Keyphrase Extraction Model: A New Design and Application on Tourism Information

Ngo Le Huy Hien
School of Built Environment, Engineering, and Computing, Leeds Beckett University, Leeds, England
E-mail: n.hien2994@student.leedsbeckett.ac.uk

Ho Minh Hoang and Nguyen Van Hieu
The University of Danang - University of Science and Technology, Danang, Vietnam
Est Rouge Technologies JSC, Vietnam
E-mail: hoanghm@tech.est-rouge.com, nvhieuqt@dut.udn.vn

Nguyen Van Tien
The University of Danang - University of Science and Technology, Danang, Vietnam
E-mail: vannt808@gmail.com

Keywords: elasticsearch, keyphrase extraction, conditional random field, BERT, BiLSTM-CRF, long short-term memory

Received: April 5, 2021

Keyphrase extraction has recently become a foundation for developing digital library applications, especially in semantic information retrieval techniques. From that context, in this paper, a keyphrase extraction model was formulated in terms of Natural Language Processing, applied explicitly in extracting information and searching techniques in tourism. The proposed process includes collecting and processing data from tourism sources such as Tripadvisor.com, Agoda.com, and vietnam-guide.com. Then, the raw data was analyzed and pre-processed with labeling keyphrase and fed data forward to Pretrained BERT model and Bidirectional Long Short-Term Memory with Conditional Random Field. The model performed the combination of Bidirectional Long Short-Term Memory with Conditional Random Field in order to solve keyphrase extraction tasks. Furthermore, the model integrated the Elasticsearch technique to enhance performance and time of looking up tourism destinations' information. The outcome extracted key phrases produce high accuracy and can be applied for extraction problems and textual content summaries.

Povzetek: Predstavljen je pristop na osnovi ključnih fraz za uporabo v turističnih sistemih.

1 Introduction

In the science of natural language processing, the analysis of sentences into phrases, labeling, and marking has been a point of interest in research and application in various aspects. Keyphrase Extraction is the process of extracting key phrases that contain important content of a document. Keyphrases are used to solve information extraction content clustering, text classification, and text summary problems [16]. Numerous studied methodologies have been widely applied in academic issues such as Key2Vec [2] - automatically extracting keywords from scientific articles, Sequence Labeling [1] - extracting keyphrase from scholarly documents. The process normally used the BiLSTM [1] model, combining a pre-trained model to extract corresponding keywords of a dataset. Then, the search engine operated through API using NoSQL Elasticsearch, which uses scoring techniques from the keyphrases of documents corresponding to the database [19].

Research that applies in traveling newspapers and documents would support tourism information searching from many traveling sites. From there, the Keyphrase Extraction method allows visitors to easily select and quickly search based on their own words without clearly understanding their desired places. Furthermore, based on official data analysis from tourist sites, visitors will avoid unreliable information of some locations related to their own needs. The method would help increase the experience and satisfaction when visitors come and learn information about the city.

From the aforementioned method and benefits, this study proposed a new design and application to assist in searching tourist information. The application can be implemented as a phone app or a tourist information website that optimally serves tourist demand for their first steps in a new destination and acquire typical characteristics of the sit shown on media. This research can play a novel and practical keyphrase extraction model and contribute to the science of extraction applications and textual content summaries.

2 Related works

2.1 Keyphrase extraction

Keyphrase Extraction is a subfield of Information extraction problem to extract keyphrases from documents according to given requirements. Currently, there are many studies and methods given in this problem with many different approaches, as mentioned below.

2.1.1 Unsupervised method

The unsupervised learning method provides probability models based on the word input that determines the frequency and importance of keywords in the text to identify keyword phrases. Some of the unsupervised learning techniques such as TextRank [3], Sgrank [4], and WikiRank [5] help extract keyphrases based on a narrow context of context (identifying the meaning of the word or by probability). Therefore, these methods' accuracy is also limited, but this is a basic method and has been applied in many problems to instantly perform labeling for supervised learning methods.

2.1.2 Self-supervised method

Recent studies focus on self-supervised learning in the field of information extraction. When the data is initially trained with the labels, the machine will then label and learn itself based on the relationships between the new input information and the previously trained information. These are superior studies such as SRES [6] - extracting information on the Web, and SelfORE [7] - extracting from natural language sentences their open-domain relation facts. Studies have introduced new approaches to training as well as solutions for information extraction.

2.1.3 Supervised method

In this method, the data is labeled corresponding to the keyphrases before training through a machine learning model. Parallel with the development in data and computing hardware; deep learning has been increasingly popular and widely used to optimize. For example, the research about Long Short-Term Memory (LSTM) [8] effectively covers the neighbor contexts [17]. BiLSTM-CRF [9] used the Glove representation model to embed input words and return positive results when experimenting in an academic dataset.

2.2 Contextual embedding

Before training, input data have to be normalized into sets of vectors. Word embedding is a form of word representation, representing words with related meanings to have similar representations.

There are many studies and experiments in implementing algorithms supporting word embedding and vocabulary modeling. And Contextual Embedding is one of the SOTA techniques to vectorize documents based on meaning and contextual relations. The enhancement of Contextual embedding compared with others embedding models such as Word2Vec [10], Glove [14] is the addition of context for vectors generated through position, thereby increasing the accuracy in terms of context and semantics.

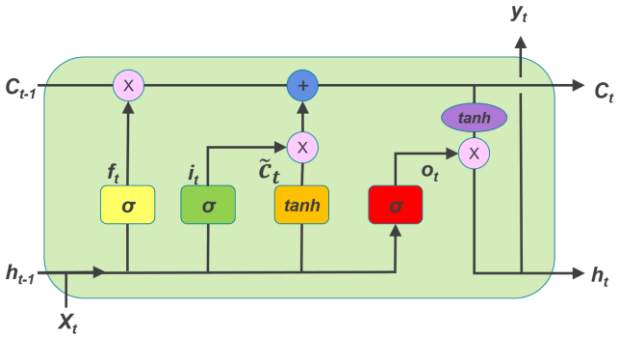

Figure 1: Structure of 1 cell LSTM.

BERT [11] was introduced as a breakthrough in the field of natural language processing, with improvements in text modeling with the application of Transformer architecture to train context-based word representations.

In this article, the authors combined the improvement of word representation models with sequence labeling techniques for extracting keyphrases in tourism documents.

3 Methodology

Let $d = \{w_1, w_2, \ldots, w_n\}$ be input document, and w_i represents the i^{th} token. Each word in d was labeled into 3 classes of set $Y = \{K_B, K_I, K_O\}$, where K_B indicates that w_i is the beginning keyphrase, K_I denotes that w_i is in the keyphrase, and K_O marks that w_i is out of the keyphrase.

3.1 Long Short-Term Memory

Long Short-Term Memory (LSTM) [8] is a form of Recurrent Neural Network (RNN) model [13] for solving problems of sequence data based on previously learned information to predict the current information in the sequence. LSTM is a solution to resolve the Vanishing Gradient issue of a primitive RNN network when information is learned from far away in the chain and lost its importance. In order to achieve this, LSTM uses several "gates" that store information remotely. Especially, Bidirectional LSTM (BiLSTM) is a generalization technique covering the context information in both directions [12].

Each word in the text was mapped for embedding size vector x_i, so that the sequence d of length n will be represented by a vector.

$x = \{x_1, x_2, \ldots, x_n\}$ was labeled accordingly with $y = \{y_1, y_2, \ldots, y_n\}$ where $y_i \in Y$.

The input of LSTM is a $[h_{t-1}, x_t]$ vector at time t, with the cell state of the network c_t, and the output vector between the two times t and $t+1$ is h_t.

LSTM unit has 4 gates: forget gate f_t, input gate i_t, output gate o_t, and memory cell c_t, which are represented by the following equations:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f); \quad (1)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i); \quad (2)$$

$$c_t = f_t * c_{t-1} + i_t * tanh(W_c \cdot [h_{t-1}, x_t] + b_c); \quad (3)$$

$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o);$ (4)

$h_t = o_t * tanh(c_t);$ (5)

in which the activation functiond are: σ (sigmoid) and *tanh*, and * is element-wise multiplication. W and b are model parameters, and h_t is hidden state.

In the BiLSTM model, 2 used LSTM architectures progress simultaneously and independently to model the input sequence in 2 directions: from left to right (direction $\overrightarrow{h}_t$ - Figure 2a) and from right to left (direction $\overleftarrow{h}_t$ - Figure 2b).

Where $\overrightarrow{h}_t$ represents the information from preceding word of $w_t\{w_1, w_2, \ldots, w_{t-1}\}$, and $\overleftarrow{h}_t$ represents the information from succeeding words of $w_t\{w_{t+1}, w_{t+2}, \ldots, w_n\}$. Vector $\overleftrightarrow{h}_t$ represents for word w_t in input sequence d when concatenating 2 vectors $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$.

$$\overleftrightarrow{h}_t = [\overrightarrow{h}_t; \overleftarrow{h}_t]$$

Then the results were mapped to vector f_t where

$$f_t = W_a \overleftrightarrow{h}_t$$

in which W_a is weight vector that has a shape of

$$|Y| \times |\overleftrightarrow{h}_t| = 3 \times |\overleftrightarrow{h}_t|$$

The output vector of BiLSTM model after multiplying weight matrix is

$$f = \{f_1, f_2, \ldots, f_n\}$$

in which f is the input of the CRF layer.

3.2 Conditional Random Field

Conditional Random Field (CRF) is a probabilistic model for structured predictive problems and has been used very successfully in machine learning areas. CRF is used in conjunction with deep learning models to increase the efficiency for segmentation and sequence data labeling [18].

As the input data in CRF is sequential, the previous context must be considered before predicting a data point, thereby increasing the model's accuracy. For example, if the previous label is B-P (begin phrase), the following tag is most likely I-P.

In this study, a 378-dimensional vector was used representation for each word following the BERT (BERT-base) model. A BERT's pre-trained model's architecture and some layers were added to match the problem. Then, the original layer parameters were fine-tuning, and the additional layer parameters were re-trained from the beginning. In this way, the proposed model could reduce the training time while ensuring its accuracy.

3.3 Elasticsearch (NoSQL)

This study used the NoSQL Elasticsearch database management system because of its ability to analyze data and statistics. A node is an Elasticsearch server, which is logically independent of each other. In fact, a node can run on one (usually in a development or test environment) or multiple physical servers (usually in a production environment). A collection of nodes working together forms a cluster; each node in the cluster contains a portion of that cluster's data. And all the data of a cluster will be divided among the nodes [20].

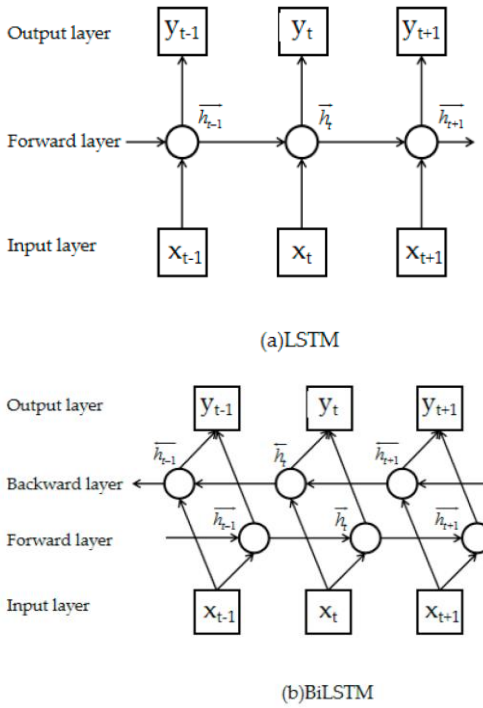

Figure 2: The architecture of LSTM and BiSLTM.

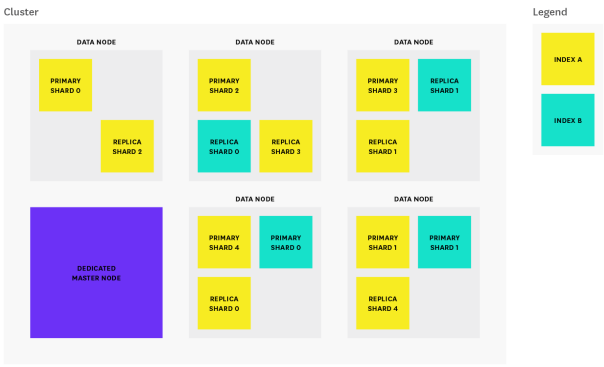

Figure 3: Elasticsearch architecture.

Nodes have three different types: master, data, and client. A cluster automatically selects a node as the master from its nodes. The master node will be responsible for coordinating the work of the cluster, such as distributing shards and creating/deleting indexes. Only the master node has the ability to update the cluster's state. In essence, Apache's Lucene - a full-text search - uses a data structure called an inverted index to perform searches with high performance. The architecture of Elasticsearch is indicated in Figure 3 [21][22].

Elasticsearch operates on a private server, communicates through RESTful APIs, and provides near real-time.

4 Proposed method and architecture

This paper reveals a new method in keyphrase extraction from travel documents, using the sequence labeling technique with pre-trained model BERT (Bidirectional Encoder Representations from Transformers). Then, the model applies BiLSTM with Conditional Random Field in the training phase to enhance its result.

4.1 Proposed model

Pre-processing: The input data was pre-processed by encoding each word (token), along with whether the word tag corresponds to the keyphrase or not. The labels are B-P (Begin Phrase), I-P (In Phrase), and O (Out Phrase). Those labels were encoded with the input data into numeric labels 0, 1, 2, respectively.

For example:

Restaurant	has	a	big	view	of	natural	landscape.
O	O	O	B-P	I-P	I-P	I-P	I-P

Pre-train model BERT: There are currently many different versions of the BERT model. All versions are based on the transformation of the Transformer architecture, focusing on 3 parameters:

L: The number of block sub-layers in transformer,

H: Embedding vector size (or hidden size),

A: The number of heads in a multi-head layer, each head operates one self-attention.

The research used the pre-trained BERT base uncase model (L = 12, H = 768, A = 12) to represent the input vocabulary into vectors containing information about the vocabulary and its context. The BERT model input consists of a sequence of coded words, and the output is a lexical vector representing each input word.

BiLSTM-CRF model: Output vectors of BERT are the inputs of the BiLSTM-CRF model. They were passed through the 2-dimensional LSTM network, and the information will be trained in two dimensions of the context, in terms of firm magnetism and context. Next, the output was passed through the CRF layer with labels marked previously to train and extract key phrase information in the text.

4.2 The process

To implement the data effectively, the proposed model applied a process that is depicted in Figure 5. In the beginning, raw data were pre-processed by filters and removed noise data, including HTML, tag, link, unrelated text, etc.

In the training phase, each sentence of the text was labeled. If the phrase is at the beginning, it would be labeled as B-P (begin phrase), the rest of the keyphrase should be labeled as I-P (in phrase), and other words considered as O-P (out phrase). After that, the processed data was fed into the BERT model in Contextual Embedding stages before forwarding to BiLSTM layers. Then the data was fed into CRF layers after attaching labels. The output weight was used in the evaluation stage

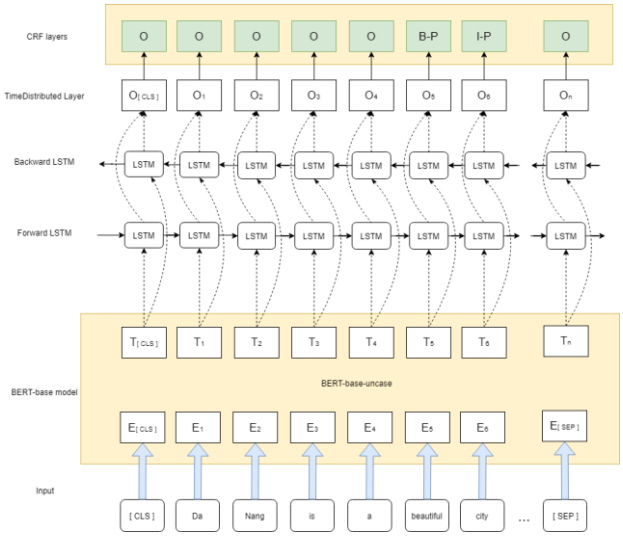

Figure 4: Proposed model BERT-BiSTM-CRF.

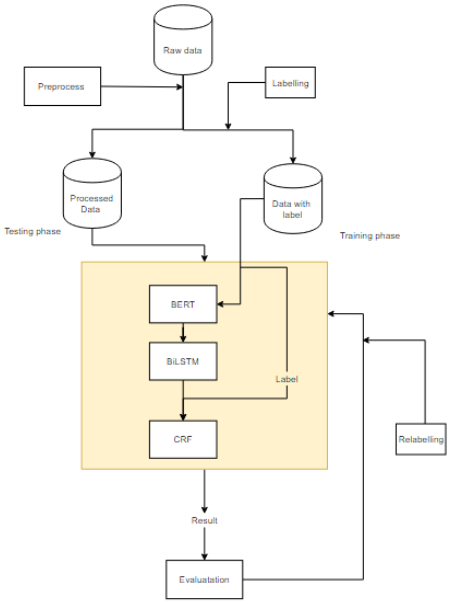

Figure 5: The proposed model process.

with new input; new input results were relabeled and became the architecture input to re-train.

In the testing phase, the real raw data was also pre-processed and then fed into the model to predict the keyphrase.

4.3 Application architecture

The output keyphrases were stored in a database and synchronized to a NoSQL database before applying the search engine of Elasticsearch. Since Elasticsearch operates on a private server through RESTful APIs, the proposed model can fit for a large feature set with the real-time processing ability of Elasticsearch.

The system architecture is presented in Figure 6 with an artificial intelligence system integrated with an API layer. The API contributes as a communication channel

between the server and the client to perform querying, processing, and returning results to UI [23]. First, the data collected from travel websites will be trained and stored in a NoSQL database (MongoDB). Then the data synchronized from Cluster Mongo, containing collected data for Elasticsearch. The APIs were used in retrieving data from Elasticsearch to fetch and return the results to the user since Elasticsearch is only effective in retrieving data.

5 Results

5.1 Training result

The study was conducted on a dataset with two models: one applies Glove embedding, and the other uses BERT embedding. After the experiment, Table 1 shows that BERT embedding as a pre-trained model indicates better results with the Recall value of about 0.891.

The two graphs in Figure 7 indicate the loss and accuracy of the proposed approach based on the number of epochs of the training phase. The Loss and Accuracy indexes at the first two epochs present the ideal trends while the next epochs take a slight improvement in the Loss and Accuracy results.

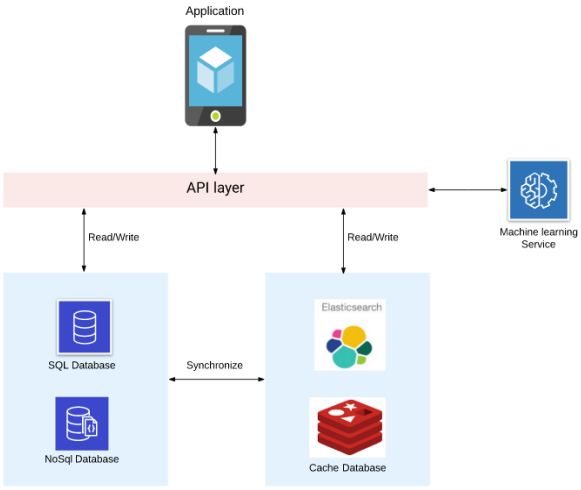

Figure 6: The proposed application architecture.

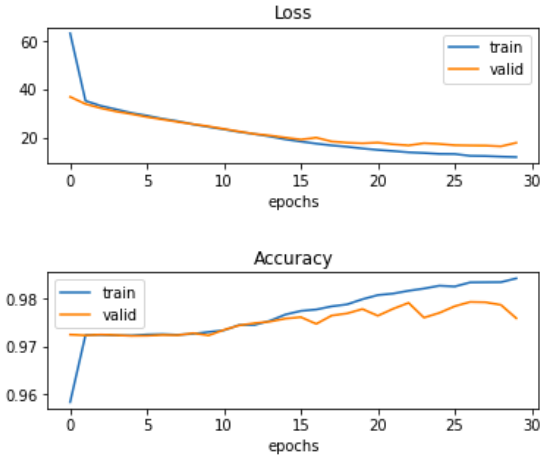

Figure 7: Training results based on number of epochs.

5.2 Test result

The study also analyzed the proposed model with travel datasets that were pre-processed and labeled. The datasets were crawled into different text blocks [15] from tourism resources, including newspapers, descriptions, documents, and comments on Tripadvisor and Agoda about destinations and attractions.

The data was pre-processed by removing noise information such as images, HTML tags, web page scripts, and other irrelevant comments. In this case, the experiment focused on English data after removing other languages' information.

The results are shown in Tables 2 and 3, corresponding to different amounts of training paragraph.

It is recognized from Tables 2 and 3 that when the training and testing data increase, the predicted

	Precision	Recall	F1 score	Exact match
BERT Embedding	0.954	0.891	0.921	0.282
Glove Embedding	0.956	0.724	0.824	0.211

Table 1: Comparison between BERT embedding and Glove embedding.

Training paragraph	Original Keyphrase (test)	Predicted Keyphrase	Accuracy	Training Time (s)
200	4681	1724	161	9.13
400	4681	3122	900	13.91
600	4681	2683	1075	17.30
873	4681	3574	2033	20.78

Table 2: Prediction results based on training data.

Training paragraph	Precision	F1 score	Recall
200	0.577	0.310	0.212
400	0.650	0.613	0.580
600	0.715	0.727	0.739
873	0.874	0.794	0.729

Table 3: Result indices corresponding to testing data based on training data.

Sgrank	BERT-BiLSTM-CRF
'Danang is a developing coastal city in the central part of Vietnam and is known as one of the largest cities alongside Hanoi and Ho Chi Minh city. Visiting Da Nang, you will be astounded by the amazing natural landscape, the friendly locals and a countless number of great attractions around the city.'	'Danang is a developing coastal city in the central part of Vietnam and is known as one of the largest cities alongside Hanoi and Ho Chi Minh city. Visiting Da Nang, you will be astounded by the amazing natural landscape, the friendly locals and a countless number of great attractions around the city.'

Table 4: Result in application[1].

keyphrases and the correct keyphrases also increase, indicating the efficiency of the model. Moreover, the training time for each experiment is applicable in real practice.

6 Application

The study examined the BiLSTM-CRF model combined with the BERT Embedding layer and compared the result with the Sgrank method.

Table 4 showcases the actual results of the model when predicting a completely new input. It is recognized that the proposed model focuses on phrases of nouns and adjectives from B-P and I-P labels. Although the Sgrank method produces many phrases, it has a high error rate and focuses on nonspecific adjectives and nouns.

7 Conclusion

In this article, a keyphrase extraction method was proposed, which uses the BiLSTM-CRF deep learning model with the BERT pre-trained model's lexical representation. The output of the BERT (encoder) model became the input of the BiLSTM-CRF model to perform the keyphrase extraction task.

With a supervised learning method, the proposed method has outweighed previous models in terms of accuracy of words' context and meaning. In addition, the study has built an API system for applications integrated with actual text extraction. The presented method has helped extract key phrases in the text with high accuracy (from 40% on sample data), thereby can be applied for extraction problems and textual content summaries.

Future work may gear towards expanding the model and proposing a software architect to conduct an application supporting tourism for different cities around the world. Based on each characteristic of each destination (from keyphrases), a recommendation system could be developed to support users in finding their next desired destinations. Furthermore, the authors aim to continue expanding the applications of the model into different languages (rather than English) and various fields, not only in tourism.

7.1 Acknowledgement

This research is funded and implemented for the Mercury project of Est Rouge Technologies JSC, Vietnam.

References

[1] Alzaidy, Rabah; Caragea, Cornelia; GILES, C. Lee. Bi-LSTM-CRF sequence labeling for keyphrase extraction from scholarly documents. In: The world wide web conference. 2019. p. 2551-2557. https://doi.org/10.1145/3308558.3313642.

[2] Mahata, Debanjan, et al. Key2vec: Automatic ranked keyphrase extraction from scientific articles using phrase embeddings. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). 2018. p. 634-639. http://dx.doi.org/10.18653/v1/N18-2100.

[3] Mihalcea, Rada; Tarau, Paul. Textrank: Bringing order into text. In: Proceedings of the 2004 conference on empirical methods in natural language processing. 2004. p. 404-411.

[4] Danesh, Soheil; Sumner, Tamara; Martin, James H. Sgrank: Combining statistical and graphical methods to improve the state of the art in unsupervised keyphrase extraction. In: Proceedings of the fourth joint conference on lexical and computational semantics. 2015. p. 117-126. http://dx.doi.org/10.18653/v1/S15-1013.

[5] Yu, Yang; Ng, Vincent. Wikirank: Improving keyphrase extraction based on background knowledge. arXiv:1803.09000, 2018.

[6] Feldman, Ronen, et al. Self-supervised relation extraction from the web. In: International Symposium on Methodologies for Intelligent Systems. Springer, Berlin, Heidelberg, 2006. p. 755-764. https://doi.org/10.1007/s10115-007-0110-6.

[7] HU, Xuming, Et Al. SelfORE: Self-supervised Relational Feature Learning for Open Relation Extraction. arXiv:2004.02438, 2020.

[8] Hochreiter, Sepp; Schmidhuber, Jürgen. Long short-term memory. Neural computation, 1997, 9.8: 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735.

[9] Alzaidy, Rabah; Caragea, Cornelia; Giles, C. Lee. Bi-LSTM-CRF sequence labeling for keyphrase extraction from scholarly documents. In: The world wide web conference. 2019. p. 2551-2557. https://doi.org/10.1145/3308558.3313642.

[10] Mikolov, Tomas, et al. Efficient estimation of word representations in vector space. arXiv:1301.3781, 2013.

[11] Devlin, Jacob, et al. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805, 2018.

[12] Graves, Alex; Fernández, Santiago; Schmidhuber, Jürgen. Bidirectional LSTM networks for improved phoneme classification and recognition. In: International conference on artificial neural

networks. Springer, Berlin, Heidelberg, 2005. p. 799-804. https://doi.org/10.1007/11550907_126.

[13] David, Rumelhart. Recurrent Neural Networks, 1986.

[14] Pennington, Jeffrey; Socher, Richard; Manning, Christopher D. Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014. p. 1532-1543. http://dx.doi.org/10.3115/v1/D14-1162.

[15] Hien, Ngo Le Huy; Tien, Thai Quang; Van Hieu, Nguyen. Web Crawler: Design and Implementation for Extracting Article-Like Contents. Cybernetics and Physics, 2020, 9.3: 144-151. https://doi.org/ 10.35470/2226-4116-2020-9-3-144-151.

[16] Witten, Ian H., et al. Kea: Practical automated keyphrase extraction. In: Design and Usability of Digital Libraries: Case Studies in the Asia Pacific. IGI global, 2005. p. 129-152. https://doi.org/10.4018/978-1-59140-441-5.ch008.

[17] Hien, Ngo Le Huy; Van Hieu, Nguyen. Recognition of Plant Species using Deep Convolutional Feature Extraction. International Journal on Emerging Technologies, 2020, 11.3: 904-910. https://doi.org/10.14445/22315381/IJETT-V68I4P205S.

[18] Zhang, Chengzhi. Automatic keyword extraction from documents using conditional random fields. Journal of Computational Information Systems, 2008, 4.3: 1169-1180.

[19] Hien, Ngo Le Huy; Huy, Luu Van; Van Hieu, Nguyen. Artwork style transfer model using deep learning approach. Cybernetics and Physics, 2021, 10.3: 127-137. https://doi.org/10.35470/2226-4116-2021-10-3-127-137.

[20] Munezero, Myriam, et al. Automatic detection of antisocial behaviour in texts. Informatica, 2014, 38.1: 3-10.

[21] Azam, Irfan, and Sajid Ali Khan. Feature extraction trends for intelligent facial expression recognition: A survey. Informatica, 2018, 42.4: 507-514. https://doi.org/10.31449/inf.v42i4.2037.

[22] Chen, Feng, and Shi Zhang. Information Visualization Analysis of Public Opinion Data on Social Media. Informatica, 2021, 45.1: 157-162. https://doi.org/10.31449/inf.v45i1.3426.

[23] Menai, Mohamed El Bachir. Word sense disambiguation using an evolutionary approach. Informatica, 2014, 38.3: 155-169.