# Open-domain Table Detection Using Large-scale PDF Files without Annotation

**Miao Fan**[†] **and Doo Soon Kim**[‡]
[†]Tsinghua University, Beijing, 100084, China.
[‡]Bosch Research, Palo Alto, CA, 95014, U.S.A.
fanmiao.cslt.thu@gmail.com



**Abstract**

Superior to state-of-the-art approaches which compete in recognizing tables among 67 annotated government reports (PDF) released by *ICDAR 2013 Table Competition*, this paper contributes a novel paradigm leveraging large-scale unlabeled PDF files to open-domain table detection. We integrate the paradigm into our latest developed system (*PdfExtra*) to detect the region of tables by means of 9,466 academic articles from the entire repository of *ACL Anthology*, where almost all papers are archived by PDF format without annotation for tables. The paradigm first designs heuristics to automatically construct weakly labeled data. It then feeds diverse evidences, such as layouts of documents and linguistic features, which are extracted by *Apache PDFBox* and processed by *Stanford NLP* toolkit, into different canonical classifiers. We finally use these classifiers, i.e. *Naive Bayes*, *Logistic Regression* and *Support Vector Machine*, to collaboratively vote on the region of tables. Experimental results show that *PdfExtra* achieves a great leap forward, compared with the state-of-the-art approach. Moreover, we discuss the factors of different features, learning models and even domains of documents that may impact the performance. Extensive evaluations demonstrate that our paradigm is compatible enough to leverage various features and learning models for open-domain table region detection within PDF files.

## Introduction

Tables are widely adopted in web pages, academic articles, online manuals, etc. They present more structural and condensed information than what plain texts do. Computer scientists who conduct research on information extraction take delight in engaging with tables that occur in electronic documents, as they are the natural sources to feed and populate relational databases.

Some formats of the electronic documents are machine-readable, such as *HTML*, *XML* and even *TEX*. These formats derive from *SGML* (Standard Generalized Markup Language) and inherit the basic principle that the language pins a pair of specific tags to mark a snippet of text. For example, *HTML* files use $\langle table \rangle$ as the start and $\langle /table \rangle$ as the end, to indicate the region of a table. AI programs can easily recognize expected regions with the help of tags, and extract the information that we want with pre-defined actions. So far, those systems have already helped us collect 147 million tables[1] from the Web.

However, it is tedious for our human beings to read the markup language, because we are sensitive to the layouts of documents, and focus more on the contents. Therefore, the *Portable Document Format* (PDF) was designed as a file format to represent a document independent of the platform it displays, and to preserve the layouts both on screen and in print. These strengths draw much attention from the online publishing. So far, many academic papers and manuals have adopted PDF as the standard format.

Unfortunately, we meet Waterloo when detecting the region of tables within PDF files, due to the lack of structural information. To the best of our knowledge, the latest off-the-shelf software, *Apache PDFBox*[2], could only provide the coordinates $(x, y)$ and the font style of each character in a PDF document. As table region detection is the fundamental and significant step for further information extraction from PDF files, fruitful approaches have been proposed in recent decades. However, they either simply design heuristic rules based on pre-defined layouts, or adopt supervised learning techniques fed by few annotated corpora from restricted domains. For instance, *ICDAR 2013* set up a competition about table detection and structure recognition within 67 annotated PDF documents posted by U.S. and E.U. governments, where each document is accompanied by a *XML* file to indicate the location of tables.

When we further apply these methods to some free access digital academic archives, such as *IEEE Xplore* and *Springer Link*, the variety of layouts and explosive amount of unannotated data expose the urgent demand on unsupervised or semi-supervised frameworks. By means of these frameworks, we do not have to spend much labor on annotation, but can leverage large-scale unlabeled PDF files. To the best of our knowledge, Klampfl et al. (Klampfl, Jack, and Kern 2014) have recently proposed unsupervised table recognition methods applied on digital scientific articles. However, their work was purely based on heuristic rules and evaluated on 109 files in total. We consider it not flexible enough to handle more PDF articles with variable layouts.



[1] http://webdatacommons.org/webtables/
[2] https://pdfbox.apache.org/

Therefore, we firstly attempt to challenge the issue of open-domain table region detection leveraging large-scale PDF files in this paper, especially without the help of labeled data. The new paradigm we design combines the advantages of heuristics and supervised learning models. It leverages heuristic rules to automatically construct weakly labeled datasets for training, and adopts multiple canonical classifiers, such as *Logistic Regression*, *Support Vector Machine* and *Naive Bayes*, to collaboratively predict the boundary of tables within PDF documents. Moreover, we integrate the paradigm as the pre-processing phase of our newly developed system (*PdfExtra*) to automatically extract tables powered by the entire repository of *ACL Anthology* (9,466 files) without annotation. Experimental results show that it takes a great leap forward, compared with the state-of-the-arts. Besides the *ACL Anthology* dataset, we use another benchmark corpus in addition, released by *ICDAR 2013 Table Competition* from a different domain to further discuss the factors of various features, learning models that may impact the performance. Extensive evaluations demonstrate that our paradigm is compatible enough to leverage various features and learning models for open-domain table region detection.

## Related Work

A comprehensive review can be found in the final report of *ICDAR 2013 Table Competition* (Gobel et al. 2013), which announced the performances of recent academic and commercial systems on either table region detection or table structure analysis. Here we restrict our survey on a number of recent methods that attempt to discover table regions within PDF files.

The first effort was the *pdf2table*[3] system (Yildiz, Kaiser, and Miksch 2005), which used heuristics to detect the table region. It assumed that a table had more than one column, and a table region was formed by merging neighboring multi-lines. However, the algorithm could only handle pages with single-column layouts.

The *PDF-TREX* system (Oro and Ruffolo 2009) considered a set of words as seeds first, and identified tables in a bottom-up manner. Specifically, words were aligned and grouped to lines based on their vertical overlap, and line segments were obtained by applying hierarchical agglomerative clustering to the words. According to the number of segments, a line was categorized into three classes: text lines, table lines, and unknown lines. Then, the table region could be found by combining contiguous table lines or unknown lines.

Supervised classification models were mainly adopted by Liu et al. (2008), who designed a table detection method that leveraged heuristics to construct lines from individual characters and to select those sparse lines that occur within a table for training. Starting from a table caption, these sparse lines were then iteratively merged to a table region. This approach is very similar to the state-of-the-art unsupervised method (Klampfl, Jack, and Kern 2014) and ours, except that it was built upon labeled text blocks instead of lines.

The up-to-date approach (Klampfl, Jack, and Kern 2014) did not rely on annotated data, but used complex heuristics to achieve comparable performances with supervise-based systems. Our system *PdfExtra* costs free on labeled data but covers large-scale PDF files with varies layouts. Therefore, we mainly compare the performance of system with the state-of-the-art unsupervised method (Klampfl, Jack, and Kern 2014).

## Paradigm

*PdfExtra* benefits a lot from the off-the-shelf software *Apache PDFBox* which can recognize almost all characters within a PDF document. Beyond the characters, the software also provides the horizontal and vertical coordinates, as well as the font style for each of them. Thus each "rich character" can be represented as a tuple: $\langle character, x-axis, y-axis, font-type, font-size \rangle$. In addition, *Apache PDFBox* can merge the characters together into words, and return words in sequence that visually lay in the same line. There is nothing more that it can do to discover tables. Therefore, we leverage the outputs from *Apache PDFBox* and engage in predicting whether each line belongs to a table or not.

Although we have formulated the table region detection task into a binary classification problem, we still suffer the lack of annotated training data. As illustrated by Figure 1, the paradigm that we have designed to fix the issue contains three phases:

### Heuristic annotation

Inspired by the idea of distant supervision (Fan et al. 2014; Fan, Zhou, and Zheng 2015), we adopt heuristics that can help automatically generate large-scale weakly labeled training examples. More specifically, we create a spider that downloads academic articles from *ACL Anthology*[4], in which almost all papers are archived in PDF format. 9,466 literatures in total the year 2000 to 2015 are collected. For a PDF article, we process each page in three steps as follows,

- *Indicator Recognition*: As all camera-ready drafts must conform to a limited number of official templates to be published, the word "Table" or "Tab." that appears in front of a line generally indicates the caption of a table. In other words, we find the lower or the upper boundary of the table region which depends on the templates.
- *Surrounding Contexts*: The caption line plays a role in separating the table from the main body. Because we do not know which portion belongs to a table, we usually extend $k$ lines up and down as the candidate context.
- *Positive v.s. Negative Examples*: After extracting these candidates, we assume that the group of lines with more blanks/margins will more likely locate in a table, rather than the other group. In this way, we can construct a balanced corpus for binary classification, even if it is weakly annotated by heuristics above.

[3] http://ieg.ifs.tuwien.ac.at/projects/pdf2table/

[4] http://aclweb.org/anthology/

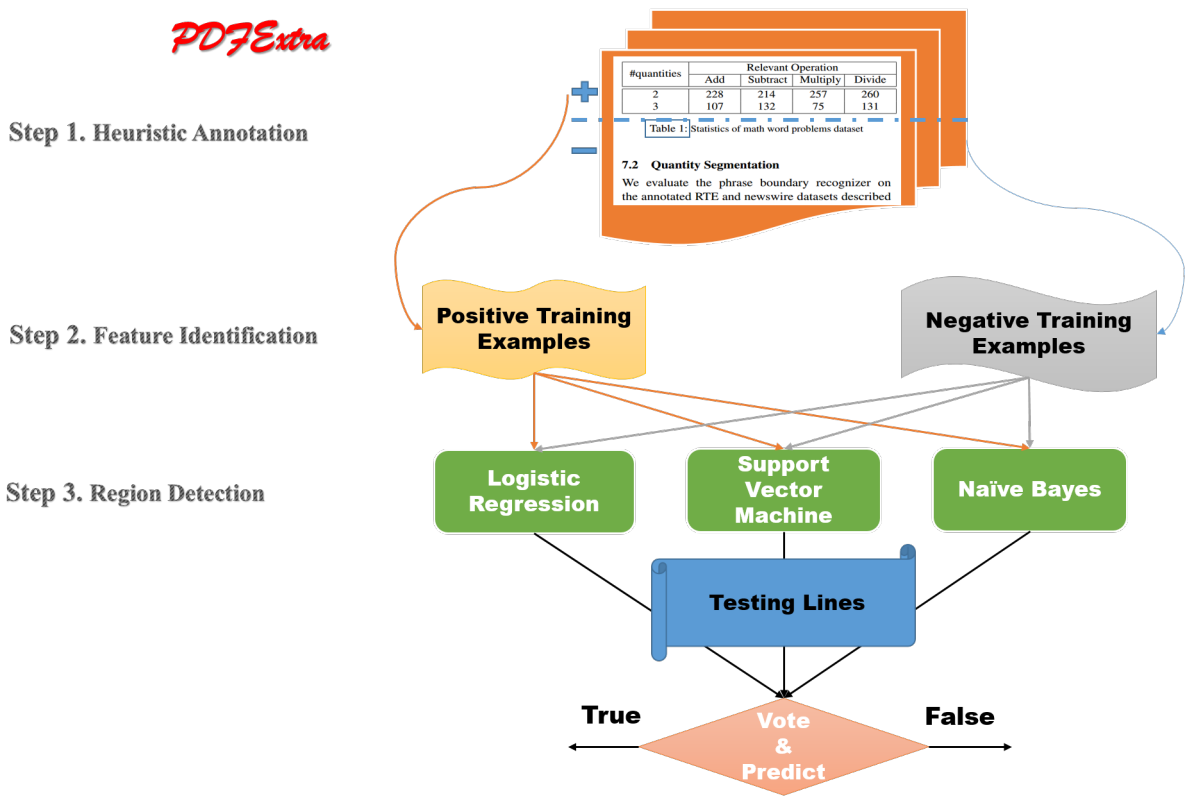


Figure 1: The proposed paradigm adopted by the *PDFExtra* system.

By means of the heuristics we have proposed, a large-scale weakly labeled dataset can be automatically constructed for training. For instance, the rules help us prepare more than 350,000 lines as training examples extracted from *ACL Anthology*. As each line is a sequence of words in which every "rich character" with its coordinates and font style, we can further process each word to mark its start and end coordinates in the horizontal direction.

## Feature identification

The state-of-the-art approach (Klampfl, Jack, and Kern 2014) only concerns about the layouts of a PDF document. It iteratively includes a sparse block into a table in the buttom-up manner, where a block is identified as "sparse" if 1) their width is smaller than $\frac{2}{3}$ of the average width of a text block, or 2) there exists a gap between two consecutive words in the block that is larger than than two times the average width between two words in the document.

However, we believe that both linguistic and layout features are significant. Therefore, we select three kinds of features based on our observation that may contribute to detecting the region of tables. They are:

- *Normalized Average Margin* (*NAM*): According to the horizontal coordinate of each word in lines, we calculate the average margin between two consecutive words, so that each line is assigned by the feature. In most cases, the average margin between two consecutive words in the main body equals to the size of a space, and that in the tables usually occupies more. However, the average margin differs along with layouts, and generally the one-column layouts generate much larger margin than the two-column formats. Therefore, we normalize the average margin within the same page to be the feature that represents the layouts.
- *POS Tag Distribution* (*PTD*): It is a common consensus that we prefer displaying information in a more structural and condensed way in tables, rather than flowery language expressed by sentences in the main body of an article. Intuitively, more noun phrases appear in tables, but less adjectives and adverbs are used. This distinction leads to the diverse distribution of the part-of-speech (POS) tags, which we concern as the second feature. There are 5 kinds of part-of-speech tags under our consideration processed by *Stanford POS Tagger*[5]: *NN*, *VB*, *JJ*, *RB* and *OTHERS*.
- *Named Entity Percentage* (*NEP*): We extend the traditional scope of named entities and include the number and the time. Therefore, 5 kinds of named entity tags, i.e. *PERSON*, *LOCATION*, *ORGANIZATION*, *NUMBER* and *TIME*, are recognized by *Stanford Named Entity Recognizer*[6]. For each kind of named entity, we compute its percentage in each line.

[5] http://nlp.stanford.edu/software/tagger.shtml

[6] http://nlp.stanford.edu/software/CRF-NER.

| Dataset | # Training files | # Testing PDF files | # Training Lines | # Testing Lines |
|---|---|---|---|---|
| *ICDAR 2013* | 50 | 17 | 804 | 224 |
| *ACL Anthology* | 9,280 | 186 | 357,892 | 346 |

Table 1: Statistics of benchmark datasets for table region detection.

## Region detection

We adopt three canonical classifiers, i.e. *Logistic Regression*, *Support Vector Machine* and *Naive Bayes*, to help decide whether a line of "rich characters" provided by *Apache PDFBox* belongs to a table or not. Before using these models, we denote the three kind of features as $x_{NAM}$, $x_{PTD}$ and $x_{NEP}$, respectively. However, most of the features are described by continuous variables. They need to be mapped into discrete variables[7], so that we can leverage *Naive Bayes* to make prediction. The three classifiers we use are as follows, based on different hypotheses that lead to various mathematical models:

- *Logistic Regression*[8] (*LR*): We can score each line to indicate whether it belongs to a table or not. The higher the score $s$, the more possible the corresponding line does. Let's assume that we can approximate $s$ as a linear function of the feature vector $\mathbf{x}$ which contains $x_{NAM}$, $x_{PTD}$ and $x_{NEP}$:

$$\begin{aligned} s_\theta(\mathbf{x}) &= \theta^T \mathbf{x} \\ &= \theta_{NAM} x_{NAM} + \theta_{PTD} x_{PTD} + \theta_{NEP} x_{NEP} + \theta_0. \end{aligned} \tag{1}$$

  Here the $\theta$ represents the vector of parameters along with the features.

  The classifier chooses the *sigmoid function* to map the score $s$ from $(-\infty, +\infty)$ into $[0.0, 1.0]$ which matches the definition of probability:

$$Pr(y = 1|\mathbf{x}) = \frac{1}{1 + e^{-s_\theta(\mathbf{x})}}, \tag{2}$$

  and

$$Pr(y = 0|\mathbf{x}) = \frac{1}{1 + e^{s_\theta(\mathbf{x})}}, \tag{3}$$

  where $y = 1$ indicates the line with feature $\mathbf{x}$ is included in a certain table.

- *Support Vector Machine*[9] (SVM): The model enhances the hypothesis of linear combination which is illustrated by Equation (1), and is described from the perspective of the functional margin $\gamma$:

$$\gamma = y(\mathbf{w}^T \mathbf{x} + b), \tag{4}$$

  where $y = \{+1, -1\}$, $\mathbf{w}$ is the vector of weights, and $b$ is the bias. Hence, a large functional margin represents a correct prediction.

- *Naive Bayes (NB)*[10] Different from the two discriminative classification model, *Naive Bayes* is a classical generative model, which originally attempts to describe the joint probability of $\mathbf{x}$ and $y$, i.e. $Pr(y, \mathbf{x})$. The key assumption of *Naive Bayes* is that all the features are independent from each other given $y$:

$$\begin{aligned} &Pr(\mathbf{x}|y = 1) \\ &= Pr(x_{NAM}|y = 1) Pr(x_{PTD}|y = 1) Pr(x_{NEP}|y = 1), \end{aligned} \tag{5}$$

  and

$$\begin{aligned} &Pr(\mathbf{x}|y = 0) \\ &= Pr(x_{NAM}|y = 0) Pr(x_{PTD}|y = 0) Pr(x_{NEP}|y = 0), \end{aligned} \tag{6}$$

  which we believe it will behave differently from the other classifiers.

# Experiment

We set experiments that conduct comparison between our paradigm and the state-of-the-art *Heuristics* approach (Klampfl, Jack, and Kern 2014) evaluated on two datasets, i.e. *ACL Anthology* and *ICDAR 2013 Table Competition*, with standard metrics.

## Datasets

We prepare two datasets from different domains. The dataset[11] of *ICDAR 2013 Table Competition* is the benchmark in which there are 67 ground-truth PDF documents of U.S. and E.U. governments. The size of the other *ACL Anthology* dataset is much larger, which contains 9,466 academic articles from the year 2000 to 2015. It covers more than 10 top-tier conferences related to *Computational Linguistics*, such as *ACL, EMNLP, COLING, NAACL*, etc. Table 1 shows the statistics of *ICDAR 2013* and *ACL Anthology* datasets for evaluation.

- *ICDAR 2013*: We divide the dataset into two parts. 75% files (50 documents) are used as training examples, and 25% files (17 documents) are prepared for testing. After processed by the *Heuristic annotation*, we get 804 lines left for training. And we manually annotate 224 lines from 17 testing documents for testing.

shtml

[7] $x_{NAM}$ generally ranges from 0.0 to 1.0. We set a step size that equals to 0.2 to map the continuous variables. For example, if $0.0 \leq x_{NAM} < 0.2$, then $x_{NAM} = 1$, and so on.

[8] To implement the classifier, we integrate the LIBLINEAR: `http://liblinear.bwaldvogel.de/` into our system.

[9] LIBSVM: `http://www.csie.ntu.edu.tw/~cjlin/libsvm/` is the well-known open-source software that can be leveraged by *PdfExtra*.

[10] We adopt `https://github.com/ptnplanet/Java-Naive-Bayes-Classifier` to implement the Naive Bayes classifier.

[11] `http://www.tamirhassan.com/files/icdar2013-competition-dataset-with-gt.zip`

| Ground-truth / Prediction | POSITIVE (+1) | NEGATIVE (-1) |
|---|---|---|
| **POSITIVE (+1)** | *True Positive (TP)* | *False Positive (FP)* |
| **NEGATIVE (-1)** | *False Negative (FN)* | *True Negative (TN)* |

Figure 2: $2 \times 2$ evaluation matrix for binary classification.

- *ACL Anthology*: The paper published in 2015 are kept, and we label 346 lines of them as the ground-truth examples for testing. In addition, we gain 357,892 lines from 9,280 academic articles as the weakly labeled training examples.

## Metrics

Since we regard the table region detection as binary classification problems, several standard metrics, such as *Accuracy*, *Precision*, *Recall*, *F1-measure*, are adopted for evaluating the performances. Each ground-truth line for testing is classified based on its features, and the output labels will be $+1$ or $-1$. As shown in Figure 2, anyone of the testing examples will fall into one of the four cells, i.e. *True Positive*. For instance, if a system assigns the positive label $(+1)$ to a ground-truth testing line which should be regarded as a negative example, that is a false positive $(FP)$.

- *Accuracy* is a metric to measure the overall performance of binary classification. It concerns about all the testing examples, including the positives and the negatives, and indicates the proportion of lines that are made correct predictions. Therefore,

$$Accuracy = \frac{\#(TP) + \#(TN)}{\#(TP) + \#(FP) + \#(FN) + \#(TN)}. \tag{7}$$

- *Precision* and *Recall* are a pair of metrics that focus on the positive ground-truth lines. Specifically, *precision* represents the proportion of correct examples regarded as the positives, i.e.,

$$Precision = \frac{\#(TP)}{\#(TP) + \#(FP)}, \tag{8}$$

and *recall* concerns about the proportion of positive predictions within all positive ground-truth examples:

$$Recall = \frac{\#(TP)}{\#(TP) + \#(FN)}. \tag{9}$$

- *F1-measure* is a trade-off between *precision* and *recall*, which measures the harmonic mean of the two metrics above:

$$F1\text{-}measure = \frac{2 \times Precision \times Recall}{Precision + Recall}. \tag{10}$$

## Performances

We use the four metrics above to measure the performances of our system *PdfExtra*, compared with the state-of-the-art approach *Heuristics* (Klampfl, Jack, and Kern 2014). Both of them are evaluated by two benchmark datasets, i.e. *ICDAR 2013* and *ACL Anthology*. Table 2 and 3 show the results of the experiments, and we find out that *PdfExtra* achieves significant improvements beyond the latest approach.

| Approach | Accuracy | Precision | Recall | F1-measure |
|---|---|---|---|---|
| *Heuristics* | 0.5491 | 0.5946 | 0.3826 | 0.4656 |
| *PdfExtra* | **0.6607** | **0.7407** | **0.5217** | **0.6122** |

Table 2: Performance comparison on *ICDAR 2013* testing set, fed by *ICDAR 2013* training examples.

| Approach | Accuracy | Precision | Recall | F1-measure |
|---|---|---|---|---|
| *Heuristics* | 0.5665 | 0.5660 | 0.3659 | 0.4444 |
| *PdfExtra* | **0.7948** | **0.7385** | **0.8780** | **0.8022** |

Table 3: Performance comparison on *ACL Anthology* testing set, fed by *ACL Anthology* training examples.

# Discussion

To deeply analyze the paradigm we have proposed, we discuss the factors that may impact the performance from three perspectives:

## Impact of features

We try different combinations of features. They are the layout feature only (*NAM*), the layout with part-of-speech feature (*NAM + PTD*) and the combination of all the features (*NAM + PTD + NEP*). We keep collaboratively using the three classification models to vote the final prediction. Both of Table 4 and 5 demonstrate that pure layout feature does not perform well on detecting the table region, as shown by the experimental results of state-of-the-art *Heuristics*(Klampfl, Jack, and Kern 2014) and *PdfExtra (NAM)*. For *ICDAR 2013* dataset, the best feature combination is *NAM + PTD*. And the other empirical study displays that the feature combination of *NAM + PTD + NEP* leads to the best performance on *ACL Anthology* dataset.

## Impact of classifiers

Besides the combinations of features, three classifiers may also perform variously, due to their different hypotheses of mathematical modeling. Therefore, we map both *ICDAR 2013* and *ACL Anthology* datasets to the same feature combination (*NAM + PTD + NEP*) schema, and iteratively select an individual classifier, such as Naive Bayes (*PdfExtra(NB)*), Logistic Regression (*PdfExtra(LR)*) or Support Vector Machine (*PdfExtra(SVM)*), to compare with the voting version (*PdfExtra*). They are the layout feature only (*NAM*), the layout with part-of-speech feature (*NAM + PTD*) and the combination of all the features (*NAM + PTD + NEP*).

| Approach | Accuracy | Precision | Recall | F1-measure |
|---|---|---|---|---|
| *Heuristics* | 0.5491 | 0.5946 | 0.3826 | 0.4656 |
| *PdfExtra (NAM)* | 0.5134 | 0.5134 | **1.0000** | 0.6785 |
| *PdfExtra (NAM + PTD)* | **0.7321** | **0.7835** | 0.6609 | **0.7170** |
| *PdfExtra (NAM + PTD + NEP)* | 0.6607 | 0.7407 | 0.5217 | 0.6122 |

Table 4: Performance comparison with different combinations of features on *ICDAR 2013* testing set, fed by *ICDAR 2013* training examples.

| Approach | Accuracy | Precision | Recall | F1-measure |
|---|---|---|---|---|
| *Heuristics* | 0.5665 | 0.5660 | 0.3659 | 0.4444 |
| *PdfExtra (NAM)* | 0.4740 | 0.4740 | **1.0000** | 0.6431 |
| *PdfExtra (NAM + PTD)* | 0.7312 | 0.6564 | 0.9085 | 0.7621 |
| *PdfExtra (NAM + PTD + NEP)* | **0.7948** | **0.7385** | 0.8780 | **0.8022** |

Table 5: Performance comparison with different combinations of features on *ACL Anthology* testing set, fed by *ACL Anthology* training examples.

| Approach | Accuracy | Precision | Recall | F1-measure |
|---|---|---|---|---|
| *Heuristics* | 0.5491 | 0.5946 | 0.3826 | 0.4656 |
| *PdfExtra (NB)* | **0.7902** | **0.7464** | **0.8957** | **0.8142** |
| *PdfExtra (LR)* | 0.6071 | 0.6956 | 0.4174 | 0.5217 |
| *PdfExtra (SVM)* | 0.6607 | 0.7407 | 0.5217 | 0.6122 |
| *PdfExtra* | 0.6607 | 0.7407 | 0.5217 | 0.6122 |

Table 6: Performance comparison with different classifiers on *ICDAR 2013* testing set, fed by *ICDAR 2013* training examples.

| Approach | Accuracy | Precision | Recall | F1-measure |
|---|---|---|---|---|
| *Heuristics* | 0.5665 | 0.5660 | 0.3659 | 0.4444 |
| *PdfExtra (NB)* | 0.7861 | 0.7206 | **0.8963** | 0.7989 |
| *PdfExtra (LR)* | 0.7948 | 0.7385 | 0.8780 | 0.8022 |
| *PdfExtra (SVM)* | 0.7919 | 0.7347 | 0.8780 | 0.8000 |
| *PdfExtra* | **0.7948** | **0.7385** | 0.8780 | **0.8022** |

Table 7: Performance comparison with different classifiers on *ACL Anthology* testing set, fed by *ACL Anthology* training examples.

| Approach | Accuracy | Precision | Recall | F1-measure |
|---|---|---|---|---|
| *Heuristics* | 0.5491 | 0.5946 | 0.3826 | 0.4656 |
| *PdfExtra (ICDAR)* | 0.6607 | 0.7407 | 0.5217 | 0.6122 |
| *PdfExtra (ACL)* | **0.7803** | **0.7683** | **0.7683** | **0.7683** |

Table 8: Cross-domain performance comparison on *ICDAR 2013* testing set, fed by *ACL Anthology* training examples.

Table 6 and 7 show the performances on *ICDAR 2013* and *ACL Anthology* datasets respectively, and *Naive Bayes* classifier behaves stably on the two benchmark datasets.

### Impact of domains

The most significant perspective of our new paradigm that needs to be discussed, is the evaluation on cross-domain datasets. It directly reflects the generality of a paradigm. If it could only outperform the state-of-the-art approaches when trained and tested by the PDF documents in the same or specific domain, the paradigm would still be a trial version that make minor contributions on the research of table region detection. Therefore, An experiment is set in which we feed the training examples of *ACL Anthology* dataset to our model, and test the performance on the testing set of *ICDAR 2013*. Fortunately, testing on files of *ICDAR 2013* achieves comparable performance with testing on *ACL Anthology* dataset. Moreover, *PdfExtra (ACL)* shows the better capability on detecting tables on government documents after trained by academic articles. The reason why our paradigm can handle cross-domain files, is that all the features and classifiers we leverage are independent from the layouts, and even the contents within diverse PDF documents.

## Conclusion

In this paper, we have contributed a novel paradigm for detecting the region of tables within PDF documents. It absorbs superiorities from both supervised and unsupervised approaches, and firstly covers almost tens of thousands PDF documents in a different domain. To be specific, it leverages different supervised learning models to adapt varies layouts and linguistic features within tables from large-scale PDF files, but costs free on labeling training corpus. We integrate the paradigm into our system *PdfExtra* which enhances the off-the-shelf software *Apache PDFBox* to predict whether a text line belongs to a table or not. Three classification models have been evaluated, which are *Logistic Regression*, *Support Vector Machine*, and *Naive Bayes*. We find out that *Naive Bayes* performs stable prediction on both two benchmark datasets, and linguistic features bring a great leap forward on the performance. What's more, we prove that our paradigm is robust to table detection on open-domain PDF documents.

However, the idea of weakly labeled paradigm can not avoid bringing noise into training data which impacts the performance of system. In the future, we look forward to exploring the correlation between tables within the same article to filter out the faults.

## Acknowledgments

The first author conducted this work during the period of summer internship (06/01/2015-08/07/2015) at Bosch Research in U.S, and this paper is dedicated to researcher Doo Soon Kim for his guidance.

## References

[Fan et al. 2014] Fan, M.; Zhao, D.; Zhou, Q.; Liu, Z.; Zheng, T. F.; and Chang, E. Y. 2014. Distant supervision for relation extraction with matrix completion. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 1, 839–849.

[Fan, Zhou, and Zheng 2015] Fan, M.; Zhou, Q.; and Zheng, T. F. 2015. Distant supervision for entity linking. *arXiv preprint arXiv:1505.03823*.

[Gobel et al. 2013] Gobel, M.; Hassan, T.; Oro, E.; and Orsi, G. 2013. Icdar 2013 table competition. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, 1449–1453. IEEE.

[Klampfl, Jack, and Kern 2014] Klampfl, S.; Jack, K.; and Kern, R. 2014. A comparison of two unsupervised table recognition methods from digital scientific articles. *D-Lib Magazine* 20(11):7.

[Liu, Mitra, and Giles 2008] Liu, Y.; Mitra, P.; and Giles, C. L. 2008. Identifying table boundaries in digital documents via sparse line detection. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, 1311–1320. New York, NY, USA: ACM.

[Oro and Ruffolo 2009] Oro, E., and Ruffolo, M. 2009. Pdftrex: An approach for recognizing and extracting tables from pdf documents. In *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*, 906–910. IEEE.

[Yildiz, Kaiser, and Miksch 2005] Yildiz, B.; Kaiser, K.; and Miksch, S. 2005. pdf2table: A method to extract table information from pdf files. In *IICAI*, 1773–1785.