Automatic Document Metadata Extraction based on Deep Networks

Runtao Liu, Liangcai Gao✉, Dong An, Zhuoren Jiang and Zhi Tang

Institute of Computer Science & Technology of Peking University
Beijing, China
{liuruntao, glc, andong, jiangzr, tangzhi}@pku.edu.cn

Abstract. Metadata information extraction from academic papers is of great value to many applications such as scholar search, digital library, and so on. This task has attracted much attention from researchers in the past decades, and many templates-based or statistical machine learning (e.g. SVM, CRF, etc.)-based extraction methods have been proposed, while this task is still a challenge because of the variety and complexity of page layout. To address this challenge, we try introducing the deep learning networks to this task in this paper, since deep learning has shown great power in many areas like computer vision (CV) and natural language processing (NLP). Firstly, we employ the deep learning networks to model the image information and the text information of paper headers respectively, which allow our approach to perform metadata extraction with little information loss. Then we formulate the problem, metadata extraction from a paper header, as two typical tasks of different areas: object detection in the area of CV, and sequence labeling in the area of NLP. Finally, the two deep networks generated from the above two tasks are combined together to give extraction results. The primary experiments show that our approach achieves state-of-the-art performance on several open datasets. At the same time, this approach can process both image data and text data, and does not need to design any classification feature.

Keywords: Information Extraction, Ensemble Modeling, Convolutional Neural Networks, Sequence Labeling, Recurrent Neural Networks

1 Introduction

Automatic metadata extraction from scientific articles is a significant prerequisite for many tasks such as scholar search, information retrieval and digital library. Manual extraction of these metadata is very time-consuming and laborious. Therefore, automatic extraction of scholar document metadata becomes an urgent problem. However, the efficient implementation of metadata extraction is not simple due to different style and scope of metadata provided by authors or publishers.

Recently deep learning has shown great power in computer vision, speech recognition, natural language processing and other fields. Therefore, in this paper

we introduce deep learning into the task, document metadata extraction. In detail, our approach contains two types of networks including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to handle image information and text information of paper headers. Finally, we combine those networks handling image and text respectively into a following Long-Short Term Memory (LSTM) to get the classifications of each line in paper headers.

Comparing to the previous work, the main contribution of our work is: (i) Besides the text information, we also utilize the vision information (e.g., layout, position, font, etc.) of the header of scientific articles. (ii) Our system achieves the best results on several public datasets, compared to other metadata extraction tools. (iii) As far as we know, this paper employs deep learning in the task of document metadata extraction at the first time.

In the following sections, we will introduce our approach and system in details. Section 2 describes the related work about metadata extraction, CNN and RNN. Sections 3-5 write about our definition of this extraction task, corresponding model design and our system. Section 6 presents the experimental results on several datasets. Section 7 draws the conclusions and discusses future work.

2 Related Work

2.1 Document Metadata Extraction

Accurate metadata information extraction is of significant importance for digital library and scholar information retrieval system. Since it's too pricey for manual extraction, building a robust and universal extraction tool can remarkably improve the efficiency and quality of the metadata extraction process.

The most widely used approach to this problem is mainly based on three kinds of methods: rules, retrieval and statical machine learning. Rule-based methods utilize a group of rules to extract the document metadata information. M. Y. Day et al. [4] constructed a hierarchical knowledge representation framework (INFOMAP) to extract citation metadata based on six major citation templates. This kind of methods need a lot of domain knowledges to write these rules, which cause it is not flexible enough in practical scenarios. Another metadata extraction method is based on Information Retrieval related techniques. Cortez E, et al. [2] construct a knowledge-base from a dataset and then they tag the blocks of splited reference strings by searching in the knowledge-base. Compared to the previous two methods, machine learning methods don't need expert knowledges and databases. Han et al.[5] utilize SVM classifier to classify every line to different tags. This task can be also regarded as a sequence labeling task. Peng et al.[12] use Conditional Radom Field (CRF) method on this task and acquire good result.

2.2 Deep Neural Networks

Deep neural networks have shown fantastic performance in many research areas such as CV and NLP. In this section, we briefly review some deep learning approaches which would be utilized in this paper.

Convolutional Neural Network in Image Classification Krizhevsky et al. [10] proposed an architecture known as AlexNet which got the champion in 2012 ILSVRC(ImageNet Large-Scale Visual Recognition Challenge) achieving a top 5 test error rate of 15.4%. It shows that CNNs are expert in image modeling and extraction of features. [8] proposed their CNN model applied on document image classifications and achieved the state-of-the-art performance, which shows that CNN is also applicable to document images. Besides, CNNs have been applied in sentence modeling tasks. Kim Y. [9] applied the CNNs on the matrix of word vectors, which achieved pretty improvements on several public benchmarks. The above studies show that CNN has a good ability to learn the local and global features separately. It could be used in not only CV tasks but also NLP tasks.

Recurrent Neural Network in sequence labeling Recurrent Neural Network could process tasks with arbitrary sequences of inputs. LSTM which is a kind of RNN and its variants has shown great ability on sequence labeling. Chiu et al. Huang Z et al. [6] proposed a variety of LSTM model named BI-LSTM-CRF to address the sequence tagging task which could utilize probability distribution of tag sequence through the top CRF layer. The above studies show that RNN is an ideal solution for sequence tasks.

3 Problem Definition

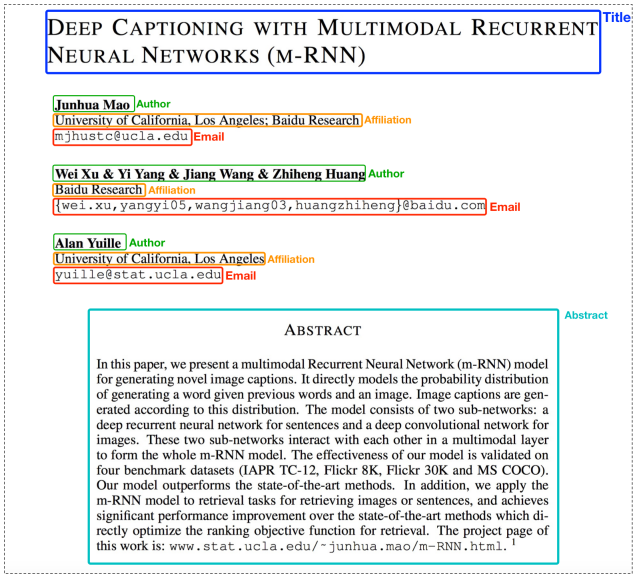

Fig. 1. Paper Header Example

As Figure 1 shows, it's an example of paper header with some metadata highlighted. Seymore et al. [13] defined 15 different tags for the scholar document header in 1999 including Title, Author, Affiliation, Address, Note, Email, Date, Abstract, Introduction, Phone, Keyword, Web, Degree, Pubnum, Page. In addition, they provided a dataset containing 935 headers of computer science research papers. The dataset had become one of the important benchmarks of the header information extraction area from then on. However, as time goes on, more and more evidences show that the 15 different classifications aren't totally suitable for being regarded as the metadata information types nowadays. Classes such as Web, Degree and Pubnum aren't the common content in the header of scholarly document now. Thus, in this paper, we only select the most common eight classes as the metadata tags, including Title, Author, Affiliation, Address, Email, Date, Abstract and Other.

Here we formulate the header metadata information extraction task as follows. Given a sequence of lines of the paper header $\boldsymbol{h} = (h_1, ..., h_n)$ and the m classifications(tags) $\boldsymbol{tag} = (tag_1, ..., tag_m)$, the problem is to match each line h_i of the header $\boldsymbol{h}$ to a specific classification $r_i \in \boldsymbol{tag}$ through a function F:

$$r_i = F(i, \boldsymbol{h})$$

Note that h_i may contain much information of the i-th line. Besides the text content information, the image, layout style, font style, character position or combination and other information may be also included. While the previous work only depends on text information of paper headers to extract their metadata, our approach tries to make full use of the multiple kinds of information from paper headers to make the extraction performance much better. Sometimes a text line contains multiple types of metadata, which are separated by whitespace. Thus, we first segment such lines into fragments according to the whitespace before tagging them. For convenient description, we also call the fragments as lines in this paper.

4 Deep Learning Model Design

This section will describe our deep learning model for extracting metadata from the header of scientific articles. With obtaining the text content and the image information of each line of the paper header, our approach can either solve it as a CV task or as a NLP task and even try to combine these methods or models.

We first explain the sentence model we adopt to map every line of the paper header to its intermediate representations. Then we describe the deep neural network architecture we use for image information and text information respectively. Finally we explain how to ensemble these two networks.

4.1 Sentence Modeling

We first map the text of header lines to corresponding intermediate representation. Inspired by Kim's [9] work, we proposed our model shown in Figure 2 to learn the representation of the text contents.

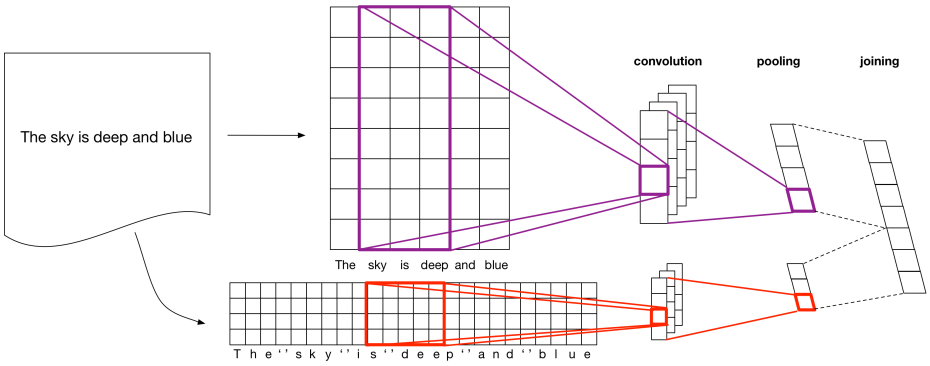

Fig. 2. Sentence Model

Formally, given a line of paper header $(w_1, w_2, ..., w_t)$, where w_i stands for the k-dimensional word vector or the character vector in this line, and by concatenating these vectors it can be represented as a $t \times k$ matrix. A convolution operation is applied to this matrix with a filter $\boldsymbol{s} \in \mathbb{R}^{l \times k}$ whose stride is l then a feature map could be obtained $\boldsymbol{c} = (c_1, ..., c_{t-l+1})$ using following formula, where $\boldsymbol{b}$ is the bias term and the α is a non-linear activation function:

$$c_i = \alpha(\boldsymbol{s} \cdot \boldsymbol{w}_{[\boldsymbol{i:i+l-1}]} + \boldsymbol{b})$$

Sigmoid, hyberbolic tangent (Tanh) and rectified linear unit(ReLU) are common non-linear activation functions. We choose ReLU as it could let the networks converge faster than standard sigmoid units[3] .

We use m' different filters with multiple strides to obtain multiple features $(\boldsymbol{c}_1, ..., \boldsymbol{c}_{m'})$. These operations are done in convolutional layers and then these results are put into pooling layers. Pooling operations is a non-linear down-sampling operation that could decrease the number of features. There we adopt the common max pooling.

Following their study, we adopt the max pooling strategy and could get the representation of the text information of i_{th} line in paper header.

$$Ptext_i = (max(\boldsymbol{c}_1), ..., max(\boldsymbol{c}_m))$$

This section introduces how to obtain the presentation of the text content of scholarly document header lines. The following section will depict the deep networks to get the representation vector of the images of header lines.

4.2 Image Modeling

Here we need to generate fixed-length vectors $\boldsymbol{I} \in \mathbb{R}^{n \times d}$ from the images that effectively represent the corresponding image information. Formally, for an image $I_i \in \mathbb{R}^{height,width}$ which stands the information of i_{th} line, our image model

$G(I_i, \theta)$ will output a representation $I_i \subset h_i$ where θ is the set of parameters of this model. Here we adopt the CNN as the model that it could learn the parameters θ itself through minimizing the following loss function in the training process where r'_i stands for the tag of i_{th} line and $\boldsymbol{h}$ stands for paper header.

$$Loss(F(i, \boldsymbol{h}), r'_i)$$

We use a minor variant of VggNet [14] as our model. As the number of classification defined in this study is far less than the number of image classification defined in original model and the document images are much simpler than natural images, we modified the VggNet to a more shallow and simpler one to make it more adaptable for this problem.

4.3 Sequence Modeling

Cho et al. [1] proposed a framework called RNN Encoder-Decoder for language translation. As shown in the right part of Figure 3, the framework consists of two components, the first one encodes a sentence to an intermediate representation and the second decodes it to a target sentence. In our model, the Bi-RNN encoder reads an element x_t in a sequence $(x_1, ..., x_n)$ at time t, and generates two hidden states:

$$h_t = f(x_t, h_{t-1}) \quad h'_t = f'(x_t, h'_{t+1})$$

where the h_t, h'_t stand for the two hidden states in two opposite directions and f, f' represent the RNN unit function. Then the decoder will generate the probability distribution on the possible tags following $p(y_i|h_t, h'_t)$. And the $\overline{y} = max\{p(y_i|h_t, h'_t)\}$ can be regarded as the most feasible tag for the element x_t. Though this approach has efficiently considered the past and the future information, it doesn't consider the transaction regulation between y_i. [11] indicates that for sequence labeling tasks considering the correlations between neighboring labels could get better result. Formally, $\beta(\boldsymbol{y})$ denotes the whole tags space for the $\boldsymbol{x}$ and the decoding task is to find the $\overline{\boldsymbol{y}}$ that meets:

$$\overline{\boldsymbol{y}} = argmax\ p(\boldsymbol{y}|\boldsymbol{x})$$

we define score and probability function of a candidate sequence $\boldsymbol{y}$ as:

$$S(\boldsymbol{x}, \boldsymbol{y}) = \prod_{i=1}^{n} \psi(y_i, \boldsymbol{x})\psi(y_i, y_{i-1})$$

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{S(\boldsymbol{x}, \boldsymbol{y})}{Z}$$

and the partition function is determined as:

$$Z = \sum_{\boldsymbol{y} \in \beta(\boldsymbol{y})} S(\boldsymbol{x}, \boldsymbol{y})$$

We regard $\psi(y_i, \boldsymbol{x})$ to be the matrix $M_1 \in \mathbb{R}^{n \times m}$,the scores between the i_{th} element and the j_{th} tag output by the bidirectional RNN network and the $\psi(y_i, y_{i-1})$ to be the transaction matrix $M_2 \in \mathbb{R}^{m \times m}$ denoting the score between i_{th} tag and j_{th} tag. As above, our sequence model considers not only the past and future information but also the joint distribution of target label sequence.

5 Information Extraction System Based on Deep Learning

The framework of our metadata extraction system is shown in Figure 3. The input of this system are PDF files. The image and text of a paper head is obtained with the help of PDFBox [1]. Based on the image model and the sentence model, the output representation vectors will be concatenated together in one representation vector. Finally, our system processes each line in the header and output the most possible tag sequence.

5.1 System Overview

As shown in Figure 3, our system has following components: image model, sentence model, joint layer and Bi-LSTM-CRF layer.

First, a document whose header contains N lines is put into our system. Then the sentence model get sentence matrices $s_i \in \mathbb{R}^{n \times k}$ for each line l_i by concatenation of the vectors output from the embedding layer. The sentence model will output a fixed length feature vector $Ptext_i \in \mathbb{R}^{len_1}$ The images of this line will be put into the image model, and the model will output a fixed length representation vector $p \in \mathbb{R}^{len_2}$. Then v_i, p_i will be concatenated together in a single representation as $rp_i = [Ptext_i, p_i]$. This vector is then put into the sequence model, Bi-LSTM-CRF network, from whose outputs we could get the final results.

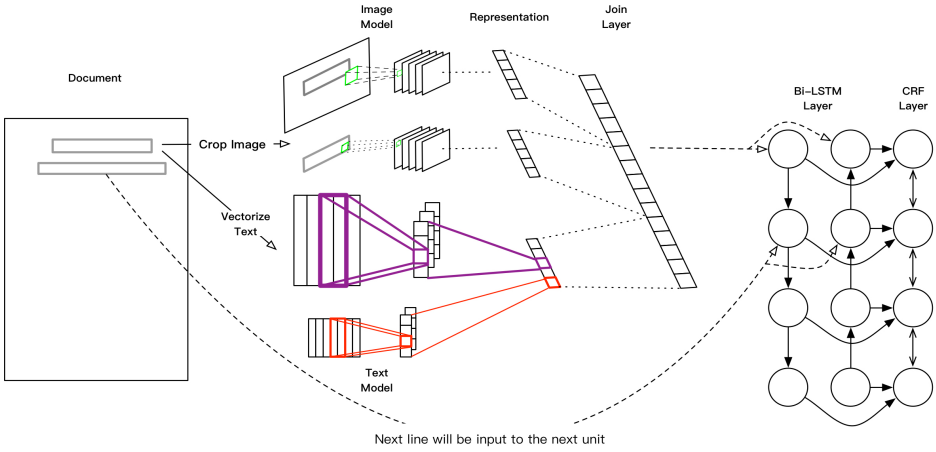

Fig. 3. The Overview of Our System

[1] http://pdfbox.apache.org

5.2 Word & Char Embedding

We use the public pre-trained word vectors named Global Vectors for Word Representation(GloVe) and there are about 400,000 words which has 200 dimensions in it. We use GloVe to initiate the word embedding layer, while the character embedding will be initialized randomly. In the training process, both the two embedding layers will be fine-tuned to promote the entire network performance.

5.3 Training and Parameters

In our system, the weights to be trained include the following parts:(i)In the sequence model, the scores matrix $M_1 = \psi(y_i, \boldsymbol{x})$ and the transition matrix $M_2 = \psi(y_i, y_{i-1})$. (ii)In the image model, the parameters $\boldsymbol{\theta}$ of the image deep ConvNet.(iii)In the sentence model, the word and character embedding matrices and the parameters $\boldsymbol{\gamma}$ of the sentence deep ConvNet. The purpose of training process is to maximize the log probability of the correct labeling sequence:

$$log(p(\boldsymbol{y}^*|\boldsymbol{x}, f)) = log(Score(\boldsymbol{y}^*, \boldsymbol{x})) - log(Z)$$

Standard back-propagation is used to learn these above parameters of this end-to-end system. In our experiments, after initiating training using Adam for some iterations, we use Stochastic gradient descent(SGD) to continue fine-tuning the model. We adopt early stopping by checking the recorded loss values on validation set for every iteration and select the lowest one as the result model.

On system configure, we retain the first 10 words and 100 characters and truncate the leftover parts for each line. We resize the height of the image of each line to 40 and the width to 200. The dimensions of the word and character embeddings is 200 and 40 respectively. The networks could learn the representations from the input. The lengths of the images representation vector, the characters representation vector and the words representation vector are 2048, 1024 and 1024 respectively. These representations will be joined and put into the two stacked Bi-LSTM networks whose hidden state dimensions are set to 512.

6 Experiment and Evaluation

We evaluate our metadata extraction system on three tasks: (i) metadata extraction only based on image information (ii) metadata extraction only based on text information (iii) metadata extraction based on both image and text information. F-measure is used for performance evaluation. Finally, we compare our system to the existing and available extraction tools and methods like, Parscit [7], SVM [5] and CRF [12]. The results show that our method has achieved much improvement in the extraction performance.

6.1 Datasets

In this section we evaluate our deep networks model on these following datasets. The proportion of each classification is shown in Table 1.

SEYMORE This dataset provided by Seymore et al. contains 935 headers with 15 classifications including Title, Author, Affiliation, Address, Note, Email, Date, Abstract, Introduction, Phone, Keyword, Web, Degree, Pubnum and Page. For each line, there is a corresponding tag at its end.

OURS The dataset contains 75,000 headers and each header contains both image and text data. It's consist of 70,000 noisy items and 5,000 correct items. It will be open and available on the web (http://www.icst.pku.edu.cn/cpdp/header /header_open.rar) . The corresponding PDF format papers are crawled from the Internet. Our dataset contains only the following categories: Title, Author, Affiliation, Address, Email, Abstract, Phone, Keyword, and Other.

Table 1. The ratio of each class in SEYMORE and OURS

Class	ratio(SEYMORE)	ratio(OURS)
Title	8.3%	9.8%
Author	7.2%	10.5%
Affiliation	10.6%	16.8%
Email	3.4%	8.8%
Abstract	50.0%	36.5%
Phone	0.6%	1.4%
Address	6.3%	9.5%
Other	13.6%	6.8%

6.2 Fine-tuning

Fine-tuning is using another dataset to train the trained networks to let back-propagation process continue to decrease the loss at this dataset. Training deep networks needs tremendous data while it's unrealistic to label hundreds of thousands of data, so we will use tools based on CRF to tag 75,000 paper headers. Even though these data contain noisy information and incorrect labels, we could first use them to train the deep networks for pre-training. Afterwards, we manually label 5,000 instances of header. Then we use 3,000 of them for fine-tuning after training with noisy data and the left 2,000 for testing.

6.3 Experiments on Image

Since the previous public dataset contained only textual information and did not have corresponding image information, the image experiments were performed on OURS dataset as shown in Table 3.

Table 3 shows the performance of the image model for each classification. The image model exhibits a good result on "Title" class because the image instances in "Title" class have significant location information and font features that they are generally located at uppermost with bold font. "Email" and "Phone" classes is also fine because "Email" instances are generally letters with "@"(at) and "Phone" instances are generally numbers with "-"(dash) that they all could be learned by image model. Owing to lack of semantic information, "Author", "Affiliation" and "Address" are not so satisfactory even though they have their respective features; semantic information is more crucial for these classes.

6.4 Experiments on Text

We adopt both word embedding and character embedding in order to catch the semantic features and the characters arrangement features simultaneously. "Title", "Author", "Affiliation" and "Address" have richer semantic features while "Email", "Phone", "Pubnum" and "Web" have more obvious character arrangements features. We test our text model on two datasets, SEYMORE and OURS. The experimental results on SEYMORE and OURS are presented in Table 2 and Table 3 respectively.

Table 2. Result on SEYMORE Dataset of Text Model

Class	Parscit(CRF)[7]	Han[5]	Peng[12]	Text Model
Title	0.968	0.945	0.971	**0.981**
Author	0.952	0.942	0.975	**0.977**
Affiliation	0.914	0.933	**0.970**	0.963
Address	0.899	0.900	0.958	**0.960**
Email	0.952	0.964	0.953	**0.969**
Abstract	0.988	0.988	**0.997**	0.995
Phone	0.913	0.762	0.979	**0.982**
Average F1	0.941	0.919	0.972	**0.975**

Experimental results show that the performance of our metadata extraction system is much better than the other systems. Existing approaches utilize semantics in the header by means of dictionaries such as name dictionary and country dictionary etc., which results in that they can only obtain very limited semantic content as the dictionaries generally didn't contain many words. Since the word embeddings could utilize the semantic information efficiently, our system shows better performance on "Title", "Author", and "Address". On the other side, char embeddings could find the pattern in the spelling of "Email" and "Phone", thus these classes perform better as well.

6.5 Experiments on Information Union

This section will attempt to evaluate the performance of converging image network and text network. As other datasets didn't contain image information of paper headers, we show the experiment results on our dataset. Our system will utilize both images and texts in our dataset while Parscit only uses the texts.

Table 3. Result on OURS dataset of other tools and our system

Class	SVM	Parscit(CRF)[7]	Image	Text	Image & Text
Title	0.891	0.944	0.950	0.969	**0.985**
Author	0.703	0.938	0.905	0.96	**0.966**
Affiliation	0.877	0.934	0.903	0.953	**0.964**
Email	0.971	0.976	0.979	0.965	**0.985**
Abstract	0.970	0.981	0.963	0.983	**0.997**
Phone	0.936	0.961	0.959	0.968	**0.974**
Address	0.745	0.923	0.895	0.947	**0.964**
Average F1	0.870	0.951	0.936	0.963	**0.976**

Table 3 shows the performance of our system which combines the image model and text model. The results show that the overall system performance has been greatly improved after combining these two models. The images carry features such as location, font and layout information, while the texts carry semantic and character arrangements information and the model integration reduces the information loss in the metadata extraction process. As [12] doesn't release corresponding implementations, here we show the results of Parscit, which is based on CRF like [12]. The competitor tool SVM is implemented employing the features referred in [5] as [5] doesn't release corresponding implementations either. The results show that our system does better in all classes than Parscit and the tool based on SVM and has achieved the state-of-art performance.

7 Conclusions

In this paper, we introduce deep learning technology to the problem, metadata extraction, at the first time. This deep networks based extraction approach can automatically learn the feature representation of metadata during the training process, which significantly reduces manual work for feature engineering compared to previous works. Furthermore, our extraction approach utilize two sources of information in paper headers: image content and text content. As a result, the deep learning model utilizing both image and text information shows the better performance on several datasets.

Compared to natural images, document images are much simpler. Thus, it is a valuable question to explore what kind of network structure is suitable for information retrieval from document images in the future. Furthermore, we plan

to explore universal architectures based on deep networks that could extract the metadata or structure information from the whole document.

8 Acknowledgement

This work is supported by the Beijing Nova Program (Z151100000315042) and the China Postdoctoral Science Foundation (No. 2016M590019), which is also a research achievement of Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology). We also thank the anonymous reviewers for their valuable comments.

References

1. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
2. Cortez, E., da Silva, A.S., Gonçalves, M.A., Mesquita, F., de Moura, E.S.: A flexible approach for extracting metadata from bibliographic citations. Journal of the American Society for Information Science and Technology 60(6), 1144–1158 (2009)
3. Dahl, G.E., Sainath, T.N., Hinton, G.E.: Improving deep neural networks for lvcsr using rectified linear units and dropout. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 8609–8613. IEEE (2013)
4. Day, M.Y., Tsai, R.T.H., Sung, C.L., Hsieh, C.C., Lee, C.W., Wu, S.H., Wu, K.P., Ong, C.S., Hsu, W.L.: Reference metadata extraction using a hierarchical knowledge representation framework. Decision Support Systems 43(1), 152–167 (2007)
5. Han, H., Giles, C.L., Manavoglu, E., Zha, H., Zhang, Z., Fox, E.A.: Automatic document metadata extraction using support vector machines. In: Digital Libraries, 2003. Proceedings. 2003 Joint Conference on. pp. 37–48. IEEE (2003)
6. Huang, Z., Xu, W., Yu, K.: Bidirectional lstm-crf models for sequence tagging. arXiv preprint arXiv:1508.01991 (2015)
7. Isaac G. Councill, C. Lee Giles, M.Y.K.: Parscit tool. http://www.comp.nus.edu.sg/entrepreneurship/innovation/osr/parscit/ (2008)
8. Kang, L., Kumar, J., Ye, P., Li, Y., Doermann, D.: Convolutional neural networks for document image classification. In: Pattern Recognition (ICPR), 2014 22nd International Conference on. pp. 3168–3172. IEEE (2014)
9. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
11. Ma, X., Hovy, E.: End-to-end sequence labeling via bi-directional lstm-cnns-crf. arXiv preprint arXiv:1603.01354 (2016)
12. Peng, F., McCallum, A.: Information extraction from research papers using conditional random fields. Information processing & management 42(4), 963–979 (2006)
13. Seymore, K., McCallum, A., Rosenfeld, R.: Learning hidden markov model structure for information extraction. In: AAAI-99 Workshop on Machine Learning for Information Extraction. pp. 37–42 (1999)
14. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR abs/1409.1556 (2014)