I.J. Image, Graphics and Signal Processing, **2021, 2, 16-24**
Published Online April 2021 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijigsp.2021.02.02

Analog Document Search Using CRNN and Keyphrase Extraction

Lokeshwar S, Vadiraja Rao M. K, Sujay Kumar P. S, Vishveshwara Guthal Gowda, Hemavathi P.
Bangalore Institute of Technology (VTU), Bengaluru, Karnataka, India
Email: {lokeshwar1998, vadiraja98rao, sujayputtu, vg931697, hemavathibitcse09}@gmail.com

Received: 30 June 2020; Revised: 23 July 2020; Accepted: 02 September 2020; Published: 08 April 2021

Abstract: There seems to be a peculiar trend in the way information is now used, moving to digital media not just for the newspapers but for books as well. With advances in Optical Character Recognition (OCR), Style Transfer Mapping (STM), and efficient key phrasing, we are now able to digitalize the document to a form that can be read across multiple platforms and searched efficiently. It provides users with the ease of searching for relevant documents without the tedious process of manual searching.

We propose a system that uses the CRNN model to detect English characters in the document with high accuracy. We then pair it with a hybrid keyphrasing technique, which uses Positional Rank as its Graph based rank and re-rank the key phrases using the C-Value method. This process allows us to automatically digitize the printed document and summarise it to provide high-quality keyphrases, which can be used to efficiently search and retrieve relevant printed documents.

Index Terms: Analog document search, CRNN, Keyphrase Extraction, Position Rank

1. Introduction

Information and historical events have always been stored in a paper document herein referred to as an analog document. Selecting the document of your interest to read before having to search through each document if it interests you is a tedious task. To make things easier, we require the document to be digitized first, i.e., to a format that can be understood by computers, so that we can efficiently perform keyphrase extraction on the document to allow searching.

Often these documents are fragile, weak, and prone to being destroyed. Preserving these documents for efficient reuse can be done using Optical Character Recognition (OCR). Optical character recognition (OCR) is the process of converting the images of printed text into machine readable or editable form. This could be from a document image or a scanned document. Using this method, the printed text can be converted to an electronically editable form. This enables us to search, store the document for further analysis by computers.

At first, have to preprocess the image of the document captured primarily to supplement accurate character recognition. The document if not aligned during capturing of the image, has to be titled either clockwise or anti-clockwise to align the text lines of the document either horizontally or vertically. The aspect ratio of the image is normalized to scale.

Feature extraction decomposes images of characters into "features" like lines and closed loops. The dimensionality of representation is reduced using Feature extraction. The recognition process can be made computationally efficient using abstract Vector representation, while comparing the features extracted to reduce to one of the image classification.

In this paper, we propose a CRNN (Convolutional Recurrent Neural Network) which performs feature extraction and then classifies the characters into words with decent accuracy. A CRNN is a combination of CNN (Convolution Neural Network) and RNN (Recurrent Neural Network). CNN is efficient at extracting features that are spatially independent, making it a good choice for image recognition. Since we are identifying words in OCR, making use of an RNN on top of the existing CNN model, makes it even more efficient as RNN can cover-up the spots where CNN might fail hence improving the efficiency of the entire network. At the end of the pre-processing phase, the original document image is segmented into words which are given as input to the neural network. The CNN extracts features from these segmented images to a certain depth. It then removes the spatial data by feeding all these features into a fully connected layer, which then feeds to RNN for further processing. At the end of the RNN step, there will be a fully connected layer again, which classifies the input into its respective characters. If the CNN model misclassifies a particular character, then RNN will be able to identify the error at once. RNN corrects those misclassifications to make sure the word is spelled correctly, thus being a better architecture than having just CNN. Since the RNN is subject to vanishing gradient problem, we use LSTM's to overcome the drawback. To avoid overfitting of the model, we also make use of dropout regularization after the LSTM, before going into the classification or the output layer.

This work is open access and licensed under the Creative Commons CC BY 4.0 License.

The growing abundance of text articles in analog documents requires automated tagging using keyphrases to extract the most relevant document for a particular search query of a user. The keyphrase has the essence of the text in a concise manner, this enables the user to grasp the main idea of the text, without the need of reading through the entire text document. People generally search for required documents through a search engine., inputting Keyphrases that are short, simple descriptors of what the document is really associated with or talking about. Without any doubt, the process of manually finding keyphrases or summarizing texts is herculean task. To allow users to access the relevant document efficiently using just keyphrases, we need to devise the keyphrases extraction model that can extract high-quality keyphrases from the documents. We need an automated approach for Generating the keyphrases for texts from all possible domains. This would reduce the time and effort and will meet the unprecedented volume of documents that is being exchanged.

Unsupervised keyphrase extraction can be implemented by either statistical method or graph-based method. Statistical methods use frequency information from the document as a whole. This results in difficultly in extracting high-quality keyphrases from a document set consisting of various domains. Graph-based models build a graph with their vertices being populated by words, and their edges being the relative measure of the co-occurrences between them. Graph-based approach has multiple drawbacks, that are discussed by Yeom [1], one of the drawbacks is that the approach is biased when scoring keyphrases within a specified window size. For example, suppose we have two similar keyphrases: "Unsupervised keyphrase extraction" and "Unsupervised extraction" in the given document. The co-occurrence frequency of "Unsupervised" and "extraction" will be the highest among the three edges. Hence, "Unsupervised extraction" will be ranked higher than "Unsupervised keyphrase extraction" which will affect the quality of keyphrases produced.

Yeom[1] proposed a hybrid keyphrase extraction technique that combines both the Graph-based and statistical methods to overcome the drawbacks. In this paper, we use positional Rank [2] as our graphical method. In this method, the position of words within the digitized text is used in the extraction of high-quality keyphrases. We then use C-value [3] as the statistical method, re-rank the keyphrases to allow faster and accurate search results. We combine these two sub-systems to provide a whole system that can provide an efficient platform for users to search for the documents by keyphrases and keywords without having to go through them manually. Thus, reducing the time of searching for the required document by multiple folds.

The remainder of the paper is organized such that in section Ⅱ, presents review of the necessary background required to implement our system effectively. The proposed system is described in Section Ⅲ. Section Ⅳ, analyses the results and system performance. In the last section of this paper, conclusions are drawn.

2. Related Work

In [4], they propose a method to improve transfer learning. Often, few visual recognition tasks have limited dataset for training, using image representations extracted from CNNs which was pre-trained of existing large datasets is highly efficient. Computation of highly efficient mid-level Image representations for images is proposed by designing a method that uses ImageNet-Trained layers in CNN. This method of transfer learning can be used to train the OCR model to classify characters from other languages that share similar features of the language it is already trained for with reduced training data.

The authors in [5], porpose STM method which allows a classifier trained for printed Chinese character to be used for classifying the historical Chinese document having variation in font styles. This method does not use nonlinear transfer functions and integration with other methods like CNN.

In [6], we can initialize another CNN model using the feature extracted and learnt by another CNN model trained for printed Chinese character samples in the source domain. The network structure and weights of source model are used to initialize target CNN model.

Deep-learning method have not been augmented with Transfer learning methods. It is computationally efficient to combine CNN-based modules along with traditional transfer learning methods. There are multiple nonlinear activation functions that can be used in a perceptron to obtain results. In this work [7], the nonlinear activation functions are approximated as a Taylor series, and the coefficients are retrained using error back-propagation framework. Since the activation functions are realized as a Taylor series, computation time is slightly higher.

The authors in [8] present an Artificial Neural Network using feedforward neural network for character recognition. Feedforward back propagation neural network can be used for classifying and recognizing Latin characters. The stages of the system are image acquisition, pre-processing, feature extraction, training of the classifier, and finally, simulation of the trained classifier.

The survey in [9] proposes various stages required to include pre-processing, Classification, Segmented Processing, Feature Extraction. Pre-processing requires noise reduction, application gaussian filters skew detection and correction. Tesseract OCR engine uses character segmentation. We can Vector passing as well to classify characters. OCR divides each character space into 4 quadrants. Then extracts features. Based on this work, we performed image processing using Tesseract before passing it as input to the OCR model.

Neural network methods appended with backpropagation can reduce errors during the training of the network. This form neural network helps the system in learning things at a faster rate.

In [10] we calculate the frequency of the compound terms which is always less than single keyword, using boosting factor (ratio of single terms to compound terms). By this we get the effective keyphrases, repetition of these in the document are more likely to be the keyphrases, which will be weighted using TF-IDF algorithm, with the assumption of important term comes sooner if is less than the threshold then it is unlikely to be a keyphrase with these hypotheses in mind we extract keyphrases using three algorithm's candidate keyphrase selection, candidate keyphrase weight calculation, final keyphrase refinement.

The authors in [11], use the information provided by the word embedding vectors as the background knowledge to rank the keyphrases generated using the graph-based method. A weighted undirected graph, where vertices are represented by the words and edges between them are based on the co-occurrence relations between the two word embedded vectors within the defined window frame. Keyphrases of the given window size are then extracted from the graph.

The authors in [3] describe the application of C-value and NC-value working together where the C-value method is applied first and then is used along NC-value for the context of the candidate, initially C-value method which has a two combined approach that is linguistic part and the statistical part where more emphasis is made on the statistical part after all the computation is done, NC-value is used, and the candidate terms are weighed using this method

An unsupervised graph-based model for keyphrase extraction is proposed in [2], PositionRank, here the scores assigned to each keyword making up the keyphrases are biased on the position of the word within the document text.

There are various techniques under OCR which provide acceptable results, but they are not effective when it comes to noisy or broken characters. This method [12] talks about overcoming this drawback in the existing system.

As seen in [13], when the size of the document is not sufficiently large, keyphrases extraction methods based on statistical models have difficulty in obtaining a highly reliable probabilities of words. Whereas, the co-occurrence information is biased on the sliding window size which is used to extract high-quality keyphrases from the edge-scored graph of the document using any Graph based model.

3. Proposed System

The system architecture is shown in Fig. 1, the system consists of two phases. The first phase is the digitizing analog document to apply keyphrase extraction. Once we have the keyphrases and digitized document, we store it in a centralized data store. The second phase allows users to search for the relevant analog document by inputting keyphrases and keywords without having to go through each of them manually.

The use of NoSQL database, resulted in the use of JSON file format throughout the system for exchange of data. This allows the use of third-party add-on softwares that work with JSON format.

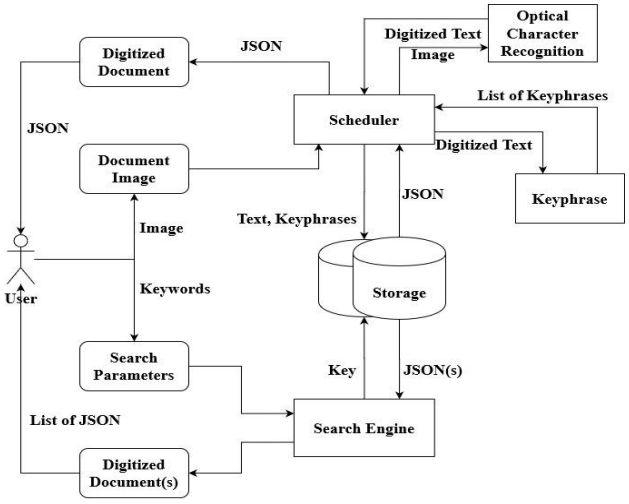

Fig. 1. System Architecture

The digitizing process starts with users capturing the pictures of the document and then supplying it to the optical character recognition (OCR) module. The document image is given as input to the system, and OCR is the first process to be applied to the input before it can be passed on to the keyphrase extraction or even to be searchable. The OCR is implemented through a neural network, but before we give the image to the neural network as the input, we have to perform certain pre-processing for it to be processed accurately and efficiently. The preprocessing involves grey-scale conversion, skewing, and segmentation. Grey-scaling is a way to normalize the image that will be fed into the neural network. Skewing has to be in case the image or the content in the image is not aligned which could lead to better

processing by the neural network. Finally, we have to segment the image by bounding the words in the input document which is then classified by the neural network.

As we discussed earlier, we are using a CRNN for OCR purpose. The architecture shown Fig. 2, involves several convolutional layers followed by several recurrent neural networks. In our application, we use LSTM's to overcome the vanishing gradient problem that is generally faced in recurrent neural networks.

The CRNN model presented in Fig. 2 is a combination of CNN and a deep bi-directional LSTM. It is well-known that the CNN architecture is well suited for image processing. The use of bidirectional LSTM is to process the sequential data part of every word. Processing the sequence from both direction provides better accuracy over single direction. Making it very efficient in predicting the words.

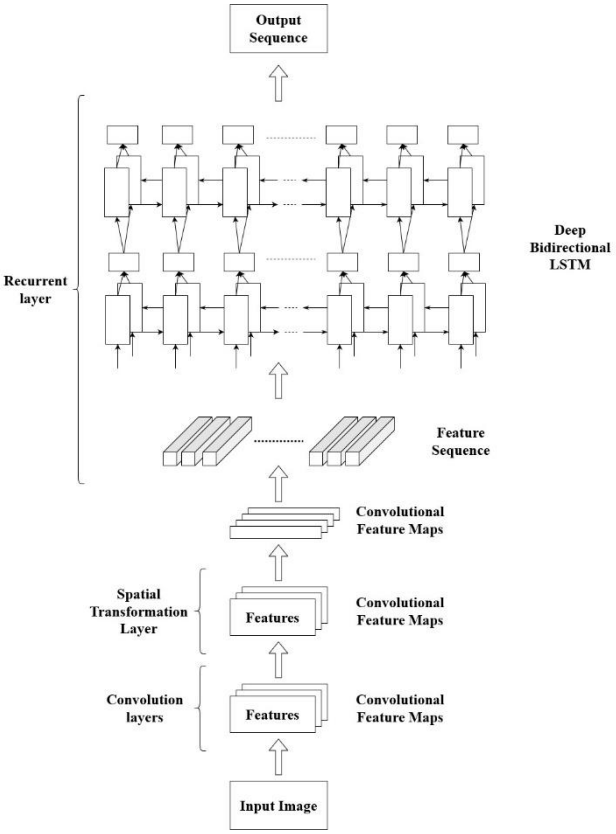

Fig. 2. The architecture of CRNN for OCR

From the architecture, we can see that the input image is first passed through the convolutional layer to extract the features, which is then used to detect the sequence by making use of the recurrent layer. The convolution layer comprises of the convolutional layer, max-pooling layer, and batch normalization. Convolutional layer and max-pooling layers are used to extract the features from the given input, and batch normalization is used to normalize the weights. At the same time, the network is being trained to avoid any deviations in the process. Once convolutional layers are done extracting features, we do not go to the recurrent layers directly. Instead, we apply a spatial transformation on the feature maps that are generated. This results in a model that learns invariance to rotation, scale, translation, and more generic warping. These feature maps are then reshaped to be fed into the bidirectional LSTM recurrent layers. This layer is used to identify the sequence in the given input to make an accurate classification.

The text of the documents has been detected by the OCR, but since OCR procedures cannot guarantee a 100% accuracy, we pass the text through a spellchecker. Misspelled words can drastically affect the accuracy of keyphrase extraction if not corrected. The spellchecker applies a best-effort policy to correct misspelled words. We then apply keyphrase to the corrected text.

In the Fig. 3, we use a hybrid system [1] of Graph-based and statistical method to perform keyphrase extraction. We use PositionRank [2] to generate a list of keyphrase candidates and then use C-value [3] method along with the damping factor to re-rank the keyphrase candidate list.

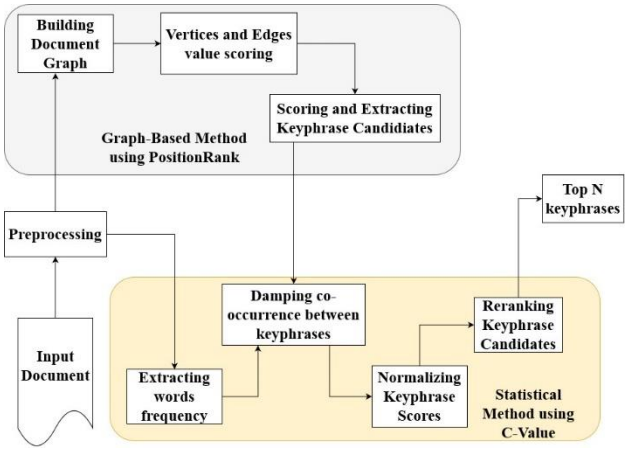

Fig. 3. Keyphrase Subsystem

The digitized document is supplied to PositionRank and C-value method, as shown above. The document text is pre-processed using StanfordNLP [14], we remove stop words from the document and then analyze each word and annotate it with a part of speech (POS) tag. The annotated document is converted to an undirected graph, where vertices represent the words and edge between represents the co-occurrence between them. We now use PositionRank [2] to score the edges of the Graph. Primarily, we use a window size that extracts keyphrases that are made of Nouns, Adjective Nouns sequences matched by the longest match strategy. We pass this scored keyphrases to the Modified C-Value [3]. Damp the keyphrase score if the keyphrase is part of another keyphrase. We then normalize the score of all keyphrase depending on the frequency of word occurrence within the text. The candidates with higher scores are extracted [1].

Once we have the keyphrases from the document. We store the keyphrases in a hash-table list, along with the digitized document text and reference to the photos used for capturing this information within centralized storage and database. So, the next time a user tries to search for documents that match their topic of interest, the search module goes through the keyphrase list of each digitized document and respond with all relevant documents that match in linear order. Since each list is hash-table the time taken to search through the list is constant.

4. Result Analysis

The results of this system were obtained through experimentation using the INSPEC dataset[15]. This dataset is a list of abstracts from various research papers and their corresponding keyphrases and keywords. It is evident that the most significant of all the components that are involved are the accuracy of the keyphrase extracted and response time concerning the search results. It is directly dependent on the accuracy of the keyphrase extraction module, which in turn depends on the accuracy of the OCR module

The experiment starts by generating an image of the abstract text from the dataset, then passing it through our proposed system. Ten keyphrases are generated for every single document. During this testing, we estimate our system performance with respect to the number of characters misspelled, the number of words misspelled by the OCR module, and then estimate the accuracy of the keyphrase module. It is important to stress that the quality of keyphrases generated directly affects the quality of searching. Hence, accuracy in the keyphrases generated for the document is always beneficial for good searching. The details of the system are shown in Table. 1, as we can see the keyphrases generated to match the datasets keyphrases 85%, i.e., for every document at least 8 out of 10 keyphrases generated to match the existing keyphrases of the document dataset.

Table 1. System Accuracy

Type of Data	Total Predicted	Miss Predicted	Accuracy
Characters	295409	17181	94.18
Words	52208	5281	89.88
Keyphrases	5048	739	85.36

The measured performance of the search is based on the time it takes to search for the particular for a specific keyphrase entered by the user. To have a clear view of comparison, we search our directory of digitized documents using a string search function based on Boyer-Moore-Harspool (BMH), then search the keyphrase extracted using just PositionRank algorithm, we then finally compare it a combined method of PositionRank and C – Value method. The

string search algorithm being used is the Boyer-Moore-Harspool implementation found in the string library functions in Python 3.6.

The observations made were plotted, as shown in Fig. 4, the time is taken to search through a single document. A search through the text algorithm takes significantly more time than searching through the keyphrase list generated by respective keyphrase algorithms. It is clear that both searching through the list of keyphrases generated by just PositionRank and a combined hybrid method of PositionRank and C-value method has better performance regular text search. The average performance is shown in the Table. 2. The average speed increase in searching due to using keyphrases is over eight times the speed of native search through text.

Table 2. Search Performance Comparison

Search Methods	Average Time (µs)	Speed Factor
Text Search	2.23	-
PositionRank Keyphrases list	0.558	**4.00**
Hybrid keyphrases list	0.268	**8.32**

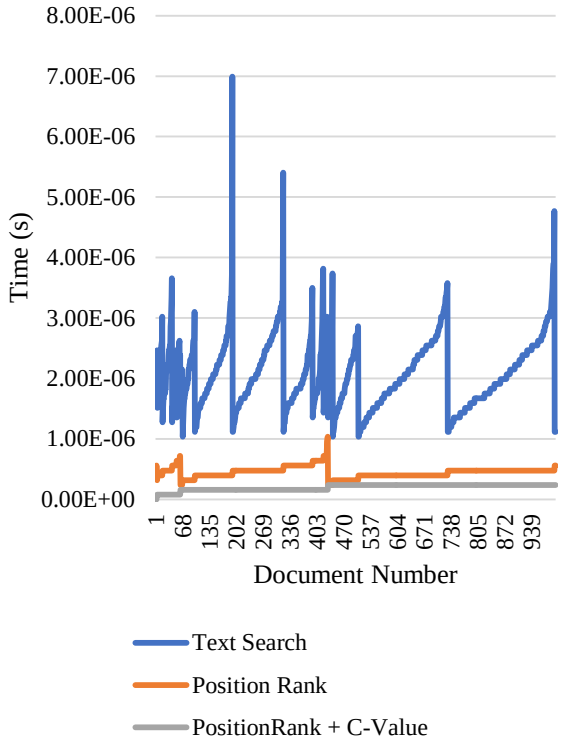

Fig. 4. Time vs Document Performance

While both the position rank and position rank along with the C-value method perform significantly better, the former had an average performance of 0.557 s while the latter had 0.268 s. On average, the proposed method is nearly eight times faster than the string search approach. We can also see that in some instances, there is a sudden fluctuation in the time taken in case of a string search. This happens because of the large size in the document through which the algorithm has to search through. The proposed method circumvents this problem by generating key phrases that drastically reduces the search space; hence it will have a consistent uniform performance irrespective of the size of the document text. The extraction of high-quality keyphrases and keywords using a Hybrid method of position rank and C-value results in constant list size for all documents. Hence, the search space is reduced drastically; also, the time taken to search for any given document becomes constant.

Fig. 5 shows the cumulative time-performance graph, where we observed the time taken to search through a directory of documents when the number of documents increased. Though we have seen that on an average the proposed method is significantly faster. When we consider the cumulative performance graph, the differences in the linear growth rate between the various method can be drastically vary. It is clearly evident that the growth of bland text search is of a much higher order than that of the proposed Hybrid Position Rank + C -value method. When we combine

this search mechanism with the ability to automatically digitize documents and extract keyphrases to populate the search directory, we have a powerful indispensable analog document retrieval system.

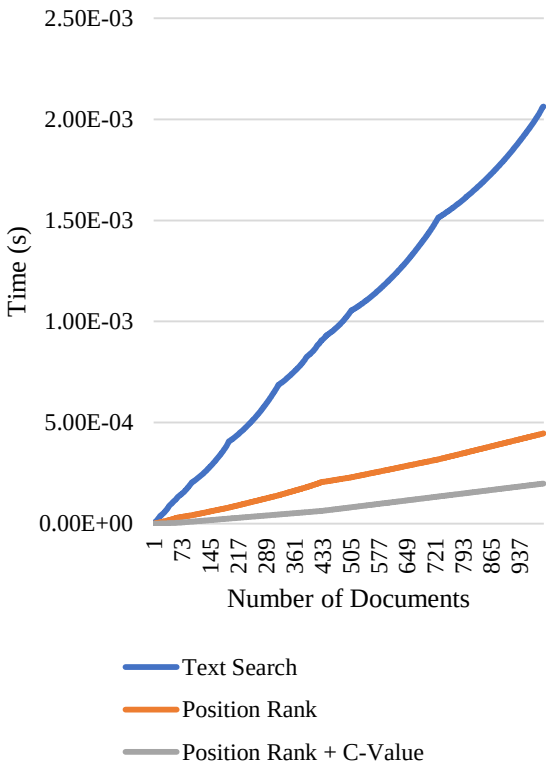

Fig.5. Cumulative performance

5. Conclusion

The proposed system presents a computationally efficient system designed for automatic keyphrase extraction of an analog document. This helps us remove the tedious task of manual searching of documents. Our work, the document image, is first skewed, and noise is removed before allowing it to processed by the OCR sub-system. The OCR sub-system splits the document into words before recognizing them. Once Text has been recognized, it passed through a dictionary layer to rectify any mis predicted words. The document text is then passed onto the Keyphrase sub-system, which extracts the keyphrases of the document. This allows us to fetch the best matching analog document for a given keyphrase or keyword without having to manually search through the vast stretch of analog document pile.

The meaningful experiment confirms the proposed system's ability to predict keyphrases of a given document with high accuracy. Hence, the system implicates its ability to digitize documents and produce a search list of documents with high accuracy. The search performance efficiency is comparatively over eight times faster than the traditional text searching. In the future, searching techniques can be tweaked and enhanced to provide an adaptive search platform, which can understand the user's input and decide the order in which the documents should be ranked. The project can be mapped to other regional languages if the OCR engine is trained using those regional language dialects. But the efficient document searching for those regional language is as limited as the availability of the Natural language Processor for the given language, which augments the Keyphrase Extraction Process.

References

[1] Yeom, H., Ko, Y., & Seo, J. (2019). Unsupervised-learning-based Keyphrase Extraction from a Single Document by the Effective Combination of the Graph-based Model and the Modified C-value Method. Computer Speech & Language

[2] Florescu, C., Caragea, C, 2017. Positionrank: an unsupervised approach to keyphrase extraction from scholarly documents. In: Proceedings of the Fifty-fifth Annual Meeting of the Association for Computational Linguitics, pp. 1105-1115

[3] Frantzi, K., Ananiadou, S., & Mima, H. (2000). Automatic recognition of multi-word terms:. the C-value/NC-value method. International Journal on Digital Libraries, 3(2), 115–130. doi:10.1007/s007999900023

[4] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and Transferring Mid-level Image Representations Using Convolutional Neural Networks," in Proceedings of the 2014 Computer Vision and Pattern Recognition (CVPR),2014, pp. 1717–1724

[5] Li, B., Peng, L., & Ji, J. (2014). Historical Chinese Character Recognition Method Based on Style Transfer Mapping. 2014 11th IAPR International Workshop on Document Analysis Systems. doi:10.1109/das.2014.33
[6] Tang, Y., Peng, L., Xu, Q., Wang, Y., & Furuhata, A. (2016). CNN Based Transfer Learning for Historical Chinese Character Recognition. 2016 12th IAPR Workshop on Document Analysis Systems (DAS). doi:10.1109/das.2016.52
[7] Hoon Chung, Sung Joo Lee, & Jeon Gue Park. (2016). Deep neural network using trainable activation functions. 2016 International Joint Conference on Neural Networks (IJCNN). Doi:10.1109/ijcnn.2016.7727219
[8] Afroge, S., Ahmed, B., & Mahmud, F. (2016). Optical character recognition using back propagation neural network. 2016 2nd International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE)
[9] Sabu, A. M., & Das, A. S. (2018). A Survey on various Optical Character Recognition Techniques. 2018 Conference on Emerging Devices and Smart Systems (ICEDSS). doi:10.1109/icedss.2018.8544323
[10] El-Beltagy, S. R., & Rafea, A. (2009). KP-Miner: A keyphrase extraction system for English and Arabic documents. Information Systems, 34(1), 132–144.
[11] Wang, R., Liu, W., McDonald, C., 2014. Corpus-independent generic keyphrase extraction using word embedding vectors. In: Proceedings of the Software Engineering Research Conference, vol. 39
[12] Wei, T. C., Sheikh, U. U., & Rahman, A. A.-H. A. (2018). Improved optical character recognition with deep neural network. 2018 IEEE 14th International Colloquium on Signal Processing & Its Applications (CSPA).
[13] Bennani-Smires, K., et al., 2018. EmbedRank: Unsupervised keyphrase extraction using sentence embeddings, [online] Available: https://arxiv.org/abs/1801.04470
[14] Manning, Christopher D., Surdeanu, Mihai, Bauer, John, Finkel, Jenny, Bethard, Steven J., and McClosky, David. 2014. The Stanford CoreNLP Natural Language Processing Toolkit In Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 55-60.
[15] Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In Proceedings of the 2003 conference on Empirical methods in natural language processing (EMNLP '03). Association for Computational Linguistics, USA, 216–223. DOI:https://doi.org/10.3115/1119355.1119383

Authors' Profiles

Lokeshwar S is pursuing his bachelors in Computer Science and Engineering from Bangalore Institute of Technology (VTU), Bengaluru**,** Karnataka, India. His area of interest includes Artificial Intelligence, Machine Learning, System Application Designs and Algorithm Designs.

Vadiraja Rao M K is pursuing his bachelors in Computer Science and Engineering from Bangalore Institute of Technology (VTU), Bengaluru**,** Karnataka, India. His area of interest includes Neural Network, Embedded Systems and Operating Systems.

Sujay Kumar P S is pursuing his bachelors in Computer Science and Engineering from Bangalore Institute of Technology (VTU), Bengaluru**,** Karnataka, India. His area of interest includes Machine Learning, Image Processing and Data Analytics.

Vishveshwara Guthal Gowda is pursuing his bachelors in Computer Science and Engineering from Bangalore Institute of Technology (VTU), Bengaluru**,** Karnataka, India. His area of interest includes Machine Learning and Computer Vision.

Dr. Hemavathi P is an assistant professor in the department of Computer Science and Engineering, Bangalore Institute of Technology, Bengaluru, Karnataka, India. She has obtained PhD, M.Tech and B.E from Jain University, Dr. Ambedkar Institute of Technology Bangalore, (VTU), Manipal Institute of Technology, Manipal (University of MAHE), respectively. She is having 16 years of experience as an academician. She has published various journals and conferences of national and international repute. Her area of interest includes Wireless networks, Internet of Things, Artificial Intelligence and Machine Learning.

How to cite this paper: Lokeshwar S, Vadiraja Rao M. K, Sujay Kumar P. S, Vishveshwara Guthal Gowda, Hemavathi P., " Analog Document Search Using CRNN and Keyphrase Extraction", International Journal of Image, Graphics and Signal Processing(IJIGSP), Vol.13, No.2, pp. 16-24, 2021.DOI: 10.5815/ijigsp.2021.02.02