applied sciences

MDPI

Article

An Interpretable Fake News Detection Method Based on Commonsense Knowledge Graph

Xiang Gao [1], Weiqing Chen [1], Liangyu Lu [2], Ying Cui [1], Xiang Dai [1], Lican Dai [1], Kan Wang [1], Jing Shen [2], Yue Wang [2], Shengze Wang [2], Zihan Yu [2] and Haibo Liu [2,*]

[1] The Southwest Institute of Electronic Technology, Chengdu 610036, China; gaoxiang9156@163.com (X.G.)
[2] College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China
* Correspondence: liuhaibo@hrbeu.edu.cn

Abstract: Existing deep learning-based methods for detecting fake news are uninterpretable, and they do not use external knowledge related to the news. As a result, the authors of the paper propose a graph matching-based approach combined with external knowledge to detect fake news. The approach focuses on extracting commonsense knowledge from news texts through knowledge extraction, extracting background knowledge related to news content from a commonsense knowledge graph through entity extraction and entity disambiguation, using external knowledge as evidence for news identification, and interpreting the final identification results through such evidence. To achieve the identification of fake news containing commonsense errors, the algorithm uses random walks graph matching and compares the commonsense knowledge embedded in the news content with the relevant external knowledge in the commonsense knowledge graph. The news is then discriminated as true or false based on the results of the comparative analysis. From the experimental results, the method can achieve 91.07%, 85.00%, and 89.47% accuracy, precision, and recall rates, respectively, in the task of identifying fake news containing commonsense errors.

Keywords: fake news detection; random walks; graph matching

Citation: Gao, X.; Chen, W.; Lu, L.; Cui, Y.; Dai, X.; Dai, L.; Wang, K.; Shen, J.; Wang, Y.; Wang, S.; et al. An Interpretable Fake News Detection Method Based on Commonsense Knowledge Graph. *Appl. Sci.* **2023**, *13*, 6680. https://doi.org/10.3390/app13116680

Academic Editor: Daeho Lee

Received: 29 March 2023
Revised: 13 May 2023
Accepted: 28 May 2023
Published: 30 May 2023

1. Introduction

With the rapid development of the mobile internet, false news can have a huge impact in a short period of time, so it is becoming increasingly challenging to identify fake news effectively and accurately. In recent years, one common method is to store external knowledge in a knowledge graph and combine external knowledge with fake news detection. External knowledge usually contains rich semantic information and objective facts. Using knowledge helps to understand the news content, and comparing the news content with external knowledge can effectively judge the authenticity of the news. Currently, existing methods that combine external knowledge with fake news detection mainly use attention mechanisms to fuse external knowledge. Ref. [1] proposes a behavioral model that can reduce the corresponding type of misinformation by identifying factors that enhance users' behavior in identifying fake news on social media. Ref. [2] proposes a blockchain-based hybrid architecture for detecting and preventing false sensing activities in mobile crowdsensing (MCS). Ref. [3] examines the relationship between exposure to and trust in COVID-19 news and information sources and belief in COVID-19 myths and false information, as well as critical verification practices before posting on social media. Ref. [4] uses the Mel-frequency Cepstral Coefficients (MFCCs) technique to obtain the most useful information from the audio while using machine learning and deep learning methods to identify deeply falsified audio. Ref. [5] explores the effects of fake news on consumer perceptions, attitudes, and behaviors. One method is to integrate visual and textual information into text representation through attention mechanisms to help the model understand the news text content [6]. Some methods use named entity recognition methods

to align entities in the text with entities in the knowledge graph and use designed multi-head attention methods to fuse news text information, entity information, and entity context information to obtain semantically rich news text modeling [7]. Some methods use graph neural networks to fuse external knowledge. Some methods construct a heterogeneous information network that includes textual information, image information, and entity information from the knowledge graph. They use Graph Convolutional Networks (GCN) to fuse information from different modalities and obtain news representations that integrate text information with external world knowledge and visual information [8]. However, these methods directly integrate external knowledge into news content rather than comparing external knowledge with news content. Although the representational learning ability of deep learning can access the underlying spatial features of things, these features are continuous vectors that are difficult for humans to understand, and humans can only understand semantic scenes. At the same time, training inference using deep learning methods takes longer than traditional methods and requires higher hardware requirements.

To address these challenges, this paper proposes a method for identifying fake news based on the random walks graph matching algorithm [9]. Graph matching algorithms come in many different types, among which a simple and effective method is using spectral relaxation to approximate Integer Quadratic Programming (IQP) [10]. The method computes the leading eigenvectors of symmetric nonnegative affinity matrices. Spectral Matching (SM) ignores the integer constraints in the relaxation step and induces them by greedy methods in the discrete step. Some methods extend SM to Spectral Matching with Affine Constraints (SMAC) by introducing affine constraints in the spectral decomposition that encodes one-to-one matching constraints [11]. The graph matching algorithm used in this paper is related to the IQP formulation associated with the SM algorithm and the SMAC algorithm, but the approach in this paper is from the perspective of random wandering. In this paper, knowledge in the news is extracted through different methods of knowledge extraction. The extracted knowledge is then compared with external knowledge using the random walks graph matching algorithm to achieve an interpretable fake news detection task.

The main contribution of this paper can be divided into three parts:

1. A method for extracting commonsense knowledge contained in news is proposed. This method is based on the Lexical Analysis of Chinese (LAC) framework [12] and the Universal Information Extraction (UIE) framework [13]. The extracted knowledge serves as the basis for the subsequent use of graph matching algorithms for fake news detection.
2. An interpretable model for fake news detection is proposed based on the random walks graph matching algorithm. Using this algorithm, the commonsense knowledge embedded in the news is compared with the relevant knowledge in the knowledge graph. The truth or falsity of the news is determined by comparing the conflicts of entities and attributes among the knowledge triads in the matching results. This approach leads to an interpretable fake news detection method.
3. The proposed method is evaluated on a fake news dataset containing commonsense errors. The experimental results demonstrate that our method outperforms the baseline method.

2. Related Work

2.1. Graph Matching

Graph matching is a process of evaluating the similarity between two graphs. The task of graph matching is to correspond the nodes in the two graphs one by one. This task requires not only a high similarity of matched nodes but also a high similarity of corresponding edges. There are two approaches to graph matching: exact and inexact graph matching. Exact graph matching requires each node and corresponding edge to be exactly matched with another graph with the highest similarity. On the other hand, inexact

graph matching allows two graphs to be somewhat different but requires that the similarity of nodes and edges be as high as possible.

Exact graph matching requires that each node and corresponding edge be exactly matched with another graph with the highest similarity. However, this approach has great limitations in the solution process, and there are few such cases in practical applications. As a result, there are few studies related to exact graph matching. On the other hand, inexact graph matching transforms the problem into an optimization problem that maximizes the sum of node similarity and edge similarity. This approach allows two graphs to be somewhat different but requires that the similarity of nodes and edges be as high as possible.

Spectral matching methods are based on the fact that the eigenvalues of a matrix remain constant regardless of the arrangement of rows and columns [13]. This method is usually effective, but it is more sensitive to noise [10]. One approach to spectral relaxation is to approximate Integer Quadratic Programming (IQP). The method computes the leading eigenvectors of symmetric nonnegative affinity matrices. The Spectral Matching (SM) technique ignores the integer constraints in the relaxation step and induces them by greedy methods in the discrete step. Some methods extend SM to Spectral Matching with Affine Constraints (SMAC) by introducing affine constraints in the spectral decomposition that encodes one-to-one matching constraints [11]. Semidefinite programming (SDP) is a general tool for solving combinatorial problems [14–16]. It relaxes the non-convex constraint to a new constraint, which is obtained through a winner-take-all strategy [16] or a randomization algorithm [15]. However, in practical applications, it can be computationally costly. To address this issue, ref. [17] proposed a maximum likelihood estimation method for the assignment matrix as a probabilistic interpretation of the spectral matching algorithm. Ref. [18] uses a random walks-based approach inspired by PageRank. Ref. [19] proposed a convex relative entropy error from probabilistic interpretation to the hypergraph matching problem.

2.2. Knowledge Graph

A knowledge graph is designed to describe the entities that exist in the real world and the relationships between them. Its initial purpose was to improve the capability of search engines and the search quality and experience of users. With the development and application of artificial intelligence, the knowledge graph has become one of the key technologies and is widely used in news recommendation [20], fact-checking [21], and intelligent Q&A [22]. Moreover, knowledge graphs can also be used for entity linking. Instead of retrieving evidence directly from news content to detect fake news, some methods use entity information extracted from entity links as auxiliary information to improve prediction accuracy [23,24]. This motivates researchers to consider external knowledge to improve detection capability.

3. Extraction of Commonsense Knowledge

Commonsense knowledge extraction is a process that extracts commonsense knowledge units from a text. These knowledge units consist of three elements: entities, relationships, and attributes. Based on these elements, a series of high-quality factual expressions are formed, which lay the foundation for the construction of the upper pattern layer [25].

3.1. Bi-GRU Based Entity Extraction

Entity extraction [26] is to automatically identify named entities from a commonsense text corpus. Since entities are the most basic elements in a knowledge graph, their extraction completeness, accuracy, recall rate, and so on will directly affect the quality of the knowledge base. Therefore, entity extraction is the most basic and key step in knowledge extraction. The entity extraction model used in this paper mainly adopts the Bi-GRU-CRF architecture [12]. The specific details are described next.

The network structure of the LAC model is shown in Figure 1. Bidirectional GRU (Bi-GRU) is an extension of GRU, which is suitable for lexical analysis tasks with input as a whole sentence. Reverse GRU combines with forward GRU to form a Bi-GRU layer. These two GRUs accept the same input, but train in different directions, and connect their results as output.

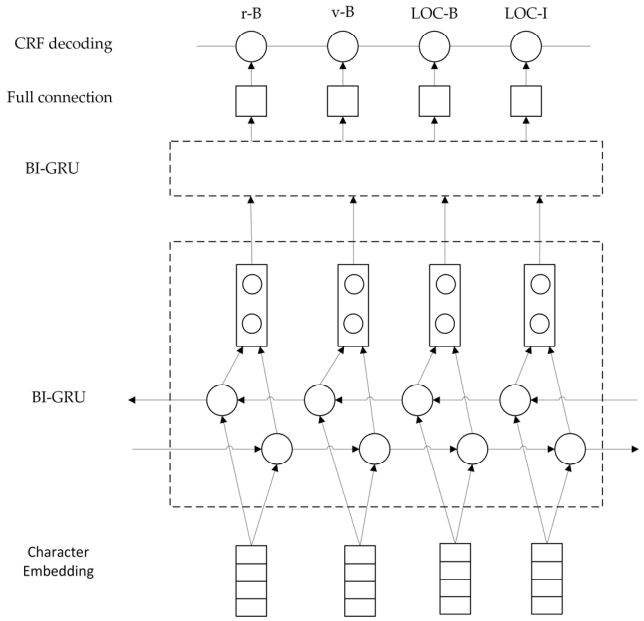

Figure 1. Illustration of Bi-GRU-CRF network.

In order to more effectively represent some functions and models' variable-length dependency relationships, two Bi-GRUs are stacked together in the LAC model to form a deep network to improve representation ability. As shown in the model structure diagram, a Conditional Random Field (CRF) [27] layer is used at the top of the GRU structure to decode the final label sequence jointly. CRF is a discriminative probability model, which is often used for tagging or analyzing sequence data. Its input is provided by a full connection layer, which converts the output of the top Bi-GRU layer into an L-dimensional vector. L represents the number of all possible labels. At the same time, hard constraints are applied to the decoding process to emphasize the dependency relationship between the output labels. Any sequence that does not conform to IOB2 transmission rules will not be accepted.

The LAC model takes the initial character sequence as the input of the model. Given a character sequence $\{c_1, c_2 \ldots, c_T\}$, each character, c_i, in the vocabulary, V, is projected to a real-valued vector, $\mathrm{e}(c_i)$, by looking up a table. Then, with the embedding of characters [28] as input, a deep GRU neural network is built to learn the structural information of a given sentence. The relevant formulas for the Bi-GRU layer are as follows:

$$u_t = \sigma_g(W_{ux}x_t + W_{uh}h_{t-1} + b_u) \tag{1}$$

$$r_t = \sigma_g(W_{rx}x_t + W_{rh}h_{t-1} + b_r) \tag{2}$$

$$\widetilde{h_t} = \sigma_c(W_{cx}x_t + W_{ch}(r_t \odot h_{t-1}) + b_c) \tag{3}$$

$$h_t = (1 - u_t) \odot h_{t-1} + u_t \odot \widetilde{h_t} \tag{4}$$

$\odot$ is the element-wise product of vectors, σ_g is the activation function, u_t is the update gate, r_t is the reset gate, and σ_c is the activation function of the hidden layer.

The CRF layer learns the conditional probability, $p(y|h)$, $h = \{h_1, h_2 \ldots, h_T\}$ is the representation sequence generated by the topmost Bi-GRU layer, and $\{y_1, y_2 \ldots, y_T\}$ is the label sequence.

The probability model of linear-chain CRF defines a conditional probability family, $p(y|h;t,s)$, over all possible label sequences, y given h, which has the following form:

$$p(y|h;t,s) = \frac{\prod_{i=1}^{T} \varphi_i(y_{i-1}, y_i, h)}{\sum_{y' \in \gamma(h)} \prod_{i=1}^{T} \varphi_i(y'_{i-1}, y'_i, h)} \quad (5)$$

and $\gamma(h)$ represent all possible label sequences. $\varphi_i(y_{i-1}, y_i, h) = \exp(\sum_{i=1}^{T} t(y_{i-1}, y_i, h) + s(y_i, h))$. t is the transition probability from $y_i - 1$ to y_i given the input sequence h. s is the output of a linear function implicitly defined by a fully connected layer, which transforms the output of the topmost Bi-RNN at step i into a score for y_i.

The CRF layer is trained using maximum conditional likelihood estimation, and the log-likelihood is:

$$L(t,s) = \sum_i log\ p(y|h;t,s) \quad (6)$$

In the decoding process, we only need to search for a sequence that maximizes the conditional probability $P(y|h)$ in $Y(h)$ by using the Viterbi algorithm:

$$y^* = argmax_{y \in \gamma(h)} p(y|h) \quad (7)$$

In the decoding process, constraints are imposed to ensure that the results conform to the IOB2 format. At the same time, the label sequence needs to satisfy certain conditions:

1. The label of the first character of a sentence cannot be an I-tag.
2. The previous tag of each I-tag can only be a B-tag or an I-tag. For example, the tag before "LOC-I" can only be "LOC-B" or "LOC-I".

3.2. Template-Based Knowledge Extraction

Knowledge extraction is an important step in building large-scale knowledge graphs, and it is an important technology for achieving the automated construction of large-scale knowledge graphs. Its purpose is to extract knowledge from data from different sources and structures and store it in knowledge graphs. Template-based knowledge extraction has the advantages of high accuracy, strong flexibility, and applicability to specific domains by defining extraction rules.

3.2.1. Knowledge Extraction Based on a Relational Word Dictionary

Template matching based on a relational word dictionary studies the sentence structure patterns that often appear in news texts, abstracts them into a template, and matches them by establishing a relational word dictionary.

The conceived general sentence structure pattern is p, which is usually composed of three parts: n, r, and v, which correspond to entity 1, relation, and entity 2 in triad, respectively. Among them, n and v need to have certain identification characteristics. Then establish a dictionary of relational words for the relations that p can identify. Firstly, the text is segmented, and then the relative words are screened out by using the AC automaton algorithm, and finally the knowledge is extracted from the text according to the identification rules in p. The process of knowledge extraction based on the relational word dictionary template matching method is shown in Figure 2.

3.2.2. Knowledge Extraction Algorithm Process

Regular expressions are often used to retrieve content from text according to certain rule patterns. With the help of artificially constructed regular expressions, information from a text can be extracted. According to different entity feature structures, regular expression extraction templates for different types of information can be efficiently constructed.

In the process of knowledge extraction based on regular expressions, the focus is on constructing regular expression templates and storing common sentence patterns as JSON files using regular expressions constructed for text content matching. Part of a regular expression template diagram is shown in Figure 3.

After the regular expression template is constructed, the input text is processed, and the entities in the text are extracted using the Bi-GRU-based entity extraction method. Then match with the regular expression template, extract and save the knowledge contained in the text in the form of triples. The process of knowledge extraction based on the regular expression template matching method is shown in Figure 4.

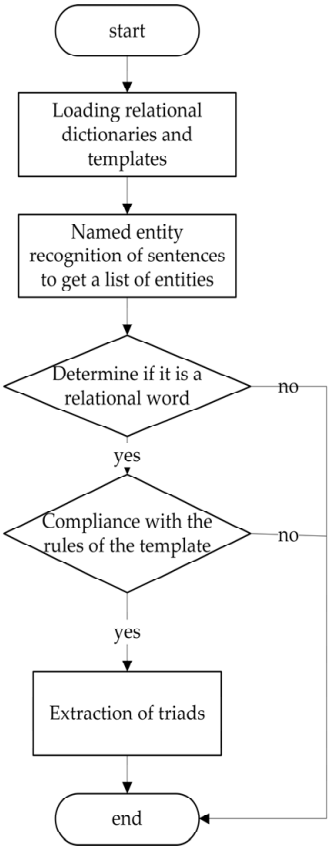

Figure 2. Knowledge extraction process based on relational word dictionary.

```
"PER":
{
  "name": "(?:^.*)(?<= O([a-zA-Z'@]*)(?=))\\b",
  "name": "(?<=english?name)[a-zA-Z'@·]*\\b",
  "alias": "(?<=[alias][name])\\w+?\\b",
  "ethnicity": "\\b\\w+?ethnicity\\b",
  "gender": "\\bwoman\lb",
  "height": "(?<=height)[\\d\\.]+?c?m",
  "weight": "(?<=weight)[\\d\\.]+?k?g",
  "birthday": "\\d*year\\d*month\\d*day(?=birth)"
}
```

Figure 3. Illustration of regular expressions.

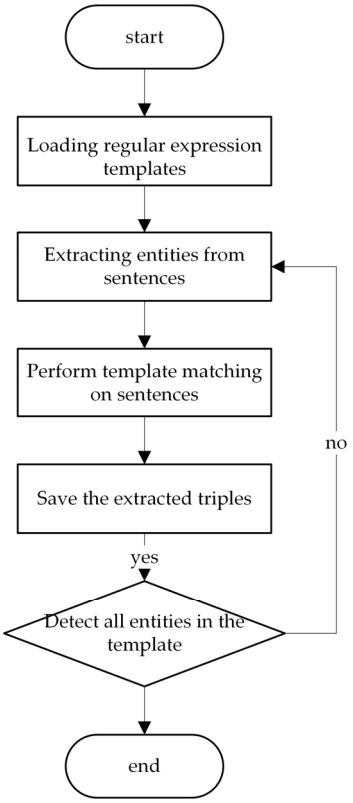

Figure 4. Knowledge extraction process based on regular expression templates.

3.3. Knowledge Extraction Based on UIE

UIE (Universal Information Extraction) is a unified framework for general information extraction that implements entity extraction, relation extraction, event extraction, sentiment analysis, and other tasks with a unified model, as shown in Figure 5. These different types of tasks are described and enable good transfer and generalization capabilities between different tasks.

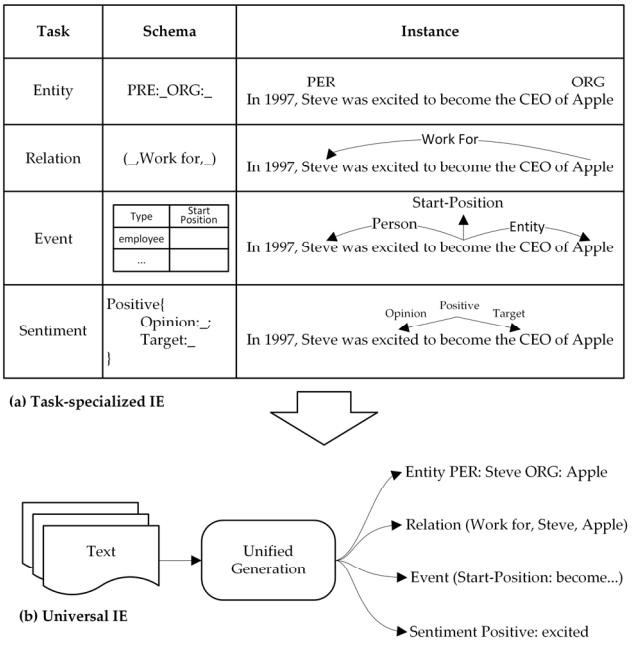

Figure 5. (**a**) Comparison of different types of tasks, (**b**) a unified modeling schematic.

UIE adapts to the target extraction by using a pattern-based prompt mechanism to achieve large-scale text-structured extraction capabilities. In order to model different structures of information extraction structures, a structural extraction language (SEL) is used, which can effectively encode different information extraction structures into a unified expression form. At the same time, a structural schema instructor (SSI), based on a pattern prompt mechanism, is used to adaptively generate target structures for different information extraction tasks.

Assuming that the input uses a predefined extraction pattern, s, and a piece of text, x, s represents the extracted model, and x represents the source of text to be extracted. The information extraction structure generation model is decomposed into two atomic operations: locating (spotting) and associating (associating). SEL encodes different sources of information extraction structures through spotting–associating structures. Each SEL expression contains three types of semantic units: (1) Spot Name represents a specific piece of information whose type spot name exists in the source text; (2) Asso Name represents that there exists a specific piece of information in the source text that is related to the upper-layer spot information in the structure; (3) Info Span represents the text block corresponding to the specific identified or associated piece of information in the source text. No matter what task it is, it can be expressed. For example, entity extraction can be viewed as locating the entity type text blocks corresponding to mentions, and event extraction can be expressed as finding trigger word text blocks with event types.

The structural schema instructor (SSI), based on a pattern-prompt mechanism, can control what information we need to discover and its related information. According to the UIE model structure diagram shown in Figure 6, the input y is composed of a text sequence x and a structural schema instructor s; the output is the structured extract language formalized: $x = x_1, x_2, \ldots, x_{|x|}$ represents the text sequence; $s = s_1, s_2, \ldots, s_{|s|}$ represents the structural schema instructor; $y = y_1, y_2, \ldots, y_{|y|}$ represents the SEL sequence where y can be converted into extracted records after calculation s and x as shown in formula (8).

$$\begin{aligned} s \oplus x &= \left[s_1, s_2, \ldots, s_{|s|}, x_1, \ldots, x_{|x|}\right] \\ &= \left[[spot], \ldots [spot] \ldots [asso], \ldots [asso] \ldots, [text], x_1, .., x_{|x|}\right] \end{aligned} \tag{8}$$

From the above formula, we know special symbols ([spot], [asso], [text]) are added before each Spot Name, Asso Name, and input text sequence template. All the tokens are concatenated together and placed before the original text sequence. For the given original texts, s, and mode indicator, x, UIE calculates a hidden vector for each token, as shown in formula (9).

$$\mathrm{H} = \mathrm{Encoder}(\mathrm{s}_1, \mathrm{s}_2, \ldots, \mathrm{s}_{|\mathrm{s}|}, \mathrm{x}_1, \ldots, \mathrm{x}_{|\mathrm{x}|}) \tag{9}$$

Here, Encoder indicates Transformer Encoder; then, UIE uses a self-regressive way to decode input texts into linearized SEL, as shown in formula (10):

$$\mathrm{y_i}, \mathrm{h_i^d} = \mathrm{Decoder}(\left[\mathrm{H}; \mathrm{h_1^d}, \ldots, \mathrm{h_{i-1}^d}\right]) \tag{10}$$

In the i-th step of decoding, UIE generates the i-th token y_i in the SEL sequence and the decoder state h_i^d. Decoder() means Transformer Decoder, which is used to predict the conditional probability of token y_i. Finally, Decoder() completes the prediction when the end symbol <eos> is output, and then converts the predicted SEL expression into an extracted information record. Compared with other information extraction tasks, the information extraction used in the UIE framework regards tags as natural language tokens, while other models regard them as special symbols. Generating labels and structures through linguistics can effectively transfer knowledge from individual pre-trained models.

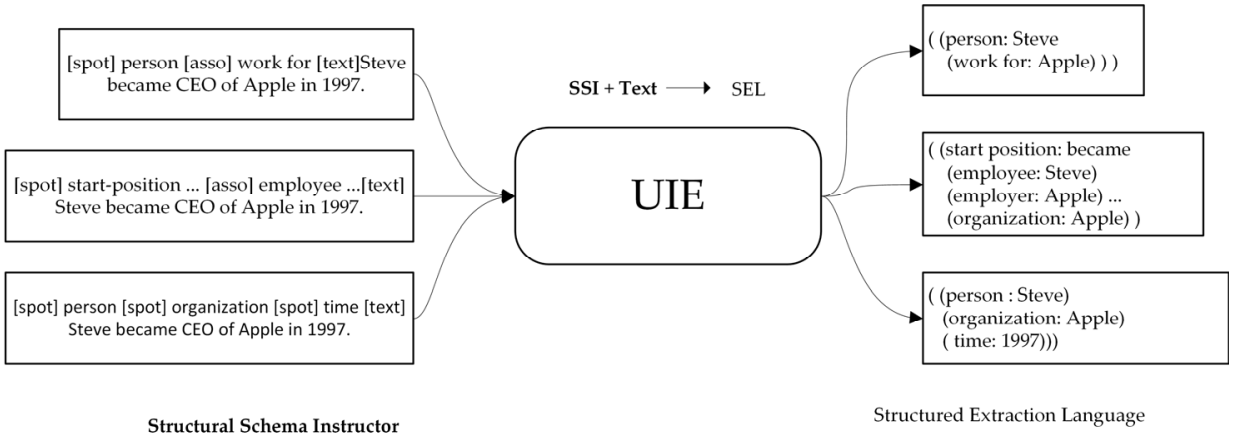

Figure 6. UIE model structure diagram.

When using the UIE framework, a list of target query entities is first created through filtering nodes from the Baidu Encyclopedia knowledge graph, and a corresponding AC automaton model is built. After getting the text data and performing named entity recognition, there are two ways to determine if the extracted entity is the target entity. One is to find the target entity in the text directly by AC automaton.

The other method first converts the target entity list into a word vector table., then compares the word vector of the extracted entity with the word vector of the target entity to determine whether they have the same semantic meaning. If the extracted entity is in the target entity list, then the entity is added to the subsequent process. In this way, the fuzzy extraction of entities can be achieved, thus improving the generalization ability of the model. After extracting the entity, the schema related to the entity is extracted from the Baidu Encyclopedia Knowledge Graph. An example diagram of the schema in text is shown in Figure 7. Then, the knowledge in the text can be extracted through the UIE framework. The whole process is shown in Figure 8.

pristine mountain forest
areajourney
duration of the
flightpower consumption
reflective area
captain of the
submarineattack method
first flight timetime of the
shootingdate of issue
age
type
power consumption
commander-in-chief of the
paradenumber of
combatants
office functions
executive mayordirector
quality
refractive index

Figure 7. Example diagram of schema in text.

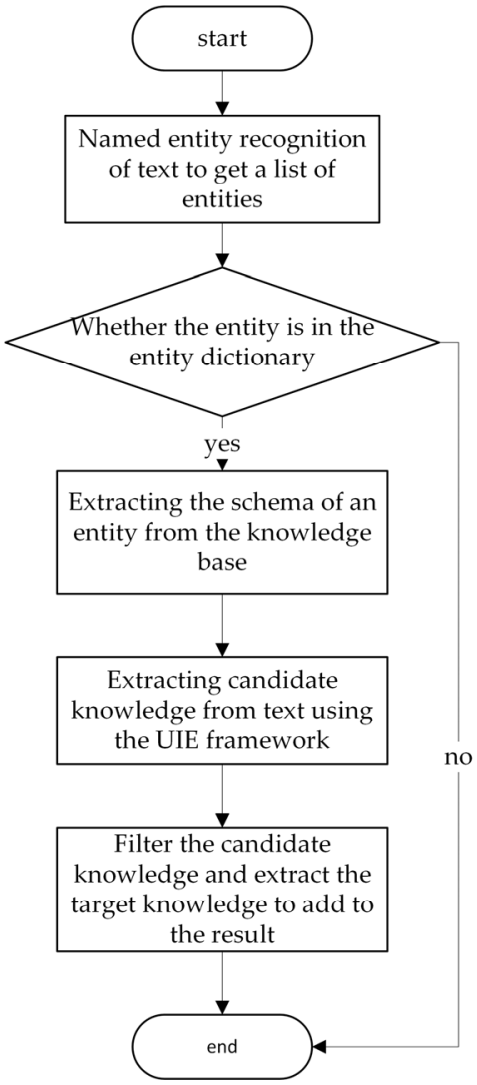

Figure 8. Knowledge extraction process based on UIE framework.

4. Fake News Identification Based on Random Walk Graph Matching

Knowledge graphs contain a large amount of external knowledge, and external knowledge contains rich semantic information, which can help us better understand news content. At the same time, external knowledge contains many objective facts, which can be compared with news content, thus identifying the falsehoods in fake news. This paper transforms the problem of fake news identification into a problem of non-exact matching between knowledge graphs about news and commonsense knowledge graphs. Then, according to the matching results, entity attribute conflicts are calculated, and according to the conflict situation, the news is identified as true or false [29]. In order to achieve non-exact matching of knowledge graphs, a graph matching algorithm based on random walk is implemented to solve the problem of knowledge graph matching [30].

4.1. Principle of Random Wandering Graph Matching Algorithm

Graph matching refers to using the similarity information of the graph structure to find node matching (first-order matching) relationships or edge matching (second-order matching) relationships between graph structures (as shown in Figure 9, nodes of the same color indicate matching nodes.), which involves graph matching research in different fields such as computer vision and pattern recognition [31].

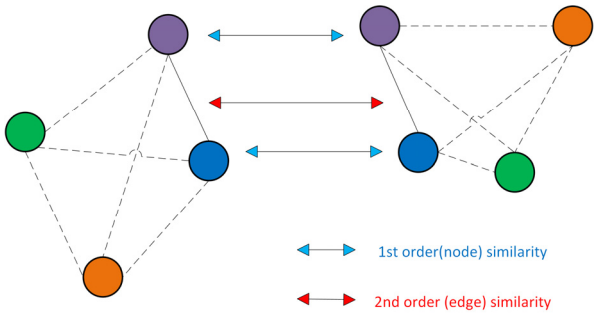

Figure 9. Graph matching schematic.

Because edges also contain information in knowledge graphs, this project needs to consider both node matching and edge matching for graph matching. The similarity matrix, W, is used to represent the matching relationship. The similarity matrix, W, contains both first-order node similarity information and second-order edge similarity information, as follows:

$$\mathrm{W} = \begin{bmatrix} w_{a_1b_1,a_1b_1} & w_{a_1b_1,a_1b_2} & \cdots \\ w_{a_1b_2,a_1b_1} & w_{a_1b_2,a_1b_2} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

In the problem shown in Figure 8, the scale of the similarity matrix is 16×16. The diagonal elements of the similarity matrix contain node-to-node similarity information, such as $w_{a_ib_k,a_ib_k}$, represents the similarity between node a_i in graph a and node b_k in graph (b). Non-diagonal elements contain edge-to-edge similarity information, such as $w_{a_ib_k,a_jb_l}$ which represents the similarity between the edge composed of nodes a_i and a_j in graph a and the edge composed of nodes b_k and b_l in graph (b). Based on the similarity matrix, W, and match matrix, X,the problem of the graph match can be modeled as:

$$x^* = arg\,max\left(x^T W x\right) \text{ s.t. } x \in \left[0,1\right]^{n^P n^Q} \tag{11}$$

to calculate the match result that maximizes the similarity. Mathematically, formula (11) is an NP-hard quadratic assignment problem, which cannot find a global optimal solution within polynomial time.

Random walk [32,33] is an efficient approximation algorithm for graph matching. The random walk algorithm constructs several random walkers. The random walker is initialized from a certain node, and then randomly visits an adjacent node of the current node in each random walk step. There are two graphs waiting for matching in a problem of a graph match, but random walk only performs on a single one. Therefore, the graph matching problem has to be transformed into the form of a single graph–accompaniment graph in order to apply the random walk algorithm to the graph matching problem. As shown in Figure 10, consider the case of two nodes (1, 2) matching three nodes (a, b, c). The two graph structures in (a) represent the original graph matching problem. The diagram in (b) is the accompanying diagram. In the graph matching problem, the node-to-node correspondence (orange dashed double arrows) translates to the nodes in the accompanying diagram; for example, the matching relationship between node 1 and node a is transformed into node 1a in the accompanying graph. The similarity information between sides (blue dashed double arrows) is transformed into edges in the accompanying graph; for example, the similarity between edge 12 and edge ab is transformed into the right edge 1a–2b in the accompanying graph. The concomitant graph is an undirected weight graph, but, in this project, it is designed as a directed graph because the knowledge graph data has the distinction between entities and attributes. With the random walk algorithm, weights can be computed for each node of the accompanying graph. The graph matching problem is then transformed into the problem of finding the number of nodes with the maximum weight in the accompanying graph [9].

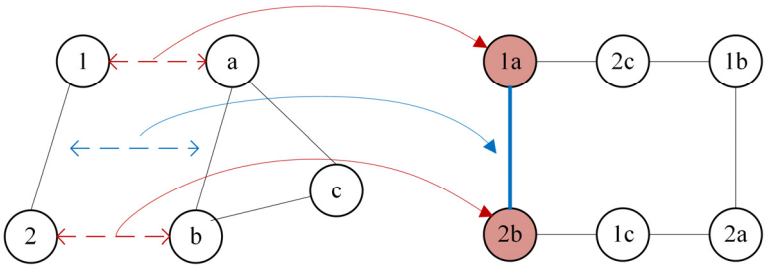

Figure 10. Graph matching schematic.

To facilitate uniform processing, absorbing nodes are added to the accompanying graph so that the out-degree of all the other nodes is equal. To add additional matching constraint information, the weight of each node is reassigned during the random wandering (allowing jump wandering).

4.2. Algorithm Implementation

4.2.1. Overall Algorithm Implementation

Figure 11 shows the process of identifying fake news based on the graph matching algorithm of random walks. Firstly, entities are extracted from the news text, and commonsense knowledge associated with the entities is extracted from the Baidu Encyclopedia knowledge graph to construct a commonsense knowledge subgraph. Then, the similarity matrix is calculated between the commonsense knowledge subgraph and the commonsense knowledge graph extracted from the news, and the similarity matrix is constructed based on the semantic similarity between the texts. Finally, the similarity matrix is input into the random walks-based graph matching algorithm to obtain the matching results of the two graphs. After obtaining the matching results, the semantic similarity between the matched head nodes, edges, and tail nodes is calculated to determine whether there is a conflict. If there is a conflict, the news is considered false, and an explanation is provided according to the specific conflict type. The conflict types are divided into three categories: a head entity conflict, a relationship conflict, and a tail entity conflict.

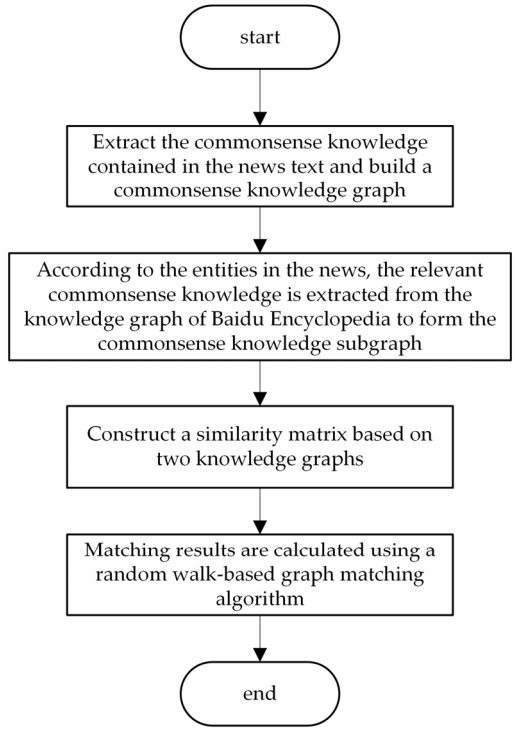

Figure 11. The process of matching two knowledge graphs.

4.2.2. Extraction of Commonsense Subgraph

The graph matching algorithm uses the similarity matrix of two graphs as input, and the size of the similarity matrix is $n^p n^q \times n^p n^q$, where n^p is the number of nodes in the intelligence graph, and n^q is the number of nodes in the commonsense graph. Too large a commonsense knowledge map can lead to difficulties in calculating the similarity matrix. Therefore, this project filters the relevant commonsense knowledge subgraph from the commonsense knowledge graph by the extracted knowledge graph in order to facilitate the construction of the similarity matrix [34].

The specific filtering strategy is as follows:

1. All the nodes in the extracted knowledge graph are treated as entities, the relationships of the entity nodes are queried in the general knowledge graph, and all the relationships and the nodes involved are recorded.
2. The nodes and relationships of records are deduplicated.
3. The entity and attribute nodes with the same value are merged, as is the relationship of two nodes with the same value.
4. The processed nodes and relations are constructed as a commonsense knowledge subgraph.

4.2.3. Construction of Similarity Matrix

The similarity calculation criterion function in this project has two designs, a character-based similarity calculation criterion function and a similarity calculation criterion function based on the word vectors generated by the Ernie model. The character-based similarity calculation criterion function uses the Sorensen–Dice similarity coefficient, which is a way to calculate the similarity between simple sets, as shown in Equation (12).

$$s = \frac{2|X \cap Y|}{|X| + |Y|} \tag{12}$$

where s is the similarity, X and Y denote the set of characters corresponding to the two strings, and $|X|$ denotes the number of characters in the set, X.

The character-based similarity calculation criterion function has the advantage of being simple, intuitive, and fast, but it does not consider the semantic relationship between strings. Therefore, we consider using the word vector generated by the Ernie model for the similarity calculation, which outputs the input string as a 768-dimensional word vector, after which we can calculate the similarity of two-word vectors. The common criterion functions used to calculate the vector similarity are Euclidean distance and cosine similarity, and it is found that the cosine similarity will output a similarity close to 1 for any two-string word vectors.

The similarity matrix, W, contains both the first-order node similarity information and the second-order edge similarity information. Each row and column in W represents a node in the concomitant graph, and each column (row) from left (top) to right (bottom) in W corresponds to the concomitant graph node composed of the first node in the intelligence map, P, and the first node in the general knowledge graph, Q, the concomitant graph node composed of the second node in the intelligence map, P, and the first node in the general knowledge map, Q, as shown below:

$$\mathrm{W} = \begin{bmatrix} w_{a_0b_0,a_0b_0} & w_{a_0b_0,a_0b_1} & \cdots \\ w_{a_0b_1,a_0b_0} & w_{a_0b_1,a_0b_1} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

The diagonal elements of the similarity matrix contain the node-to-node similarity information, such as $\mathrm{W}[i \times n^p + j][i \times n^p + j]$ denotes the similarity between node a_j in the intelligence map, P, and node b_i in the commonsense knowledge map, Q. Its index in the matrix is $w_{a_jb_i,a_lb_k}$, where n^p is the number of nodes in intelligence map P. The non-diagonal elements contain the similarity information between edges, such as $w_{a_jb_i,a_lb_k}$, denotes the

similarity between the edges composed of nodes a_j and a_l in the intelligence map, P, and the edges composed of nodes b_i and b_k in the commonsense knowledge map, Q. Its index in the matrix is $W[i \times n^p + j][k \times n^p + l]$.

4.2.4. Reasoning Explanation of Identification Results

The white-box inference technique employs knowledge graph matching technology to reason and identify authenticity. This approach can directly uncover the reasons for detecting forged intelligence and differences in entity attributes.

The algorithm based on random walks graph matching matches the commonsense knowledge embedded in the news content with the relevant commonsense knowledge in the knowledge graph of the Baidu Encyclopedia. It then calculates the entity or attribute conflicts between them, based on the matching results between the commonsense knowledge graph extracted from the news and the commonsense knowledge subgraph of the Baidu Encyclopedia. By analyzing the cases of conflict, the fake news is identified and interpreted accordingly, and the conflicts are divided into head node conflict, edge conflict, and tail node conflict. The flow of inference and interpretation is shown in Figure 12.

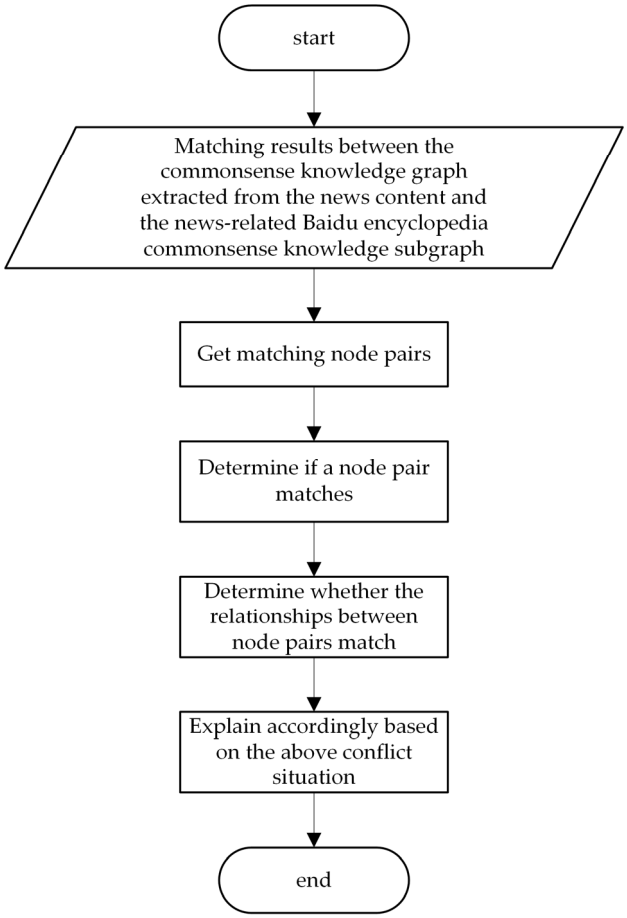

Figure 12. The process of matching two knowledge graphs.

5. Experiment and Analysis

5.1. Experimental Result Comparison Analysis

The experimental environment is based on the Windows operating system; the development language is Python and Cypher; the knowledge extraction model framework uses the UIE framework; the entity extraction model framework uses the LAC framework. The experimental data mainly comes from relevant news crawled from the Internet using crawler scripts, as well as various fake news rumor websites, and also uses fake news generated based on Ernie.

The LAC model consists of two Bi-GRU layers consisting of two forward GRUs and two reverse GRUs, and the hidden layer has a dimension of 256. The weight matrix in each layer of the model is first initialized randomly. For the Bi-GRU layer, sigmoid is used as the gating activation function, and *tan h* is used as the activation function of the hidden layer. During the model training, parameter optimization is performed using stochastic gradient descent. The learning rate of the base is 1e-3. The learning rate of the embedding layer is 5e-3. The size of the batch size is set to 250.

5.2. Experimental Result Comparison Analysis

The fake news detection task dataset used in this paper mainly has two parts: obtained by crawling various fake news websites and news websites using crawlers and obtained by using a fake news generation method based on Ernie, finally obtaining a fake news dataset with 40,000 entries.

At present, most methods for fake news detection are based on deep learning; although deep learning's representation learning ability can obtain objects' underlying spatial features, these features are obtained through a black box and are a continuous vector; humans cannot understand them at all. At the same time, most deep learning methods only rely on contextual relationships in news text for identification, without fully utilizing commonsense knowledge contained in the news text. In response to these problems, a comparative experiment was designed based on the interpretability of the method, and the three graph matching methods of RRWM, SM, and SMAC were compared.. The algorithm based on random wandering graph matching used in this paper is also compared with several benchmark methods, namely SVM-TS, CNN, EANN, and GRU. The evaluation metrics used in this experiment are accuracy, precision, and recall.

The example used in the experiments is described below: "On Nov. 10, local time, the U.S. Department of Defense announced another $400 million worth of military assistance to Ukraine to meet its critical security and defense needs. According to the statement, the assistance program includes weapons such as the Hawk air defense missile. The Hawk air defense missiles, which cost up to $250,000 each to develop, have an effective range of 41 km and can be used for medium-range, low- and medium-altitude defense. Since President Biden took office, the U.S. has pledged more than $19.3 billion in military assistance to Ukraine." According to the knowledge extraction method introduced in the previous section, the background knowledge triads extracted from them are: ['United States', 'President', 'Biden'], ['Biden', 'Nationality', 'United States'], ['"Hawk" anti-aircraft missile', 'Effective range', '41 km'], ['"Hawk" anti-aircraft missile', 'Development cost', '$250,000']. According to the analysis based on the matching results, the news mentions that the development cost of the Hawk anti-aircraft missile is USD300,000, which conflicts with the knowledge in the commonsense knowledge graph, so the news is considered to be fake news, and the error lies in the fact that the development cost of the Hawk anti-aircraft missile does not match the actual one. Thus, the interpretable recognition of false news texts is completed.

From the experimental results in Table 1, it can be seen that the effect of the RRWM algorithm is better than the SM algorithm and the SMAC algorithm in terms of the evaluation indexes such as accuracy, correctness, and recall. This is because in the RRWM algorithm, compared with the SM algorithm and SMAC algorithm, additional matching constraint information is added in each step of the random wandering, and the weight of each node is reassigned during the random wandering, which makes the matching accuracy significantly improved.

According to the experimental results in Table 2, it can be seen that the fake news recognition method based on the random wandering graph matching algorithm used in this experiment outperforms several benchmark methods based on deep learning in terms of accuracy, correctness, and recall while possessing interpretability and being understandable to humans. CNNs outperform SVM-TS because they can only analyze the local semantics of the news text and cannot fully exploit the features of the full text of the news, sequences

of words with different lengths, and cannot better consider the global semantics. The fake news recognition method based on the random wandering graph matching algorithm extracts the commonsense knowledge contained in the news text, and, by comparing it with the commonsense knowledge map of the Baidu Encyclopedia, it can more effectively identify the knowledge that does not conform to commonsense in the news, and thus has good recognition ability.

Table 1. Comparison results of graph matching algorithms.

Matching Algorithm	RRWM	SM	SMAC
Accuracy	91.07%	82.14%	78.57%
Precision	86.36%	76.47%	68.75%
Recall	90.48%	68.42%	61.11%

Table 2. Comparison results of fake news identification methods.

Method	Accuracy	Precision	Recall
SVM-TS	63.12%	63.29%	63.01%
CNN	71.12%	71.30%	71.12%
GRU	79.27%	81.39%	79.27%
RRWM	91.07%	85.00%	89.47%

For the ablation experiments, the Ernie-based text similarity comparison module is used in the knowledge extraction process, and the module serves to fuzzy match the entities extracted from the news text and the target entities to enhance the ability of knowledge extraction from the text. Meanwhile, in the process of graph matching the background knowledge graph and the general knowledge graph extracted from the news text using the random walk-based graph matching algorithm for the two knowledge graphs, the Ernie-based text similarity matching module is utilized in constructing the similarity matrix to improve the generalization performance of the graph matching results by using text fuzzy matching. In order to verify the effectiveness of the two-part text similarity matching module in this experiment, an ablation experiment is designed to compare and analyze the results in terms of accuracy, correctness, and recall before and after removing the two-part semantic matching module.

The experimental results are shown in Table 3. From the experimental results in the figure, we can see that using the Ernie-based text similarity comparison module to calculate the similarity of entities in the process of knowledge extraction and graph matching can effectively improve the experimental results. When the module is not added, the accuracy of the model is 80.36%, and the correct rate is 83.33%. After adding this module in the knowledge extraction part, the accuracy of the model increased to 83.93%, and the correctness rate increased to 84.21%. By adding this module to the graph matching process, the model accuracy increased to 83.93%, and the correctness increased to 85.71%. When both modules were added simultaneously, the accuracy and correctness increased to 91.07% and 86.36%, respectively. Thus, it can be learned that the Ernie-based text similarity comparison module can effectively improve the generalization performance of the model after adding knowledge extraction and graph matching, and thus improve the effectiveness of the model.

Table 3. Results of ablation experiments.

Model	Accuracy	Precision	Recall
No semantic matching model	80.36%	83.33%	65.22%
Knowledge extraction with semantic matching	83.93%	84.21%	72.73%
Graph matching with semantic matching	87.50%	85.71%	81.82%
Semantic matching is added to both parts	91.07%	86.36%	90.48%

6. Conclusions

The existing deep learning-based fake news identification methods can obtain the underlying spatial features of news, but these features are obtained through a black box and are a continuous vector that humans simply cannot understand; humans can only understand the semantic scenarios, which have the problem of poor interpretability. Meanwhile, the existing methods of combining external knowledge for fake news detection exist only to directly integrate external knowledge into news content without considering the comparison between external knowledge and news content. In this paper, on the one hand, we extract the commonsense knowledge background knowledge contained in the news through knowledge extraction technology and form the news background knowledge map. On the other hand, based on the graph matching algorithm of random walk, the extracted news background knowledge map and the knowledge map of the Baidu Encyclopedia are subjected to graph matching, and the knowledge in the news knowledge map is judged to be consistent with the knowledge in the knowledge map of the Baidu Encyclopedia through the analysis and processing of the matching results; then, the truth or falsity of the news is judged.

Although the method proposed in this paper can compare and analyze the commonsense knowledge contained in the news text with the Baidu Encyclopedia commonsense knowledge graph to determine the truthfulness of the news, there are still shortcomings in that the method is not able to identify errors in the news text when they are non-commonsense knowledge errors. The graph matching algorithm used in this paper can only use second-order graph structure information. Therefore, we will consider using a super-graph matching algorithm to perform graph matching, so that higher-order graph structure information can be utilized, and false news identification can be performed more accurately and efficiently. Methods to solve these problems will be investigated in more depth in subsequent work.

Author Contributions: Conceptualization, X.G., X.D. and L.D.; methodology X.G., H.L. and S.W.; Software, W.C., S.W., Y.W., Z.Y. and L.L.; validation, J.S.; investigation, Y.C., H.L. and J.S.; writing, L.L.; funding acquisition, K.W. and H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Innovation Theory Technology Group Fund of China Electronics Tian'ao Co., Ltd. and the Natural Science Foundation of Heilongjiang Province of China under Grant LH2019F011.

Institutional Review Board Statement: Not Applicable.

Informed Consent Statement: Not Applicable.

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

Acknowledgments: The authors are thankful for the support of the staff member of China Electronics Tian'ao Co., Ltd.

Conflicts of Interest: There is no conflict of interest between co-authors in this article.

References

1. Barakat, K.A.; Dabbous, A.; Tarhini, A. An empirical approach to understanding users' fake news identification on social media. *Online Inf. Rev.* **2021**, *45*, 1080–1096.
2. Arafeh, M.; El Barachi, M.; Mourad, A.; Belqasmi, F. A blockchain based architecture for the detection of fake sensing in mobile crowdsensing. In Proceedings of the 2019 4th International Conference on Smart and Sustainable Technologies (SpliTech), Split, Croatia, 18–21 June 2019; pp. 1–6.
3. Melki, J.; Tamim, H.; Hadid, D.; Makki, M.; El Amine, J.; Hitti, E. Mitigating infodemics: The relationship between news exposure and trust and belief in COVID-19 fake news and social media spreading. *PLoS ONE* **2021**, *16*, e0252830. [CrossRef] [PubMed]
4. Hamza, A.; Javed, A.R.R.; Iqbal, F.; Kryvinska, N.; Almadhor, A.S.; Jalil, Z.; Borghol, R. Deepfake Audio Detection via MFCC Features Using Machine Learning. *IEEE Access* **2022**, *10*, 134018–134028. [CrossRef]

5. Mahdi, A.; Farah, M.F.; Ramadan, Z. What to believe, whom to blame, and when to share: Exploring the fake news experience in the marketing context. *J. Consum. Mark.* **2022**, *39*, 306–316. [CrossRef]
6. Zhang, H.; Fang, Q.; Qian, S.; Xu, C. Multi-modal knowledge-aware event memory network for social media rumor detection. In Proceedings of the 27th ACM international conference on multimedia, Nice, France, 21–25 October 2019; pp. 1942–1951.
7. Dun, Y.; Tu, K.; Chen, C.; Hou, C.; Yuan, X. Kan: Knowledge-aware attention network for fake news detection. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtually, Palo Alto, CA, USA, 2–9 February 2021; Volume 35, pp. 81–89.
8. Wang, Y.; Qian, S.; Hu, J.; Fang, Q.; Xu, C. Fake news detection via knowledge-driven multimodal graph convolutional networks. In Proceedings of the 2020 International Conference on Multimedia Retrieval, Dublin, Ireland, 8–11 June 2020; pp. 540–547.
9. Cho, M.; Lee, J.; Lee, K.M. Reweighted random walks for graph matching. In Proceedings of the Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, 5–11 September 2010.
10. Leordeanu, M.; Hebert, M. A spectral technique for correspondence problems using pairwise constraints. In Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, Beijing, China, 17–21 October 2005; Volume 2, pp. 1482–1489.
11. Cour, T.; Srinivasan, P.; Shi, J. Balanced graph matching. *Adv. Neural Inf. Process. Syst.* **2006**, *19*, 313–320.
12. Jiao, Z.; Sun, S.; Sun, K. Chinese lexical analysis with deep bi-gru-crf network. *arXiv* **2018**, arXiv:1807.01882.
13. Lu, Y.; Liu, Q.; Dai, D.; Xiao, X.; Lin, H.; Han, X.; Sun, L.; Wu, H. Unified structure generation for universal information extraction. *arXiv* **2022**, arXiv:2203.12277.
14. Kang, U.; Hebert, M.; Park, S. Fast and scalable approximate spectral graph matching for correspondence problems. *Inf. Sci.* **2013**, *220*, 306–318. [CrossRef]
15. Torr, P.H. Solving markov random fields using semi definite programming. In Proceedings of the International Workshop on Artificial Intelligence and Statistics, Key West, FL, USA, 3–6 January 2003; pp. 292–299.
16. Schellewald, C.; Schnrr, C. Probabilistic subgraph matching based on convex relaxation. In Proceedings of the Energy Minimization Methods in Computer Vision and Pattern Recognition: 5th International Workshop, EMMCVPR 2005, St. Augustine, FL, USA, 9–11 November 2005; pp. 171–186.
17. Egozi, A.; Keller, Y.; Guterman, H. A probabilistic approach to spectral graph matching. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1798–1828. [CrossRef] [PubMed]
18. Gori, M.; Maggini, M.; Sarti, L. Exact and approximate graph matching using random walks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1100–1111. [CrossRef] [PubMed]
19. Zass, R.; Shashua, A. Probabilistic graph and hypergraph matching. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 1–8.
20. Wang, H.; Zhang, F.; Xie, X.; Guo, M. DKN: Deep knowledge-aware network for news recommendation. In Proceedings of the 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018; pp. 1835–1844.
21. Zhong, W.; Xu, J.; Tang, D.; Xu, Z.; Duan, N.; Zhou, M.; Wang, J.; Yin, J. Reasoning Over Semantic-Level Graph for Fact Checking. In *ACL*; Association for Computational Linguistics: Cedarville, OH, USA, 2020; pp. 6170–6180.
22. Shi, X. Automatic Commonsense Knowledge Base Construction and Completion for Chinese. Master's Thesis, East China Normal University, Shanghai, China, 2022.
23. Grishman, R. Twenty-five years of information extraction. *Nat. Lang. Eng.* **2019**, *25*, 677–692. [CrossRef]
24. Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C.D.; Ng, A.Y.; Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, WA, USA, 18–21 October 2013; pp. 1631–1642.
25. Sak, H.; Senior, A.; Beaufays, F. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. *Interspeech* **2014**, 338–342. [CrossRef]
26. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv* **2014**, arXiv:1412.3555.
27. Lafferty, J.; McCallum, A.; Pereira, F.C. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of the 18th International Conference on Machine Learning 2001, Williamstown, MA, USA, 28 June–1 July 2001.
28. Bengio, Y.; Ducharme, R.; Vincent, P. A neural probabilistic language model. *Adv. Neural Inf. Process. Syst.* **2000**, *13*, 1137–1155.
29. Yan, J. Algorithmic Studies and Design on Graph Matching. Ph.D. Thesis, Shanghai Jiao Tong University, Shanghai, China, 2015.
30. Zhou, L. Research and Implementation of Commonsense Reasoning Technology Based on Path Mining. Master's Thesis, University of Electronic Science and Technology of China, Chengdu, China, 2022.
31. Conte, D.; Foggia, P.; Sansone, C.; Vento, M. Thirty years of graph matching in pattern recognition. *Int. J. Pattern Recognit. Artif. Intell.* **2004**, *18*, 265–298. [CrossRef]
32. Xu, X. Random Walk Learning on Graph. Ph.D. Thesis, Nanjing University of Aeronautics and Astronautics, Nanjing, China, 2008.

33. Wang, Y. Research and Application of Random Walk Algorithm Based on Distance. Master's Thesis, Beijing Jiao Tong University, Beijing, China, 2012.
34. Li, Y.; Gao, D. Research on Entities Similarity Calculation in Knowledge Graph. *J. Chin. Inf. Process.* **2017**, *31*, 140–146+154.