sustainability

MDPI

Article

Graph Embedding-Based Domain-Specific Knowledge Graph Expansion Using Research Literature Summary

Junho Choi

Division of Undeclared Majors, Chosun University, Gwangju 61452, Korea; xdman@chosun.ac.kr

Abstract: Knowledge bases built in the knowledge processing field have a problem in that experts have to add rules or update them through modifications. To solve this problem, research has been conducted on knowledge graph expansion methods using deep learning technology, and in recent years, many studies have been conducted on methods of generating knowledge bases by embedding the knowledge graph's triple information in a continuous vector space. In this paper, using a research literature summary, we propose a domain-specific knowledge graph expansion method based on graph embedding. To this end, we perform pre-processing and process and text summarization with the collected research literature data. Furthermore, we propose a method of generating a knowledge graph by extracting the entity and relation information and a method of expanding the knowledge graph using web data. To this end, we summarize research literature using the Bidirectional Encoder Representations from Transformers for Summarization (BERTSUM) model based on domain-specific research literature data and design a Research-BERT (RE-BERT) model that extracts entities and relation information, which are components of the knowledge graph, from the summarized research literature. Moreover, we proposed a method of expanding related entities based on Google news after extracting related entities through the web for the entities in the generated knowledge graph. In the experiment, we measured the performance of summarizing research literature using the BERTSUM model and the accuracy of the knowledge graph relation extraction model. In the experiment of removing unnecessary sentences from the research literature text and summarizing them in key sentences, the result shows that the BERTSUM Classifier model's ROUGE-1 precision is 57.86%. The knowledge graph extraction performance was measured using the mean reciprocal rank (MRR), mean rank (MR), and HIT@N rank-based evaluation metric. The knowledge graph extraction method using summarized text showed superior performance in terms of speed and knowledge graph quality.

Keywords: knowledge graph expansion; knowledge graph embeddings; relationship extraction

Citation: Choi, J. Graph Embedding-Based Domain-Specific Knowledge Graph Expansion Using Research Literature Summary. *Sustainability* **2022**, *14*, 12299. https://doi.org/10.3390/su141912299

Academic Editors: Hyunchul Ahn and Luigi Aldieri

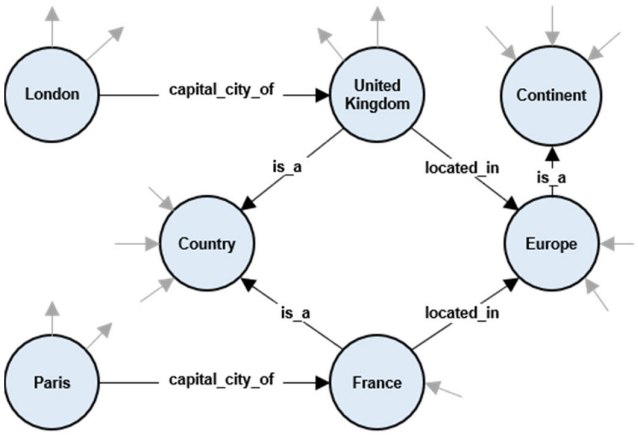
Received: 30 July 2022
Accepted: 23 September 2022
Published: 27 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

Recently, research has been conducted on how to build a knowledge base automatically using natural language processing technology and artificial neural network technology [1]. The process of building a knowledge base can be divided into the entity linking process and the relation extraction process. The first is concerned with finding an entity's needs in the knowledge base from texts expressed in natural language, and the latter identifies the relationships between entities [2]. The knowledge base has to be updated continuously with new knowledge, which results in the problem that experts have to add or modify rules. To solve this problem, there is an increasing number of studies on deep learning-based knowledge graph expansion methods [3]. A knowledge graph shows a graph of relationships between concepts based on a method of representing the entities and relations extracted from various texts. Figure 1 shows an example of how to represent an entity-relationship of a knowledge graph as a graph. Knowledge graphs are used as an important resource in the knowledge processing field and the deep learning field [4].

Specifically, they are applied to various systems, such as question answering (QA) systems, recommendation systems, and knowledge inference systems. Knowledge graphs can be used to combine and manage knowledge data from various sources [5].



A sample part of the knowledge graph

	Head	Relation	Tail
1	London	capital_city_of	United Kingdom
2	United Kingdom	is_a	Country
3	United Kingdom	located_in	Europe
4	Europe	is_a	Continent
5	Paris	capital_city_of	France
6	France	is_a	Country
7	France	located_in	Europe

Triple Instances from a sample part of the knowledge graph

Figure 1. An example of an entity-relationship of a knowledge graph.

Furthermore, knowledge graphs provide a method for reconstructing new knowledge through link predictions using inference for entity-relation. Previous studies on the generation of knowledge graphs used rule-based or neural network model-based methods [6]. Studies on named entity recognition (NER) have usually used a text string-matching method with a pre-built entity name dictionary or used word class patterns, such as noun + noun and a pronoun, through morphological analysis based on natural language processing techniques. Furthermore, for relation extraction, people have primarily used rule-based methods, which identify relationships of entities extracted through the NER process based on sentence structure analysis, such as dependency parsing, or extract the relationship by comparison with a specific pattern through semantic role labeling [7]. However, these traditional methods have disadvantages. They require a huge amount of time and manpower, and the pre-processing process has a significant impact on the performance or the quality of the results [8]. To overcome these disadvantages, many studies have been conducted recently on methods of generating a knowledge base by embedding the knowledge graph's triple information into a continuous vector space [9]. Knowledge graph embedding refers to training performed by representing the relations between the entities of a knowledge graph as vectors to satisfy a specific function. In knowledge graph embedding, the knowledge base's triple ($Entity^{Head}$, Relation, $Entity^{Tail}$) is embedded into a K-dimensional vector space.

In this paper, we propose a domain-specific knowledge graph expansion method based on graph embedding using a research literature summary. To this end, we perform the pre-processing process and text summarization for the collected research literature data. Furthermore, we propose a method for generating a knowledge base by extracting the entity and relation information from the summarized text and a method for expanding the knowledge graph using web data. The structure of this paper is as follows. Section 2 describes methods and deep learning-based trained language models used in previous studies for automatic knowledge graph generation and relation extraction. Section 3 explains the domain-specific knowledge graph expansion method proposed in this study. In Section 4, we measure the proposed method's performance to validate the methodology proposed in this paper, and Section 5 presents the conclusion.

2. Related Work

2.1. Automatic Knowledge Graph Generation and Expansion

A knowledge graph consists of a relation triple in the form of <subject-relation-object> to represent the relationship between two entities. Each entity in the knowledge graph is represented as a node, and the relation is represented as an edge linking two nodes. Recently, various large-scale knowledge graphs, such as DBpedia, Wikidata, and Freebase, have been constructed. Large-scale knowledge graphs are constructed using information that can be accessed through the web, such as news, blogs, social media, and Wikipedia. These knowledge graphs are used as key resources for services in a variety of application fields, such as recommendation systems, QA, and conversation systems [10,11]. Automatic knowledge graph generation and expansion methods are essential techniques for extracting new knowledge from various and vastly added data and continuously updating existing knowledge graphs. Methods of automatically generating and expanding knowledge graphs can be classified into link prediction and relation prediction [12,13]. Link prediction is a method of finding an object by learning the vector representation of the subject and its relation in a relation triple of <Subject-Relation-?>. In contrast, relation prediction is a method of finding the relationship by learning the vector representation of the subject and object in a relation triple of <Subject-?- Object>. Link prediction is a method of generating a new triple. In this method, a new relation between entities included in the knowledge graph is found, and it is then used to generate a new triple. Although link prediction cannot find new entities, it can find missing relations in an existing knowledge graph, thereby supplementing insufficient knowledge in the knowledge graph. Relation prediction generates a new triple by predicting a missing relation from among the relations between entities in a given knowledge graph. It is used as an essential technique for automatic knowledge graph generation and expansion because it can find missing relations with only a given knowledge graph without relying on any external corpus [14,15]. Furthermore, various studies are underway on knowledge completion for the generation and expansion of knowledge graphs. In particular, knowledge graph embedding is a typically used model in link prediction problems.

The knowledge graph embedding model expresses entities and relations as vectors in a low-dimensional space and a matrix, respectively. The vector values are learned to optimize the score function defined in the knowledge graph embedding model [16]. A new triple is generated through the learned vectors and the score function. Among the knowledge graph embedding methods, translation-based embedding models have shown excellent performance. Typical models of translation-based embedding include TransE, TransH, ComplEx, and Distmult [17]. Translation-based embedding models represent all entities and relations as vectors in the embedding space, and a relation is a vector that has the role of an operator for the transition of a subject entity into an object entity. When <Subject-Relation-Object>, the components of a triple, are given, a translation-based graph embedding model finds vectors for which the vector value of the object entity in the embedding space is the same value as the sum vector of the subject entity and relation vectors [18]. The example shown in Figure 2 illustrates the TransE learning method, which is a basic knowledge graph embedding method [19].

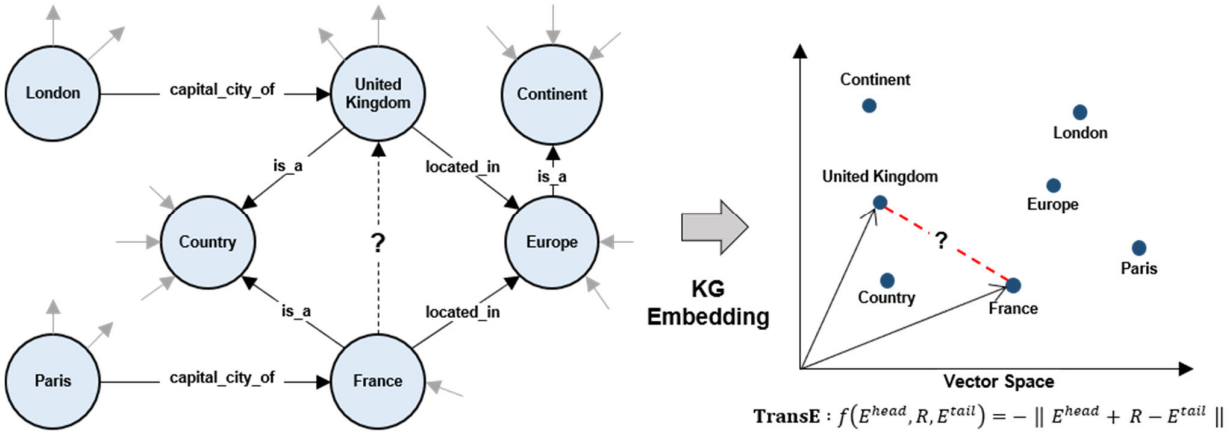


Figure 2. An example of a knowledge graph embedding for relationship inference.

2.2. Deep Learning-Based Pre-Trained Language Model

Typical techniques required for knowledge graph generation are NER and relation extraction between entities [20]. Named entity recognition refers to the task of recognizing an entity that has a unique name in a text [21]. It recognizes the category that the entity belongs to from among pre-defined categories, such as Person, Organization, Location, and Event. Relation extraction is the process of identifying the relationship between two entities after performing NER. Before deep learning technology was applied, a large entity name dictionary constructed in advance was used for NER, or a named entity was recognized using a combination of words formed with a specific pattern, such as noun + noun. For relation extraction, a rule-based task was usually performed, which identifies the relation by identifying the sentence structure through dependency parsing of the sentence for the extracted entities or comparing it with a specific pattern through semantic role labeling [22]. These traditional methods require a long time and a huge amount of manpower because a large dictionary has to be constructed, or all patterns of the sentence structures have to be defined. Furthermore, they have the disadvantage that the manual pre-processing method has a significant impact on the knowledge graph generation result [23]. Deep learning-based language models have the advantage that human intervention can be minimized because the neural network models learn the language patterns on their own during the training process, and they outperform traditional methods that analyze patterns or use dictionaries [24]. Among the deep learning-based pre-trained language models, the Bidirectional Encoder Representations from the Transformers (BERT) model is a model that is pre-trained using a large corpus with a volume of about 16 GB, such as Wikipedia, web documents, and books [25]. Further, it is a bidirectional language model that uses multiple encoder structures of the transformers model [26]. The pre-training process of the BERT model refers to the method of learning similar data in a large volume for solving problems that need to be solved in the future [27]. In this paper, the BERTSUM language model is used for the research literature text summarization [28]. BERTSUM is a structure proposed for summarizing documents using the BERT model and is a pre-trained model. The output result is processed at the token level, not the sentence level, and the (CLS) token (Special Classification token), which is input before all tokens, is used. In document summarization, semantic summarization can be performed at the sentence level as sentences that contain key content are selected by identifying the semantic relationships between texts [29]. In the BERTSUM model, the (CLS) token is attached to the input text to learn the representative sentences, and segment embedding is entered as an input to distinguish each text.

3. Graph Embedding-Based Domain-Specific Knowledge Graph Expansion

3.1. Overall Framework

This section describes the graph embedding-based domain-specific knowledge expansion framework. This framework consists of the following: a data preparation layer, in which texts are extracted and pre-processed from research literature for a specific domain; a text summarization layer, which is a BERTSUM-based text summarization layer, which removes sentences that are not needed in the knowledge graph generation, while preserving the topics of the collected text in the specialization field; a knowledge graph construction layer, a Research-BERT (RE-BERT) model-based knowledge graph generation layer, which extracts entities and relation information—the components of the knowledge graph—from the summarized research literature text data; a knowledge graph expansion layer, which extracts related entities from Wikipedia pages for the entities of the generated knowledge graph and then adds the knowledge graphs for those entities through Google News. Figure 3 illustrates the overall framework of this graph embedding-based domain-specific knowledge graph expansion method.

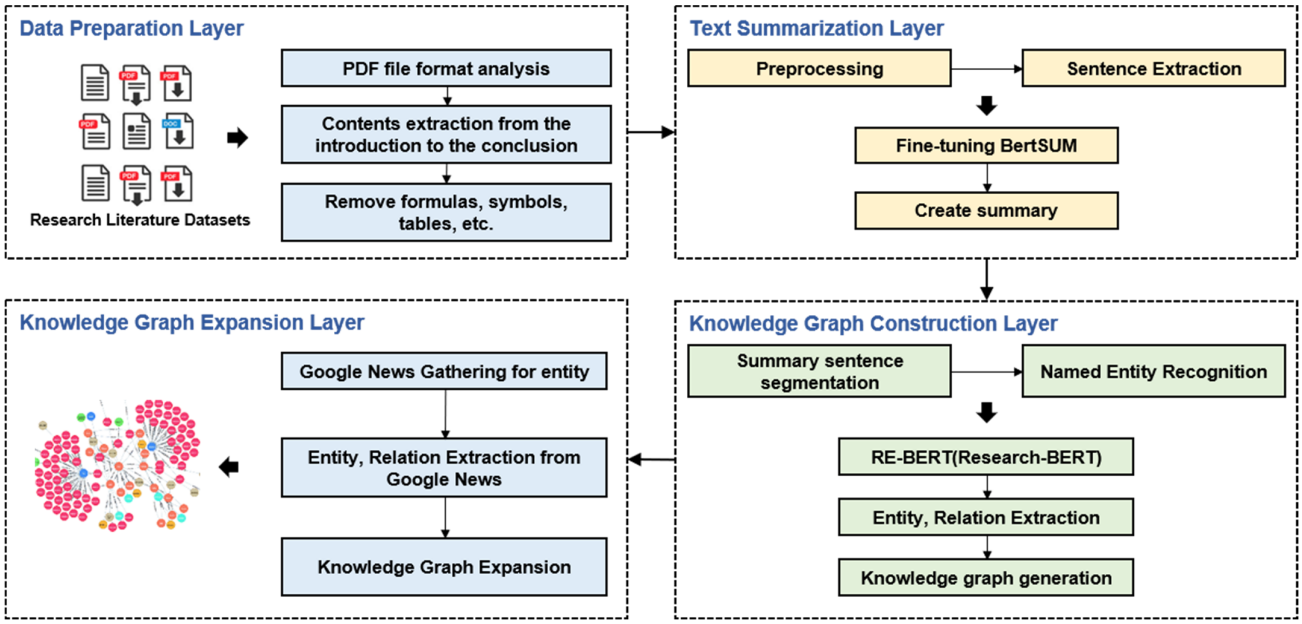


Figure 3. The overall framework.

3.2. Pre-Processing and Summarization of Research Literature

In this paper, we need a document summarization method that considers semantic relations between texts while maintaining the characteristics of the research topics in a large volume of research literature for efficient knowledge graph generation from research literature texts. To achieve this, we use a document summarization method that can effectively include the topics of research literature using BERTSUM. In general, as most of the research literature is stored online in pdf format, the process of extracting and pre-processing text from pdf format documents is a very important stage for data quality. The content from the introduction to the conclusion of the paper is extracted so as to extract only the core content of the research literature. The order of the words in the text is also important for the content summarization of research literature. Therefore, clauses in parentheses, reference labels in square brackets, equations, and special characters are removed. Moreover, information unnecessary for paper summarization, such as tables, figures, and relevant text in a pdf document, is deleted. The content summary of the research literature should maintain the important topics and content characteristics of the entire document. Content summarization of research literature for knowledge graph extraction

is performed through the extraction of important sentences. A BERTSUM-based abstractive summarization model is used for this. BERTSUM accepts text divided at the sentence level as input and produces an importance score for each sentence as output. BERTSUM is a pre-trained BERT-based summarization model. Figure 4 shows the architecture of BERTSUM, in which six layers of transformer encoder are added to the traditional BERT for research literature summarization [30]. BERTSUM inserts a (CLS) token at the beginning of the input data sentence. The output (CLS) token vector is selected through BERT and sent to the transformer encoder. The (CLS) token vector of each sentence is output with a value between 0 and 1. The larger the output value is, the better the summarization performance and the more important the sentence [31]. In the segment embedding layer, sequences in the text are classified using interval segment embeddings. Depending on whether a sentence is an even or odd number, it is assigned to E_A or E_B. BERTSUM assigns a score to the sentence set according to the value added to the whole text and then rearranges the sentence with the highest score to generate the text's summary [31]. Because the encoder is pre-trained, and the decoder is trained from the beginning, data may be overfitted in one of them and underfitted in the other one.

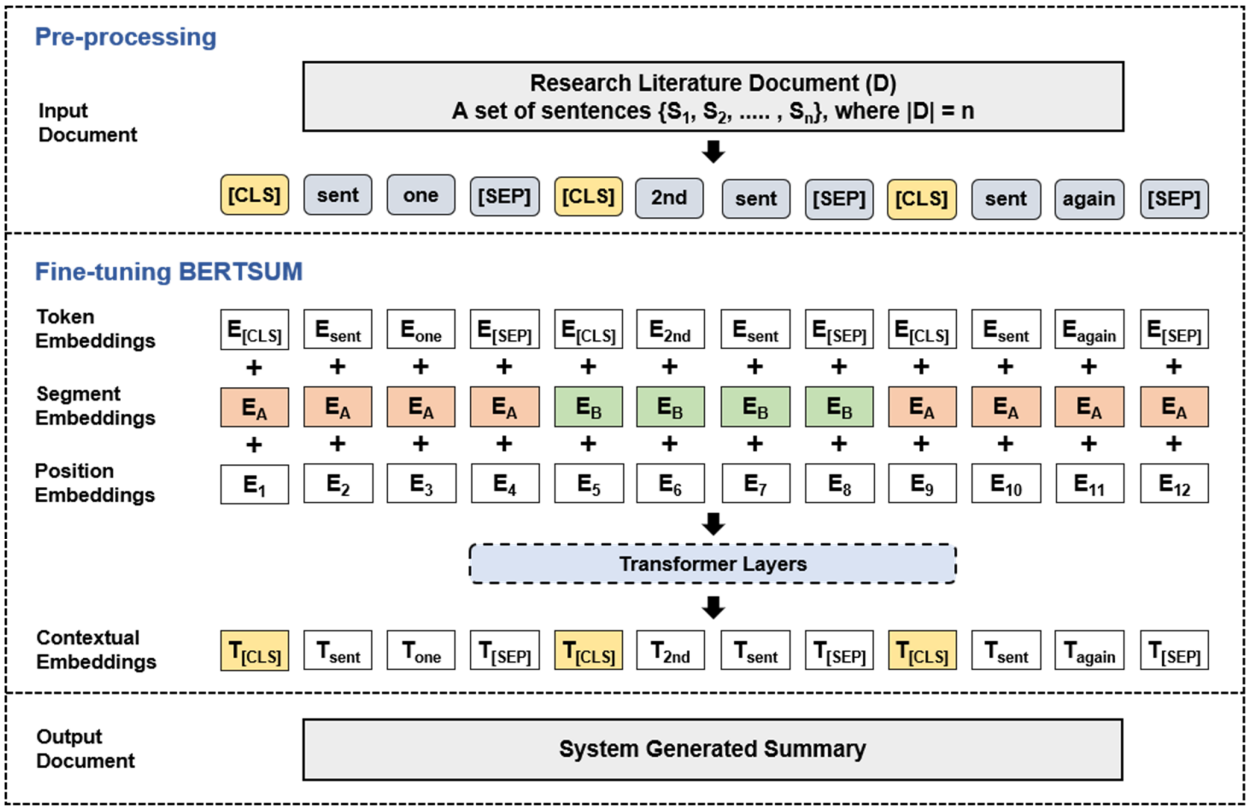


Figure 4. The architecture of BERTSUM.

For fine-tuning, we use Adam optimizers and use $\beta_1 = 0.9$ in the encoder and $\beta_2 = 0.999$ in the decoder. The following Equations (1) and (2) are used for the learning rates of the encoder and the decoder.

$$lr_E = \tilde{l}r_E \cdot min\left(step^{-0.5}, step.warmup_E^{-1.5}\right) \quad (1)$$

$$lr_D = \tilde{l}r_D \cdot min\left(step^{-0.5}, step.warmup_D^{-1.5}\right) \quad (2)$$

The pre-trained encoder is fine-tuned with a low learning rate. In the case of the encoder, $warmup_E = 20{,}000$ and $\tilde{l}r_E = 2 \times 10^{-3}$, and in the case of the decoder, $warmup_D = 10{,}000$ and $lr_D = 0.1$. Figure 5 shows the result of summarizing a text of the research

literature data using the BERTSUM model. The original text containing a total of 250 words is summarized in 98 words, showing a summarization rate of 61%. The goal of text summarization in this paper is to generate an efficient knowledge graph using research literature texts that contain substantial amounts of technical terms and numerical information. To this end, the text summarization process is performed to extract keywords and sentences from the large amount of jargon used in the research literature and delete sentences and numerical information that are not required for the knowledge graph generation.

Source Text :

Recently, network intrusion attacks, particularly new unknown attacks referred to as zero-day attacks, have become a global phenomenon. Zero-day network intrusion attacks constitute a frequent cybersecurity threat, as they seek to exploit the vulnerabilities of a network system. Previous studies have demonstrated that zero-day attacks can compromise a network for prolonged periods if network traffic analysis (NTA) is not performed thoroughly and efficiently. NTA plays a crucial role in supporting machine learning (ML) based network intrusion detection systems (NIDS) by monitoring and extracting meaningful information from network traffic data. Network traffic data constitute large volumes of data described by features such as destination-to-source packet count. It is important to use only those features that have a significant impact on the performance of an NIDS. The problem is that most existing ML models for NIDS employ features such as Internet protocol (IP) addresses that are redundant for detecting zero-day attacks and therefore negatively impact the performance of these ML models. The solution proposed in this study demonstrates that the law of anomalous numbers, famously known as Benford's law, is a viable technique that can effectively identify significant network features that are indicative of anomalous behaviour and can be used for detecting zero-day attacks. Finally, our study illustrates that semi-supervised ML approaches are effective for detecting zero-day attacks if significant features are optimally chosen. The experimental results demonstrate that one-class support vector machines achieved the best results (Matthews correlation coefficient of 74% and F1 score of 85%) for detecting zero-day network attacks.

Summary Results :

Recently, network intrusion attacks, particularly new unknown attacks referred to as zero-day attacks, have become a global phenomenon. Zero-day network intrusion attacks constitute a frequent cybersecurity threat, as they seek to exploit the vulnerabilities of a network system. Previous studies have demonstrated that zero-day attacks can compromise a network for prolonged periods if network traffic analysis (NTA) is not performed thoroughly and efficiently. Network traffic data constitute large volumes of data described by features such as destination-to-source packet count. It is important to use only those features that have a significant impact on the performance of an NIDS.

Figure 5. The result of summarizing a text of the research literature data using the BERTSUM model.

3.3. Knowledge Graph Generation and Expansion Using Research Literature

Two entities and the relation information between these two entities—the essential elements of a knowledge graph—are extracted from the summary dataset of the research literature text. There is a high possibility that the entities in the research literature text are jargon in the pertinent field. Therefore, we need a NER model trained with jargon for the pertinent field. The entity extraction model consists mainly of a NER model, which extracts words from sentences, and a model that combines the extracted words to generate entity pairs. The NER model extracts all identifiable entities from the collected sentences, and the pair generation model combines all pairs of the entities to create an entity pair set. A pre-trained BERT model, which has been trained with a large amount of data, is created for NER in the summary text set.

To create the pre-trained BERT model, pre-processing data are generated after creating a vocabulary using the research literature text. Then, BERT is pre-trained using the pre-processing data to create a pre-trained RE-BERT model for research literature. The research literature text summary data set is divided into sentences for input to the pre-trained BERT model. Through the tokenization process, (CLS) and (SEP)—the special tokens that represent the beginning and end of a sentence—are added to the beginning and end of each sentence, which are then used as input values [32]. The vectors of the tokens, which are matched with the vocabulary constructed using the research literature summary text, are converted into a token embedding. For a token corresponding to a special

token ("(CLS)", "(SEP)") contained in the input data, the index is returned, and for a token that does not exist in the vocabulary, the index corresponding to "(UNK)" is returned, based on which token embedding is generated. As for the model's output value, after applying average pooling and max-pooling, which use the average value and the largest value, respectively, the result is produced by tagging the entity name through the fully connected layer and the SoftMax layer. Figure 6 shows the architecture of the named entity recognition module.

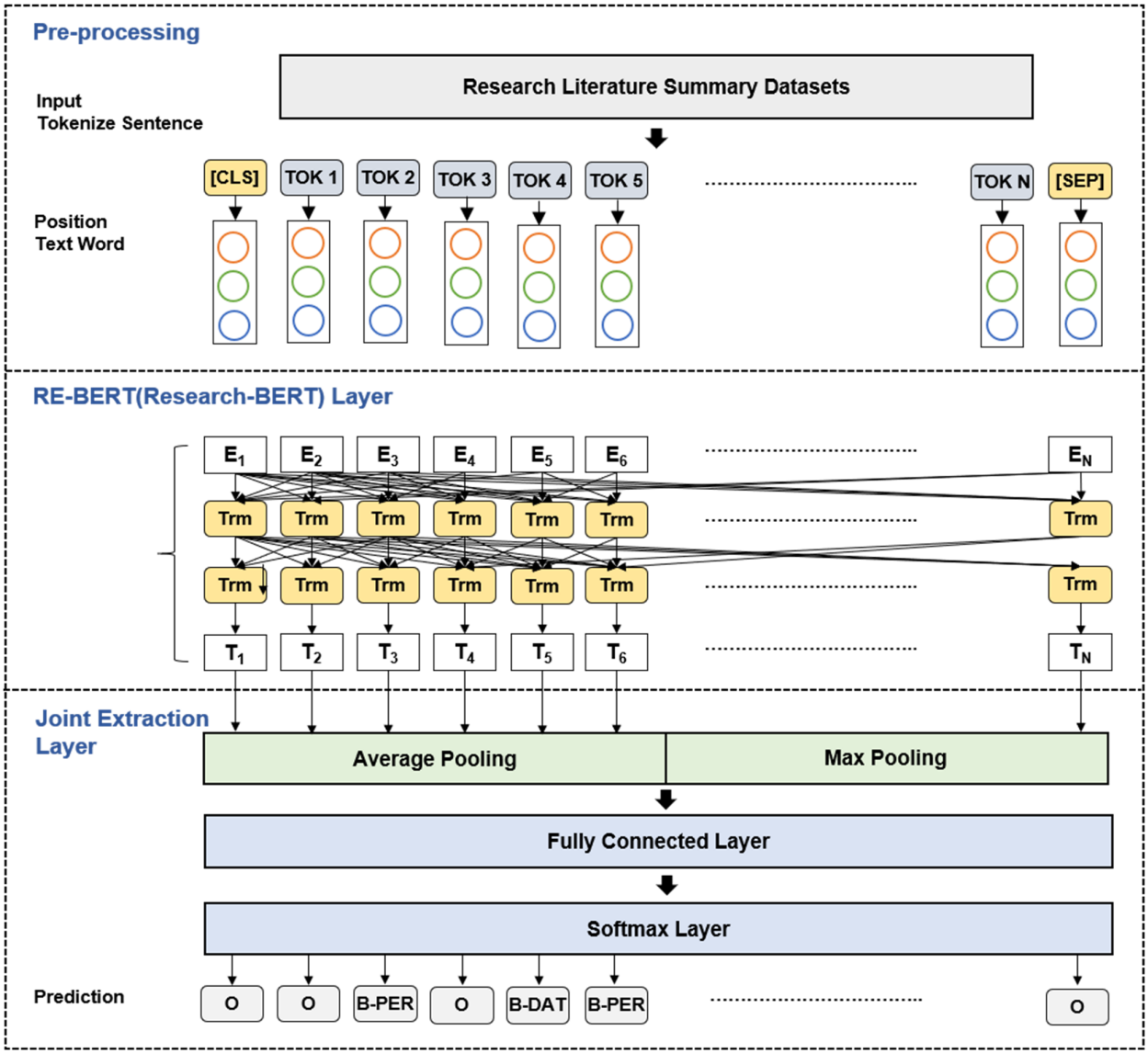


Figure 6. Entity Extraction Module.

As a knowledge graph is connected based on relations, the relation information between two entities contained in the sentence must be extracted [33]. To this end, the entity extraction results are used as input values in the relation extraction model. $Entity^{Head}$, $Entity^{Tail}$, $Entity^{HeadType}$, and $Entity^{TailType}$ are extracted from a sentence. The sentence, $Entity^{Head}$, and $Entity^{Tail}$ from the extracted results are input into the pre-trained language model, while an embedding process is performed separately for $Entity^{HeadType}$ and $Entity^{TailType}$. The research literature summary and the entity extraction model's results are saved as data, and if $Entity^{Head}$, an entity that becomes the subject word, appears more

than a certain number of times, it is sent to the model that extracts relation information. The sentence, $Entity^{Head}$ and $Entity^{Tail}$ are tokenized for application in the BERT model. To distinguish the tokenized results, a special token, (SEP), is used. For a tokenized input value, an output hidden state is obtained for each token through the BERT model. Based on this output value, average pooling and max-pooling are used to calculate the results. As the relation extraction is closely related to the entity type, an embedding process is performed for converting the $Entity^{Head}$ type and the $Entity^{Tail}$ type into 64-dimensional vectors, respectively, and for improving the performance of relation extraction. The vector values and the type-embedding values obtained by applying the average pooling and the max-pooling, respectively, to the BERT model's output values, are all connected. The connected results go through the fully-connected layer, and then the SoftMax operation is finally performed to classify the relation. Figure 7 shows the architecture of the named entity recognition module.

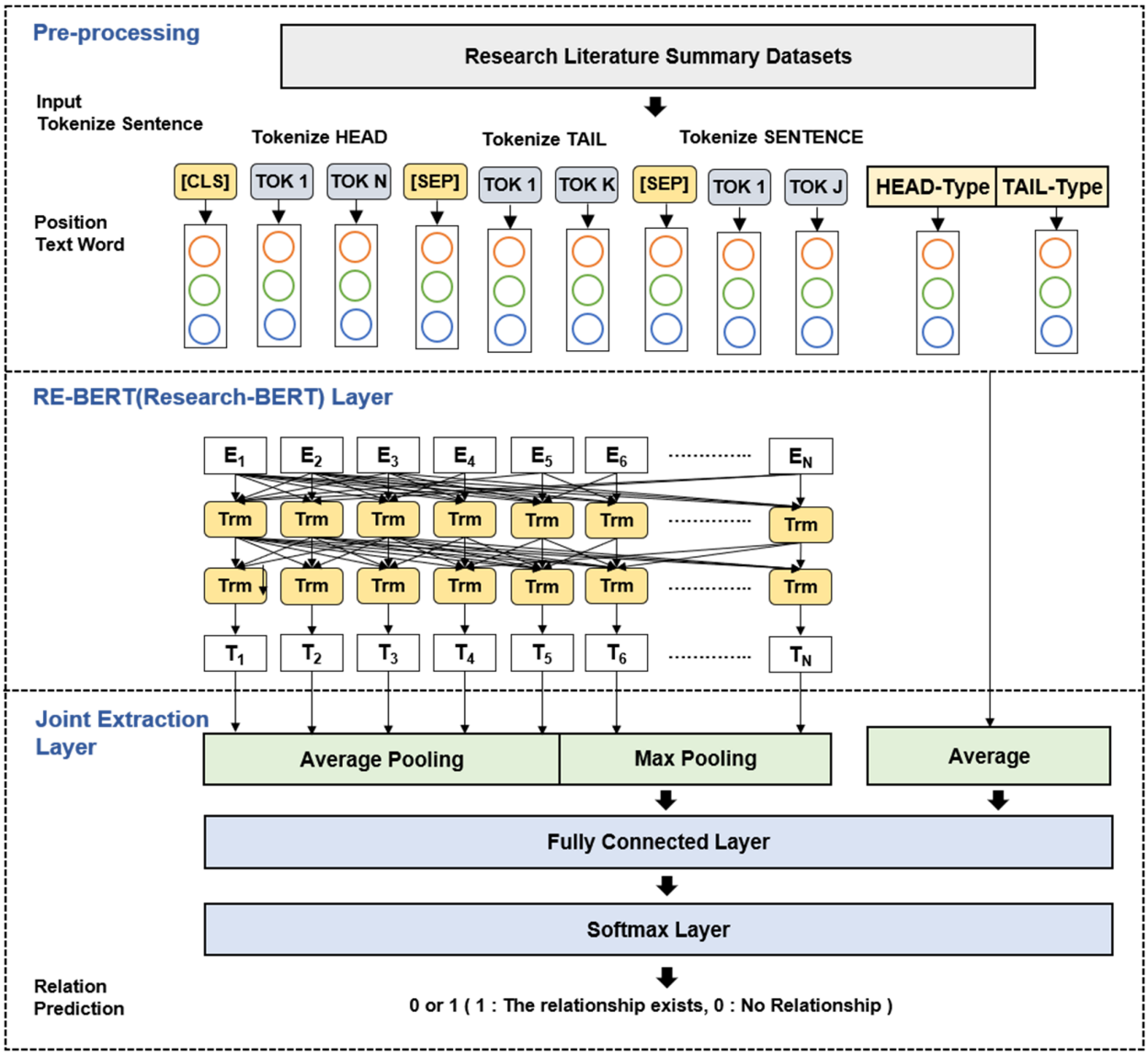


Figure 7. Relationship Extraction Module.

Figure 8 shows an example of extracting entity-entity pairs using the BERT model from the original text and summary text presented in Figure 5. In Figure 8, the relation

associated with the "Benford's law" entity was extracted from the original text of the research literature, but the entity was not extracted from the summary text. As the content of the original text is related to "network security", the "Benford's law" entity cannot be seen as a representative word of the whole text. Therefore, the summarization process of the knowledge graph generation for a specific domain using research literature text can have a significant impact on the quality of the entity-relation extraction. The knowledge graph expansion is performed based on the entities and relations extracted and using Wikipedia to expand the knowledge graph extracted from the research literature summary text. To this end, the Wikipedia page for the extracted entity is searched to check whether the entity is a registered entity in Wikipedia, and if it is a registered entity, the Wikipedia document for the entity is summarized and saved.

Source Text :

Input has 319 tokens
Input has 3 spans
Span boundaries are [[0, 128], [95, 223], [190, 318]]
Relations:
{'head': 'zero-day attacks', 'type': 'subclass of', 'tail': 'cybersecurity threat', 'meta': {'spans': [[0, 128]]}}
{'head': 'zero-day attack', 'type': 'subclass of', 'tail': 'cybersecurity threat', 'meta': {'spans': [[0, 128]]}}
{'head': 'zero-day attacks', 'type': 'instance of', 'tail': 'cybersecurity threat', 'meta': {'spans': [[0, 128]]}}
{'head': 'Benford's law', 'type': 'instance of', 'tail': 'law of anomalous numbers', 'meta': {'spans': [[95, 223], [190, 318]]}}
{'head': 'Benford's law', 'type': 'instance of', 'tail': 'law', 'meta': {'spans': [[95, 223]]}}
{'head': 'law of anomalous numbers', 'type': 'named after', 'tail': 'Benford's law', 'meta': {'spans': [[95, 223], [190, 318]]}}
{'head': 'one-class', 'type': 'subclass of', 'tail': 'support vector machine', 'meta': {'spans': [[190, 318]]}}

Summary Results :

Input has 125 tokens
Input has 1 spans
Span boundaries are [[0, 128]]
Relations:
{'head': 'zero-day attacks', 'type': 'subclass of', 'tail': 'cybersecurity threat', 'meta': {'spans': [[0, 128]]}}
{'head': 'zero-day attack', 'type': 'subclass of', 'tail': 'cybersecurity threat', 'meta': {'spans': [[0, 128]]}}
{'head': 'zero-day attacks', 'type': 'facet of', 'tail': 'cybersecurity', 'meta': {'spans': [[0, 128]]}}

Figure 8. An example of extracting entity-entity pairs using the RE-BERT model from the original text and summary text presented in Figure 5.

After extracting entities from the Wikipedia document summary text for the pertinent entity, there is an attempt to search for the extracted entities in Google News in order to extract candidate entities and candidate relations for the specific entity from a news article. Figure 9 is an example showing a knowledge graph expanded using five pages of linked news sites after extracting information from 20 links from Google News to expand the knowledge graph for the "Computer Security" entity.

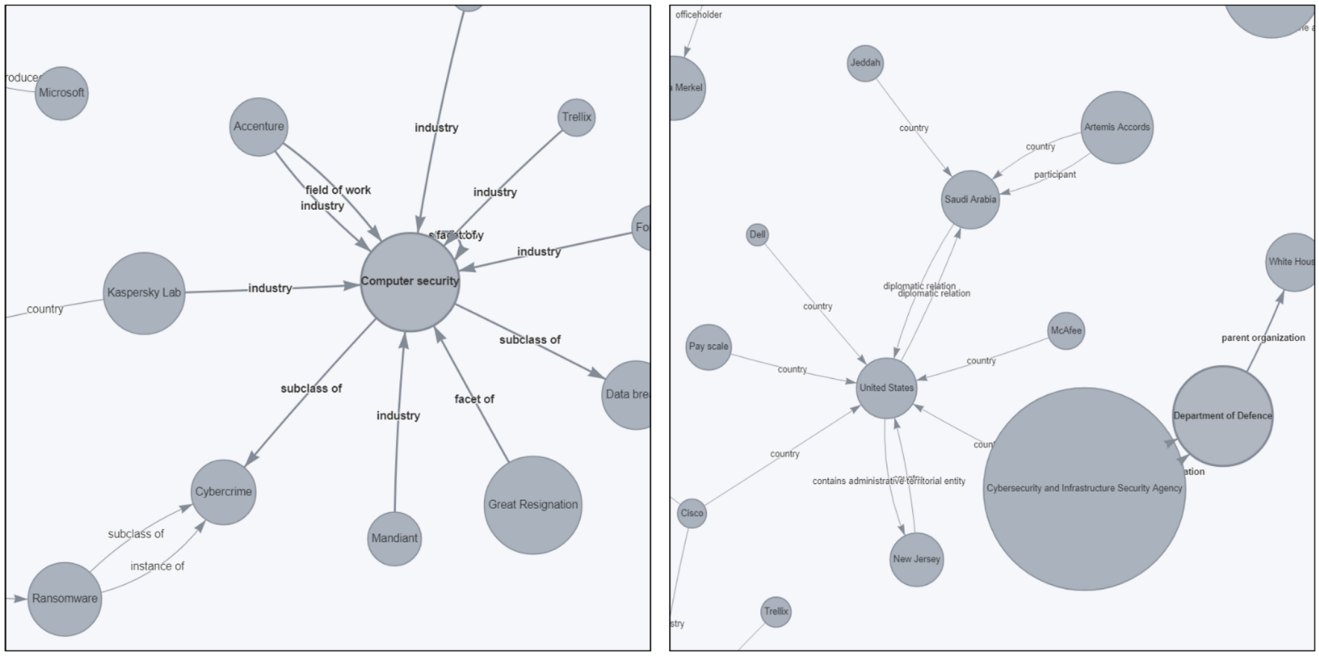


Figure 9. An example of knowledge graph expanding using Google News for the "Computer Security" entity.

4. Experiment and Evaluation

4.1. Research Literature Summarization Experiment

For the experiment in this study, we constructed a research literature training dataset to expand domain-specific knowledge graphs. As for the research literature data, we collected 3542 documents with a text size of five pages or more for each year based on documents collected from the web. To assess the summarization performance, we used the Recall-Oriented Understudy for Gisting Evaluation (ROUGE-N) method. ROUGE-N is a method of counting the number of repeated tokens in a tokenized ground-truth summary text and the generated summary text based on N-gram. ROUGE-1 is the percentage of each word (unigram) repeated in the generated summary text and the ground-truth summary text, and ROUGE-2 is the percentage that two consecutive words (bigram) are repeated. ROUGE-N refers to an evaluation of whether the summarized text contains important information from the original text sufficiently. ROUGE-L measures the longest common subsequence in the summarized text and the original text. The summarization model is evaluated by the recall, precision, and F1-score values. The equation for each measurement value is as follows:

$$Recall = \frac{number\ of\ overlapping\ words}{Total\ number\ of\ words\ in\ Reference\ summary} \times 100\% \quad (3)$$

$$recision = \frac{number\ of\ overlapping\ words}{Total\ number\ of\ words\ in\ System\ summary} \times 100\% \quad (4)$$

$$F1 - Score = \frac{recall \times precision}{recall + precision} \times 100\% \quad (5)$$

Table 1 shows the ROUGE recall, precision, and F1-score results for three fine-tuned models. The RNN model has limitations in that the longer the sentence, the slower the calculation speed, and the farther the distance, the more difficult it is to accurately represent the relationship. In addition, Transformer is a model that utilizes attention techniques to correct the poor accuracy of long sentences to solve the problems of the RNN model

[34]. The BERTSUM classifier model's ROUGE-1 precision shows that 57.86% of the words are in the generated summary data. Furthermore, it is shown that the BERTSUM classifier model outperforms the RNN-based BERTSUM.

Table 1. The ROUGE recall, precision, and F1-score results for three fine-tuned models.

Model	Metrics	Recall	Precision	F1-Score
BERTSUM Classifier	ROUGE-1	9.34%	57.86%	16.08%
	ROUGE-2	4.25%	16.53%	6.76%
	ROUGE-L	6.67%	39.12%	11.40%
Transformer	ROUGE-1	9.56%	58.34%	16.43%
	ROUGE-2	4.87%	14.91%	7.34%
	ROUGE-L	5.89%	39.87%	10.26%
RNN	ROUGE-1	8.65%	53.26%	14.88%
	ROUGE-2	4.35%	14.61%	6.70%
	ROUGE-L	5.79%	35.18%	9.94%

4.2. Accuracy of Knowledge Graph Relation Extraction Model

We used the proposed RE-BERT embedding model for the entity-relation extraction experiment using the research literature data. The gelu transformer encoder was used, and the LAMB optimizer was used as an optimizer. The values of the parameters used for training were 128 for embedding_size, 1024 for hidden_size, 12 for layer, and 12 for attention_heads. A total of 12,641 entities and 873 relations were extracted from 3542 collected research documents, and a total of 213,311 triples ($Entity^{Head}$, Relation, $Entity^{Tail}$) were created. From the summary data of the same research documents, we extracted a total of 7278 entities and 459 relations and created a total of 173,098 triples ($Entity^{Head}$, Relation, $Entity^{Tail}$). For the evaluation in this experiment, we used the MRR, MR, and HIT@N rank-based evaluation metrics, which are commonly used. MRR is the harmonic mean of the rank values. MR is the mean rank value of all triples. The HIT@N metric shows the ratio of the top N rank triples to all text triples. Q is the test triple set. A list of predicted triples is generated for each triple by the learning model. The rank of a triple refers to the correct indexing in the list.

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank(s,p,o)_i} \quad (6)$$

$$MR = \frac{1}{|Q|}\sum_{i=1}^{|Q|} rank(s,p,o)_i \quad (7)$$

$$Hits@N = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\begin{cases}1\ if(rank(s,p,o)_i \le N)\\ 0\ otherwise\end{cases} \quad (8)$$

For the N value in the HIT@N metric, 1, 3, and 10 were used: the larger the value, the better the result. In experiment 1, the model's performance was measured by constructing the training data and the test data with the triples ($Entity^{Head}$, Relation, $Entity^{Tail}$) extracted from the original texts and summary data of the research literature. In experiment 2, the model's performance was measured by constructing the expanded triples ($Entity^{Head}$, Relation, $Entity^{Tail}$) through Google News.

Table 2 shows the prediction results of the relationship between TransE, HolE, and ConvE and the RE-BERT model proposed in this paper. The TransE model is based on distance measuring. In the scoring function of TransE, the similarity between embeddings is calculated using the distances of the embedding vectors. The HolE model uses a circular correlation between entity embeddings as a scoring function. Additionally, the ConvE model has a convolution layer, a projection layer for embeddings, and another layer to multiply embedding vectors and obtain scores. In Experiment 1, RE-BERT's mean rank recorded 218.91, and Hits@10 recorded 0.53, which was the best value. Experiment 2

showed similar performance when comparing RE-BERT and TransE models. Through experiments, it has been proven that the method of extracting concepts and relationships using a summary of the research literature can exhibit sufficiently good performance.

Table 2. The prediction results of the relationship using RE-BERT model.

Model		MRR	MR	HITS@10	HITS@3	HITS@1
RE-BERT	Experiment-1	0.38	218.91	0.53	0.42	0.37
	Experiment-2	0.47	131.67	0.61	0.57	0.42
TransE	Experiment-1	0.29	531.87	0.46	0.36	0.31
	Experiment-2	0.44	152.31	0.68	0.42	0.45
HolE	Experiment-1	0.26	198.46	0.48	0.31	0.28
	Experiment-2	0.37	156.14	0.42	0.39	0.26
ConvE	Experiment-1	0.24	763.56	0.39	0.28	0.21
	Experiment-2	0.28	356.10	0.43	0.34	0.21

5. Conclusions

In this paper, we proposed a graph embedding-based domain-specific knowledge graph expansion method using a research literature summary. To this end, we performed a pre-processing process and text summarization for the collected research literature data and proposed a method for generating a knowledge graph by extracting entities and related information from the summarized text and a method for expanding the knowledge graph through web data. In the experiment, we measured the performance of summarizing research literature using the BERTSUM model and the accuracy of the relation extraction model. According to the results of the experiment, from which unnecessary sentences in the research literature text had been removed and the text was summarized with key sentences, the BERTSUM classifier model's ROUGE-1 precision was 57.86%. The knowledge graph extraction performance was measured using MRR, MR, and HIT@N rank-based evaluation metrics, and the knowledge graph extraction method using summarized text demonstrated a better performance in terms of knowledge graph quality.

Funding: This study was supported by research fund from Chosun University, 2018.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The author declares no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BERT	Bidirectional Encoder Representations from Transformers
BERTSUM	Bidirectional Encoder Representations from Transformers for Summarization
NER	Named Entity Recognition
CLS	Special Classification Token
SEP	Special Separator Token
UNK	Unknown Token
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
RNN	Recurrent Neural Network
MRR	Mean Reciprocal Rank
MR	Mean Rank

References

1. Khurana, D.; Koli, A.; Khatter, K.; Singh, S. Natural language processing: State of the art, current trends and challenges. *Multimed. Tools Appl.* **2022**, 1–32. https://doi.org/10.1007/s11042-022-13428-4.
2. Wang, R.J.; Yan, Y.C.; Wang, J.L.; Jia, Y.T.; Zhang, Y.; Zhang, W.N.; Wang, X.B. AceKG: A Large-scale Knowledge Graph for Academic Data Mining. In Proceedings of the Cikm'18: Proceedings of the 27th Acm International Conference on Information and Knowledge Management, New York, NY, USA, 22–26 October 2018; pp. 1487–1490. https://doi.org/10.1145/3269206.3269252.
3. Nayyeri, M.; Vahdati, S.; Zhou, X.; Shariat Yazdi, H.; Lehmann, J. Embedding-based recommendations on scholarly knowledge graphs. In Proceedings of the European Semantic Web Conference, Heraklion, Greece, 31 May-4 June 2020; pp. 255–270.
4. Rossi, R.A.; Zhou, R.; Ahmed, N.K. Deep inductive graph representation learning. *IEEE Trans. Knowl. Data Eng.* **2018**, *32*, 438–452.
5. Ferré, S. Link prediction in knowledge graphs with concepts of nearest neighbours. In Proceedings of the European Semantic Web Conference, Portorož, Slovenia, 2–6 June 2019; pp. 84–100.
6. Rossanez, A.; dos Reis, J.C. Generating Knowledge Graphs from Scientific Literature of Degenerative Diseases. In Proceedings of the SEPDA@ ISWC, Auckland, New Zealand, 27 August 2019; pp. 12–23.
7. Paulheim, H. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semant. Web* **2017**, *8*, 489–508.
8. Liu, Y. DKG-PIPD: A Novel Method About Building Deep Knowledge Graph. *IEEE Access* **2021**, *9*, 137295–137308.
9. Dai, Y.; Wang, S.; Xiong, N.N.; Guo, W. A survey on knowledge graph embedding: Approaches, applications and benchmarks. *Electronics* **2020**, *9*, 750.
10. Jaradeh, M.Y.; Oelen, A.; Farfar, K.E.; Prinz, M.; D'Souza, J.; Kismihók, G.; Stocker, M.; Auer, S. Open research knowledge graph: Next generation infrastructure for semantic scholarly knowledge. In Proceedings of the 10th International Conference on Knowledge Capture, Los Angeles, CA, USA, 19–21 November 2019; pp. 243–246.
11. Kim, J.; Kim, K.; Sohn, M.; Park, G. Deep Model-Based Security-Aware Entity Alignment Method for Edge-Specific Knowledge Graphs. *Sustainability* **2022**, *14*, 8877.
12. Kejriwal, M. *Domain-Specific Knowledge Graph Construction*; Springer: Berlin/Heidelberg, Germany, 2019.
13. Chen, X.; Xie, H.; Li, Z.; Cheng, G. Topic analysis and development in knowledge graph research: A bibliometric review on three decades. *Neurocomputing* **2021**, *461*, 497–515.
14. Berrendorf, M.; Faerman, E.; Vermue, L.; Tresp, V. Interpretable and Fair Comparison of Link Prediction or Entity Alignment Methods. In Proceedings of 2020 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), Melbourne, Australia, 14–17 December 2020; pp. 371–374.
15. Lissandrini, M.; Pedersen, T.B.; Hose, K.; Mottin, D. Knowledge graph exploration: Where are we and where are we going? *ACM SIGWEB Newsl.* **2020**, 1–8. https://doi.org/10.1145/3409481.3409485.
16. Sun, Z.; Huang, J.; Hu, W.; Chen, M.; Guo, L.; Qu, Y. Transedge: Translating relation-contextualized embeddings for knowledge graphs. In Proceedings of International Semantic Web Conference, Auckland, New Zealand, 26–30 October 2019; pp. 612–629.
17. Zhu, Q.; Zhou, X.; Zhang, P.; Shi, Y. A neural translating general hyperplane for knowledge graph embedding. *J. Comput. Sci.* **2019**, *30*, 108–117.
18. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Philip, S.Y. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 4–24.
19. Zhang, W.; Deng, S.; Chen, M.; Wang, L.; Chen, Q.; Xiong, F.; Liu, X.; Chen, H. Knowledge graph embedding in e-commerce applications: Attentive reasoning, explanations, and transferable rules. In Proceedings of The 10th International Joint Conference on Knowledge Graphs, Bangkok, Thailand, 6–8 December 2021; pp. 71–79.
20. Lakshika, M.; Caldera, H. Knowledge Graphs Representation for Event-Related E-News Articles. *Mach. Learn. Knowl. Extr.* **2021**, *3*, 802–818.
21. Sun, Z.; Deng, Z.-H.; Nie, J.-Y.; Tang, J. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv* **2019**, arXiv:1902.10197.
22. Nguyen, D.Q. A survey of embedding models of entities and relationships for knowledge graph completion. *arXiv* **2017**, arXiv:1703.08098.
23. Ma, J.; Qiao, Y.; Hu, G.; Wang, Y.; Zhang, C.; Huang, Y.; Sangaiah, A.K.; Wu, H.; Zhang, H.; Ren, K. ELPKG: A high-accuracy link prediction approach for knowledge graph completion. *Symmetry* **2019**, *11*, 1096.
24. Yao, L.; Mao, C.; Luo, Y. KG-BERT: BERT for knowledge graph completion. *arXiv* **2019**, arXiv:1909.03193 .
25. Kazemi, S.M.; Poole, D. SimpleE embedding for link prediction in knowledge graphs. *Adv. Neural Inf. Processing Syst.* **2018**, *31*. https://doi.org/10.48550/arXiv.1802.04868.
26. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
27. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. Albert: A lite bert for self-supervised learning of language representations. *arXiv* **2019**, arXiv:1909.11942.
28. Liu, Y.; Lapata, M. Text summarization with pretrained encoders. *arXiv* **2019**, arXiv:1908.08345.

29. Liu, Y.; Luo, Z.; Zhu, K. Controlling length in abstractive summarization using a convolutional neural network. In Proceedings of Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 4110–4119.
30. Khatri, C.; Singh, G.; Parikh, N. Abstractive and extractive text summarization using document context vector and recurrent neural networks. *arXiv* **2018**, arXiv:1807.08000.
31. Mao, X.; Yang, H.; Huang, S.; Liu, Y.; Li, R. Extractive summarization using supervised and unsupervised learning. *Expert Syst. Appl.* **2019**, *133*, 173–181.
32. Kim, T.; Yun, Y.; Kim, N. Deep learning-based knowledge graph generation for COVID-19. *Sustainability* **2021**, *13*, 2276.
33. Chen, X.; Jia, S.; Xiang, Y. A review: Knowledge reasoning over knowledge graph. *Expert Syst. Appl.* **2020**, *141*, 112948.
34. Guo, L.; Zhang, Q.; Ge, W.; Hu, W.; Qu, Y. DSKG: A deep sequential model for knowledge graph completion. In Proceedings of the China Conference on Knowledge Graph and Semantic Computing, Tianjin, China, 14–18 August 2018; pp. 65–77.