express the formal context in the field. The implicit correlations between concepts can be extracted to form the conceptual hierarchy, that is, concept lattice, and finally generate the semantic information that computers can understand. FCA uses mathematical symbols to represent all concepts, which achieves the effect of a formal conceptual model. Ref. [8] points out that FCA will not cause the given information to become coarse-grained, as other data analysis methods, and can contain all data details. Therefore, FCA can directly and effectively provide the correct knowledge source, which is meaningful for improving the knowledge reliability of a domain knowledge graph.

In this paper, we consider a novel perspective to conduct the knowledge-graph-related research, which connects traditional knowledge engineering and knowledge graphs. We propose a theory from Attribute Implications to Knowledge Graphs (AIs-KG), which can construct a knowledge graph through attribute implications. The theory gets through the attribute implication, the formal context, the concept lattice, ontology and the knowledge graph. The specific procedure is shown in Figure 1. Specially, we propose to obtain implications from an expert system, which can ensure the correctness of the data source. We conduct our algorithm on implications from the rule base of an animal recognition expert system to prove that the algorithm is feasible.

![Figure 1. From attribute implication to the knowledge graph.](image)

The contributions of our theory are summarized below:

1. The theory connects formal concept analysis and a knowledge graph, which provides a novel method to construct knowledge graphs based on attribute implications.
2. We analyze the mutual transformation between the attribute implication, the formal context and the concept lattice, which realizes the closed cycle between the three.
3. We put forward an IFC-A to generate an implication formal context, which can supplement the domain knowledge based on attribute implications.
4. We show that the theory we present has many application possibilities, and have given one application as an illustration. We apply the theory on knowledge graph
completion to CN-DBpedia and Probase datasets, which shows that the theory is applicable and effective.

The rest of the paper is organized in the following way. Related work and methodology are discussed in Sections 2 and 3, respectively, followed by the experiment and the conclusion in Sections 4 and 5, respectively.

2. Related Work

In this section, we describe the related work in two aspects: related researches in the field of formal concept analysis and the research on ontology construction and the application to knowledge graphs.

2.1. Formal Concept Analysis

FCA was proposed by Wille [6] in 1982, and its function is to mine, sort and represent concepts by using mathematical models. In FCA, the set of objects belonging to a concept is called the extent of the concept, and the set of attributes corresponding to all objects is called the intent of the concept. According to the determination of extent and intent, we can use the binary relationship to express all concepts and corresponding relationships, and extract the concept hierarchical relationship—concept lattice.

Common concept lattice construction algorithms can be divided into three categories: batch processing algorithm, incremental algorithm and parallel algorithm.

The main idea of a batch processing algorithm is to find all the concepts in the formal context and establish the direct predecessor-successor relationship between all the concepts. Batch processing algorithms can be divided into three categories according to different construction methods of concept lattice: bottom-up algorithm [9], top-down algorithm [10] and enumeration algorithm [11,12].

An incremental algorithm mainly adds the objects in the formal context to the concept lattice one by one. Compared with a batch processing algorithm, the advantage is to avoid the reconstruction of the concept lattice caused by a dynamic update due to the increase in the number of objects in the formal context. The main step is to insert the object step by step after initializing the concept lattice to be empty. When the \(i\)-th object is inserted, the existing concepts in the concept lattice are intersected with the newly inserted object, and the concept lattice is updated in different ways according to the operation results. The most representative progressive algorithms are the Godin algorithm [13] and the AddIntent algorithm [14].

A parallel algorithm is a new research direction with the development of computer technology and network technology. The idea is that when inserting a new object, it does not need to traverse all nodes in the concept lattice, but check the nodes that have at least one common attribute with the newly inserted object, which can significantly improve the construction efficiency.

2.2. Mapping from Concept Lattice to Knowledge Graph

This section may be divided by subheadings. It should provide a concise and precise description of the experimental results, their interpretation and the experimental conclusions that can be drawn.

From the concept lattice to the knowledge graph, there are two main steps: (1) mapping from concept lattice to ontology; (2) mapping from ontology to knowledge graph. Ontology is an important link between formal concept analysis and the knowledge graph.

Ontology is a clear formalization of a conceptual model, defined by Gruber [15] in 1993. In practice, most formalizations are not clear, but vaguely exist in documents, the human brain or the brain’s activities. Therefore, for ontology generation, the difficulty lies in how to materialize the hidden knowledge and make it clear in ontology, that is, how to find out all possible abstract concepts. Considering the common points of ontology and formal concept analysis about formal expression, many ontology construction methods based on FCA [4,16–18] have attracted the attention of many scholars. The representative
Methods can be divided roughly into four categories: methods of Cimiano, methods of GuTao, methods of Haav, methods of Marek Obitko.

Methods of Cimiano [19,20]: Cimiano et al. proposed a domain ontology construction method which uses formal concept analysis to analyze the usage of words in the text, in order to obtain the corresponding background knowledge and then generate ontology. It realizes the automatic construction of a domain ontology and improves the construction efficiency and the formalization of ontology.

Methods of GuTao [21]: GuTao et al. developed the “Fcatab” plug-in to automatically obtain the formal context through the corresponding relationship between the ontology and FCA, and can be converted into the formal context input format required by the concept lattice tool ConExp. It can effectively realize a semi-automatic domain ontology construction with the participation of domain experts.

Methods of Haav [22,23]: Haav et al. put forward a semi-automatic domain ontology construction method, which combines formal concept analysis with rule-based language. The two preconditions of this method are: (1) The content of the domain text is relatively short; (2) The supposition that the domain text describes an entity, which contains terms describing the domain.

Methods of Marek Obitko [24]: Marek Obitko et al. put forward a method of applying formal concept analysis to domain ontology construction in the GACR project. The method convention: concepts are described by attributes; Attributes determine the hierarchy of concepts; When the attributes of two concepts are the same, the two concepts are the same; Ontology representation can be directly described by the modified concept lattice.

The limitations of these four classical researches have been shown in Table 1. And Table 2 illustrate the result of our comparison with four classical methods.

| Classical Research | Limitations                                                                 |
|--------------------|-----------------------------------------------------------------------------|
| Cimiano et al. [19,20] | (1) Generated verb–object dependencies cannot be all correct.               |
|                    | (2) This method is only suitable for English texts.                         |
|                    | (3) This method only considers the object–attribute relation formed by the verb–object phrase. |
| GuTao et al. [21]  | (1) Relying on a protégé ontology modeling tool.                            |
|                    | (2) “Fcatab” only supports single value formal context.                     |
| Haav et al. [22,23] | (1) Insufficient formalization of the domain ontology concept.              |
|                    | (2) The expression vacancy of the domain ontology concept extent.           |
|                    | (3) The attribute expression of the ontology also becomes a vacancy.       |
| Marek Obitko et al. [24] | (1) The generation of a formal context needs to be done manually.        |
|                    | (2) This method starts with empty objects and attributes every time, which requires a lot of work and is only suitable for the construction of a small domain ontology. |
Table 2. Comparison with our method.

| Research                          | Completeness of Documents | Mapping Rules                        |
|----------------------------------|---------------------------|--------------------------------------|
| Cimiano et al. [19,20]           | Incomplete                | Intent → Ontology concept            |
|                                  |                           |                                      |
| GuTao et al. [21]                | Incomplete                | Extent → Ontology sub-concept        |
| Haav et al. [22,23]              | Incomplete                | Intent → Ontology concept            |
| Marek Obitko et al. [24]         | Incomplete                | Node concept → Ontology concept      |
|                                  |                           |                                      |
| Our method                       | More complete             | Node concept → Ontology concept      |
|                                  |                           | Intent → Ontology concept            |
|                                  |                           | Extent → Ontology sub-concept        |
|                                  |                           | Relation → Ontology relation         |

After introducing formal concept analysis into domain ontology construction, researchers have solved some key problems in domain ontology construction to a certain extent [25–30], including the semi-automatic construction and the formalization of ontology; the relation mining between implicit concepts; the degree of subjective influence and ontology sharing and reuse; rich ontology semantics; and the visualization of an ontology model based on a concept lattice.

Based on the constructed ontology, researchers continue to consider the close relationship between ontology and knowledge graphs. In fact, ontology describes the basic framework of people’s cognition of a domain, and, in contrast, a knowledge graph is rich in entities and relation instances. So far, ontology has been widely used in the research of knowledge graphs [31–34]. As noted by Paulheim [35], ontology is mainly involved in the construction and data introduction phases of KGs. They are useful for these phases, as they provide restrictions for instances and relations.

Therefore, inspired by the previous work, our paper proposes to organically connect formal concept analysis and knowledge graphs, and makes use of their respective advantages to make up for their shortcomings.

3. Methodology

In this section, we start by briefly introducing some basic concepts of FCA, and then give the definition of implication and related theorems. Based on these definitions, we put forward the Asls-KG theory and introduce each part in detail. We introduce the IFC-A algorithm, which can generate an implication formal context based on the augment method. Then, we discuss the mutual transformation methods among the attribute implication, the formal context and the concept lattice. Furthermore, we present the mapping rules among the concept lattice, ontology and knowledge graph.

3.1. Basic Concept of FCA

We firstly provide a brief introduction about FCA, in order to allow for a better understanding. The basic concepts of FCA are formal context, formal concept, concept lattice and attribute implication.

The main definitions are as follows:

**Definition 1.** Let \( G \) be the set of objects, \( M \) be the set of attributes and \( I \) be the binary relationship between object set \( G \) and attribute set \( M \). Then, the triple \( (G, M, I) \) is a formal context.

For \( g \in G, m \in M, (g, m) \in I \) (or denoted as \( gIm \)), it means “object \( g \) owns attribute \( m \)”. Generally speaking, the formal context can be expressed in the form of a two-dimensional table, in which each row is an object and each column is an attribute. In the formal context, if the intersection of \( g \) rows and \( m \) columns is 1, it means that the object \( g \) has attribute \( m \); if it is 0, it means that the object \( g \) does not have attribute \( m \).
Definition 2. Given a formal context \( K = (G, M, I) \), for \( A \subseteq G, B \subseteq M \), the following operations are defined:
\[
A' = \{m \in M | \exists g \in A \},
\]
\[
B' = \{g \in G | \exists m \in B \}.
\]
\( A' \) is the set of attributes shared by all objects in \( A \); \( B' \) is the set of objects corresponding to all attributes in \( B \).

If \( A' = B, B' = A \), then the binary \((A, B)\) is a concept in the formal context, where \( A \) is the intent of concept \((A, B)\) and \( B \) is the extent of concept \((A, B)\).

Example 1. A formal context \( K = (G, M, I) \) is shown in Table 3, where the object set \( G = \{1, 2, 3, 4, 5\} \) and the attribute set \( M = \{a, b, c, d, e\} \).

Table 3. A formal context.

|   | \(a\) | \(b\) | \(c\) | \(d\) | \(e\) |
|---|------|------|------|------|------|
| 1 | 1    | 1    | 0    | 0    | 0    |
| 2 | 0    | 1    | 0    | 1    | 1    |
| 3 | 0    | 1    | 1    | 0    | 0    |
| 4 | 0    | 0    | 1    | 0    | 1    |
| 5 | 0    | 0    | 0    | 1    | 1    |

According to Definition 2, We can get all the concepts in the formal context \( K: (G, \emptyset), (1, ab), (2, bde), (3, bcd), (4, ce), (23, bd), (245, e), (235, d), (135, b), (\emptyset, M) \).

Definition 3. Let \((A_1, B_1)\) and \((A_2, B_2)\) be two concepts on the formal context \( K = (G, M, I) \), and \( A_1 \subseteq A_2 \) (equivalent to \( B_1 \supseteq B_2 \)), then we call \((A_1, B_1)\) the subconcept of \((A_2, B_2)\) and \((A_2, B_2)\) the parent concept of \((A_1, B_1)\), denoted as \((A_1, B_1) \leq (A_2, B_2)\).

Definition 4. Let \( K = (G, M, I) \) be a formal context. The set of all concepts in \( K \) and the ordered set of partial order relations on the set are called the concept lattice of \( K \), denoted as \( L(K) \).

Definition 5. Let \( S \) be a set and \( R \) be called a partial order relation. If \( a, b, c \in S \), then the following properties are satisfied.
1. Reflexivity: \( \forall a \in S, aRa \);
2. Anti-symmetry: \( \forall a, b \in S, aRb, bRa \Rightarrow a = b \);
3. Transitivity: \( \forall a, b, c \in S, aRb, bRc \Rightarrow aRc \).

The concept lattice can be visualized by a Hasse diagram. The corresponding concept lattice in Example 1 is shown in Figure 2.

Figure 2. Concept lattice of Example 1.
3.2. The AIs-KG Theory

As shown in Figure 3, the specific procedure of the AIs-KG theory has been described in detail.

Figure 3. Specific progress from the attribute implication to the knowledge graph.

Firstly, through the implication relationship between attributes, the initial formal context can be augmented to generate the implication formal context, which contains more complete information. And then, the concept lattice can be constructed by the existing concept lattice construction algorithm and corresponding tool. Furthermore, we analyze the closed cycle among attribute implications, formal contexts and concept lattices. Finally, taking ontology as a bridge, we realize the mapping from concept lattice to knowledge graph according to the mapping rules. The details of our AIs-KG theory are introduced below.

3.2.1. Formal Context Generation Algorithm Based on Attribute Implication

1. Attribute implications

FCA can be seen as a kind of attribute logic, aiming at studying possible attribute combinations. Researchers generally believe that attributes can be independent of each other, or that there can be an interdependent relation, described as implication.

The definition of implication is as follows:

**Definition 6.** Let \( K = (G, M, I) \) be a formal context and \( A, B \subseteq M \), \( A \rightarrow B \) be an attribute implication in \( K \). If every object with attributes in \( A \) also has attributes in \( B \), \( A \) is called the premise of \( A \rightarrow B \) and \( B \) is called the conclusion of \( A \rightarrow B \).

**Theorem 1.** Let \( K = (G, M, I) \) be a formal context, \( A, B \subseteq M \). Then, these relations are equivalent:

\[
A \rightarrow B \text{ is an implication of } K \Leftrightarrow A' \subseteq B' \Leftrightarrow B \subseteq A''.
\]  

(3)
Proof of Theorem 1. \( A \rightarrow B \) is an implication of \( K \). According to the definition of implication and the relationship between extent and intent, we can know that \( A \) has all the attributes of \( B \). So, \( B \subseteq A'' \). Then it is easy to know \( B' \supseteq A''' \), that is, \( B' \supseteq A' \). \( \square \)

2. Implication formal context generation algorithm IFC-A

In the process of formal context generation, due to different understandings of the data or limited acquisition methods, there is the phenomenon of incomplete attributes corresponding to objects. In addition, based on the dynamic change of data, formal contexts and concept lattices also need to be updated synchronously. Thus, we propose the augment method IFC-A to generate the implication formal context based on the given implication set. The method applies the idea of an incremental concept lattice construction algorithm to generate the implication formal context, which can improve the generation efficiency and adapt to the dynamic increase of the number of implications by adding more attributes corresponding to the objects one by one.

Before the algorithm procedures are given, implication formal context and other corresponding concepts are firstly defined.

Definition 7. Let \( P = \{p_1, p_2, \ldots, p_r\} \ (p_i = A_i \rightarrow B_i, \ i \in \{1, 2, \ldots, r\}) \) be an implication set of domain knowledge, and \( K = (G, M, I) \) a formal context constructed by \( P \), then \( K \) is called the implication formal context of \( P \). If and only if \( \forall g \in G, \ g \in A'_i \) and \( g \in B'_i \) hold.

Definition 8. Let \( K = (G, M, I) \) be a formal context and \( P = \{p_1, p_2, \ldots, p_r\} \ (p_i = A_i \rightarrow B_i, \ i \in \{1, 2, \ldots, r\}) \) an implication set of domain knowledge. All instances in the implication set constitute the object base.

The core idea of the augment algorithm is to gradually increase the missing attributes according to the attribute implication relations, and finally get the implication formal context. The process of the algorithm is divided into two parts: (1) generate the object base from instances in implication relations, (2) generate the implication formal context gradually by using the initial formal context composed of the object base and corresponding attributes.

For a given implication set \( P = \{p_1, p_2, \ldots, p_r\} \), the specific process of the augment algorithm is as follows:

- **Input**: Implication set \( P = \{p_1, p_2, \ldots, p_r\} \).
- **Output**: Implication formal context \( K_p \).
- **Step1**: Obtain the attribute set \( M = \{m_1, m_2, \ldots, m_n\} \) and object base \( S_o \) according to \( P \).
- **Step2**: The objects in \( S_o \) are combined with the attributes to generate the initial formal context \( K_o \).
- **Step3**: For each implication \( p_i = A_i \rightarrow B_i \ (i \in \{1, 2, \ldots, r\}) \), take it out in turn.
- **Step4**: Determine whether the attributes in \( B_i \) and \( A_i \) are in \( K_o \) in turn. For \( \forall m_i \in B_i \), if \( m_i \) is in \( K_o \), judge whether all the attributes in \( A_i \) are in \( K_o \). If not, complete them.
- **Step5**: Repeat step 4 until the final implication decision in \( P \) is completed and generate the implication formal context \( K_p \).

The pseudocode of the augment algorithm has been shown in Algorithm 1.
Algorithm 1. Pseudocode of the augment algorithm.

Algorithm implementation:

Input: Attribute implication set: \( P = \{ p_1, p_2, \ldots, p_r \} \)

Implication formal context \( K_p \)

\( K_p = \emptyset, m = 0, M = \emptyset \)

For \( p_i \in P \)

\( \text{count}(m) \) // Count the number of attributes in each implication.

\( \text{add } m \text{ into } M \) // Add attributes into \( M \).

End For

obtain \( S_o \) from \( P \) // Obtain object sets.

obtain \( K_o \) // Generate initial formal context according to \( S_o \).

For \( p_i = A_i \rightarrow B_i \in P \)

If \( \forall m_i \in B_i \text{ is in } K_o \)

If \( \forall m_i \in A_i \text{ is not in } K_o \)

update \( K_o \)

Else

Continue // The formal context remains the same.

Else

Continue // The formal context remains the same.

End If

End For

\( K_p = K_o \) // Get the implication formal context according to the last updated \( K_o \).

End

Example 2. Table 3 is an initial formal context \( K = (G, M, I) \), where the object set \( G = \{1, 2, 3, 4, 5\} \) and the attribute set \( M = \{a, b, c, d, e\} \). Given a set of implications: \( \{a\} \rightarrow \{f\}, \{b, c, d\} \rightarrow \{g\}, \{e\} \rightarrow \{h\} \), the implication formal context can be generated by the implications, which has been shown in Table 4.

Table 4. The implication formal context.

|     | a | b | c | d | e | f | g | h |
|-----|---|---|---|---|---|---|---|---|
| 1   | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 2   | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 3   | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 4   | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 5   | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |

3.2.2. Mutual Transformation

According to the IFC-A and the existing concept lattice construction algorithms, we can realize the process of generating a concept lattice through attribute implication. In reverse thinking, when the concept lattice is known, we can also generate a formal context through specific algorithms, and mine the relationship between attributes. We analyzed the mutual transformation based on the ideas of symmetry with a strict proof among the attribute implication, the formal context and the concept lattice.

1. Formal context generation based on a given concept lattice

Each node in the concept lattice represents a concept, which is composed of extent and intent. Therefore, based on the given concept lattice, each concept node of the concept lattice can be traversed through the traversal algorithm. The extent and intent of each node can be mined in turn, which constitutes the objects and attributes of the formal context, respectively. Taking hierarchy traversal as an example, the flow chart and pseudocode of the algorithm are shown in Algorithm 2 and Figure 4.
### Algorithm 2. Pseudocode of the augment algorithm.

**Algorithm implementation:**

**Input:** Concept lattice.

**Output:** Formal context.

```java
array = ∅; //Define an empty table to store the formal context.
queue_1 = ∅; queue_2 = ∅;
Queue<TreeNode> queue = new LinkedList<>();
List<TreeNode> list = new LinkedList<>();
Queue.add(treeNode);
while(queue.size()>0):
    TreeNode node = queue.poll(); //node contains extent and intent.
    if(queue.size() = 1 and node_extent = ∅):
        return array;
    if(node_intent ! = ∅):
        queue_1.add(node_extent);
        queue_2.add(node_intent);
        while(queue_1.size()>0):
            while(queue_2.size()>0):
                array[queue_1.poll()][queue_2.poll()] = 1;
        list<TreeNode> childrens = node.childrens;
        for(TreeNode childNode: childrens):
            Queue.add(childNode);
    end
end
```

**2. Implication rules mining**

This paper adopts the NextClosure algorithm [7], which was proposed by Ganter, to get a list of implications from a concept lattice. The specific algorithm has been shown in Algorithm 3.

![Figure 4](image-url)

**Figure 4.** Specific progress from the attribute implication to the knowledge graph.

We give a table filling method to add the extent and intent into the formal context. When the extent and intent of the node are obtained by hierarchy traversal, the table-filling method will tick off the corresponding intent of the extent. Take the concept lattice in Figure 2 as an example: for the node (1, ab), the table-filling method will tick off the attributes {a} and {b} of object {1}.

2. Implication rules mining

This paper adopts the NextClosure algorithm [7], which was proposed by Ganter, to get a list of implications from a concept lattice. The specific algorithm has been shown in Algorithm 3.
Algorithm 3. Pseudocode of the augment algorithm.

Algorithm implementation:
Input: A concept lattice $L$.
Output: A set of implications.
1. Function GenerateRulesForNode $(N = (X, X'))$. //Returns the complete set of rules
2. generated from the node $N$.
3. $Δ := ∅$;
4. if $X ≠ ∅$ and $||X'|| > 1$ then //discard some trivial rules such as $P ⇒ ∅$
5. for each nonempty set $P ∈ \{ P(X') ∼ X' \}$ in ascending $||P||$ do;
6. if $\exists M = (Y, Y')$ parent on $N$ such that $P ⊆ Y'$ then
7. if $\exists P' ⇒ Q ∈ Δ$ such that $P' ⊂ P$ then
8. $Δ := Δ ∪ \{ P ⇒ X' − P \}$
9. end if
10. end if
11. end for
12. end if
13. return $(Δ)$
14. end
15. $∑ = ∅$; //the cumulative set of implication rules.
16. for each node $H = (X, X') ∈ L$ in ascending $||X'||$ do
17. $Rules[H] = GenerateRulesForNode(H)$
18. $∑ = ∑ ∪ Rules[H]$;
19. end for
20. return $(∑)$
21. end

3. Implication rules mining

Combining with the augment method of generating the implication formal context based on the given implication set, this paper proposes an FCA ontology construction model based on constraints, which is actually the mapping1 in Figure 3. The ontology construction model has been shown in Figure 5, which also indicates the connection with the traditional skeleton method [36].

![Figure 5. Ontology construction model and the connection with the skeleton method.](image-url)

The specific process is as follows:
1. Concept lattice generation
   The concept lattice is constructed by the algorithms mentioned above.
2. Define classes and levels
   In order to satisfy the application requirements of the knowledge graph, the core classes and their levels are extracted from the concept lattice according to the attribute
implications. For the node with only one attribute in the concept lattice, the attribute is defined as the concept of the node. For the node with multiple attributes, the relationship between the attributes is analyzed according to the attribute implications “\(A \rightarrow B\)”, and \(A\) is defined as the concept of the node. Then, extract the core classes from the concepts and define the levels from the concept lattice.

3. Define the attributes of the classes
Summarize the concepts and attributes outside the core classes, divide them into corresponding classes as attributes and clarify the relationship between the classes.

4. Ontology representation
Ontology can be represented by the ontology modeling tool Protégé 4.3 and stored as an OWL format file.

5. Ontology evaluation
Taking the five ontology construction principles [15] proposed by Thomas as the evaluation criteria.

6. Ontology construction
If the ontology does not meet the evaluation criteria, update the ontology and finally the output ontology; or else, output the ontology directly.

In the process of ontology construction, the following necessary conditions should also be followed:

1. In the process of constructing the initial ontology, we must strictly follow the principles and methods of ontology construction; in the process of ontology update, in order to prevent the error of ontology update caused by artificial subjective interference during the ontology update, we should also strictly abide by the ontology update method proposed in this paper, and cannot ignore any link due to human intervention.

2. The construction of the domain ontology does not depend on one person, but on collective efforts, and the constructed domain ontology needs to be recognized by everyone.

3. In the process of transforming the concept lattice into an ontology, the mapping rules should be strictly followed, and the visualization of the concept lattice and the ontology should be realized by computer.

4. Mapping from ontology to knowledge graph
As a form of knowledge representation, a knowledge graph is a large-scale semantic network, including entities, concepts and their semantic relations. The semantic network is a way to express knowledge through points and edges in the form of graphics, and points and edges are the basic elements. The points in a semantic network can be entities, concepts and values. The edges can be divided into attributes and relations. Therefore, the basic elements between ontology model and the Neo4j graph model of the knowledge graph can be mapped as shown in Figure 6, which is actually the mapping2 in Figure 3, and the specific mapping rules are explained as follows.

![Figure 6. Mapping from ontology to knowledge graph.](image-url)
① Node mapping

Node mapping mainly includes concept mapping and entity mapping. The nodes stored in the Neo4j graph database are the concepts and concrete instances in the ontology model. Concepts and concrete instances with clear meaning are regarded as nodes in Neo4j.

② Relation mapping

The relation in the Neo4j graph database is used to connect different concepts in the ontology model. When building an ontology model, a self-defined relation between two concepts is regarded as the relation between concepts, so the relations between concepts in ontology are transformed into the relations between nodes in the graph database.

4. Experiment

This section proves the feasibility of the AIs-KG theory through experiments, and is organized as follows. First, we generate an implication formal context based on the proposed augment algorithm IFC-A. Then, we construct an animal recognition ontology and map it into a knowledge graph. Finally, we introduce one application direction of the AIs-KG theory.

4.1. Implication Formal Context Generation Based on IFC-A

According to the theory proposed in this paper, the premise of generating an implication formal context is to find attribute implications. How can the attribute implications be obtained? As we know, the expert system contains a lot of knowledge and experience of experts in a certain field, which is stored in the knowledge base as a set of rules to form a knowledge base system. The facts for a knowledge base must be acquired from human experts in various domains through interviews and observations. The knowledge is then usually represented in the form of “If-then” rules (production rules): “If some condition is true, then the following inference can be made (or some action taken).” [37]. “If-then” rules actually are the attribute implications. Therefore, this paper proposes to obtain the implications from existing expert system rule bases, which can provide the highest level of professional knowledge and a good reliability.

This paper selects the rule base of an animal recognition expert system [38] as the basic data source. The system has a certain knowledge about animal characteristics. When the user inputs the description of an animal, it can automatically judge what animal is described according to the existing knowledge. Some expert rules are as follows:

Rule1: If (animal has hair), then (animal is mammal).
Rule2: If (animal gives milk), then (animal is mammal).
Rule3: If (animal has feathers), then (animal is bird).
Rule4: If (animal flies) (animal lays egg), then (animal is bird).
Rule5: If (animal eats meat), then (animal is carnivore).
Rule6: If (animal has sharp teeth) (has claws) (has forward eyes), then (animal is carnivore).
Rule7: If (animal has hair), then (animal is mammal).
Rule8: If (animal is mammal) (animal chews cud), then (animal is ungulate).
Rule9: If (animal is mammal) (animal is carnivore) (animal has tawny color) (animal has dark spots), then (animal is cheetah).
Rule10: If (animal is mammal) (animal is carnivore) (animal has tawny color) (animal has black stripes), then (animal is tiger).
Rule11: If (animal is ungulate) (animal has long neck) (animal has long legs) (animal has dark spots), then (animal is giraffe).
Rule12: If (animal is ungulate) (animal has black stripes), then (animal is zebra).
Rule13: If (animal is bird) (animal does not fly) (animal has long neck) (animal has long legs) (animal is black and white), then (animal is ostrich).
Rule14: If (animal is bird) (animal does not fly) (animal swims) (animal is black and white), then (animal is penguin).
Rule 15: If (animal is bird) (animal flies well), then (animal is albatross).

We consider the animal instances as the object base in the formal context, and the corresponding attribute as the initial attributes. Then, the initial formal context can be generated according to relevant expert rules in [38], shown in Table 5. For convenience, we denote {‘animal’, ‘mammal’, ‘carnivore’, ‘bird’, ‘ungulate’, ‘has tawny color’, ‘has dark sports’, ‘has black stripes’, ‘has long neck’, ‘has long leg’, ‘black and white’, ‘can swim’, ‘fly well’, ‘hairy’, ‘give milk’, ‘has feather’, ‘can fly’, ‘lay egg’, ‘eat meat’, ‘has sharp teeth’, ‘has claw’, ‘has forward eyes’, ‘has hoof’, ‘chew cud’} as {‘a’, ‘b’, ‘c’, ‘d’, ‘e’, ‘f’, ‘g’, ‘h’, ‘i’, ‘j’, ‘k’, ‘l’, ‘m’, ‘n’, ‘o’, ‘p’, ‘q’, ‘r’, ‘s’, ‘t’, ‘u’, ‘v’, ‘w’, ‘x’}.

Table 5. Initial formal context.

|   | a | b | c | d | e | f | g | h | i | j | k | l | m |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Cheetah | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| Tiger | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| Giraffe | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| Zebra | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| Ostrich | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| Penguin | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Albatross | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

And then, some attribute implication relations can be extracted from the remaining rules:

$\{\text{animal}, \text{hairy}\} \rightarrow \{\text{mammal}\}$

$\{\text{animal}, \text{gives milk}\} \rightarrow \{\text{mammal}\}$

$\{\text{animal}, \text{has feathers}\} \rightarrow \{\text{bird}\}$

$\{\text{animal}, \text{can fly}, \text{lays eggs}\} \rightarrow \{\text{bird}\}$

$\{\text{animal}, \text{eats meat}\} \rightarrow \{\text{carnivore}\}$

$\{\text{animal}, \text{has sharp teeth}, \text{has claws}, \text{has forward eyes}\} \rightarrow \{\text{mammal}\}$

$\{\text{animal}, \text{mammal, has hoofs}\} \rightarrow \{\text{ungulate}\}$

$\{\text{animal}, \text{mammal, chews cud}\} \rightarrow \{\text{ungulate}\}$

According to these implications, we can generate the implication formal context based on IFC-A, in which the appropriate effective attributes have been selected to add into the initial formal context. The implication formal context is shown in Table 6.

Table 6. Initial formal context.

|   | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Cheetah | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| Tiger | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| Giraffe | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| Zebra | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| Ostrich | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Penguin | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| Albatross | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |

4.2. Mapping from Implication Formal Context to Knowledge Graph Based on AIs-KG

Firstly, by using the concept lattice construction tool Concept Explorer 1.3 and inputting the implication formal context generated by IFC-A, the concept lattice based on a formal context can be obtained. A Hasse diagram is a concept lattice description tool, which can realize the visualization of concepts and their hierarchical relations. The concept lattice
Based on a Hasse diagram obtained by Table 6 has been shown in Figure 7. In fact, the process of constructing a concept lattice using a formal context is the process of clustering various concepts.

![Figure 7. Constructed concept lattice.](image)

According to the mapping rule from concept lattice to ontology, the specific steps can be conducted as follows:

Firstly, according to the concept lattice and attribute implications, we define the different classes in the ontology model. For example, for the node with the attributes {“carnivore”, “hairy”, “gives milk”, “eats meat”, “has sharp teeth”, “has forward eyes”, “has tawny color”, “has claws”}, we can analyze the relationships between attributes according to attribute implications: {carnivore}→{hairy, gives milk, has sharp teeth, has tawny color, has forward eyes, has claws}.

Therefore, we can regard “carnivore” as the concept of this node, and core classes can be extracted from the concepts, which are {“animal”, “mammal”, “carnivore”, “bird”, “ungulate”}.

Then, the concepts and attributes outside the core classes can be divided into corresponding classes as attributes, and we also need to define the relation between classes, which is “is_A”, and the relation between classes and attributes, which is “Attribute_Label”.

Finally, the concepts, attributes and instances obtained by FCA can be constructed by the ontology editing tool Protégé 4.3 and OWL (Web Ontology Language) to obtain the ontology, which is shown in Figure 8.

Taking the five ontology construction principles [15] proposed by Thomas as the evaluation criteria, it is proved that the ontology constructed in this paper satisfies the objective facts and evaluation standards, follows the ontology construction principles and can satisfy the requirements of practical application.

Based on the constructed ontology and the Neo4j graph database, the entities, relations and attributes in the domain can be designed and compiled, so as to map into the domain knowledge graph. The mapped knowledge graph has been shown in Figure 9. Specific mapping rules are based on Figure 6, and finally, the data is stored in the form of Neo4j.
4.3. One Application of the AIs-KG Theory

The AIs-KG theory can be effectively applied in many different ways. We propose to apply AIs-KG to knowledge graph completion. The experiment has been evaluated on the commonly used knowledge graph datasets—CN-DBpedia [39] and Probase [40]. Details of the datasets have been summarized in Table 7.

Table 7. Details of the datasets. #E and #R denote the number of entities and relations.

| Dataset       | #E              | #R              |
|---------------|-----------------|-----------------|
| CN-DBpedia    | 9,000,000+      | 67,000,000+     |
| Probase       | 5,376,526       | 85,101,174      |
According to the results in Figure 9, we adopt the existing completion algorithm to complete datasets, in which we also need to consider the attribute alignment and entity alignment. All the completed new attributes are shown in Tables 8 and 9. Surprisingly, the results sufficiently illustrate the theory’s effectiveness, as more obligatory properties have been completed.

Table 8. All completed new attributes in DBpedia.

| Entity       | Attribute      | Entity       | Attribute      | Entity       | Attribute      |
|--------------|----------------|--------------|----------------|--------------|----------------|
| Cheetah      | hairy          | Cheetah      | has dark sports| Tiger        | eats meat      |
| Cheetah      | gives milk     | Zebra        | chew cud       | Tiger        | has sharp teeth|
| Cheetah      | has sharp teeth| Tiger        | hairy          | Tiger        | has claws      |
| Cheetah      | has forward eyes| Tiger       | gives milk     | Tiger        | has forward eyes|
| Cheetah      | has tawny color|              |                |              |                |

Table 9. All completed new attributes in Probase.

| Entity       | Attribute      | Entity       | Attribute      | Entity       | Attribute      |
|--------------|----------------|--------------|----------------|--------------|----------------|
| Cheetah      | hairy          | Tiger        | gives milk     | Zebra        | chews cud      |
| Cheetah      | give milk      | Tiger        | eats meat      | Zebra        | has black stripes|
| Cheetah      | eats meat      | Tiger        | has sharp teeth| Ostrich      | lays eggs      |
| Cheetah      | has sharp teeth| Tiger        | has forward eyes| Ostrich      | has long legs  |
| Cheetah      | has forward eyes| Tiger      | has tawny color| Ostrich      | black and white|
| Cheetah      | has tawny color| Giraffe      | has black stripes| Penguin      | lays eggs      |
| Cheetah      | has dark sports| Giraffe      | chews cud      | Penguin      | black and white|
| Tiger        | hairy          | Giraffe      | has black stripes| Albatross    | has feathers   |

As we know, the datasets of CN-DBpedia and Probase are large open-knowledge graphs that have been widely used by many universities and enterprises. The application of our algorithm to these two datasets can achieve a good completion effect with high efficiency, which can prove that our algorithm can adapt to large-scale datasets.

In fact, in addition to the knowledge graph completion, the knowledge graph constructed according to our theory can also be applied to the combination of knowledge graphs through a knowledge fusion algorithm, which can also make the existing knowledge graph more complete. Furthermore, the theory can be widely applied to the industrial field. Take fault information in the manufacturing industry as an example: given the equipment fault information, the corresponding formal context of the “equipment-fault” can be generated, and then the fault information knowledge graph can be naturally constructed according to our theory, which can be used as a reference for future fault diagnoses.

5. Conclusions

This paper first shows that a domain knowledge graph, especially with subject and industry, requires a highly reliable knowledge. However, the knowledge extracted by the information extraction system often has some errors or omissions. This paper proposes a new complete AIs-KG theory. AIs-KG realizes the connection between traditional knowledge engineering and the knowledge graph, making full use of the advantages of traditional knowledge engineering to make up for the problem of the knowledge graph.

Our theory preliminarily explores the combination of the two domains, only to prove the feasibility through algorithms. Compared to others, our algorithm is not mature enough and needs to be iteratively optimized in the future. For further improvement, we will continue to study more efficient mapping rules from the concept lattice to the knowledge graph. This paper only gives a detailed description of one application method for illustration. In fact, the AIs-KG theory can be effectively applied in many directions, and we will explore more industrial applications in the future. Moreover, we will construct a large attribute implication database based on the existing formal context and conduct
regular maintenance checks to ensure the accuracy and timeliness of the data in the implication database.

**Author Contributions:** N.L. performed the experiments, contributed to the development of the theory, executed the detailed analysis and wrote some sections. S.Y. set the objectives of the research, contributed to the design of the algorithms, designed the paper, wrote some sections and performed the final corrections. L.Y. and Y.G. contributed to the design of the algorithms. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Key R&D Program of China, grant number 2020AAA0109400 and National Natural Science Foundation of China, grant number 61802251. And The APC was funded by 61802251.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Restrictions apply to the availability of these data. Data were obtained from Gadaleta et al. [39].

**Conflicts of Interest:** The authors declare no conflict of interest.

**References**

1. Chen, X.; Jia, S.; Xiang, Y. A review: Knowledge reasoning over knowledge graph. *Expert Syst. Appl.* 2020, 141, 112948. [CrossRef]
2. Namata, G.M.; London, B.; Getoor, L. Collective graph identification. *ACM Trans. Knowl. Discov. Data* 2016, 10, 2818378. [CrossRef]
3. Grangel-González, I. A Knowledge Graph Based Integration Approach for Industry 4.0. Ph.D. Thesis, Universitäts-und Landesbibliothek Bonn, Bonn, Germany, 2019.
4. Yang, S.; Shu, K.; Wang, S. Unsupervised fake news detection on social media: A generative approach. *Artif. Intell.* 2019, 33, 5644–5651. [CrossRef]
5. Zhao, X.; Jia, Y.; Li, A. Multi-source knowledge fusion: A survey. *World Wide Web* 2020, 23, 2567–2592. [CrossRef]
6. Ganter, B.; Wille, R.; Franzke, C. *Formal Concept Analysis: Mathematical Foundations*; Springer: Berlin/Heidelberg, Germany, 1997.
7. Ganter, B.; Wille, R.; Wille, R. *Formal Concept Analysis*; Springer: Berlin/Heidelberg, Germany, 1999.
8. Ganter, B.; Wille, R. Applied lattice theory: Formal concept analysis. In *General Lattice Theory*; Grätzer, G., Ed.; Birkhäuser: Basel, Switzerland, 1997.
9. Chein, M. Algorithme de recherche des sous-matrices premières d’une matrice. *Bull. Math.* 1969, 13, 21–25.
10. Bordat, J.P. Calcul pratique du treillis de Galois d’une correspondance. *Math. Inform. Et Ences Hum.* 1986, 96, 31–47.
11. Ganter, B. Two Basic Algorithms in Concept Analysis. In *Formal Concept Analysis*; Springer: Berlin/Heidelberg, Germany, 2010.
12. Zhi, H.; Qi, J.; Qian, T. Three-way dual concept analysis. *Int. J. Approx. Reason.* 2019, 114, 151–165. [CrossRef]
13. Zhi, H.; Qi, J.; Qian, T. Conflict analysis under one-vote veto based on approximate three-way concept lattice. *Inf. Sci.* 2020, 516, 316–330. [CrossRef]
14. Merwe, D.; Obiedkov, S.; Kourie, D. AddIntent: A New Incremental Algorithm for Constructing Concept Lattices. In *Formal Concept Analysis*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 372–385.
15. Gruber, T.R. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *Int. J. Hum.-Comput. Stud.* 1995, 43, 907–928. [CrossRef]
16. Jindal, R.; Seega, K.R.; Jain, S. Construction of domain ontology utilizing formal concept analysis and social media analytics. *Int. J. Cogn. Comput. Eng.* 2020, 1, 62–69. [CrossRef]
17. Shubhra, G.J.; Arvinder, K. Information Retrieval from Software Bug Ontology Exploiting Formal Concept Analysis. *Comput. Sist.* 2020, 24, 3368.
18. Priya, M.; Aswani, K.C. A novel method for merging academic social network ontologies using formal concept analysis and hybrid semantic similarity measure. *Lbr. Hi. Tech.* 2020, 38, 399–419. [CrossRef]
19. Cimiano, P.; Stumme, G.; Hotho, A.; Tane, J. Conceptual knowledge processing with formal concept analysis and ontologies. *Form. Concept Anal. 2004, 11, 189–207.*
20. Cimiano, P.; Staab, S.; Tane, J. Automatic Acquisition of Taxonomies from Text: FCA meets NLP. *Workshop Adapt. Text. Extr. Min. 2003, 11, 10–17.*
21. Tao, G. Using Formal Concept Analysis for Ontology Structuring and Building. Master’s Thesis, Nanyang Technological University, Singapore, 2003.
22. Haav, H.M. A Semi-automatic Method to Ontology Design by Using FCA. *CLA 2004, 13–15.* Available online: [http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.92.6057&rep=rep1&type=pdf](http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.92.6057&rep=rep1&type=pdf) (accessed on 3 November 2021).
23. Haav, H.M. An application of inductive concept analysis to construction of domain-specific ontologies. In Proceedings of the VLDB Pre-conference Workshop on Emerging Database Research in East Europe, Berlin, Germany, 9–12 September 2003; pp. 63–67.
24. Obitko, M.; Snasel, V.; Smid, J. Ontology Design with Formal Concept Analysis, Edited by Vaclav Snasel, Radim Belohlavek. Workshop Concept Lattices Appl. Ostrav. 2004, 11, 111–119.
25. Fawei, B.; Pan, J.Z.; Kollingbaum, M. A Semi-automated Ontology Construction for Legal Question Answering. New Gener. Comput. 2019, 9, 145. [CrossRef]
26. Moreira, A.; Filho, J.L.; Oliveira, A. Automatic Creation of Ontology Using a Lexical Database: An Application for the Energy Sector; Springer: Cham, Switzerland, 2016.
27. Han, P.; Li, Y.; Yin, Y. Ontology Construction for Eldercare Services with an Agglomerative Hierarchical Clustering Method; Springer: Cham, Switzerland, 2019.
28. Vairavasundaram, S.; Logesh, R. Applying Semantic Relations for Automatic Topic Ontology Construction. Dev. Trends Intell. Technol. Smart Syst. 2018, 14, 48–77.
29. Geng, Q.; Deng, S.; Jia, D. Cross-domain ontology construction and alignment from online cust-omer product reviews. Inf. Sci. 2020, 531, 47–67. [CrossRef]
30. Al-Aswadi, F.N.; Chan, H.Y.; Gan, K.H. Automatic ontology construction from text: A review from shallow to deep learning trend. Artif. Intell. Rev. 2020, 53, 3901–3928. [CrossRef]
31. Zeng, W.; Liu, H.; Feng, H. Construction of Scenic Spot Knowledge Graph Based on Ontology. Int. Symp. Distrib. Comput. Appl. Bus. Eng. Sci. 2019, 75, 120–123. [CrossRef]
32. Varma, S.; Shivam, R.; Jamaiyar, A.; Anukriti, S.; Kashyap, A.S. Link Prediction Using Semi-Automated Ontology and Knowledge Graph in Medical Sphere. India Counc. Int. 2020, 5, 9342301.
33. Dou, J.; Qin, J.; Jin, Z. Knowledge graph based on domain ontology and natural language processing technology for Chinese intangible cultural heritage. J. Vis. Lang. Comput. 2018, 48, 19–28. [CrossRef]
34. Amador-Dominguez, E.; Hohenecker, P.; Lukasiewicz, T. An ontology-based deep learning approach for knowledge graph completion with fresh entities. In International Symposium on Distributed Computing and Artificial Intelligence; Springer: Cham, Switzerland, 2019; pp. 125–133.
35. Paulheim, H. Knowledge graph refinement: A survey of approaches and evaluation methods. Semant. Web 2017, 8, 489–508. [CrossRef]
36. Uschold, M. Ontologies Principles, Methods and Applications. Knowl. Eng. Rev. 1996, 11, 93–155. [CrossRef]
37. Zwass, V. Expert System. Encyclopedia Britannica. 2016. Available online: https://www.britannica.com/technology/expert-system (accessed on 12 June 2021).
38. Xia, M. Implementation of an animal recognition expert system by Prolog. J. Chengdu Univ. Inf. Technol. 2003, 14, 245.
39. Bo, X.; Yong, X.; Liang, J. CN-DBpedia: A Never-Ending Chinese Knowledge Extraction System. Ind. Eng. Other Appl. Appl. Intell. Syst. 2017, 52, 428–438.
40. Wu, W.; Li, H.; Wang, H. Probase: A probabilistic taxonomy for text understanding. Manag. Data 2012, 119, 481–492.