Document Segmentation Labeling Techniques for Court Filings

Alex Lyte, Karl Branting

The MITRE Corporation

{alyte,lbranting}@mitre.org

ABSTRACT

Arguments, motions, and decisions in courts of the United States of America are recorded in PDF documents filed in each court's docket. Utilization of these documents as data requires accurate and efficient information extraction methods. We take a supervised machine learning approach to a portion of this task, predicting metadata labels in court filings. On a dataset of about 2500 annotated scanned PDF images with 21 labels, we found that traditional classifiers such as MaxEnt achieved an average F1-score of 0.44 (micro-averaged across labels), with the highest label (Body) at 0.88. However, a 1-dimensional sequences in the text, Mallet's CRF implementation, achieved an average F1-score of 0.6 across all labels, with some labels as high as 0.91. These results demonstrate the value of using sequence models over traditional classifiers in labeling the types of information in court filings.

In: Proceedings of the Third Workshop on Automated Semantic Analysis of Information in Legal Text (ASAIL 2019), June 21, 2019. Montreal, QC, Canada.

1. INTRODUCTION

A court filing is a legal document submitted to a court that triggers an event in a legal proceeding. Court filings indicate what the case is about, why it should be in that court, and the grounds for the legal dispute. The legal effect of a filing depends critically on the event that it is intended to trigger (e.g., dismissal, answer to complaint, substitution of counsel), the role of the filer (plaintiff, defendant, court, intervener, etc.), the context of the filing (e.g., the previous filing, if any, that it is intended to respond to), whether it has been properly signed, and other document characteristics. Any process for automated analysis of court filings must determine the contents of these fields, which we refer to as "metadata", to distinguish it from the content of the body of a document.

A simple example of importance of automated metadata extraction is automated document quality control; that is, detection of discrepancies between the document metadata (such as the case number) and the metadata specified by the filer (e.g., the number of the case that the document was filed into). The shift to electronic filing systems, such as the US Federal Judiciary's CM/ECF system, by increasing numbers of courts means that filings are no longer inspected for errors by an intake clerk. Instead, this function is often performed by quality-control staff. Automating this process would free limited court resources for more productive purposes (Branting 2017).

However, automated extraction of document metadata requires identifying the type and location of the fields in the case caption and footer. This process could be assisted by machine transcription, but there are several challenges. For one, many documents are first printed on paper and then scanned into PDF form. Thus, a common format for these documents is an image, rather than plain text or XML. Moreover, recovering the layout of native PDF documents can itself be challenging, as described below.

There are tools available for image analysis, as well as for converting documents to plain text or XML, such as Apache Tika. But further challenges arise in how the information is laid out on the page. There is some structure in the layout of a court filing; the court is at the top of the page, with the parties below it, and the document number to the right of the parties. However, the actual physical position of this information can vary based on the amount of text and the conventions of the court. Many courts have small variations in how the information is presented, such as right-justifying vs. centering the court, or putting the document number at the top of the page.

Since there is no fixed location of information on each page, and rarely any indicative metadata, it becomes very difficult to automatically determine which piece of text is the court, the parties, and the document number. Additionally, things like stamps and signatures are often placed arbitrarily on the page, introducing noise in any image-to-text conversion.

When a document image is converted into XML via a conversion tool like Apache Tika, there are a number of features that can be taken from the new structure. In this paper, we attempt to assign a label to each word using both lexical and positional features. Positional features include the x and y position of each word, the quadrant of the page it is in, and the distance from other words around it. Lexical features include the word itself, the word case, the word type, and indicators of the word matching typical words in each type.

In our analysis, we find that positional features alone are not sufficient to classify most words, but reasonable performance can be obtained by including both lexical and positional features.

2. RELATED WORK

Several research communities have been active in document analysis, including historians, librarians, scientists, legal technologists, and those in government. Each community comes with a different set of data and goals, but all follow a similar processing framework.

There are several ways approach the information extraction from documents. One of the first tasks is separating the elements of the

page, a process called segmentation. (Mao, Rosenfeld, & Kanungo, 2001) distinguish between physical and logical layout segmentation. Physical segmentation includes identifying the lines, spaces, blocks, and other elements on the page. Logical segmentation seeks categorize these elements by their function (e.g. headers, footers, content trees). Methods of logical segmentation include rule-based approaches, comparison against knowledge-bases, and unsupervised learning.

More recently, researchers have approached the problem by converting the elements on the page into vectors and using supervised machine-learning models to classify the logical function of each element on the page. (Souafi-Bensafi, Parizeau, Lebourgeois, & Emptoz, 2001), for example, identified a hierarchy of geometric text blocks in various publications, and, along with typographical information, constructed a vector representation for each word. They then used a Bayesian network classifier to label the logical function of each word.

Standard classifiers, such as SVMs, Bayes nets, and random forests, can be considered 0-dimensional models, in that they only consider the features of each token, but not the sequence of tokens around it, Sequence learning algorithms, such as Conditional Random Fields (CRF), can be considered 1-dimensional classifiers, in that they consider the features of the elements before and after each token. (Trompper & Winkels) used a CRF model to classify header types in Dutch court documents from XML and found that CRFs outperformed a deterministic tagger.

Two-dimensional sequence learners can consider sequences of tokens in multiple directions and can thus exploit horizontal and vertical relationships between elements in documents. In '2D Conditional Random Fields for Web Information Extraction', (Zhu, Nle, Wen, Zhang, & Ma) successfully used a 2D CRF to classify sections of web pages.

In this paper, we focus on assigning logical labels to words in each court filing. We converted each scanned PDF into hierarchical OCR (XML) using Apache Tika and developed positional and linguistic features for each word token. We then compared 0-,1-, and 2-dimensional models to identify the relevant sections of the page.

3. APPROACH

In this work, a labeled dataset was constructed from scanned PDFs of court filings. This was done using an annotation tool called the MITRE Annotation Tool (MAT), developed by The MITRE Corporation. This tool contains resources for creating, maintaining and scoring annotated corpora of page images. The tool contains a set of annotation guidelines which we settled on after a number of rounds of pilot annotation. These guidelines focus on the first and last pages of court filings and legal letters. The annotator is asked to locate the major, non-nested sections of these pages (signatures, caption, court, body, etc.), as well as non-text stamps (such as received stamps), which are annotated for future reference. The annotation tool is Web-based and provides a graphical tool for identifying blocks and labeling them. In comparison mode, the tool can compare two annotators' efforts to each other.

The tool exploits a position-aware OCR output format known as hOCR, which presents each word along with its pixel-level location block on the page from which it was extracted. This position awareness allows us to score annotator blocks against each other, by determining which words are within each annotator block and how many of the words are in common between blocks. This allows the scorer to ignore slight variations in the actual x/y locations of the blocks and focus on how much content is in common.

Once the documents were annotated and converted into XML with labels, a toolchain was constructed to build models for automated inference of the textual (non-stamp) blocks given the hOCR output.

The fundamental problem with standard text-based approaches is that the text on these pages is not running text, but rather in blocks, so serializing the blocks in a standard line-oriented way may obscure the structure of the document and lead to problems applying standard structural techniques. Our hypothesis has been that using a graphical modeling inference strategy, allowing us to create much more structurally sophisticated contextual dependencies among elements, including 2-dimensional geometry, would enhance our ability to learn the location of these blocks.

Our strategy is an enhancement of the standard classification approach. Our goal has been to be able to compare multiple strategies to each other, including these strategies which build on these sophisticated contextual dependencies. Therefore, we've built a general-purpose experimentation harness for this family of classifiers.

First, from the hOCR output for a given page, the tool constructs a set of features for each token in the document. These features can be atomic features, string-valued features, or float-valued features. These features include:

- case features, related to capitalization pattern of the token
- digit and garbage features, related to the distribution of digits and non-alphabetic characters in the token
- word and ngram features, related to the character sequence of the token
- tag features, derived from applying the Stanford toolkit named entity tagger to the linearized text (these features are not likely to do much work for us, given the known problems with simply serializing this text line-by-line)
- similarity features which identify the best reasonably close match between the token and some of the metadata for the case for the document (e.g., the names of the parties or attorneys)
- 2-dimensional location features which indicate the position of the token on the page (what quadrant its in, and what percentage from the origin it is)
- margin features indicating words on the margin and whether they're indented

They can also be features on links between tokens, e.g., whether two tokens are farther apart than the average or median distance between tokens in the horizontal direction, or whether two tokens are more than one line apart in the vertical direction or indicating whether two tokens are on the same line.

This array of features, then, provides two levels of position sensitivity: first, on the token level, with the 2-dimensional location features, and second, with links between the tokens, for engines which recognize such features.

We explored three classes of algorithms:

- 0-dimensional token classifiers, represented by a maximum-entropy algorithm, implemented separately by the MALLET[1] engine, and by the Mandolin[2] engine.
- 1-dimensional linear CRF, also implemented with the MALLET and Mandolin engine.
- 2-dimensional CRF, where the dimension here refers not to geometric dimensions but abstract properties of the engine. Our goal, however, has been to use these properties to encode context dependencies in two dimensions. This was implemented only with Mandolin; a MALLET-equivalent (GRMM) implementation was attempted but unsuccessful.

Only the Mandolin engine explicitly represents links between tokens. We model our 2 geometric dimensions by computing unobstructed overlap between tokens in the vertical direction, as well as using line adjacency in the horizontal direction. Only the 2-dimensional model captures feature information in the vertical direction in our approach.

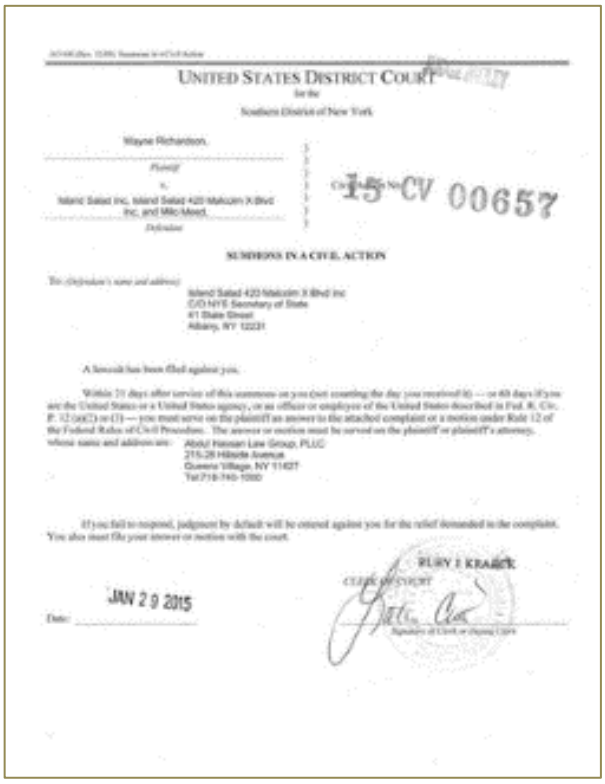

Figure 1: Examples of the varied structure of court filings

4. DATA

Our corpus consists of the first and last pages drawn from approximately 2500 court filings, PDFs typified by Figure 1, amounting to about 3500 annotated pages (some documents are only one page long).

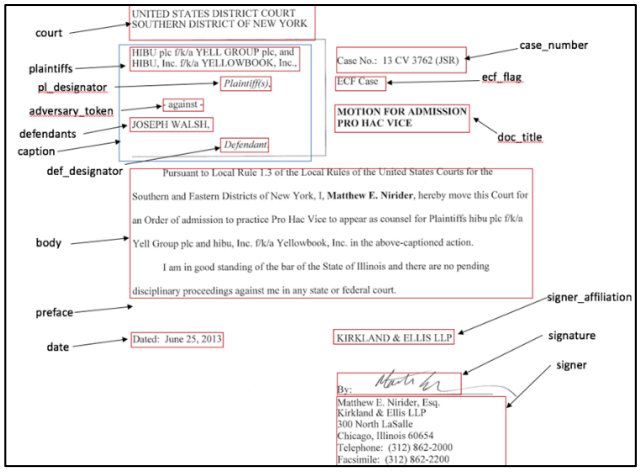

Figure 2: Examples of a case caption and footer with labeled fields.

Each court document contains the name of the court, the parties in the case, the case number, and the document title. Each word in the document is extracted, and positional and lexical features are determined from the words and their context. Several machine learning algorithms were then used to construct models to predict the labels based on the training data.

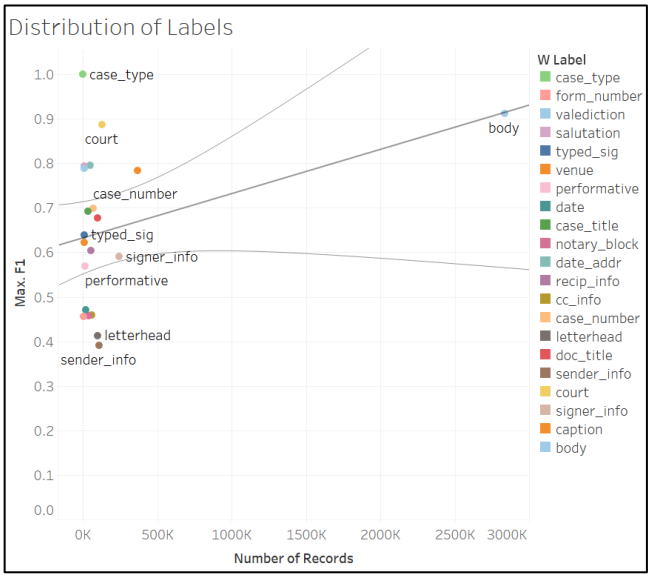

Figure 3: Number of occurrences of each type of word

The data was separated into batches, with each batch containing about 150 documents. Overall, about 22 batches were used for training, and 2 batches were used for testing. Within each batch, each document was divided into words, with features assigned to each word based on its positional and lexical elements. The number of words with each label vary, with the body containing the most words on average, and the caption a distant second, as illustrated in Figure 3.

[1] http://mallet.cs.umass.edu/

[2] http://project-mandolin.github.io/mandolin/index.html

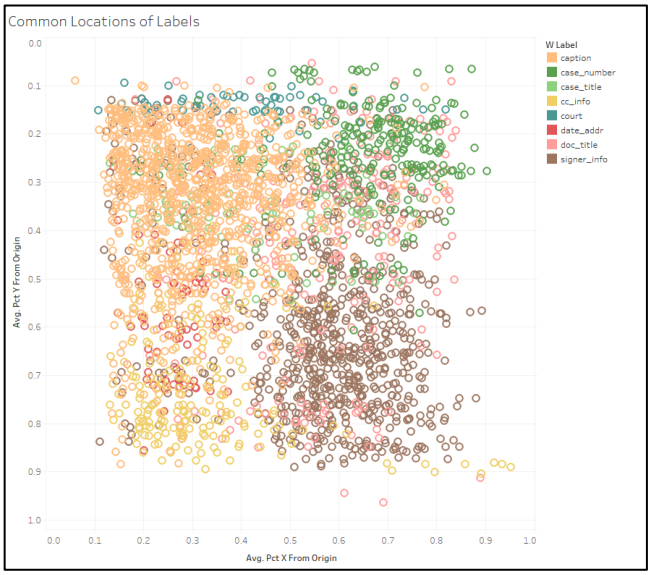

Figure 4: Sample locations of words, colored by label

Labels tend to occupy certain regions consistently, though their actual position can vary greatly. As an illustration of this, we plotted the X and Y coordinates of a sample of words, colored by label, in Figure 4.

5. FEATURE EXTRACTION

Around 25 features of the data were identified and extracted for each token, characterizing its positional, linguistic, and contextual information.

Using Weka's 'Information Gain' evaluator, the features were ranked according to the predictive value they provide. The most highly ranked features, pct_y_from_origin and pct_y_from_origin, represent the position of the token on the page. After that, entryType, stanford_lemma, and uncacheableAtomicFeatures deal with the linguistic properties of the data. Finally, otherWText and right_indent deal with the relative positional information of the token to other tokens in the text.

Each type of feature, positional, lexical, and contextual, help the model determine the role of the text on the page. While that doesn't necessarily mean that is the information humans use to make the determination, it is a relatively intuitive result: the position, type of word, and relation to the words around it, all indicate the function of each word in the text.

Info Gain	Type	Feature	Description
0.44	Pos	pct_y_from_origin	Vertical Distance from Origin
0.41	Pos	pct_x_from_origin	Horizontal Distance from Origin
0.41	Lex	entryType	Named Entity Type
0.41	Lex	otherWText	Word following token
0.40	Lex	stanford_lemma	Lemmatized token
0.37	Lex	uncacheableAtomicFeatures	n-grams of token text
0.22	Pos	right_indent	Indentation from Right Margin
0.16	Pos	v_half	Top or Bottom of Document
0.13	Lex	wText	Token Text
0.12	Lex	stanford_pos	Part-of-Speech

Figure 6: Information Gain metrics for top 10 features

6. EXPERIMENTS

Our hypothesis is that a combination of positional, lexical, and contextual information can be used to determine the function of each word on the page. To test this, a dataset of case metadata was developed, with features extracted about each token, including positional and lexical information. Each token was then assigned to the metadata label of the field where it occurred, and a machine learning algorithm was trained to predict the label based on the features of the token. This was treated as a multi-label training task in which the F_1 score was calculated separately for all tokens occurring in each documents field, i.e., for each label (document title, case caption, etc.). This may enable future researchers, who are interested in only a subset of the metadata, to get a baseline for the difficulty of extraction.

An ablation study was performed in which each model was evaluated with each type of feature, lexical or positional, present or absent in order to determine its relative contribution to classification accuracy (s. Finally, the predictive accuracies of several alternative predictive models were compared, including standard classifiers with 1D and 2D sequence models. The results of each of these experiments in terms of mean F_1-score across all labels is shown in Figures 7-9.

7. RESULTS

As Figure 7 shows, some metadata labels are reliably predictable using a combination of positional and lexical features. The degree of accuracy on some labels, as high as 90%, could be useful in many extraction tasks. Further, results were significantly improved by adding both positional and lexical features, and by using models that consider sequences, such as CRFs.

Label	F_1	Label	F_1
body	0.91	doc_title	0.68
court	0.9	case_type	0.67
date_addr	0.84	typed_sig	0.64
valediction	0.82	form_number	0.63
salutation	0.79	venue	0.62
caption	0.78	signer_info	0.59
recip_info	0.72	date	0.55
case_number	0.72	cc_info	0.46
performative	0.71	notary_block	0.46
case_title	0.71	letterhead	0.42

Figure 7: Top 20 metadata labels by max F_1 score

In particular, when the word value of a token (i.e., Token Text) was the sole feature, non-sequence models classified most tokens as 'Body,' with a small proportion tagged as 'Court'. This appears to be due to the fact that 'Court' words are a very small and specific set, including 'UNITED', 'STATES', 'DISTRICT', and 'COURT'. Curiously, adding positional features to a standard classifier did little to improve the results. However, when other lexical information was included, the F_1-measure increased greatly for most other labels. Including both lexical and positional features improves the results even more, as shown in Figure 8. This is consistent across each of the model types and shows that while the token's position on the page is important, the lexical properties of that token also play a significant role in identifying its label.

Lexical/Positional (F1)	**No Lex**	**Lex**
No Pos	0.23	0.089
Pos	0.45	0.52

Figure 8: Average model F_1 scores across all labels and models, with and without lexical and positional features

In comparing the types of classifiers, the CRF's outperformed standard classifiers in all cases. We used two different implementations of CRFs: Mallet CRF, and Mandolin CRF. We chose to compare Mallet to Mandolin because Mandolin could be used for standard classification, 1D and 2D analysis, while Mallet only included the standard classifier and 1D CRF. However, the Mallet CRF has been around for quite some time, and likely benefitted from significant tuning. Consistent with this surmise, Mallet outperforms Mandolin's 1D and 2D CRFs, as shown in Figure 9.

Models/Dimensions (F1)	**MaxEnt**	**CRF**	**2D CRF**
Mallet	0.27	0.44	
Mandolin	0.28	0.37	0.39

Figure 9: Average model F_1 scores across all labels and features, organized by algorithm and dimensionality

However, comparing Mandolin's 1D to its 2D CRFs, we see that most labels had an improved F-measure with the 2D. That leads us to believe that with further tuning, the 2D CRF could do quite well, but the Mallet 1D CRF had the best results overall in these experiments.

8. CONCLUSIONS

In these experiments, we found that the metadata labels (i.e., fields) of case captions and footers in US Federal court filings can be predicted using a combination of positional and lexical information. Accuracy was higher for much higher for some fields, such as body, case type, and court, than others, such as the sender and signer info, are harder to identify. The best performance was observed from the Mallet CRF, indicating that sequence-learning techniques perform better than 1D classifiers in the domain of court filings. While the utility of 2D sequence models has an intuitive appeal, we did not find that they increased accuracy over the 1D sequence model.

9. FUTURE WORK

There are several areas for improvement in this task. In general, some rigorous error analysis could be performed to identify major classes of errors. Further tuning of the models may also improve results, and additional training data may allow for other models such as Neural Nets. Additionally, a more mature 2D CRF implementation, such as GRMM might improve performance.

Finally, while initial work aims to label each word in a document, using these labels to predict the label of the 'block' that the text is in, is the longer-term objective. This would facilitate information extraction from the entire block.

10. ACKNOWLEDGMENTS

Thanks to Ben Wellner for providing the Mandolin models and support, and to Stacy Petersen, Grace Sullivan, and Ariana Kellogg for annotating the court filings. Special thanks to Sam Bayer for developing the training and testing framework.

11. REFERENCES

Apache Tika - a content analysis toolkit. https://tika.apache.org/.

Clausner, C., Pletschacher, S., & Antonacopoulos, A. (2014). Document Representation Refinement for Precise Region Description. DaTeCH. Madrid, Spain: ACM.

Eskenazi, S., Gomez, P., and Jean-Ogier, M., A comprehensive survey of mostly textual document segmentation algorithms since 2008, Pattern Recognition 64 (2017) 1-14.

Gabdulkhakova, A., & Hassan, T. (2012). Document Understanding of Graphical Content in Natively Digital PDF Documents. DocEng'12 (pp.137-140). Paris, France: ACM.

Klampfl, S., & Kern, R. (2015). Machine Learning Techniques for Automatically Extracting Contextual Information from Scientific Publications. SemWebEval 2015, 105-116.

Klampfl, S., Granitzer, M., Jack, K., & Kern, R. (2014). Unsupervised document structure analysis of digital scientific articles. Int.J.Digit.Libr.,14, 3-4 (August 2014), 83-99.

Konstas, I., & Lapate, M. (2013). Inducing Document Plans for Concept-to-text Generation. Proceedings of EMNLP 2013, 1503-1514. Seattle, Washington: Association for Computational Linguistics.

Lebourgeois, F. (1996). Localisation de textes dans un image a` niveaux. Colloque National sur l'Ecrit et le Document.

Mao, S., Rosenfeld, A., & Kanungo, T. (2003). Document Structure Analysis Algorithms: A Literature Survey. SPIE Electronic Imaging 5010:197-207.

Mencia, E. L. (2009). Segmentation of Legal Documents. ICAIL'09, 88-97. Barcelona, Spain: Association of Computing Machinery.

O'Gorman, L. (1993). The Document Spectrum for Page Layout Analysis. IEEE Trans. on Pat. Analysis and Machine Intelligence, Vol 15 No. 11.

Ramakrishnan, C., Patnia, A., Hovy, E., & Burns, G. A. (2012). Layout-aware text extraction from full-text pdf of scientific articles. Source Code for Biology and Medicine, 7:7.

Souafi-Bensafi, S., Parizeau, M., Lebourgeois, F., & Emptoz, H. (2001). Logical Labeling using Bayesian Networks. 6th Int. Conf. on Doc. Anal.

Trompper, M., & Winkels, R. (2016). Automatic Assignment of Section Structure to Texts of Dutch Court Judgements, JURIX 2016, 167-172.

Zhu, J., Nle, Z., Wen, J.-R., Zhang, B., & Ma, W.-Y. (2005). 2D Conditional Random Fields for Web Information Extraction. Proceedings of the 22nd International Conference on Machine Learning. Bonn, Germany.