Investigating Transfer Learning for Title Detection in Table of Contents Generation

Dhruv Premi, Amogh Badugu and Himanshu Sharad Bhatt
{Dhruv.Premi, Amogh.Badugu, Himanshu.S.Bhatt}@Aexp.com

Abstract

We present a transfer learning approach for Title Detection in FinToC 2020 challenge. Our proposed approach relies on the premise that the geometric layout and character features of the titles and non-titles can be learnt separately from a large corpus, and their learning can then be transferred to a domain-specific dataset. On a domain-specific dataset, we train a Deep Neural Net on the text of the document along with a pre-trained model for geometric and character features. We achieved an F-Score of 83.25 on the test set and secured top rank in the title detection task in FinToC 2020 (Bentabet et al., 2020)

1 Introduction

Title detection and table of contents generation are important sub-parts of the bigger problem, known as, document structure analysis. Understating the inherent document layout and structure benefits several downstream document AI tasks such as search, summarizing, entity extraction and table detection etc. from a document. Humans glance at a document and comprehend the document structure including the titles vs non-titles as well as the overall hierarchy of the titles. Many reasons can be attributed to it, like the sequential nature of the document, geometrical features or the semantic meaning of the sentences. We have tried to incorporate these basic human instincts into our model.

Humans have intuitive notions of how a document is structured and the assumption is confirmed after reading a text block. Transfer learning can be used to model the structural properties of a general document. We use Arxiv documents[1] available in the open-domain to learn the structural model of a general document. Semantic properties are learned using LSTM (Hochreiter and Schmidhuber, 1997) at title level for a domain specific document. The final model is trained on a domain-specific dataset with structural weights pre-trained from Arxiv documents. We see considerable improvements by applying transfer learning to the title detection task. Combining Deep neural networks based on manual features and Character CNN(Zhang et al., 2015) on the starting eight characters helps us model the structural signature of a general document.

2 Related Work

The literature on title detection can be classified broadly into three categories: works that deal with ToC page of documents, works that use images of document pages and works which use the geometrical and textual features of the text blocks.

In the approaches dealing with ToC pages of documents, after the ToC pages are detected, the title entries are extracted and mapped to the pages by finding links between title and corresponding pages. El Haj et al.(2014) used this approach in detecting titles in UK Financial Reports. As they rely on ToC pages, they cannot be applied to documents that do not have ToC pages.

Other approaches use computer vision to fragment the page image into entities such as text, title and table. Yang et al.(2017) used Multimodal Fully Convolutional Neural Networks for this task. Li et

[1] https://arxiv.org/help/bulk_data

Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation, pages 153–157
Barcelona, Spain (Online), December 12, 2020.

al.(2020b) used Convolutional neural networks combined with graphical models to identify the entities in a document page. Few datasets for these tasks are publicly available.(Zhong et al., 2019; Li et al., 2020a)

Finally, some approaches use learning or rule-based methods to detect headers based on textual and geometrical features(Bentabet et al., 2019; Gopinath et al., 2018; Liu et al., 2011; Budhiraja and Mago, 2018; Klampfl and Kern, 2013). These methods are usually used in digitally generated documents like webpages and native PDF documents.

3 Methodology

We pre-process the PDF files by converting them to XML documents by Poppler[2] library. These files are then parsed to merge elements similar in styling and located in close proximity. Headers and Footers are identified and removed by page association (Lin, 2003) as they would hinder the process of title detection.

Our proposed title detection method has three components; Pre-trained neural network to model general structural information, Sequential Network to learn domain-specific text and training of both the networks combined.

3.1 Pre-trained Neural Network to Model Structural Information on Arxiv Documents

The network composes of two key components geometrical and a character network.

3.1.1 Geometrical Network

The network comprises 22 manual features as depicted in Table 1. Model is trained by multi-layer neural network as described by the architecture in Table 2.

Alignment	**Distance**
-Center Alignment with parent text block -Left Alignment with parent text block -Right Alignment with parent text block -Center Alignment with child text block -Left Alignment with child text block -Right Alignment with child text block	-Normalized vertical distance to the Child text Block -Normalized vertical distance to the Parent text Block
Font	**Extra**
-Font Difference between current and child -Font Difference between current and parent -Font Size -All first word in caps -Is Bold -Is Italic -Number of Fonts -Font Change -Begins with numbering	-Number of New Lines -Number of Poppler blocks -Number of words -Majority of the characters in the start are in Bold -Has Verb

Table 1: Features for Geometric Model

3.1.2 Character Network

Input to the Character CNN is the first eight characters. The aim is to extract patterns that denote start of a title like 1.1, a.1, (a). Architecture is mentioned in Table 3. The trained module did not achieve as high F Score as expected. However, our focus was to capture patterns for a general document. Network benefits can be utilized in steps further.

[2] https://poppler.freedesktop.org/

Hyper-parameters	Value (After Validation)
Input Layer	22
First Hidden Layer	15
Second Hidden Layer	4
Epochs	10
Batch Size	100
Dropout (Between 1st and 2nd Layer)	0.2
Activation	ReLU
Loss Function	binary cross entropy
optimizer	Adam*

Table 2: Hyper Parameters for Geometrical Model

Hyper-parameters	Value (After Validation)
Vocabulary length	71
sequence of characters length	8
convolutions (number of kernels, kernel size, pool size)	[256, 3, 2] , [256, 2, 2]
Dense 1	50
Dense 2	10
Loss Function	binary cross entropy
Dropout (Between 1st and 2nd Dense)	0.5
Activation	ReLU
optimizer	Adam

Table 3: Hyper Parameters for Character Model

3.1.3 Dataset and Training

We take around 6000 Arxiv documents from the annotated documents provided by Muhammad Mahbubur Rahman and Tim Finin (2017). The data split is shown in Table 4. The training was done for three models, namely, geometric, character and character plus geometric. Character plus geometric model performed the best as expected. Intuition being features from geometric and character will complement each other when trained together. We got a significant rise of 5% as compared to the geometric model. Training metrics are depicted in Table 6.

	Train		Validate		Test	
	Title	**Non-Title**	**Title**	**Non-Title**	**Title**	**Non-Title**
Arxiv	59656	536910	3314	29828	3314	29828
FinToC	6666	64719	952	12341	694	6609

Table 4: Train,Test and Validate sizes for Arxiv and FinToC datasets

3.2 Sequence network to learn domain-dependent semantics

We use LSTM as a sequence classification model. Intuition being common phrases that are part of financial titles can be learnt by a sequence network such as LSTM. We use Glove word vector embedding. Last word cell state is passed as input to two dense layers. Final layer after the dense layer performs title detection Table 5. Out Best F Score on the test dataset was at 73.

3.2.1 Dataset and Training

The architecture is mentioned in Table 5. Dataset Split can be seen in Table 4. Our best performing model on validation set gave 73% F-Score on test set. Due to time constraints, we could not explore bidirectional and attention mechanisms (Abi Akl et al., 2019) .

Hyper-parameters	Value (After Validation)
Embedding Size	300
Number of words (From Start)	12
cell state size	13
Dense 1	10
Dense 2	5
optimizer	Adam
Epochs	30
Batch Size	500

Table 5: Hyper Parameters of Sequential Model

3.3 Joint Trained combination of both of the above networks

Full network comprises of pre-trained weights from Character plus Geometric Model trained on Arxiv PDF and Sequence Model trained on FinToC dataset. Last dense layer from Sequence Model and Character Plus Geometric Model are concatenated. One more and last dense Layer of 10 units is added after that. Loss function is binary cross entropy. Total trainable parameters are $2,26,335$. No layers is freezed for subsequent training.

3.3.1 Dataset and Training

FinToC Dataset as mentioned in Table 4 was used. Adam optimizer, epochs equal to 30 and batch size of 500 were used as hyperparameters in the model. The code was written in Tensorflow v1.15 (Abadi et al., 2015) Jointly trained final architecture got the F-Score of 83.25 on the test set.

	Fscore (Test)	
Models	**Arxiv**	**FinToC**
Geometric	87.38	-
CharCNN	58.56	-
Geometric+CharCNN	91.5	-
XGBoost	-	73.01
LSTM	-	73.02
Joint Trained LSTM and Geometric+CharCNN	-	83.25

Table 6: F-Scores of models on Arxiv and FinToC datasets

4 Results & Conclusion

4.1 Investigations

Final results are shown in Table 6. Two highlights of the final model are

- Pre-trained weights captured the generic structure of documents, giving a boost to accuracy. This transfer learning approach can be improved further by using better architectures and features which are domain-independent.This procedure achieved a 10% increase in F Score.
- Combination of geometric and character-based features complemented each other to attain higher accuracy compared to either of them separately.

4.2 Submitted system

We submitted two systems for the final evaluation.

- First one is the Joint Trained LSTM and Geometric+CharCNN network.
- Second one was the ensemble of the first one and an XGBoost(Chen and Guestrin, 2016) model as shown in Table 6.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Hanna Abi Akl, Anubhav Gupta, and Dominique Mariko. 2019. Fintoc-2019 shared task: Finding title in text blocks. In *Proceedings of the Second Financial Narrative Processing Workshop (FNP 2019)*, pages 58–62.

Najah-Imane Bentabet, Rémi Juge, and Sira Ferradans. 2019. Table-of-contents generation on contemporary documents. *arXiv preprint arXiv:1911.08836*.

Najah-Imane Bentabet, Rémi Juge, Ismail El Maarouf, Virginie Mouilleron, Dialekti Valsamou-Stanislawski, and Mahmoud El-Haj. 2020. The Financial Document Structure Extraction Shared task (FinToc 2020). In *The 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation (FNP-FNS 2020*, Barcelona, Spain.

Sahib Singh Budhiraja and Vijay Mago. 2018. A supervised learning approach for heading detection. *Expert Systems*, page e12520.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Mahmoud El-Haj, Paul Rayson, Steven Young, and Martin Walker. 2014. Detecting document structure in a very large corpus of uk financial reports.

Abhijith Athreya Mysore Gopinath, Shomir Wilson, and Norman Sadeh. 2018. Supervised and unsupervised methods for robust separation of section titles and prose text in web documents. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 850–855.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Stefan Klampfl and Roman Kern. 2013. An unsupervised machine learning approach to body text and table of contents extraction from digital scientific articles. In *International Conference on Theory and Practice of Digital Libraries*, pages 144–155. Springer.

Minghao Li, Yiheng Xu, Lei Cui, Shaohan Huang, Furu Wei, Zhoujun Li, and Ming Zhou. 2020a. Docbank: A benchmark dataset for document layout analysis. *arXiv preprint arXiv:2006.01038*.

Xiao-Hui Li, Fei Yin, and Cheng-Lin Liu. 2020b. Page segmentation using convolutional neural network and graphical model. In *International Workshop on Document Analysis Systems*, pages 231–245. Springer.

Xiaofan Lin. 2003. Header and footer extraction by page association. In *Document Recognition and Retrieval X*, volume 5010, pages 164–171. International Society for Optics and Photonics.

Caihua Liu, Jiajun Chen, Xiaofeng Zhang, Jie Liu, and Yalou Huang. 2011. Toc structure extraction from ocr-ed books. In *International Workshop of the Initiative for the Evaluation of XML Retrieval*, pages 98–108. Springer.

Muhammad Mahbubur Rahman and Tim Finin. 2017. Understanding the logical and semantic structure of large documents. *CoRR*, abs/1709.00770.

Xiao Yang, Ersin Yumer, Paul Asente, Mike Kraley, Daniel Kifer, and C Lee Giles. 2017. Learning to extract semantic structure from documents using multimodal fully convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5315–5324.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. 2019. Publaynet: largest dataset ever for document layout analysis. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1015–1022. IEEE.