Bulletin of Electrical Engineering and Informatics
Vol. 12, No. 1, February 2023, pp. 303~313
ISSN: 2302-9285, DOI: 10.11591/eei.v12i1.4130

Automatic keyphrases extraction: an overview of deep learning approaches

Lahbib Ajallouda[1], Fatima Zahra Fagroud[2], Ahmed Zellou[1], El habib Benlahmar[2]
[1]National School of Computer Science and Systems Analysis (ENSIAS), Mohammed V University, Rabat, Morocco
[2]Department of Mathematics and Computer Science, Faculty of Sciences Ben M'sik, Hassan II University, Casablanca, Morocco

Article Info

Article history:

Received May 24, 2022
Revised Aug 6, 2022
Accepted Sep 27, 2022

Keywords:

Artificial neural networks
Deep learning algorithms
Keyphrases extraction
Natural language processing

ABSTRACT

Automatic keyphrases extraction (AKE) is a principal task in natural language processing (NLP). Several techniques have been exploited to improve the process of extracting keyphrases from documents. Deep learning (DL) algorithms are the latest techniques used in prediction and extraction of keyphrases. DL is one of the most complex types of machine learning, relying on the use of artificial neural networks to make the machine follow the same decision-making path as the human brain. In this paper, we present a review of deep learning-based methods for AKE from documents, to highlight their contribution to improving keyphrase extraction performance. This review will also provide researchers with a collection of data and information on the mechanisms of deep learning algorithms in the AKE domain. This will allow them to solve problems encountered by AKE approaches and propose new methods for improving key-extraction performance.

Corresponding Author:

Lahbib Ajallouda
National School of Computer Science and Systems Analysis (ENSIAS), Mohammed V University
Avenue Mohamed Ben Abdellah Regragui, Madinat Al Irfane, BP 713, Agdal Rabat, Morocco
Email: lahbib_ajallouda@um5.ac.ma

1. INTRODUCTION

Keyphrase is an expression that identifies one of the main topics or one of the main ideas of a document. Automatic keyphrase extraction is essential for numerous natural language processing (NLP) tasks such as clustering [1] information retrieval [2], and text summarization [3]. Deep learning (DL) is one of the most important solutions proposed for automatic keyphrase extraction, because DL algorithms have the ability to understand the complex relationships between a large number of interrelated variables [4]. Moreover, traditional machine learning (ML) algorithms are not able to process raw input data, but DL algorithms helped overcome this limitation [5]. Recently several DL algorithms have been proposed [6]. These algorithms have been widely used to improve many tasks, such as keyphrase extraction [7], machine translation [8], sentiment analysis [9], question-answer systems [10], words recognition system [11], and recommender system [12]. In contrast, we found that most of the reviews that focused on the use of DL algorithms did not discuss at length their use of the keyphrase extraction task, but rather on the basis of a set of tasks [6], [13]. This makes it difficult for new researchers to understand how to use DL algorithms to improve keyphrases extraction performance.

The objective of this article is to review DL-based keyphrases extraction approaches in order to provide an overview of the best algorithms for this task. As well as the datasets used to train and test these approaches. This review will enable researchers to gain a better understanding of how deep learning algorithms can be used to extract keyphrases and propose new approaches that outperform the current one.

The content of our paper will be as follows. In section 2, we will introduce deep learning algorithms, especially those used in NLP tasks. In section 3, we will discuss deep learning-based keyphrase extraction approaches. The empirical results of these methods are then discussed in section 4. We will then have a general discussion in section 5. Finally, we will conclude our paper in section 6 as well as future research directions.

2. DEEP LEARNING ALGORITHMS

Currently, DL algorithms are the most efficient among all machine learning algorithms [14]. In this section we will present the most important and well-known deep learning algorithms and their fields of application. Generally, most of the reviews like [6] classify these algorithms into several models which are convolutional neural network, auto-encoder, deep belief network, recurrent neural network (RNN), generative adversarial network, and deep reinforcement learning. While [15] classifies them into supervised, unsupervised and hybrid algorithms.

2.1. Multilayer perceptrons

Multi-layered perceptron (MLP) [16], is a neural network made up of several layers. In addition to input and output layers, MLP contains many hidden layers (by default, MLP has three hidden layers). Each layer is made up of a variable number of neurons. A neuron has inputs, which are real values, denoted by x1, . . . xn, and an output, denoted y, see Figure 1.

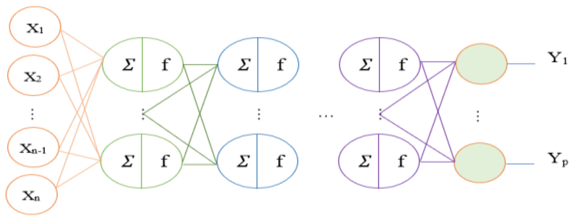


Figure 1. The multilayer perceptron architecture

To solve any problem by MLP it is necessary to determine the best weights of the lines connecting the neurons. For this, MLP uses backpropagation [17] as a training method. This requires the use of a differentiable activation function such as the sigmoid, rectified linear unit (ReLU) and tanh functions (Table 1), in order to iteratively define the weights in the network, with the aim of minimizing the deviation at the targeted output. MLP is used in various applications such as machine translation, speech recognition, and image classification. Moreover, the performance of some methods based on machine learning algorithms [18], [19] can be improved by using MLP.

2.2. Convolutional neural networks

Convolutional neural networks (CNN) [20] are types of artificial neural networks based on the idea that neurons in the visual cortex search for features. Pre-processing in CNN is much less compared to other algorithms. For example, CNN receives an input text and defines the learnable features in the text. CNNs are more often used in several fields, especially classification, document analysis, computer vision tasks, and images segmentation [21]. The architecture of CNN is constituted by three main types of layers, which are convolutional layers, mutualization layers and fully connected (FC) Layer. Figure 2 presents this architecture.

2.2.1. The convolution phase

In this step, a feature detector is applied to an area of the image to introduce the necessary features. To introduce non-linearity into the model after each convolution, CNN applies a linear transformation via the ReLU activation function to the feature matrix. CNN uses multiple convolutional layers, giving us a network that has a full understanding of the images in the dataset.

2.2.2. The pooling phase

A second layer called pooling is used to group convolutional features to reduce dimensions for easier preprocessing. Two types of pooling can be used (average and max pooling). Convolution and pooling can be considered as the first layer that allows CNN to accurately understand the features of the text, taking into account that complex texts require the multiplication of these layers.

2.2.3. The fully connected phase

Once the convolution and the pooling are complete. The obtained matrix must be transformed into a vector to include it in an artificial neural network. The improvement of the network is done by a flow of information until the desired state is reached.

2.3. Recurrent neural network

RNN [13] differ from other neural networks in that they have internal memory which allows them to store information associated with an input. This allows RNN to define sequential properties of data to use them to predict future scenarios. This allows us to predict very accurately what will happen. Especially in NLP tasks, which is one of the most important application areas of RNN. It also uses long short-term memory (LSTM), to provide long-term memory, see Figure 3.

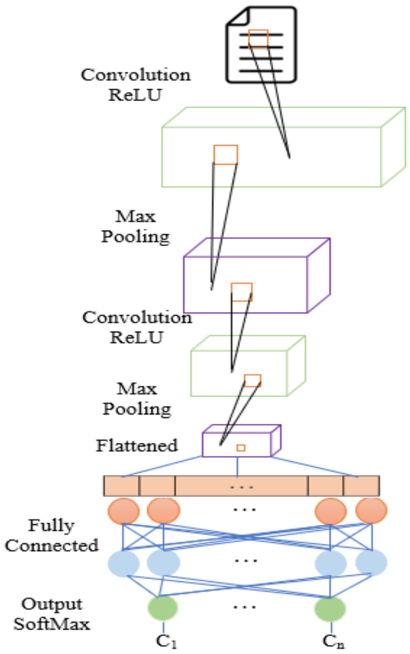


Figure 2. The convolutional neural network architecture

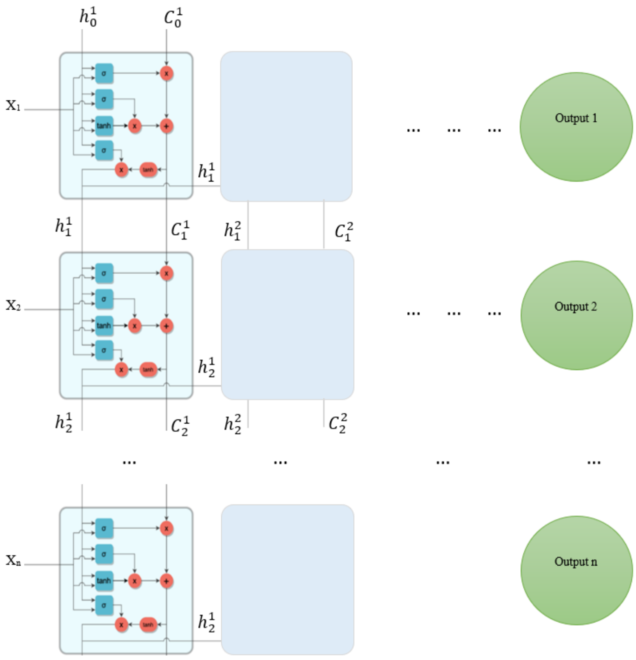


Figure 3. RNN s architecture with LSTM cell

The RNN uses two inputs for each neuron, the current input and the output of the preceding neuron. The decision is always tied to the current entry and what he has learned in the past. The weights are modified either by gradient descent [22] or backpropagation through time (BPTT) [23]. When RNN has a large number of time steps, it is better to use gradient descent because it is less computationally expensive than BPTT. RNNs are used in several NLP applications such as text generation, machine translation, and text summarization. RNN can also be used to predict the content of ancient manuscripts that have lost some of their content, which may perform better than some methods that predict handwritten words [24].

2.4. Autoencoder

Autoencoder [25] is a type of unsupervised neural network. It is characterized by the fact that the input data is the same as the output data. An auto-encoder consists of an encoder, decoder, artificial neural networks (ANN), and code, which is a single layer of ANN that summarizes input data, it is also called representation latent space.

Building an encoder requires an encoding method, a decoding method, and a loss function to compare the output to the target. The structure of the decoder is often the mirror image of the encoder as shown in Figure 4. But this is not necessary. The prerequisite is that the dimensions of the inlet and the outlet are identical. There are several types of autoencoders including, convolutional autoencoders, sparse autoencoders, and deep autoencoders [26]. The autoencoder can be used for solving tasks such as data analysis, information retrieval, or keyphrase extraction.

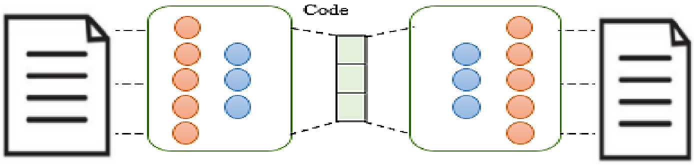


Figure 4. Autoencoder architecture

2.5. Deep belief networks

The deep belief network (DBN) [27] is one of the most important types of unsupervised deep neural networks. The DBN architecture consists of several layers of restricted Boltzmann machines (RBM) [28]. It is a stochastic RNN consisting of a layer of visible units, v, and a layer of hidden units, h, where each layer is connected to the previous and next layers to act as a hidden layer for the nodes that precede it and the role of the input layer for the nodes that follow. Figure 5 shows the architecture of the DBN. DBN remains a solution to many tasks. It can be used to reduce feature dimensions and to recognize images. It can also be used for handwriting recognition and speech recognition.

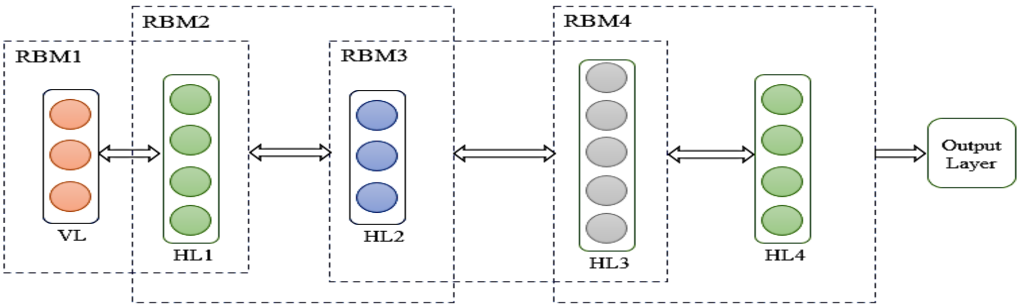
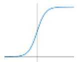
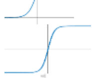
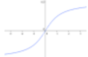
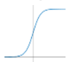


Figure 5. Deep belief networks architecture

2.6. Activation function

All artificial neural networks use non-linear functions to be able to bound the result of a summation in a neuron. Generally, its role is to determine whether or not to activate a neural response. These are also called activation functions that we perform before sending the value of the neuron to the next layer. Several types of activation functions are used by DL algorithms. Table 1 presents the most popular functions.

Table 1. Activation functions used by DL algorithms

Name	Function	Output value	Plot
Heaviside	$f(x) = \begin{cases} 0 & for\ x < 0 \\ 1 & for\ x \geq 0 \end{cases}$	(0,1)	
Sigmoïde	$f(x) = \dfrac{1}{1 + e^{-x}}$	[0,1]	
TanH	$f(x) = \dfrac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$	[-1,1]	
ArcTan	$f(x) = \tan^{-1}(x)$	$[-\frac{\pi}{2}, \frac{\pi}{2}]$	
Signum	$f(x) = \begin{cases} 1 & for\ x > 0 \\ 0 & for\ x = 0 \\ -1 & for\ x < 0 \end{cases}$	[-1,1]	
ReLU	$f(x) = \begin{cases} 0 & for\ x < 0 \\ x & for\ x \geq 0 \end{cases}$	$[0, \infty]$	
Gaussian	$f(x) = e^{-x^2}$	[0,1]	
Softmax	$f(x_i) = \dfrac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}} \quad i \in \{1, \ldots, N\}$	[0,1]	

3. KEYPHRASES EXTRACTION APPROACHES

Keyphrase extraction approaches are varied depending on the techniques used, where Nikzad-Khasmakhi *et al*. [29] categorizes them into textual, graph-based, and hybrid models. Chi and Hu [30] classifies them into supervised, unsupervised, and deep learning. In this section, we will present methods for extracting keyphrases that rely on deep learning.

We will divide our presentation into two parts. The first includes methods that only extract keyphrases mentioned in the text. While the second part presents methods that also predict keyphrases that are not mentioned in the text.

3.1. Keyphrases extraction

The approach proposed [31] is the first to use deep learning techniques to extract keyphrases from the text. It is based on a neural network trained to determine whether a candidate phrase is a keyphrase or not, using the values of four features, which are term frequency, inverted document frequency, appearances in the document title, and the frequency of appearance in the paragraphs. The disadvantage of this method is that it gives all keyphrases the same importance. Thus, we cannot choose a specific number of the most important keyphrases. For this, Sarkar *et al.* [32] proposes to use a trained multilayer neural network to classify the candidate phrases according to their probability of being keyphrases or not. To choose the number of desired keyphrases. Other deep learning algorithms are also used like RNN which was exploited [33] to propose an automatic keyphrases extraction (AKE) method from tweets. The RNN model used has two hidden layers. The first is used to identify keyphrase information, while the second extracts the keyphrase using a sequence labeling approach.

3.2. Keyphrase generation

Some methods attempt in addition to extracting the keyphrases mentioned in the text, to predict the keyphrases not mentioned in the text. Meng *et al.* [34] propose a supervised approach to predicting keyphrases based on an auto-encoder that captures the semantic meaning of content via the RNN method. The approach focuses on compressing the original text into a hidden layer using an encoder and predicting a keyphrase using a decoder. However, this approach suffered from some problems, the most important of which is the prediction of keyphrases that express the same meaning, so the extracted keyphrases do not cover the topics of the document.

To overcome these problems, Chen *et al.* [35] corrected the previous approach, by using CorrRNN, to predict keyphrases that do not have the same meaning and cover the topics of the document. This correction requires a large amount of labeled data for training. Ye and Wang [36] attempted to propose a method that reduces the amount of data prepared for training using term frequency-inverse document frequency (TFIDF) and TextRank [37], to obtain the set of keyphrases used to train a multitasking pattern. Wang *et al.* [38] also proposed creation of a topic-based adversarial neural network (TANN) that uses both labeled and unlabeled data to reduce the amount of data used for training. Basaldella *et al.* [39] believe that exploiting the preceding and following context of a given phrase can help predict keyphrases, for this, propose a bidirectional long short-term memory (BiLSTM) RNN network predicts keyphrases.

Other methods not only use deep learning techniques, but also add other techniques such as conditional random field (CRF) [40], and sentence embedding [41] techniques. Alzaidy *et al.* [42] propose the combination of BiLSTM and CRF. The first captures the semantics of the phrase and the second gives a probability distribution over the phrase using the dependencies between the labels (keyphrase or non-keyphrase). Zhang and Xiao [43] propose a model based on seq2seq RNN, which can extract both keyphrases present and predict others not existing in the document by capturing the semantic, linguistic, and statistical information. Santosh *et al.* [44] propose document-level attention for keyphrase extraction (DAKE), a model that combines BiLSTM and CRF which is enhanced with interest at the document level and a gateway mechanism to improve the extraction of key phrases from scientific documents. Also, Huanqin *et al.* [45] propose a method that relies on the use of keyphrases mentioned in the text, to construct keyphrases not mentioned in the text using a mask-predict method.

3.3. Deep learning techniques for AKE

The DL techniques used by the keyphrase extraction methods that we presented in the previous paragraph were divided into two sets. Traditional techniques like multilayer feed-forward neural network and multilayer perceptron neural network, and the modern techniques as encoders and RNN variants, such as LSTM, BiLSTM, and bidirectional gated recurrent unit (BiGRU). Table 2 shows the DL techniques used by each AKE method.

Table 2. DL techniques used by the ake method

Approach	DL techniques	Activation function	Type
Wang *et al.* [31]	Multilayer feed-forward neural network	Sigmoid	Supervised backpropagation algorithm
Sarkar *et al.* [32]	Multilayer perceptron neural network	Sigmoid	Supervised backpropagation algorithm
Zhang *et al.* [33]	RNN	Sigmoid, Softmax	Supervised stochastic gradient descent [46]
Meng *et al.* [34]	RNN	Sigmoid	Supervised BiGRU [47]
Chen *et al.* [35]	RNN	Sigmoid, Softmax	Supervised BiGRU
Ye and Wang [36]	BiLSTM model LSTM model	Sigmoid, Tanh Softmax	Semi-supervised self-learning algorithm [48]
Wang *et al.* [38]	BiLSTM network CNN	Sigmoid, Tanh, ELU	Supervised adversarial learning technique
Basaldella *et al.* [39]	BiLSTM RNN	Softmax, Tanh, Sigmoid	Supervised root mean square propagation optimization algorithm [49]
Alzaidy *et al.* [42]	BiLSTM	Tanh, Sigmoid	Supervised stochastic gradient descent
Zhang and Xiao [43]	RNN BiGRU Unidirectional GRU	Tanh Sigmoid, Softmax	Supervised skip-gram [50]
Santosh *et al.* [44]	BiLSTM	Tanh, Sigmoid	Supervised adam optimization method [49]
Wu *et al.* [45]	Prefix LM (encoder-decoder) [51]	Softmax	Supervised multitask learning [52]

We also noted that only four activation functions remain preferred by the AKE methods, namely sigmoid tanh, exponential linear unit (ELU) and Softmax. These functions are suitable for the DL techniques used, especially the RNN technique and its variants which have been used by 70% of AKE methods studied as shown in Figure 6. The biggest problem with AKE methods that rely on DL techniques is that they are either supervised or semi-supervised, which requires providing datasets for training, which is not always available.

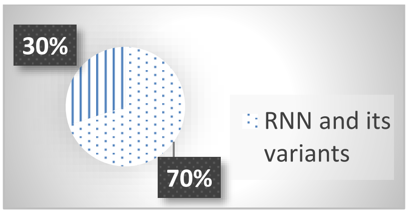


Figure 6. Percentage of use of DL techniques

4. EMPIRICAL RESULTS

In this section, we will describe the training and test datasets that were used by the studied AKE methods. In addition, the most commonly used evaluation metrics. Then, we discuss the results obtained by these methods.

4.1. Datasets

To train or evaluate AKE methods, several datasets are used. Table 3 presents the datasets used by the methods studied. However, methods based on DL techniques require large datasets for training. Unfortunately, before 2017 the largest database available contained only 2,304 scientific articles [53]. This is insufficient to train RNN. But with the construction of KP20K which contains 527,430 documents, it became the favorite of the methods that appeared after 2017. In addition, most of the studied methods relied on five datasets to evaluate their performance, Table 3 presents the performance evaluation datasets for AKE methods. There is also a recent dataset, KPTimes [54] that has not been used which provides 259,923 training documents.

Table 3. Datasets used by the studied ake methods

Dataset	Documents	Training documents	Test documents	Validation documents	Usage rate (%)
Inspec	2,000	1,000	500	500	50
Krapivin	2,304	1,900	404	-	42
NUS	211	-	211	-	50
SemEval	288	188	100	-	58
KP20K	527,430	527,030	20,000	20,000	58
KDD	755	-	755	-	8
WWW	1,330	-	1,330	-	8

4.2. Evaluation metrics

All the methods studied were based on three metrics of performance evaluation. Most of AKE methods evaluate its performance based on the results of these metrics. They are:

$$Precision = \frac{True\ Keyphrases}{True\ Keyphrases + False\ Keyphrases} \quad (1)$$

$$Recall = \frac{True\ Keyphrases}{True\ Keyphrases + False\ NonKeyphrase} \quad (2)$$

$$F1\_Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (3)$$

The results of these measurements remain relative because they are affected by the number of phrases extracted and the nature and length of the document. Also, methods that do not predict phrases that do not exist in the document will have fewer results when using these measures. It will therefore be necessary to think about other ways of evaluating performance that go beyond these constraints.

4.3. Performance

To evaluate the performance of the studied approaches. The authors relied on the evaluation metrics discussed in the previous paragraph by applying them to five datasets. Table 4 compares the average performance of methods that extract only keyphrases found in the document with methods that also predict keyphrases that are not mentioned in the document.

Table 4. Comparison of the kp extraction and kp generation methods

Dataset	F1-score	KP extraction	KP generation
Inspec	F@5	0.27	0.36
	F@10	0.21	0.33
Krapivin	F@5	0.12	0.22
	F@10	0.15	0.26
NUS	F@5	0.15	0.35
	F@10	0.19	0.41
SemEval	F@5	0.14	0.29
	F@10	0.17	0.31
KP20K	F@5	-	0.38
	F@10	-	0.34

Thus, from these results, it is clear that methods that extract only the keyphrases mentioned in the document perform less well than methods that extract the keyphrases mentioned or not in the document. This can be explained by the fact that most of the datasets used to evaluate AKE methods contain documents in which keyphrases not mentioned in the document are specified [55]. Figure 7 shows the distribution of present and absent keyphrases according to each dataset.

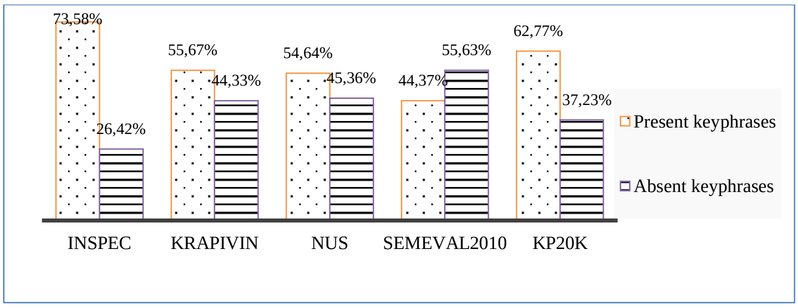


Figure 7. Percentage of present and absent keyphrases in datasets

We then analyzed the performance according to the DL technique used. We calculated the average performance of the methods studied according to the DL technique used. Table 5 shows the results obtained for each dataset.

Despite the need to provide a large dataset for training BiLSTM-based approaches. The results presented in Table 5 clearly show that these methods perform better than the others. Therefore, it is recommended to use the BiLSTM technique to extract and predict keyphrases from a document, especially with the availability of datasets for training and validation [54].

Table 5. Performance of each dl technique according to each dataset

Dataset	F1-score	MLP	Simple RNN	LSTM RNN	BiLSTM RNN	BiGRU RNN
Inspec	F@5	0.17	0.22	0.25	0.29	0.27
	F@10	0.21	0.24	0.24	0.31	0.30
Krapivin	F@5	0.15	0.19	0.29	0.32	0.31
	F@10	0.13	0.15	0.22	0.28	0.26
NUS	F@5	0.28	0.32	0.38	0.46	0.39
	F@10	0.30	0.31	0.33	0.41	0.32
SemEval	F@5	0.16	0.21	0.27	0.29	0.27
	F@10	0.13	0.18	0.30	0.32	0.32
KP20K	F@5	-	-	0.31	0.37	0.33
	F@10	-	-	0.33	0.35	0.29

5. DISCUSSION

Recently, there has been a lot of interest in using deep learning techniques in several fields. In this study, we highlight deep learning techniques to give an idea of the techniques that correspond to each domain to know which techniques are exploitable in NLP tasks. Especially the process of extracting keyphrases from the document. When analyzing the studied AKE methods, we found that the authors limit themselves only to RNNs and their variants because they are a good solution to sequential data problems, such as speech and language processing [56].

Through the results obtained, we found that the methods which extract the keyphrases present only in the text are less efficient than the methods which also predict the absent keyphrases. One of the reasons for the superiority of these models is due to the evaluation method used, which is based on datasets in which half of the keyphrases are not mentioned in the documents. Thus, AKE models that only extract the keyphrases mentioned in the document remain less performant. Empirical results also showed that models based on BiLSTM have higher extraction and prediction ability than other techniques. On the other hand, training these models requires a large amount of data. Therefore, it is recommended to use the BiLSTM technique to extract and predict keyphrases, especially with datasets available for training and validation.

CNNs are more efficient for data consisting of matrices such as images and videos, which explains the dominance of this technique on computer vision tasks [16]. However, it also performed well when used in NLP tasks [57]. Since CNN has a great capacity for classification, its use to predict key phrases not mentioned in the document can improve the performance of AKE approaches. Additionally, we encourage researchers interested in keyphrase extraction and prediction, to further research into deep learning techniques, especially regarding the amount of input, number of hidden layers, and discover key phrase features, loss functions, and activation functions. This will inevitably lead to better-performing keyphrase extraction and prediction methods.

6. CONCLUSION

Keyphrases are one of the solutions exploited to improve the performance of NLP tasks such as information retrieval, summarizing, classifying, and clustering documents. Our article presents a review of keyphrase methods that use deep learning techniques to understand how to use deep learning in the extraction and prediction of keyphrases. Most AKE models that used DL techniques, chose the RNN or one of its variants such as LSTM, BiLSTM, and BiGRU. Our review also included an evaluation of the performance of the AKE methods studied. Through the results obtained, it is shown that the BiLSTM technology performed better than the other techniques and that the methods which predict the absent keyphrases performed better than the methods which only extract the keyphrases mentioned in the document. Generally, the performance of AKE methods based on deep learning remains better than other methods, especially unsupervised methods, but on the other hand, their weak point remains that they require a large amount of data for learning and validation. In the future, we will expand our study to develop an unsupervised system that takes advantage of deep learning techniques and focuses on predicting keyphrases in documents whether they are present or absent.

REFERENCES

[1] X. Li and M. Daoutis, "Unsupervised key-phrase extraction and clustering for classification scheme in scientific publications," *CEUR Workshop Proceedings*, vol. 2831, pp. 1–8, Jan. 2021, [Online]. Available: http://arxiv.org/abs/2101.09990

[2] F. Boudin and Y. Gallina, "Redefining Absent Keyphrases and their Effect on Retrieval Effectiveness," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Mar. 2021, pp. 4185–4193, doi: 10.18653/v1/2021.naacl-main.330.

[3] H. Li, J. Zhu, J. Zhang, C. Zong, and X. He, "Keywords-guided abstractive sentence summarization," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 5, pp. 8196–8203, Apr. 2020, doi: 10.1609/aaai.v34i05.6333.

[4] S. Varastehpour, H. Sharifzadeh, and I. Ardekani, "A comprehensive review of deep learning algorithms," New Zealand, 2021, doi: 10.34074/ocds.092.

[5] R. K. Mishra, G. Y. S. Reddy, and H. Pathak, "The understanding of deep learning: A comprehensive Review," *Mathematical Problems in Engineering*, pp. 1–15, Apr. 2021, doi: 10.1155/2021/5548884.

[6] H. Abdel-Jaber, D. Devassy, A. A. Salam, L. Hidaytallah, and M. EL-Amir, "A review of deep learning algorithms and their applications in healthcare," *Algorithms*, vol. 15, no. 2, pp. 1–55, Feb. 2022, doi: 10.3390/a15020071.

[7] P. Yang, Y. Ge, Y. Yao, and Y. Yang, "GCN-based document representation for keyphrase generation enhanced by maximizing mutual information," *Knowledge-Based Systems*, vol. 243, p. 108488, May 2022, doi: 10.1016/j.knosys.2022.108488.

[8] T. Munz, D. Väth, P. Kuznecov, N. T. Vu, and D. Weiskopf, "Visualization-based improvement of neural machine translation," *Computers & Graphics*, vol. 103, pp. 45–60, Apr. 2022, doi: 10.1016/j.cag.2021.12.003.

[9] N. R. Bhowmik, M. Arifuzzaman, and M. R. H. Mondal, "Sentiment analysis on Bangla text using extended lexicon dictionary and deep learning algorithms," *Array*, vol. 13, pp. 1–14, Mar. 2022, doi: 10.1016/j.array.2021.100123.

[10] M. Pota, M. Esposito, G. D. Pietro, and H. Fujita, "Best practices of convolutional neural networks for question classification," *Applied Sciences*, vol. 10, no. 14, pp. 1–27, Jul. 2020, doi: 10.3390/app10144710.

[11] S. Hamida, B. Cherradi, and H. Ouajji, "Handwritten Arabic words recognition system based on hog and gabor filter descriptors," in *2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, Apr. 2020, pp. 1–4, doi: 10.1109/IRASET48871.2020.9092067.

[12] S. Lee and D. Kim, "Deep learning based recommender system using cross convolutional filters," *Information Sciences*, vol. 592, pp. 112–122, May 2022, doi: 10.1016/j.ins.2022.01.033.

[13] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *IEEE Computational Intelligence Magazine*, vol. 13, no. 3, pp. 55–75, Aug. 2018, doi: 10.1109/MCI.2018.2840738.

[14] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006, doi: 10.1126/science.1127647.

[15] P. Dixit and S. Silakari, "Deep learning algorithms for cybersecurity applications: A technological and status review," *Computer Science Review*, vol. 39, pp. 1–15, Feb. 2021, doi: 10.1016/j.cosrev.2020.100317.

[16] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.

[17] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986, doi: 10.1038/323533a0.

[18] O. Terrada, S. Hamida, B. Cherradi, A. Raihani, and O. Bouattane, "Supervised machine learning based medical diagnosis support system for prediction of patients with heart disease," *Advances in Science, Technology and Engineering Systems Journal*, vol. 5, no. 5, pp. 269–277, 2020, doi: 10.25046/aj050533.

[19] O. Terrada, B. Cherradi, A. Raihani, and O. Bouattane, "Atherosclerosis disease prediction using supervised machine learning techniques," in *2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, Apr. 2020, pp. 1–5, doi: 10.1109/IRASET48871.2020.9092082.

[20] F. Emmert-Streib, Z. Yang, H. Feng, S. Tripathi, and M. Dehmer, "An introductory review of deep learning for prediction models with big data," *Frontiers in Artificial Intelligence*, vol. 3, pp. 1–23, Feb. 2020, doi: 10.3389/frai.2020.00004.

[21] H. Moujahid, B. Cherradi, and L. Bahatti, "Convolutional neural networks for multimodal brain mri images segmentation: A comparative study," in *Smart Applications and Data Analysis*, Cham: Springer International Publishing, 2020, pp. 329–338, doi: 10.1007/978-3-030-45183-7_25.

[22] S. Ruder, "An overview of gradient descent optimization algorithms," pp. 1–14, Sep. 2016, [Online]. Available: http://arxiv.org/abs/1609.04747

[23] J. Guo, "Backpropagation through time," *Unpubl. ms., Harbin Institute of Technology*, vol. 40, pp. 1–6, 2013, [Online]. Available: http://ir.hit.edu.cn/$~$jguo/docs/notes/bptt.pdf

[24] S. Hamida, B. Cherradi, O. Terrada, A. Raihani, H. Ouajji, and S. Laghmati, "A novel feature extraction system for cursive word vocabulary recognition using local features descriptors and gabor filter," in *2020 3rd International Conference on Advanced Communication Technologies and Networking (CommNet)*, Sep. 2020, pp. 1–7, doi: 10.1109/CommNet49926.2020.9199642.

[25] M. Sewak, S. K. Sahay, and H. Rathore, "An overview of deep learning architecture of deep neural networks and autoencoders," *Journal of Computational and Theoretical Nanoscience*, vol. 17, no. 1, pp. 182–188, Jan. 2020, doi: 10.1166/jctn.2020.8648.

[26] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders," pp. 1–22, Mar. 2020, [Online]. Available: http://arxiv.org/abs/2003.05991

[27] Y. Hua, J. Guo, and H. Zhao, "Deep belief networks and deep learning," in *Proceedings of 2015 International Conference on Intelligent Computing and Internet of Things*, Jan. 2015, pp. 1–4, doi: 10.1109/ICAIOT.2015.7111524.

[28] N. Zhang, S. Ding, J. Zhang, and Y. Xue, "An overview on restricted Boltzmann machines," *Neurocomputing*, vol. 275, pp. 1186–1199, Jan. 2018, doi: 10.1016/j.neucom.2017.09.065.

[29] N. Nikzad-Khasmakhi *et al.*, "Phraseformer: Multimodal key-phrase extraction using transformer and graph embedding," pp. 1–15, 2021, [Online]. Available: http://arxiv.org/abs/2106.04939

[30] L. Chi and L. Hu, "ISKE: An unsupervised automatic keyphrase extraction approach using the iterated sentences based on graph method," *Knowledge-Based Systems*, vol. 223, pp. 1–12, Jul. 2021, doi: 10.1016/j.knosys.2021.107014.

[31] J. Wang, H. Peng, and J. Hu, "Automatic keyphrases extraction from document using neural network," in *Advances in Machine Learning and Cybernetics*, Berlin: Springer, 2006, pp. 633–641, doi: 10.1007/11739685_66.

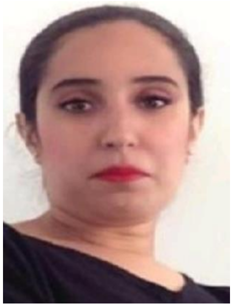
[32] K. Sarkar, M. Nasipuri, and S. Ghose, "A new approach to keyphrase extraction using neural networks," pp. 16–25, 2010, [Online]. Available: http://arxiv.org/abs/1004.3274

[33] Q. Zhang, Y. Wang, Y. Gong, and X. Huang, "Keyphrase extraction using deep recurrent neural networks on Twitter," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 836–845, doi: 10.18653/v1/D16-1080.

[34] R. Meng, S. Zhao, S. Han, D. He, P. Brusilovsky, and Y. Chi, "Deep keyphrase generation," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 582–592, doi: 10.18653/v1/P17-1054.
[35] J. Chen, X. Zhang, Y. Wu, Z. Yan, and Z. Li, "Keyphrase generation with correlation constraints," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 4057–4066, doi: 10.18653/v1/D18-1439.
[36] H. Ye and L. Wang, "Semi-supervised learning for neural keyphrase generation," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 4142–4153, doi: 10.18653/v1/D18-1447.
[37] R. Mihalcea and P. Tarau, "Textrank: Bringing order into text," in *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004, pp. 404–411. [Online]. Available: https://aclanthology.org/W04-3252/
[38] Y. Wang *et al.*, "Exploiting topic-based adversarial neural network for cross-domain keyphrase extraction," in *2018 IEEE International Conference on Data Mining (ICDM)*, Nov. 2018, vol. 2018-Novem, pp. 597–606, doi: 10.1109/ICDM.2018.00075.
[39] M. Basaldella, E. Antolli, G. Serra, and C. Tasso, "Bidirectional LSTM recurrent neural network for keyphrase extraction," in *Digital Libraries and Multimedia Archives*, vol. 806, Cham: Springer International Publishing, 2018, pp. 180–187, doi: 10.1007/978-3-319-73165-0_18.
[40] H. M. Wallach *et al.*, "Conditional random fields: An introduction," 2004. [Online]. Available: https://repository.upenn.edu/cgi/viewcontent.cgi?article=1011&context=cis_reports
[41] L. Ajallouda, K. Najmani, A. Zellou, and E. H. Benlahmar, "Doc2Vec, SBERT, InferSent, and USE which embedding technique for noun phrases?," in *2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, Mar. 2022, pp. 1–5, doi: 10.1109/IRASET52964.2022.9738300.
[42] R. Alzaidy, C. Caragea, and C. L. Giles, "Bi-LSTM-CRF sequence labeling for keyphrase extraction from scholarly documents," in *The World Wide Web Conference on - WWW '19*, 2019, pp. 2551–2557, doi: 10.1145/3308558.3313642.
[43] Y. Zhang and W. Xiao, "Keyphrase generation based on deep seq2seq model," *IEEE Access*, vol. 6, pp. 46047–46057, 2018, doi: 10.1109/ACCESS.2018.2865589.
[44] T. Y. S. S. Santosh, D. K. Sanyal, P. K. Bhowmick, and P. P. Das, "DAKE: Document-level attention for keyphrase extraction," in *Advances in Information Retrieval*, Cham: Springer International Publishing, 2020, pp. 392–401, doi: 10.1007/978-3-030-45442-5_49.
[45] H. Wu, B. Ma, W. Liu, T. Chen, and D. Nie, "Fast and constrained absent keyphrase generation by prompt-based learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Jun. 2022, vol. 36, no. 10, pp. 11495–11503, doi: 10.1609/aaai.v36i10.21402.
[46] N. Ketkar, "Stochastic gradient descent," in *Deep Learning with Python*, Berkeley, CA: Apress, 2017, pp. 113–132, doi: 10.1007/978-1-4842-2766-4_8.
[47] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," Jun. 2014, [Online]. Available: http://arxiv.org/abs/1406.1078
[48] X. Li, P. Lu, L. Hu, X. Wang, and L. Lu, "A novel self-learning semi-supervised deep learning network to detect fake news on social media," *Multimedia Tools and Applications*, vol. 81, no. 14, pp. 19341–19349, Jun. 2022, doi: 10.1007/s11042-021-11065-x.
[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," pp. 1–15, Dec. 2014, [Online]. Available: http://arxiv.org/abs/1412.6980
[50] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," pp. 1–9, Oct. 2013, [Online]. Available: http://arxiv.org/abs/1310.4546v1%0Apapers2://publication/uuid/FB1742A3-202C-44CA-98F5-6EA51EC019D2
[51] C. Raffel *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, Oct. 2019, [Online]. Available: https://www.jmlr.org/papers/volume21/20-074/20-074.pdf?ref=https://githubhelp.com
[52] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 7482–7491, doi: 10.1109/CVPR.2018.00781.
[53] M. Krapivin, A. Autaeu, and M. Marchese, "Large dataset for keyphrases extraction," 2009. [Online]. Available: http://eprints.biblio.unitn.it/1671/
[54] Y. Gallina, F. Boudin, and B. Daille, "KPTimes: A large-scale dataset for keyphrase generation on news documents," in *Proceedings of the 12th International Conference on Natural Language Generation*, 2019, pp. 130–135, doi: 10.18653/v1/W19-8617.
[55] J. Ye, T. Gui, Y. Luo, Y. Xu, and Q. Zhang, "One2Set: Generating diverse keyphrases as a set," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 4598–4608, doi: 10.18653/v1/2021.acl-long.354.
[56] J. Hirschberg and C. D. Manning, "Advances in natural language processing," *Science*, vol. 349, no. 6245, pp. 261–266, Jul. 2015, doi: 10.1126/science.aaa8685.
[57] M. Xia, A. Anastasopoulos, R. Xu, Y. Yang, and G. Neubig, "Predicting performance for natural language processing tasks," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 8625–8646, doi: 10.18653/v1/2020.acl-main.764.

BIOGRAPHIES OF AUTHORS

Lahbib Ajallouda Ph.D student at Computer Science and Systems Analysis School (ENSIAS), Mohamed V University, Rabat, Morocco. His research interests are primarily in the area of internet of things, search engines, cloud computing, and machine learning, where he is the author/co-author of over 6 research publications. He can be contacted at email: lahbib_ajallouda@um5.ac.ma.



Fatima Zahra Fagroud Ph.D student at Faculty of Sciences Ben M'sick, Hassan II University of Casablanca, Morocco. Her research interests are primarily in the area of internet of things, search engines, cloud computing, machine learning, where she is the author/co-author of over 14 research publications. She can be contacted at email: fagroudfatimazahra0512@gmail.com.

Ahmed Zellou Received his Ph.D. in Applied Sciences at the Mohammedia School of Engineers, Mohammed V University, Rabat, Morocco 2008. He is currently a coordinator of the IWIM Web Engineering & Mobile Computing branch at ENSIAS Mohamed V university in Rabat, Morocco. His research interests include parallel computing, information systems (business informatics), and distributed computing, where he is the author/co-author of over 72 research publications. He can be contacted at email: ahmed.zellou@um5.ac.ma.

EL Habib Benlahmar received his Ph.D, computer science at Computer Science and Systems Analysis School (ENSIAS), University Mohamed V, Rabat, Morocco. He is currently a coordinator of the master data science & Big Data at FSBM Hassane II University Casablanca Morocco. His research interests include educational technology, software engineering, information systems (business informatics), and human-computer interaction, where he is the author/co-author of over 165 research publications. He can be contacted at email: h.benlahmer@gmail.com.