# RESEARCH

# **Open Access**

# A knowledge-graph based text summarization scheme for mobile edge computing



Zheng Yu<sup>1</sup>, Songyu Wu<sup>1</sup>, Jielin Jiang<sup>1</sup> and Dongqing Liu<sup>2\*</sup>

# Abstract

As the demand for edge services intensifies, text, being the most common type of data, has seen a significant expansion in data volume and an escalation in processing complexity. Furthermore, mobile edge computing (MEC) service systems often faces challenges such as limited computational capabilities and difficulties in data integration, requiring the development and implementation of more efficient and lightweight methodologies for text data processing. To swiftly extract and analysis vital information from MEC text data, an automatic generation scheme of multi-document text summarization based on knowledge graph is proposed in this paper, named KGCPN. For the text data from MEC devices and applications, the natural language processing technology is used to execute the data pre-processing steps, which transforms the MEC text data into a computationally tractable and semantically comprehensible format. Then, the knowledge graph of multi-document text is constructed by integrating the relationship paths and entity descriptions. The nodes and edges of the knowledge graph serve to symbolize the semantic relationships within the text, and the Graph Convolution Neural network (GCN) is used to understand the text and learn the semantic representation. Finally, a pointer-generator network model accepts the encoding information from GCN and automatically generate a general text summarization. The experimental results indicate that our scheme can effectively facilitate the smart pre-processing and integration of MEC data.

Keywords Mobile edge computing, Artificial intelligence, Knowledge graph, Natural language processing

# Introduction

With the accelerated advancement of the Internet of Things (IoT) and cloud computing, individuals are inundated with an abundance of textual information daily. The continuous monitoring and data collection from mobile edge computing (MEC) devices have generated an even larger volume of data. Although MEC aims to extend the edge of cloud networks to reduce latency and network

and Technology, Nanjing 210044, China

Nanjing 210093, China

congestion [1], how to process MEC data in an economical and secure way remains a fundamental challenge. In recent years, deep learning-driven artificial intelligence (AI) has gradually grown into a key technology for realizing intelligent data analysis [2], capable of addressing issues such as limited computing power and fast-changing context environment. To efficiently and intelligently process text data from MEC devices and applications, text summarization and knowledge graphs, which are pivotal technologies in the field of AI, offer promising approaches to address this issue.

To swiftly comprehend and assimilate massive information, text summarization has emerged as an extremely pragmatic tool [3]. However, traditional summarization methodologies predominantly rely on single documents, failing to fully exploit the information



© The Author(s) 2024. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

<sup>\*</sup>Correspondence:

Dongqing Liu

dqliu1986@sina.com

<sup>&</sup>lt;sup>1</sup> School of Software, Nanjing University of Information Science

<sup>&</sup>lt;sup>2</sup> Institute of Special Weather Climate (ISWC), Nanjing University,

redundancy across multi-documents. This limitation hinders the ability to provide a comprehensive and accurate summary of the article's theme [4]. Furthermore, with the advent of knowledge graph technology, an increasing number of studies are focusing on how to utilize this technology for text summarization. The automatic generation algorithm based on knowledge graph is studied to improve the accuracy and efficiency of multi-document text summarization [5]. The knowledge graph is a kind of knowledge base presented in a graphical way, which expresses the complex relationship between knowledge through entities, relationships and attributes [6]. As the demand for knowledge grows, traditional standalone information formats such as text, images, and videos are no longer sufficient to satisfy people's need for diverse and integrated knowledge [7]. Knowledge graphs can visually present a wealth of knowledge information, enabling users to more intuitively understand the relationships between different pieces of knowledge [8]. This not only enhances comprehension but also allows for more effective application in real-world scenarios.

Therefore, this study aims to overcome the limitations of traditional MEC text data processing methods by proposing a new scheme named KGCPN that utilizes a summarization model based on knowledge graphs and Graph Convolution Neural Network (GCN) [9]. This approach is designed to capture the contextual information of MEC text data, extract entities and relationships, and construct the knowledge graph. The nodes and edges in the knowledge graph represent semantic relationships. Through GCN for text comprehension and semantic representation learning, the corresponding semantic information is input into the Pointer-Generator Network (PGN) to automatically generate multi-document text summarization. The contributions of this paper are as follows:

- 1. Automated feature extraction from MEC data: This research employs NLP techniques such as word segmentation, part-of-speech tagging, and named entity recognition (NER) to integrate MEC data and construct the knowledge graph.
- 2. **Optimization of text summarization**: Leveraging the constructed knowledge graph, the semantic sequences generated by the GCN are fed into a PGN model to produce summarization. This approach enhances the generation of multi-document summarization.
- 3. Effective integration of MEC data and AI: The fully trained text summarization model is deployed directly on edge devices, obviating the need to transmit vast quantities of raw text back to the central server. This approach enables a rapid response to

user requests, enhancing the overall efficiency of the system.

The related work will be presented and discussed. Then we elaborate on the construction of the knowledge graph. The design details of our text summarization model are introduced later. Our comparative experiments and performance evaluation are described then. Finally, we conclude the article with prospects.

# **Related work**

There are various processing methods for data and resources in edge cloud environments currently. Xu et al. [10] proposed a dynamic offloading strategy based on game theory and convolutional neural network partitioning, which can determine the optimal offloading decision in dynamic and heterogeneous edge-cloud environment. Li et al. [11] proposed a knowledge driven anomaly detection framework that effectively solves the problem of feature loss in distributed environments by dynamically adjusting feature attention. To address the potential load imbalance issues that traditional edge server deployment strategies may face, Yan et al. [12] designed a deployment strategy of edge servers based on the stateaction-reward-state-action (SARSA) learning, which optimized response latency and processing energy consumption. To fully utilize AI technology and edge devices distributed at road edges, The advantages of machine learning methods represented by CNN and federated learning in improving model efficiency and stability have also been recognized [13-15]. Tian et al. [16] achieved a MEC-enabled distributed cooperative microservice caching scheme by integrating deep reinforcement learning and Markov decision process. However, issues such as how to effectively integrate text data from MEC devices and applications, and how to effectively extract semantic information from MEC text data still need to be explored and studied.

The goal of automatic text summarization is to generate concise and accurate abstracts, extracting key information from multi-documents, which simplifies the understanding of the main points across a collection of texts. This task can effectively summarize and organize vast amounts of text information in fields such as search engines, recommendation systems, and automatic translation. This enhances the efficiency and quality of information processing, thereby improving the overall user experience and system performance [17]. In the field of machine learning, the task of automatically generating multi-document text summarization can be perceived as a distinct text classification task [18]. In the domain of data mining, this endeavor can be considered as a text clustering mining task, where the goal is to extract meaningful patterns and information from an extensive array of texts. The investigation into algorithms for the text summarization from multi-documents encounters numerous challenges and intricacies [19]. Redundancy and conflict of information among multi-documents need to be fully dealt with to avoid duplicate or contradictory information. Quality and accuracy need to be fully guaranteed [20] to ensure that the generated abstract is concise and comprehensive. Knowledge differences and language barriers between documents in different fields also need to be fully considered and solved [21]. At present, many researchers have studied the method of generating text summarization. Sanchez-Gomez et al. [22] take medical texts as an example to study the query-oriented algorithm of generating text abstracts. This method better considers the professional knowledge and needs in the medical field and has certain pertinence. It can effectively compress and sort out the important information in medical texts, making the abstracts more informative. Mojrian et al. [23] use the heuristic method of quantum computing and genetic algorithm to extract multi-document text summarization. Quantum heuristic genetic algorithm can find better abstracts in a shorter time, has strong robustness and can deal with various input documents and noise data. However, this method is difficult to implement, requires more technology and resources, and affects the real-time performance and efficiency of the algorithm. Gupta et al. [24] used two-level feature extraction method to generate abstracts. This method used two-level feature extraction, firstly understood the report at a macro level, and then extracted more detailed features, which helped to grasp the core content of the text and clearly show how each element in the abstract was extracted from the original text, so that readers could better understand the abstract. However, because this method needs multi-stage processing, it may take more time to extract abstracts than other single-stage methods.

Aiming at the problems existing in the above methods, we propose an automatic multi-document text summarization scheme based on knowledge graphs. Unlike traditional summarization strategies, this study delves into the deep processing and analysis of multi-document texts, fully utilizing semantic information and entity relationships within the knowledge graph. This approach facilitates efficient processing and abstract generation of multi-document texts, offering enhanced support to swiftly comprehend and assimilate vast amounts of MEC text data information.

# **Data integration**

The MEC data in this paper mainly comes from roadside intelligent devices such as cameras and traffic lights [25], vehicles, and mobile devices such as smartphones. After these data are cleaned and preprocessed into multidocument texts, they are subjected to feature extraction using NLP methods. This primarily involves word segmentation, part-of-speech tagging, and NER. The result of word segmentation is directly used for the part-ofspeech tagging task, where the generated word sequence is used as input and corresponds to the tagging sequence. The part-of-speech tagging sequence is used for training the NER model. Once the features, such as entities and relationship paths within the text, are identified, the automatic generation of a multi-document knowledge graph is initiated. Subsequently, the generated knowledge graph is utilized for text comprehension and semantic representation learning based on GCN. Finally, the text summarization is automatically produced using the PGN which leverages the encoding information from the GCN. Contrasted with traditional data integration methods that require substantial manpower to manually extract entities and relations from text, NLP technology accomplishes the task of automatically constructing a knowledge graph with minimal cost, which is achieved by modeling and integrating the semantic relations of textual entities. The resultant knowledge graph, endowed with attributes such as accuracy and interpretability, can significantly enhance the efficacy of text summarization (Fig. 1).

#### Word segmentation

In the data integration, a word serves as the basic semantic unit and an integral element of the text. The word segmentation is employed to establish the boundaries of words within documents, which is essential as it enables the execution of deep text processing. In the realm of natural language processing (NLP), the N-gram model is often utilized for word segmentation, which allows for the straightforward matching of text sentences according to a predefined dictionary [26], thereby enabling the search for all potential candidate words within the matching results. Utilizing the candidate words derived from text sentences, an N-gram segmentation graph is formulated. In this graph, all candidate words are set as nodes and weights are set for the edges of the word segmentation graph to represent the cost of different edges. The Viterbi algorithm is employed to identify the path with the minimum cost in the word segmentation graph, thereby deriving the final word segmentation sequence. This approach combines both statistical and rule-based methods, which can effectively execute the word segmentation task using the word separator.

# Part-of-speech tagging

The part-of-speech tagging requires the establishment of a set of tagging rules. Firstly, continually mark the



intermediate results [27], count the incorrect tags, then build a rule learner, and use the corrected marked content to generate new tagging rules. Then, iterate repeatedly, and stop when no new rules can reduce the number of incorrectly tagged instances, obtaining a set of final tagging rules. Finally, the number of incorrect part-of-speech tags can be minimized to the greatest extent based on these tagging rules.

The rule iteration and model training process mentioned above are carried out by the Hidden Markov Model (HMM), combining the pre-trained word embeddings (such as Word2Vec [28], GloVe [29], etc.) and language models (such as BERT, ELMo [30], etc.) to extract rich semantic features of vocabulary, which can help better represent the context of vocabularies in the sentences. The word sequences  $\{x_1, x_2, ..., x_n\}$  and part-of-speech markers  $\{y_1, y_2, ..., y_n\}$  are introduced to represent observation sequence and state sequence, the expression of text part-of-speech tagging is as follows:

$$y = \underset{y \in Y}{\operatorname{argmax}} \prod_{i=1}^{n} P(x_i | y_i) \cdot P(y_i | y_{i-1})$$
(1)

In the formula (1),  $P(x_i|y_i)$  indicates the probability that the word sequence  $x_i$  marked as the label of word  $y_i$ .

Before the parameter training of the HMM, the relevant parameters of the dictionary information constraint model are set, and the state generation probability and state transition probability parameters are trained by large-scale corpus. After completing the model training, the Viterbi algorithm is selected to mark the part-of-speech of multi-document texts.

# NER

NER, also known as entity labeling, is an information extraction technology. From the multi-document text, the category marking of key information of named entities is processed [27]. Conditional random field (CRF) probability model is selected to identify named entities in multidocument texts. According to the input text sequence, this method constructs a discriminant model to obtain the output sequence. While training the CRF model, the previous part-of speech tagging results are used as additional feature inputs, which means that each word is not only represented as its original form, but also as its partof-speech tagging label. The part-of-speech tagging can help the model understand the grammatical role of words in sentences, thereby recognizing named entities more accurately. In the NER of multi-document texts, we use  $X = \{x_1, x_2, ..., x_n\}$  and  $Y = \{y_1, y_2, ..., y_n\}$  represent two sets of random text variables, X and Y represent a structurally identical observed sequence as well as the hidden states, respectively. The constructed linear chain CRFs are represented by  $p(y_i|X, y_{i-1}, y_{i+1})$ , and the constraints are added to the CRF model to reduce the probability of errors in the output label. The expression of the CRF is as follows:

$$score(s, w) = \sum_{i=0}^{n} (A_{w_i, w_{i+1}} + P_{i, w_i})$$
(2)

$$P(w \mid s) = \frac{e^{score(s,w)}}{\sum\limits_{w \in W_i} e^{score(s,w)}}$$
(3)

In the above formula, score(s, w) and P respectively represent the comprehensive evaluation score and the emission score matrix of the text, A and  $w_i$  respectively represent the transfer matrix and all possible tag sequences, P(w|s) represents the probability that the input sequence of text corresponds to the tag sequence. The input and output of CRF model are text sequence and label sequence respectively. The Viterbi algorithm is used to complete the prediction task, and the complexity of the named entity identification path is reduced by the dynamic programming solution method, and the final text NER result is output.

### Multi-document text knowledge graph

By identifying the relationship paths between entities, we can enhance the connectivity and expressive power of the knowledge graph. On the other hand, transforming the entity description information in the text into semantically represented vectors can enrich the attribute information of entities in the knowledge graph. Therefore, the representation of relationship path information and entity description information in multi-document texts aims to enrich and improve the knowledge graph from different perspectives, thereby enhancing the effectiveness of knowledge acquisition and inference.

#### Representation of path information

After completing NER, we begin to represent the relationship path information. The TransE model [31] is employed for knowledge graph embedding tasks above due to three pivotal reasons. Firstly, the TransE model boasts a comparatively simple structural design, which significantly enhances the computational efficiency during both training and prediction phases. Secondly, it articulates relationships within the knowledge graph as vector translations between entities, offering an intuitive geometric perspective for comprehending and interpreting the knowledge graph. Lastly, the translation characteristic of TransE can be harnessed for sophisticated tasks such as link prediction and entity alignment, thereby broadening the application spectrum of the knowledge graph.

The relationship between entities h and t is represented by relationship paths connected by N bars, and the relationship paths between connecting h and t are included in the set  $P = \{p_1, p_2, ..., p_N\}$ . The relationship path is represented as a triplet (h, p, t), defining the energy function for it. Considering the relationship between entities and path information in the multi-document, the accuracy of knowledge reasoning of knowledge graph is improved. All the vectors in the text [32] are then added into the path combination vector p\*, which is expressed as follows:

$$p* = r_1^* + e_1^* + \dots + e_{l-1}^* + r_l^*$$
(4)

In the formula (4),  $r^*$  represents structured vector representation of text relations. When there is a multi-step relationship path between the entity pairs, based on the energy function of the TransE model, the expression of the triple energy function that defines the relationship path is as follows:

$$E(h, p, t) = ||h^* + p^* - t^*|| = ||p^* - r^*|| = E(p, r)$$
(5)

From formula (5), the smaller the relationship r between the relationship path p, the higher the similarity between them, and the probability of calculating the relationship r is greater by using the path p.

According to the path-constrained resource allocation algorithm applied to the fact triplet (h, r, t), considering the relationship between multiple paths, the comprehensive energy function expression of multiple paths in the text is constructed as follows:

$$E_{p}(h, P, t) = \frac{1}{Z} \sum_{p \in P(h, t)} R(p|h, t) E(h, p, t)$$
(6)

In the formula (6),  $p \in P(h, t)$  represents the reliability of the full relationship path and the relationship path pbetween entity pairs (h, t) is expressed as R(p|h, t). Z and E(h, p, t) respectively represent the normalization factor and the energy function of the path. Based on the above process, the representation of path information of multidocument text relationship is completed.

## Representation of descriptive information

To reduce text information loss, the entity description information of multi-document texts is input into the BERT model [33] and converted into semantic representation. The entity information in the text is described using methods such as word embedding, segment embedding, and position embedding. The description result of entity information is processed by vector concatenation [34], and the concatenation result is input into the BERT model to obtain sentence vectors. After taking the average value of all sentence vectors, we can obtain the entity description information vector. The expression of the entity description information vector in the text is as follows:

$$h_d^* = \frac{\sum_{i=1}^n S_i}{n} \tag{7}$$

In the formula (7), n indicates the number of text statement,  $S_i$  represents the sentence vector of the statement.

The representation method based on structure and entity description can better determine the information of factual triples in the knowledge graph. When similar entities exist in the text, the description information is similar to the keywords [35], and the relationship between entities cannot be directly obtained through structural information, but can be clarified by analyzing the internal relationship of keywords. The expression of energy function expressed by text entity description information is defined as follows:

$$E_d = E_{dd} + E_{ds} + E_{sd} \tag{8}$$

In the formula (8),  $E_{dd}$  and  $E_{ds}$  mean that the head and tail are described by entities and structural information respectively, which is helpful to obtain the best vector representation results of entities and relationships in texts, thereby constructing a more comprehensive and accurate knowledge graph.

### Automatic construction

Prior to the actual deployment of the KGCPN in the edge environment, the nodes and edges within the knowledge graph bear no direct correlation to edge computing itself. Instead, the nodes in the knowledge graph symbolize various entities found within the text, such as "Ocean" and "Forest". Conversely, the edges denote the relationships between these textual entities, expressing associations like "Belongs to" and "Similar to". Based on the entities from the previously acquired multi-document texts, when automatically constructing a knowledge graph, it is necessary to establish the index of the knowledge graph according to the attributes of the entities. The process of building multi-document knowledge graph is as follows:

(1) Input the stored entities. A random entity X is selected from the text entities as the initial center point of the knowledge graph.  $Y_i$  and D respectively represent the text words and the set of words added to the graph. The attribute values corresponding to entity X are retrieved to form a set  $\Phi$ . Using the representation of entity description information to determine the initial central node, the boundary set  $E_d = \{Y_i\}$  of central node X is created with the newly added node  $X_d$ . Based on the representation of relationship path information, the relationship between the new node to be added and the central node is determined through the existing path information in multi-document texts. Then  $E_d$  contained in  $\Phi$  is gathered. At the same time, if a collection of words D is not empty, merge edge sets, otherwise proceed to the next step.

(2) Set  $X = X_k$ ,  $X_k$  represents the next entity in the knowledge graph that is associated with X. Repeat the above steps until all the entities associated with the current entity have been traversed, and proceed to the next step.

(3) Select other nodes in multi-document texts. Take this node as the starting node of the knowledge graph. Repeat the above steps until all texts of multi-document are traversed, and the multi-document knowledge graph is expanded to complete the construction.

In the process of merging node and edge sets, the edge set  $E_i$  of current node  $X_i$  and central node X is used to determine whether  $Y_i$  exists in  $E_i$ . If  $Y_i$  is not present in  $E_i$ , then  $Y_i$  is added to  $E_i$ , otherwise the addition is skipped.  $Y_i$  represents the current query conditions during the construction. By merging the node boundaries, the information of the current node and the added node edges is merged to solve the conflict of the knowledge graph edge and the redundant information between the nodes.

# KGCPN

# Text understanding and semantic representation

According to the knowledge graph constructed for multidocument texts, the GCN method is selected for text understanding and semantic representation learning, which uses convolution kernel with parameter sharing function to convolution receptive field, realizing the text feature extraction. According to the convolution theorem, the GCN completes the graph convolution operation of each node information in the knowledge graph (Fig. 2).

Suppose that the knowledge graph contains N nodes, in Fourier domain, the graph convolution operation is performed by multiplying the convolution kernel  $g_{\theta}$ , and the node feature  $x \in \mathbb{R}^N$ , which is expressed as follows:

$$g_{\theta} * x = U g_{\theta}(\Lambda) U^T x \tag{9}$$

In the formula (9), U and \* respectively represent the eigenvector matrix of the normalized Laplacian graph matrix L as well as the convolution operation,  $\Lambda$  and  $g_{\theta}(\Lambda)$  respectively represent the diagonal matrix composed of the node features of the knowledge graph and the convolution kernel associated with the function of the matrix  $\Lambda$ . By updating convolution kernel  $g_{\theta}$ , the expression of the convolution operation of knowledge graph is as follows:

$$g_{\theta'} * x \approx \sum_{k=0}^{K} \theta'_k T_k \left(\frac{2}{\lambda_{max}} L - 1\right) x \tag{10}$$

In the formula (10),  $\lambda_{max}$  and  $\theta' \in \mathbb{R}^{K+1}$  respectively express the maximum eigenvalue of L and Chebyshev coefficient vector, K represents the order. The operation procedure of the graph convolution is related to the polynomial of order K of the Laplacian graph matrix, making the GCN have strong K-local connectivity.

To further simplify the propagation rules of node information of GCN, *K* is set as 1, by reducing the number of



Fig. 2 The scheme of proposed KGCPN

parameters, over-fitting can be avoided.  $L^{j}$  represents the *j*-th layer node GCN and the number of layers is j + 1, the expression of node convolution operation in this layer is as follows:

$$z_i = \sigma \left( \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} L^j W_j \right) \tag{11}$$

In the formula (11),  $\tilde{A} = A + I$  represents an adjacency matrix with a self-ring,  $\tilde{D}$  and I respectively represent the degree matrix and the identity matrix of the diagonal matrix,  $W_j$  and  $\sigma$  respectively represent the weight matrix and the activation function. The GCN is used to convolution the node features in the knowledge graph to obtain the final text sentence features.

The text features obtained above will be translated into  $z'_i$  after the average pooling and combining features, and then input into the fully connected layer, where the activation function is set to Softmax, and finally get the labels  $Y_i$  of the multi document texts. The calculation formula of the fully connected layer is as follows:

$$Y_{i} = argmax \left( Softmax \left( W_{s} z_{i}^{'} + b \right) \right)$$
(12)

In the formula (12),  $Y_i \in \mathbb{R}^C$ , *C* and *b* indicate that number of text label and the offset respectively. When dealing with multi-document texts, the GCN selects cross entropy as the loss function, and the expression of the loss function is as follows:

$$loss = \sum y log\hat{y} + \lambda ||\theta||^2$$
(13)

In the formula (13),  $\lambda$  and  $\theta$  respectively represent the L2 regularization coefficient and regularization parameter, y and  $\hat{y}$  respectively represent the real label of the text and the prediction result.

The GCN layer is used to understand and learn the semantic representation of multi-document texts, and the multi-dimensional semantic feature coding results are output, which are subsequently utilized to automatically generate text summarization.

## Automatic generation

The PGN model [36] is selected to be the generation model for the KGCPN. In general PGN models, the parameters of the word embedding matrix are randomly initialized, which may lead to poor performance and elevated training costs. In addition, the issue of polysemy is also urgently needed to be addressed. To tackle these challenges, we incorporate the BERT model for parameter initialization. The multi-head attention mechanism is employed to obtain contextual semantic information, making it more dynamic and enabling more effective feature extraction. In addition, BERT also distinguishes multiple meanings of a word by combining different contextual information features to represent distinct semantics. After completing the above initialization operations, the encoding results of multi-dimensional semantic features output by GCN and BERT are input into the encoder of long short-term memory (LSTM) network to generate the hidden layer state sequence  $h_i$ . At the *t* time, the word vector generated at the last moment is input into the LSTM decoder to obtain the decoding state sequence  $s_t$ , which is the generation results of multi-document text summarization. By harnessing the structural information of the knowledge graph, the KGCPN is enabled to more effectively comprehend the relationships and structural data among nodes.

The PGN not only copies words from the original document but also generates words from a fixed-size vocabulary, which effectively solve the OOV problem (difficulty in handling out-of-vocabulary words). The generation probability  $p_{gen} \in [0, 1]$  for timestep t is calculated from the context vector  $h_{t}^*$ , the decoder state  $s_t$  and the decoder input  $x_t$ :

$$p_{gen} = \sigma \left( w_{h^*}^T h_t^* + w_s^T s_t + w_x^T x_t + b_{ptr} \right)$$
(14)

In the formula (14),  $W_{h^*}$ ,  $W_s$ ,  $W_x$  and scalar  $b_{ptr}$  are learnable parameters and  $\sigma$  is the sigmoid function. Next,  $p_{gen}$ will decide whether to generate words from the vocabulary by sampling from  $P_{vocab}$ , or copy words from the input sequence by sampling from attention distribution  $a^t$ . The probability distribution over the extended vocabulary can be specified as:

$$P(w) = p_{gen} P_{vocab}(w) + (1 - p_{gen}) \sum_{i:w_i = w} a_i^t$$
(15)

To reduce the repetition problem, the coverage mechanism is introduced. In each decoding process, PGN calculates a new coverage vector, which represents the cumulative total of attention distributions from all preceding steps. Subsequently, in the loss function computation, a coverage loss is also factored in, which is the sum of the minimum values of the coverage vector and the attention distribution of the current step. Coverage loss encourages PGN to avoid giving too much attention to the same input vocabulary as much as possible. Through this methodology, the PGN can effectively mitigate the issue of duplicate generation and enhance the quality of text summarization. The coverage vector *c<sup>t</sup>* tracks the words that have been generated and applies a certain penalty to these words to minimize the duplication in the generated summary.  $c^{t}$  can be understood as the degree of coverage that the words have received from the attention mechanism, then the coverage vector is regarded as extra input into the attention mechanism.

$$c^{t} = \sum_{t'=0}^{t-1} a^{t'},$$
(16)

$$e_i^t = v^T tanh \left( W_h h_i + W_s s_t + W_c c_i^t + b_{attn} \right)$$
(17)

In the formula (17),  $W_c$  is a learnable parameter vector of same length as v. Accordingly, the coverage loss must be introduced. Therefore, the final loss function consists of two parts: the cross entropy and the coverage loss, in which the weight of the coverage loss is determined by  $\lambda$ :

$$covloss_t = \sum_i min(a_i^t, cr_i^t),$$
 (18)

$$loss_t = -logP(w_t^*) + \lambda \times covloss_t$$
(19)

In conclusion, the composition of KGCPN is as follows: GCN parses the entity information and dependency relationships transmitted from the knowledge graph into encoded results and transmits them to PGN, and finally PGN completes the summarization output.

## **Experimental results and analysis**

We gathered 2746 documents from MEC devices like traffic lights, cameras, mobile phones, and vehicles, covering situations such as traffic and travel. Additionally, we included 582 authoritative meteorological forecast documents from the Jiangsu Provincial Meteorological Bureau into our dataset. To encompass daily situations like dialogue and news, we collected 1056 documents using a theme crawler program from sources like Wikipedia, Baidu Baike, and major news websites. After a rigorous screening process, we utilized 4097 documents to construct the final dataset, with an average word count of 1052 per document. Based on this amalgamated dataset, we employ NLP methods such as word segmentation, part of speech tagging, and NER to finish in feature extraction and data integration, ultimately constructing a knowledge graph. The MEC data collected in this paper includes not only textual data, but also device status, network traffic, user behavior, etc., most of which are semi-structured. There are three primary methods for converting these data into textual format: first, convert each item or group of data into descriptive sentences. For example, the device status can be converted to "Device A has 80% battery and a signal strength of -60dBm". Network traffic can be converted to "User B's traffic consumption between 12:00 and 13:00 is 200MB". Alternatively, convert the data into a series of event logs. For instance, user behavior can be translated as "User C visited website D at 14:00 and then downloaded file E at 14:05.". Finally, if certain log tables do require storage, OCR technology can be used to extract textual data from them.

In the stage of text preprocessing, it's essential to eliminate special tables and symbols that have low contributions to the analysis of multi-document texts. Following this, stemming is performed on the text to retain the root word. Stem processing allows us to normalize various forms of vocabulary (like plurals, past tense, etc.) into the same stem, which significantly reduces the size of the vocabulary space, thereby diminishing the complexity and computational cost of the model. Similarly, in the field of information retrieval, we can use any form of vocabulary to search for information. For instance, if a user queries "running", the system is capable of returning documents that contain "run" and "runner", thereby enhancing the efficiency of information retrieval. Subsequently, word segmentation is carried out, breaking down the sentences within the document into sets of words. To gain a more comprehensive understanding of the document's content and structure, a length coefficient is introduced to manage sentences that are excessively long or unusually short within the experimental dataset. The formula for calculating this length coefficient is as follows:

$$C_L = L/L_M \tag{20}$$

In formula (20), L and  $L_M$  respectively represent that sentence length and the longest sentence length in the text.

In the experimental dataset, the statistical results of the length coefficient of some texts are shown in Fig. 3. As can be seen from Fig. 3, the length coefficients of the document text in the experimental data set are distributed in different areas in the [0,1] interval. When the text statement is too long or too short, it cannot be set as the candidate statement. Based on the calculation results of the text length coefficient, sentences with a coefficient greater than 0.8 or less than 0.2 are removed. The resulting text is ultimately used for the automatic generation of multi-document text summarization in the experiment.

To evaluate the performance of the model on tasks such as part-of-speech tagging, NER, and text summarization, distinct metrics are employed for each module. The definitions and computation methodologies of these specific indicators are elucidated as follows:

- 1. **Part-of-Speech Tagging**: The indicator for measuring the effectiveness of part-of-speech tagging is the accuracy, which is a ratio of the number of correctly labeled samples to the total number of samples. The higher the accuracy, the better the word segmentation performance of the model.
- 2. NER: The following three common metrics have been selected as the key criteria: precision, recall, and F1-score. Precision metric indicates the proportion of predicted results that are truly positive. Recall metric indicates the proportion of all positive cases that are correctly predicted. F1-score combines the precision and recall, and can analyze the entity classification performance more comprehensively. In the text summarization experiments, the semantic similarity index examines whether the information conveyed in the generated summarization is similar to the reference summarization, not just the similarity of vocabulary or sentence structure. The higher the value of meaning similarity, the better the model performance.
- 3. **Text Summarization**: Rouge (Recall-Oriented Understudy for Gisting Evaluation) [37] indicator is one of the most commonly used evaluation tools for evaluating automatic summarization or text summarization. Rouge-1, Rouge-2 and Rouge-L are three



Fig. 3 Statistical results of length coefficient

# Table 1 Tagging accuracy of part-of-speech

	Accuracy(	%)		
Type of Words	BERT	НММ	BERT-HMM	
Noun	90.35	76.52	92.12	
Verb	86.14	80.65	87.50	
Adverb	88.46	71.87	89.02	
Adjective	86.93	82.54	88.67	
Preposition	82.56	77.15	86.21	
Pronoun	87.45	72.68	88.25	
Conjunction	82.31	73.42	87.38	

main versions of Rouge series, which evaluate the repetition of one word, two words and long words respectively. Rouge-1 only considers the repetition of a single word or vocabulary. Rouge-2 considers the repetition of two consecutive words or words. Rouge-L considers not only the repetition, but also the relative positions of words in sentences.

In this paper, the part-of-speech tagging task is executed by BERT pre-trained model and HMM (BERT-HMM). To verify the enhancement by BERT-HMM, we respectively conducted part-of-speech tagging task on the same text data based on BERT and HMM methods, comparing their final accuracies. The comparison results of part-of-speech tagging are shown in Table 1.

According to the Table 1, BERT-HMM surpasses BERT and HMM in the performance of tagging all main types of words, achieving a peak accuracy of 92.12% for Nouns. Even for the lowest-scoring category, Prepositions, our model still exceeds an accuracy of 86%. Firstly, the bidirectional comprehension of BERT equips our model with enhanced representational learning abilities, thereby harvesting richer contextual information. Secondly, in Markov processes, the emergence of a particular state is contingent solely on one or a handful of preceding states, and remains unaffected by other states. This characteristic empowers HMM to manage first-order dependencies within sequences. Furthermore, through the utilization of the state transition matrix and observation probability matrix, we can derive the most probable state sequence. In summary, our model delivers the best performance in word segmentation, providing vital support for the follow-up NER task. Subsequently, NER experiments are conducted based on the results of part-of-speech tagging.

The NER model presented in this paper is primarily composed of three key components: a pre-trained BERT model, part-of-speech tagging results serving as additional input features, and a CRF for predicting entity types, named BERT-POS-CRF. In terms of parameter settings, the batch size is configured to 32, and a learning rate of 0.0001 is established. The number of epochs is set to 25 and Adam was selected as the optimizer. Meanwhile, the BERT model comprises 12 Transformer layers, 768-dimensional hidden layers, and employs 12 multihead attention mechanisms. Our model is compared against three benchmark models, the CRF, BiLSTM-CRF, and BERT-CRF. Table 2 details the comparison results.

From Table 2, compared to the single CRF and baseline model BiLSTM-CRF, the BERT-POS-CRF has improved the precision by 10.13% and 2.59% respectively. Moreover, compared to the BERT-CRF model that does not involve part-of-speech tagging, our model also shows an improvement of 4.49% in the precision. There are three main reasons why our model performs best in the NER task. Firstly, when the results of part-of-speech tagging are input as additional features, they can assist the model in understanding the grammatical role of words within a sentence. Secondly, part-of-speech tagging can aid in comprehending the context of sentences. For instance, verbs often precede nouns, and prepositions usually come before noun phrases. The contextual information can help the model identify named entities more accurately. Lastly, if a noun is followed by a verb, it likely indicates the end of an entity, thereby making the boundaries of the entity clearer.

After the NER is completed, we introduce the py2neo toolkit to read the entities and relationships obtained using NER, relationship path and entity description, traverse the file to generate node trees and store them in the list, then convert them into subgraph instances. Using the tool classes written according to rules and query logic, a large amount of knowledge is efficiently written into the knowledge graph. We select the travel topic knowledge graph as the construction instance, and the display effect is shown in the Fig. 4. The "Travel" is divided into two categories: "Era" and "Resource". The "Era" is divided into two parts: "History" and "Modern", among which "Folk Custom", "Religion", "Traditional Chinese Opera", "Minority Culture", "Painting and Calligraphy Art" and "Landscape Architecture" all belong to the "History", while "Contemporary Art", "Popular Art", "Natural Science", "Film and Television" and "Modern Architecture"

 Table 2
 Comparison of experimental results of NER

	Results			
Model	Precision(%)	Recall(%)	F1(%)	
BERT-POS-CRF	89.78	88.31	89.04	
CRF	79.65	77.18	78.40	
BiLSTM-CRF	87.19	88.24	87.71	
BERT-CRF	85.29	84.36	84.82	



Fig. 4 Display effect of knowledge graph

all belong to the "Modern". Meanwhile, the "Resource" is divided into two parts: "Natural Landscape" and "Artificial Facility". "Natural Landscape" includes tourist places such as "Ocean", "Mountain Range", "Steppe", "Forest", "Shadow" and "Inland Waters", and "Artificial Facility" includes tourist places such as "Zoo", "Park", "Museum", "Ancient Town", "Holiday Village" and "Snack Street". Based on the effectiveness of the graph construction, our MEC data integration scheme intuitively presents the content related to various topics across multi-documents, which successfully encompasses entities and relationships within MEC text data, thereby laying a solid foundation for the subsequent automatic generation of text summarization.

Given that the number of documents in multi-document summarization can vary, to demonstrate that our model can sufficiently preserve the original semantics, we calculated the semantic similarity of the summarizations generated by KGCPN across a range of 2 to 8 documents in three categories: travel, weather, and news. The comparison results of the semantic similarity are depicted in Fig. 5.

As can be seen from the experimental results in Fig. 5, the semantic similarity of the text summarization generated by KGCPN for different types of texts of 2-8 documents is higher than 0.8. The experimental results show that KGCPN can effectively reflect the themes that need to be expressed in different types of texts, and have good generalization performance.

To further verify the reliability of our scheme, we select the text evaluation index Rouge to evaluate the generated abstracts to check their quality and accuracy. The experimental results of text summarization are shown in Fig. 6. It can be seen from Fig. 6 that the Rouge-1, Rouge-2 and Rouge-L of the KGCPN are all higher than 0.9. The experimental results show that the KGCPN takes



Fig. 5 Comparison results of semantic similarity



Fig. 6 Comparison of experimental results of KGCPN on Rouge

into account the repetition of a single word, two consecutive words or long words, which makes the automatically generated summarization more superior and suitable for a wider range of applications. To illustrate the superiority of KGCPN over traditional methods, we compared its performance with the extractive TextRank method, the abstractive pure PGN model, and the BERT-PGN model that incorporates pre-trained models, all tested on the dataset used in this paper. The comparative results, as demonstrated by their respective Rouge scores, are presented in Table 3. While extractive methods exemplified by TextRank have a lower error rate, the sentences they extract are often redundant and struggle to succinctly encapsulate the original meaning, which clearly does not meet the requirements for multi-document summarization in the MEC environment. In comparison to abstractive models like PGN, KGCPN extracts semantic features

Table 3 Comparison of experimental results on Rouge

	Results			
Model	Rouge-1	Rouge-2	Rouge-L	
KGCPN	0.95	0.90	0.91	
TextRank	0.64	0.58	0.62	
PGN	0.78	0.70	0.75	
BERT-PGN	0.89	0.79	0.86	

of entities from the data integration stage, enriching the knowledge graph based on relationship paths and descriptive information within the multi-document texts. Subsequently, the information from the knowledge graph is parsed by the GCN and transmitted to the PGN, enabling our model to better capture the structure and dependencies between nodes in multi-documents, thereby achieving the best Rouge scores and summarization results. In addition, the difference in performance between the extractive and generative methods is mainly due to the requirements for information coherence and language fluency. When summarizing multi-documents, key information may be scattered across multiple sections, which may not be easily extracted and consolidated to form a cohesive summarization. Generative methods can rephrase and assimilate this scattered information to generate more coherent and scattered summarization. Finally, enhanced language fluency and readability within dialogue, traffic, weather, and other situations in MEC environments align more closely with human reading preferences, which also contributes to the superior performance of generative methods under the demands of the research presented in this paper.

To ascertain the superior generation efficiency of our model in MEC environments, we simulated the

Model	Normal			Edge		
	BERT	PGN	KGCPN	BERT	PGN	KGCPN
Travel	3.62 s	3.54 s	<b>3.24</b> s	2.61 s	2.37 s	<b>2.15</b> s
Traffic	5.98 s	5.91 s	<b>5.64</b> s	4.14 s	3.95 s	<b>3.82</b> s
Weather	3.48 s	3.35 s	<b>3.07</b> s	3.36 s	3.14 s	<b>2.99</b> s
Dialogue	2.10 s	1.97 s	<b>1.82</b> s	1.88 s	1.75 s	<b>1.68</b> s
News	3.16 s	3.05 s	<b>2.81</b> s	2.92 s	2.87 s	<b>2.46</b> s

 Table 4
 Performances of each model under varying environments and situations

The generation time is the total time required to calculate from the start of generation to the completion of output

prediction process of three models: BERT, PGN, and KGCPN, under the influence of continuously evolving multi-document contexts in both normal single node and multi-node edge environments. We recorded the total time for generating complete summarization. The experiments encompassed five situations: Travel, Traffic, Weather, Dialogue, and News. KGCPN necessitates two additional preprocessing steps before the deployment. Initially, to expedite model computations, particularly on MEC devices capable of low-precision calculations, we reduce model size through quantization operations to accelerate computation, which is manifested by converting the floating point weights in the model to low precision numbers (such as 8-bit integers). Subsequently, to accommodate the formats of mobile and embedded devices, we employ the TensorFlow Lite tool to transform the model into a format capable of independent operation. Table 4 illustrates the performance of each model under varying environments and situations.

From Table 4, it is evident that our model outperforms the others under all situations. However, there is a significant difference in model performances between dialogue and traffic situations, mainly due to the data characteristics and processing methods [38] in the two situations. Generally, the length of the text being processed is directly proportional to the time required for generation; the longer the text, the more time is needed. Because the model needs to read and comprehend a larger amount of content to produce an accurate summarization. Therefore, in the context of dialogue situations, the generation of text summarization is typically faster due to the shorter length of the input text. Conversely, data pertaining to traffic environments can encompass a variety of information types, including road conditions, weather, traffic accidents, and traffic rules, among others, which may originate from diverse sources, and the format and structure can vary significantly. Moreover, traffic information usually requires real-time updates, and processing these multifaceted data requires more time and computational resources, thereby increasing the difficulty and time required to generate the summarization.

In the normal environment, KGCPN exhibits an average improvement of 7.37% across all situations when compared to the PGN model. This underscores the significant enhancements brought about by the knowledge graph and GCN methods, which deeply extract semantic and entity information from multi-documents. Furthermore, in edge environments, our model demonstrates an average performance improvement exceeding 7.12% in various situations. This improvement is not solely due to the model itself but also because it is directly deployed on MEC devices. The elimination of the process of transmitting large amounts of original text back to the central server facilitates faster user request responses and enhances the overall system efficiency.

## Conclusion

This article introduces KGCPN, a text summarization scheme based on knowledge graphs and GCN, designed for MEC text data preprocessing and integration. By constructing a knowledge graph and encoding the GCN representation of text data derived from MEC devices and applications, the generation quality of multi-document text abstracts has been significantly enhanced. Moreover, performance evaluations conducted in edge environments demonstrate that KGCPN outperforms existing schemes in terms of generation efficiency. The approaches in this paper offer room for further enhancement. For instance, during the graph embedding process, consideration could be given to more diverse embedding methods, such as TransA and TransD. Additionally, optimizing the scope and volume of MEC data collection could prove beneficial. Since we have trained KGCPN within five common situations, future work will involve adapting the model to more situations. A promising work is employing generative large models in encoding. Furthermore, the exploration of multimodal MEC data mining also presents a valuable direction for future work.

#### Authors' contributions

The authors confirm contributions to the paper as follows: study conception and design: Zheng Yu and Songyu Wu; drawing and grammar guidance: Jielin Jiang and Dongqing Liu. All authors reviewed the results and approved the final version of the manuscript.

#### Funding

This study was supported by the National Natural Science Foundation of China (No. 42050102) and the Future Network Scientific Research Fund Project (No.FNSRFP-2021-YB-18).

### Availability of data and materials

The data are not publicly available due to privacy restriction and public security.

#### Declarations

#### **Competing interests**

The authors declare no competing interests.

Received: 9 November 2023 Accepted: 28 December 2023 Published online: 06 January 2024

### References

- Zhang L, Wang S, Chang RN (2018) QCSS: a qoe-aware control plane for adaptive streaming service over mobile edge computing infrastructures. https://doi.org/10.1109/ICWS.2018.00025
- Ongsulee P (2017) Artificial intelligence, machine learning and deep learning. In: 2017 15th international conference on ICT and knowledge engineering (ICT &KE), IEEE, pp 1–6
- Weis JW, Jacobson JM (2021) Learning on knowledge graph dynamics provides an early warning of impactful research. Nat Biotechnol 39(10):1300–1307
- Rassil A, Chougrad H, Zouaki H (2023) Deep multi-agent fusion q-network for graph generation. Knowl-Based Syst 269:110509
- Tempelmeier N, Demidova E (2021) Linking openstreetmap with knowledge graphs—link discovery for schema-agnostic volunteered geographic information. Futur Gener Comput Syst 116:349–364
- 6. Alharbi M, Roach M, Cheesman T, Laramee RS (2021) VNLP: Visible natural language processing. Inf Vis 20(4):245–262
- Akay H, Kim SG (2021) Reading functional requirements using machine learning-based language processing. CIRP Ann 70(1):139–142
- 8. Hanxiao Q, Bin H (2022) Entity relation retrieval and simulation of low redundancy knowledge atlas. Comput Simul 39(6):4
- 9. Kipf TN, Welling M (2016) Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907
- Xu X, Tang S, Qi L, Zhou X, Dai F, Dou W (2023) CNN partitioning and offloading for vehicular edge networks in web3. IEEE Commun Mag 61(8):36–42. https://doi.org/10.1109/MCOM.002.2200424
- Li Z, Xu X, Hang T, Xiang H, Cui Y, Qi L, Zhou X (2022) A knowledge-driven anomaly detection framework for social production system. IEEE Trans Comput Soc Syst 1–14. https://doi.org/10.1109/TCSS.2022.3217790
- 12. Yan H, Bilal M, Xu X, Vimal S (2022) Edge server deployment for health monitoring with reinforcement learning in internet of medical things. IEEE Trans Comput Soc Syst 1–11. https://doi.org/10.1109/TCSS.2022.3161996
- Li Z, Xu X, Cao X, Liu W, Zhang Y, Chen D, Dai H (2022) Integrated CNN and federated learning for covid-19 detection on chest x-ray images. IEEE/ACM Trans Comput Biol Bioinforma 1–11. https://doi.org/10.1109/ TCBB.2022.3184319
- Liu W, Xu X, Wu L, Qi L, Jolfaei A, Ding W, Khosravi MR (2023) Intrusion detection for maritime transportation systems with batch federated aggregation. IEEE Trans Intell Transp Syst 24(2):2503–2514. https://doi. org/10.1109/TITS.2022.3181436

- Yao L, Xu X, Bilal M, Wang H (2023) Dynamic edge computation offloading for internet of vehicles with deep reinforcement learning. IEEE Trans Intell Transp Syst 24(11):12991–12999. https://doi.org/10.1109/TITS.2022. 3178759
- Tian H, Xu X, Lin T, Cheng Y, Qian C, Ren L, Bilal M (2022) Dima: Distributed cooperative microservice caching for internet of things in edge computing by deep reinforcement learning. World Wide Web 25(5):1769–1792. https://doi.org/10.1007/s11280-021-00939-7
- Ayetiran EF, Sojka P, Novotný V (2021) Eds-membed: Multi-sense embeddings based on enhanced distributional semantic structures via a graph walk over word senses. Knowl-Based Syst 219:106902
- Koloski B, Perdih TS, Robnik-Šikonja M, Pollak S, Škrlj B (2022) Knowledge graph informed fake news classification via heterogeneous representation ensembles. Neurocomputing 496:208–226
- Tiwari P, Zhu H, Pandey HM (2021) DAPath: distance-aware knowledge graph reasoning based on deep reinforcement learning. Neural Netw 135:1–12
- Dethlefs N, Schoene A, Cuayáhuitl H (2021) A divide-and-conquer approach to neural natural language generation from structured data. Neurocomputing 433:300–309
- Keshan N, Fontaine K, Hendler JA (2022) Semiautomated process for generating knowledge graphs for marginalized community doctoralrecipients. Int J Web Inf Syst 18(5/6):413–431
- Sanchez-Gomez JM, Vega-Rodríguez MA, Pérez CJ (2022) A multi-objective memetic algorithm for query-oriented text summarization: medicine texts as a case study. Expert Syst Appl 198:116769
- Mojrian M, Mirroshandel SA (2021) A novel extractive multi-document text summarization system using quantum-inspired genetic algorithm: Mtsqiga. Expert Syst Appl 171:114555
- Gupta S, Gupta SK (2021) An approach to generate the bug report summaries using two-level feature extraction. Expert Syst Appl 176:114816
- Xu X, Liu Z, Bilal M, Vimal S, Song H (2022) Computation offloading and service caching for intelligent transportation systems with digital twin. IEEE Trans Intell Transp Syst 23(11):20757–20772. https://doi.org/10.1109/ TITS.2022.3190669
- Singh S, Siwach M (2022) Handling heterogeneous data in knowledge graphs: a survey. J Web Eng 21(4):1145–1186
- Borrego A, Ayala D, Hernández I, Rivero CR, Ruiz D (2021) Cafe: Knowledge graph completion using neighborhood-aware features. Eng Appl Artif Intell 103:104302
- Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781
- Pennington J, Socher R, Manning CD (2014) Glove: Global vectors for word representation. In: Alessandro M, Bo P, Walter D (eds) Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Doha, Qatar, pp 1532–1543. https://aclanthology.org/D14-1162, https://doi.org/10.3115/ v1/D14-1162
- Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L (2018) Deep contextualized word representations. CoRR abs/1802.05365. arXiv:1802.05365
- Bordes A, Usunier N, Garcia-Duran A, Weston J, Yakhnenko O (2013) Translating embeddings for modeling multi-relational data. Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2. Curran Associates Inc., Lake Tahoe, Nevada, pp 2787–2795
- Sitar-Tăut DA, Mican D, Buchmann RA (2021) A knowledge-driven digital nudging approach to recommender systems built on a modified Onicescu method. Expert Syst Appl 181:115170
- Devlin J, Chang M, Lee K, Toutanova K (2018) BERT: pre-training of deep bidirectional transformers for language understanding. CoRR abs/1810.04805. arXiv:1810.04805
- Oliveira D, D'aquin M, (2022) Extracting data models from background knowledge graphs. Knowl-Based Syst 237:107818
- Žagar A, Robnik-Šikonja M (2022) Cross-lingual transfer of abstractive summarizer to less-resource language. J Intell Inf Syst 58:153–173. https:// doi.org/10.1007/s10844-021-00663-8
- See A, Liu PJ, Manning CD (2017) Get to the point: Summarization with pointer-generator networks. arXiv preprint arXiv:1704.04368
- 37. Lin CY, Hovy E (2003) Automatic evaluation of summaries using n-gram co-occurrence statistics. Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics

on Human Language Technology - Volume 1. Association for Computational Linguistics, Edmonton, Canada, pp 71–78. https://doi.org/10.3115/ 1073445.1073465

 Xu X, Gu J, Yan H, Liu W, Qi L, Zhou X (2023) Reputation-aware supplier assessment for blockchain-enabled supply chain in industry 4.0. IEEE Trans Ind Inform 19(4):5485–5494. https://doi.org/10.1109/TII.2022.3190380

# **Publisher's Note**

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

# Submit your manuscript to a SpringerOpen<sup>⊗</sup> journal and benefit from:

- Convenient online submission
- ► Rigorous peer review
- ► Open access: articles freely available online
- ► High visibility within the field
- ► Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com