Reference Extraction using Conditional Random Fields

Klim Zaporojets
Student Id: 28057963
Department of Computer Science
University of Massachusetts
Amherst, MA 01002
klim@cs.umass.edu

Abstract

In this report I will describe how Conditional Random Fields (CRF) probabilistic graphical model can be used to identify logical structure of the document. As a proof of concept, I will train a CRF classifier to identify references in "References" section of any scholarly document.

1 Introduction

In this work I will describe how Conditional Random Fields (CRFs) can be used to identify automatically *logical* and *physical* structures in scholarly documents. By *physical* structures I refer to components such as pages, columns, paragraphs, words, tables, figures, etc.; by *logical* structures, on the other hand, I refer to components such as authors, sections, titles, affiliations, references, abstracts, etc. Furthermore, a CRF classifier will be trained to identify each of the references inside a particular publication. I consider that solving this problem can be very useful as a previous step to multiple dependent tasks. For example, in academic search engines such as google scholar, the information inside each of the references can be used to solve issues such as those related to mining similar publications as well as counting the number of references to a particular paper. More generally, authors in [16] and [15] have already identified other important applications such as key phrases and concept extraction as well as prediction of plagiarism by identifying citations. Finally, I run experiments to determine the performance of different optimization algorithms to train the CRF.

2 Related Work

There is numerous amount of work done related to identification of structures and layout extraction in scholarly documents. In the following points I will describe some areas of research when solving this problem:

1. **Rule Based Approach:** some researches have applied rule-based approach to identify physical and logical structures in the documents. Authors in [11] (physical) and [13] (physical and logical) do a good job describing all kinds of rule-based algorithms that can be used to identify the extract different aspects of layout in digital publications. Most of this algorithms work either in bottom-up or top-down way. The *bottom-up* algorithms start from document image pixels, and cluster the pixels into connected components such as characters which are then clustered into words, lines and so on. The *top-down* algorithms, on the other hand, start from whole document and iteratively split it into smaller ranges. The splitting procedure stops when some criterion is met and the ranges obtained at that stage constitute the final segmentation results. It is also interesting how some authors such as in [22] use language grammar (in form of context free grammars) in order to detect logical

structures such as sentences.
From my perspective, this approach can be very powerful specially when identifying physical components such as lines and columns. However, when it comes to the identification of logical components, it can be very cumbersome to design rules that adapt to all kind of layouts easily. Some of the authors (such as in [17]) tackled this problem by implementing a rule-based system that uses pre-defined templates to identify more precisely the structure of a particular document.

2. **Supervised Machine Learning Approach:** this approach is currently the most used to detect logical structure of the document. Two machine learning techniques that proved very successful are Hidden Markov Models (proposed in [20], [21] and [2] among others), and Conditional Random Fields (proposed in [23], [24] and [6] among others). Both of this approaches are not used as much on pixel level, but rather starting with already pre-processed document usually using information from parsers such as pstotext that allows access attributes of text such as those related to geometric position of a particular character, its size, font, colors, and so on. The difference between these two graphical models is described in detail in [26] and [25], and basically boils down to the fact that despite both represent sequential structures, HMM model joint probability with linear-sequence structure whereas CRFs model conditional probability and can be arbitrarily structured (though linear-chain CRFs are the most common and are used in this work as well).
   Finally, a number of researches tried using other supervised machine learning methods to perform document layout extraction. Belaid et al. in [12], for instance, describe a Perceptive Structured NN (PSNN) that can identify logical structure in a document by using local and contextual features of a particular element (such as word) in a document. Han et al. in [4], on the other hand, propose using Support Vector Machines to extract logical structure. It is done by assigning a big number of features to a particular line that are not only determined by that line, but also dependent on the N previous and N posterior lines. The result is 1100 dimensional feature space associated to each line that is used to train and predict logical structure of the documents using SVM.
3. **Unsupervised Machine Learning Approach:** finally, there is a small amount of work to extract some of the physical and logical structures from the documents using unsupervised approach. More specifically, Klampfl et al. in [3] proposes to use a series of clustering techniques to identify different components in a document. This is done by gathering a set of attributes that can identify a particular element. For example, one of these attributes is related to text sparsity that allows to discriminate elements such as tables from the rest of the text.

3 Proposed Solution

The solution developed in this work consists in using Conditional Random Fields (CRFs) in order to detect and classify logical components inside any scholarly document. More particularly, I train a classifier to segment references inside any paper. The basic element my classifier takes as the input are lines in "references" section and all sections below that. Then, it classifies each of the lines in whether: 1) it is part of prologue (e.g. "References" title) using Begin-Inside-Outside-Last (BIOL) labeling convention, 2) it is part of a reference also using BIOL convention, 3) it is part of epilogue (e.g. some papers have more sections after "references" such as the one related to the authors) also using BIOL convention. The concrete mathematical expression of the CRF model used in this work is as follows:

$$P_W(\mathbf{y}_i|\mathbf{x}_i) = \frac{\exp\left(\sum_{j=1}^{L_i} \phi_j^F(y_{ij}, \mathbf{x}_{ij}) + \sum_{j=1}^{L_i-1} \phi_j^T(y_{ij}, y_{ij+1})\right)}{\sum_{\mathbf{y}'_i} \exp\left(\sum_{j=1}^{L_i} \phi_j^F(y_{ij}, \mathbf{x}_{ij}) + \sum_{j=1}^{L_i-1} \phi_j^T(y_{ij}, y_{ij+1})\right)} \tag{1}$$

Where $\phi_j^F(y_{ij}, \mathbf{x}_{ij})$ is a feature potential and $\phi_j^T(y_{ij}, y_{ij+1})$ is a transition potential. Both of this potentials are assigned to the edges between features $\mathbf{x}$ and labels $\mathbf{y}$ and between labels $\mathbf{y}$ respectively. The following is the formal definition:

$$\phi_j^F(y_{ij}, \mathbf{x}_{ij}) = \sum_{c=1}^{C} \sum_{f=1}^{F} W_{cf}^F [y_{ij} = c] x_{ijf} \tag{2}$$

$$\phi_j^T(y_{ij}, y_{ij+1}) = \sum_{c=1}^{C} \sum_{c'=1}^{C} W_{cc'}^T [y_{ij} = c][y_{ij+1} = c'] \tag{3}$$

Next I will briefly describe each of the elements in this notations in the context of the problem to solve. $\mathbf{y}_i$ represents the labels assigned to each of the lines (using BIOL convention explained before). $\mathbf{x}_i$ represents all the set of features for each of the lines in a reference section in ith document. L_i represents each of the lines inside "references" section in ith document. C represents each of the output labels (using BIOL convention). F are a set of features that can be assigned to any line. W^F is a *feature parameter* and can be considered as $|C| \times |F|$ array where each entry describes how likely is for a label $c \in C$ to appear with a feature $f \in F$. Similarly, W^T is a *transition parameter* and can be conceptually thought of as a $|C| \times |C|$ array where each entry describes how likely a label $c \in C$ is followed by the label $c' \in C$. For a better understanding Figure 1 represents this formulations as a Markov graph.

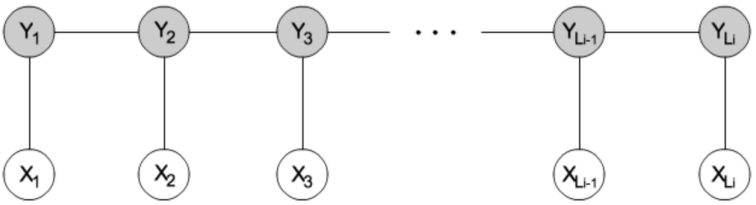

Figure 1: Conditional Random Fields

To learn the weight matrices W^F and W^T, on the other side, instead of using the traditional approach to compute the derivatives taking all the samples and calculate the gradients based on that, I use online algorithm AdaGrad (Adaptive Subgradient Methods) that can be consulted more in detail in [27]. Basically, this online learning algorithm allows to "dynamically incorporate knowledge of the geometry of the data observed in earlier iterations to perform more informative gradient based learning."[1] This is specially useful in case of having infrequently occurring features that are highly informative. In the case of references, there are features that don't appear in every paper such as those related to epilogue identification [2] . Furthermore, in this work I will compare the performance of AdaGrad with other online optimization algorithms such as RDA.
The following is the mathematical expression of the average log likelihood function derivative with respect to the parameter W^F used for learning:

$$\frac{\partial \mathcal{L}(W|\{\mathbf{x}_i, \mathbf{y}_i\}_{1:N})}{\partial W_{cf}^F} = \frac{1}{N} \sum_{i=1}^{N} \left(\sum_{j=1}^{L} ([y_{ij} = c] - P_W(y_{ij} = c|\mathbf{x}_i)) x_{ijf} \right) \tag{4}$$

The following is the mathematical expression of the average log likelihood function derivative with respect to the parameter W^T used for learning:

$$\frac{\partial \mathcal{L}(W|\{\mathbf{x}_i, \mathbf{y}_i\}_{1:N})}{\partial W_{cc'}^T} = \frac{1}{N} \sum_{i=1}^{N} \left(\sum_{j=1}^{L-1} ([y_{ij} = c][y_{ij+1} = c'] - P_W(y_{ij} = c, y_{ij+1} = c'|\mathbf{x}_i)) \right) \tag{5}$$

In both of the cases, a regularization term can be added (e.g. $l1$, $l2$, etc).

[1] Definition taken from [27]

[2] e.g. lexical features that look certain keywords such as "Acknowledgements" that may strongly indicate that the section that follows is no longer part of "References", but there maybe only a few papers in training set with this feature

4 Data Set

The data set consists in a set of papers (24 papers) extracted from NIPS conference web site that I tagged as part of the project using the PDF to SVG parser developed at IESL lab called `iesl-pdf-to-text` . The reference section in these papers have similar format and I expect that the CRF classifier will work fairly well on it. On the other side, I also pre-tagged a set of miscellaneous papers (38 papers) from different sources. The references in these papers are less structured (e.g. some of them are even not numbered and intermixed with other elements such as charts and epilogues with pictures). I expect that the CRF classifier will work worse in this last set and one of my experiments described in section 5 will consist in determine this performance.
With respect to the features, these are divided into *lexicon features* and *layout features*. The *lexicon features* capture the lexical properties of each line such as whether the line starts with capitalized letter or number, whether it ends in dot, and so on. The *layout features* capture the layout properties of the line such as the font size, the distance to the previous and next lines, etc. There is a total of 52 features (20 layout and 32 lexicon features).

5 Experiments and Results

In this section I will describe and show results of some of the experiments I have executed. First of all, as I already mentioned in section 4, I tagged two types of papers: the ones presented at NIPS conference, and some other miscellaneous papers from different disciplines. The first experiment I tried consists in determine the performance of CRF classifier using AdaGrad optimization algorithm on these different sets. For this purpose I performed random resampling validation tests (with $S = 50$) using different validation-train ratios. Figure 2 shows the results.

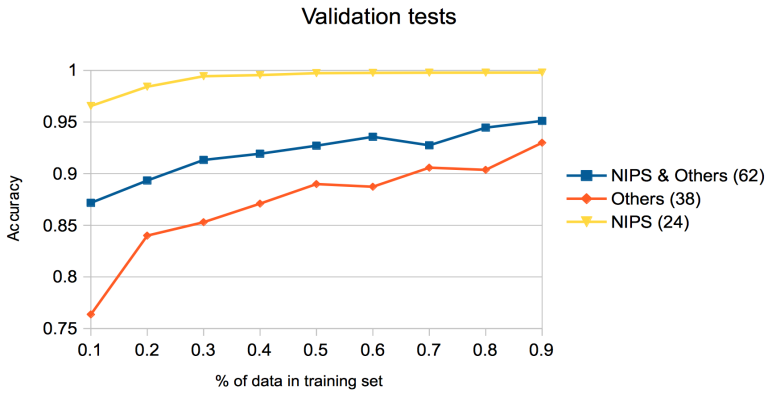

Figure 2: Validation test using AdaGrad optimization on different data sets

These results are expected. The classifier converges very fast to accuracy close to 1 on NIPS papers after training on around 12 papers which represent 50% of all the papers tagged from NIPS. This is because these papers follow a well defined layout template published at NIPS web page with only few exceptions. However, for more complicated layouts in "Others" scholarly publications, the classifier still doesn't converge to a particular accuracy value and probably needs more training data to produce better results.
Another experiment I have executed as part of this project consists in comparing different state of the art online gradient optimization algorithms. Besides AdaGrad, I worked with other two algorithms called RDA (Regularized Dual Averaging, look into [28] for details) and FTRL-Proximal (Follow-the-Regularized-Leader Proximal, look into [29] for details). Figure 3 shows accuracy for random re-sampling running on mix of NIPS and other papers with $S = 10$ and training on 80% of all the data in each iteration. The experiment was run for different $L1$ regularization values. Figure 4, on the other side, shows the effect of L2 regularization on these algorithms. As it can be observed, AdaGrad outperforms the other two optimization algorithms, which shows its better performance where the number of instances with very discriminative attributes is very small (as it already was explained in section 3), such as in this case. Probably the other algorithms may work better with higher amount of training data.

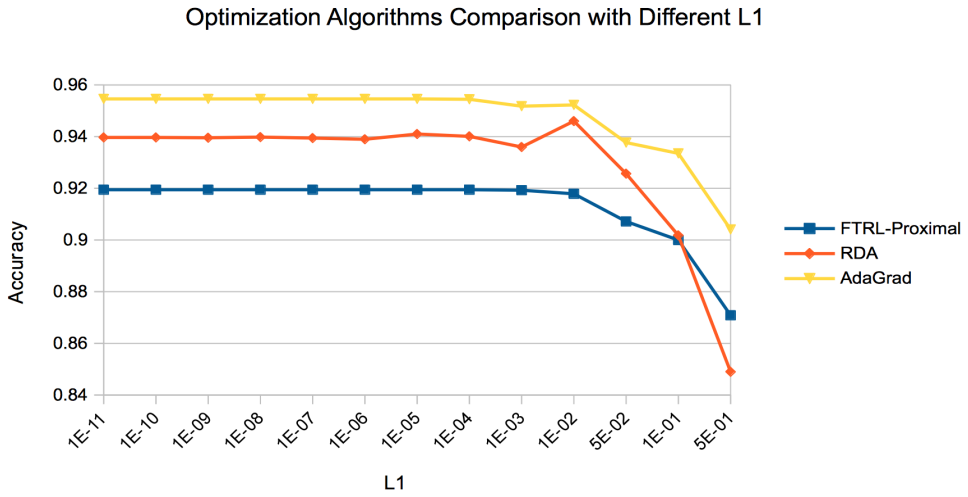

Figure 3: Validation tests using a set of L1 regularization settings to compare different online optimization algorithms performance (AdaGrad, RDA and FTRL-Proximal).

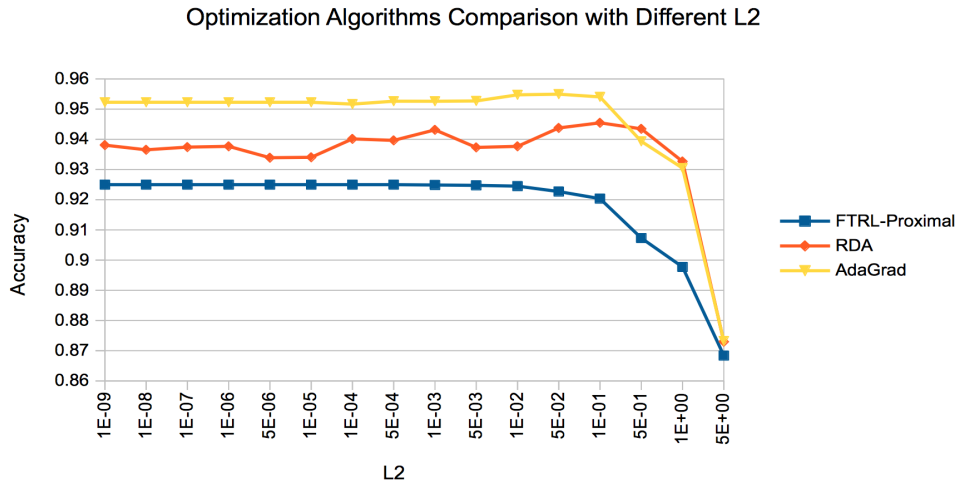

Figure 4: Validation tests using a set of L2 regularization settings to compare different online optimization algorithms performance (AdaGrad, RDA and FTRL-Proximal).

6 Configuration

In this section I will indicate briefly where the labeled files and the model is located. All the labeled files are uploaded to blake server in the following files:

1. `blake.cs.umass.edu:/iesl/canvas/klim/svg_other_annotated.zip`
2. `blake.cs.umass.edu:/iesl/canvas/klim/svg_nips_annotated.zip`

The manual references annotator is implemented in `https://github.com/iesl/ref-segmentation-annotator`. The model, on the other hand, is implemented in `https://github.com/iesl/ref-segmentation-model`.

7 Discussion and Conclusions

In this work I presented one possible way of identifying physical and logical layouts inside the document. This approach consists in training Conditional Random Fields and using optimization algorithms to learn the model. As it turned out in the studied case, the performance depends on a particular optimization algorithm used. Possible aspects to tackle in order to improve the reference identification accuracy include labeling more data on papers whose reference section is not well formatted (e.g. with epilogue sections following it, charts, acknowledgments, non-enumerated and un-tabbed references, etc.). Another point to consider in future work is to check the BIOL label distribution, I have noticed several examples where the classifier labeled correctly, but only using the I tag, without the B and L labels at the beginning and the end. This can be fixed easily having a rule-based routine to check this assignment.

References

[1] Bounhas, I. & Slimani, Y. (2010) A hierarchical Approach for Semi-Structured Document Indexing and Terminology Extraction. *International Conference on Information Retrieval and Knowledge Management*

[2] Tkaczyk, D., Bolikowski, L., Czeczko, A. & Rusek, K. (2012) A modular metadata extraction system for born-digital articles. *In: Proceedings of the 10th IAPR International Workshop on Document Analysis Systems.*

[3] Klampfl, S. & Kern, R. (2013) An Unsupervised Machine Learning Approach to Body Text and Table of Contents Extraction from Digital Scientific Articles. *International Conference on Theory and Practice of Digital Libraries*

[4] Han, H., Giles Lee, C., Manavoglu, E. & Zha, H. (2003) Automatic Document Metadata Extraction using Support Vector Machines. *In Proceedings of the 3rd ACM/IEEE-CS Joint Conference on Digital libraries*, (pp. 3748)

[5] Stoffel, A., Spretke, D., Kinnemann, H., & Keim, D.A. (2010) Enhancing Document Structure Analysis using Visual Analytics. *In Proceedings of the 2010 ACM Symposium on Applied Computing* (pp. 812). New York: ACM Press.

[6] Do, H.H.N., Chandrasekaran, M.K., Cho, P.S. & Kan, M.Y. (2013) Extracting Authors and Affiliations in Scholarly Documents. *In Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries*, (pp. 219-228)

[7] Luong, M. T., Nguyen, T. D., & Kan, M.Y. (2010). Logical structure recovery in scholarly articles with rich document features. *International Journal of Digital Library Systems* (pp. 1-23)

[8] Ramakrishnan C., Patnia A., Hovy E. & Burns G.A. Layout-Aware Text Extraction from Full-text PDF of Scientific Articles. (2012) *Source Code for Biology and Medicine* Vol. 7, No. 1

[9] Lee K.H. et al., Logical structure analysis and generation for structured documents: A syntactic approach. (2003) *IEEE Transactions on Knowledge and Data Engineering* Vol. 5, No. 5 (pp. 1277-1294)

[10] Di Iorio, A., Peroni, S., Poggi, F., Vitali, F. & Shotton, D. (2013). Recognising document components in XML-based academic articles. *In Proceedings of DocEng*

[11] Shafait F., Keysers D. & Breuel T. M. Performance comparison of six algorithms for page segmentation. (2006) *7th IAPR Workshop on Document Analysis Systems* (pp. 368-379).

[12] Belad, A. & Rangoni, Y. (2008) Structure Extraction in Printed Documents Using Neural Approaches. *Machine Learning in Document Analysis and Recognition* (pp. 21-43).

[13] Mao S., Rosenfeld A. & Kanungo T. (2003) Document structure analysis algorithms: a literature survey. *Document Recognition and Retrieval X* (pp. 197-207).

[14] Langer, H., Lungen, H. & Bayerl, P. S. (2004) Text type structure and logical document structure. *In Proceedings of the ACL Workshop on Discourse Annotation* (pp. 49-56)

[15] Alzahrani, S., Palade, V., Salim, N., & Abraham, A. (2012) Using structural information and citation evidence to detect significant plagiarism cases in scientific publications. *JASIST* (pp. 286 - 312).

[16] Nguyen, T.D. & Luong, M.T. (2010). WINGNUS: Keyphrase extraction utilizing document logical structure. *ACL 2010 Workshop on Evaluation Exercises on Semantic Evaluation*

[17] Flynn P., Zhou L., Maly K., Zeil S., & Zubair M. (2010) Automated Template-Based Metadata Extraction Architecture. *In Proceedings of the 10th international conference on Asian digital libraries* (pp. 327-336).

[18] Giuffrida G., Shek E., & Yang J. (2000) Knowledge-based metadata extraction from PostScript files. *in Proc. of the fifth ACM conference on Digital libraries* (pp. 77-84).

[19] Mao S., Kim J. W., & Thoma G. R. (2004) A Dynamic Feature Generation System for Automated Metadata Extraction in Preservation of Digital Materials. *Proceedings of the First International Workshop on Document Image Analysis for Libraries*

[20] Seymore K., McCallum A., & Rosenfeld R. (1999) Learning Hidden Markov Model Structure for Information Extraction. *in AAAI 99 Workshop on Machine Learning for Information Extraction*

[21] Wang Y., Phillips I., and Haralick R. (2006) Document zone content classification and its performance evaluation. *Pattern Recognition* Vol. 39, No. 1 (pp. 57-73)

[22] Conway A. et al. (1993) Page grammars and page parsing: A syntatic approach to document layout recognition. *in Proceedings of International Conference on Document Analysis and Recognition* (pp. 761-764)

[23] Pinto, D., McCallum, A., Wei, X. and Bruce, W. (2003) Table Extraction Using Conditional Random Fields. *SIGIR-03*

[24] Peng F. & McCallum A. (2004) Accurate information extraction from research papers. *In Proc. of HLT/NAACL*

[25] Klinger R. & Tomanek K. (2007) Classical Probabilistic Models and Conditional Random Fields. *Technical Report, Dept. of Computer Science, Dortmund Univ. of Technology*

[26] Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *In Proceedings of ICML 2001*

[27] Duchi, J., Hazan, E., and Singer, Y. (2011) Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research* (pp. 2121-2159)

[28] Xiao L. (2010) Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, Vol 11, (pp. 2543-2596)

[29] McMahan H. B. (2011) Follow-the-regularized-leader and mirror descent: Equivalence theorems and L1 regularization. *In AISTATS*