Political Analysis (2025), **00**, 1–7
doi:10.1017/pan.2025.10006

LETTER

Detecting Formatted Text: Data Collection Using Computer Vision

Jonathan Colner[1,2]

[1]Center for Data Science, New York University, New York, NY, USA; [2]Center for Urban Research, City University of New York, Graduate Center, New York, NY, USA
E-mail: jonathanpcolner@gmail.com

(Received 11 July 2024; revised 17 January 2025; accepted 21 January 2025)

Abstract
Research in political science has begun to explore how to use large language and object detection models to analyze text and visual data. However, few studies have explored how to use these tools for data extraction. Instead, researchers interested in extracting text from poorly formatted sources typically rely on optical character recognition and regular expressions or extract each item by hand. This letter describes a workflow process for structured text extraction using free models and software. I discuss the type of data best suited to this method, its usefulness within political science, and the steps required to convert the text into a usable dataset. Finally, I demonstrate the method by extracting agenda items from city council meeting minutes. I find the method can accurately extract subsections of text from a document and requires only a few hand labeled documents to adequately train.

Keywords: computer vision; object detection; meeting records; local politics

Edited by: Daniel J. Hopkins and Brandon M. Stewart

1. Introduction

Foundation models, a subset of AI neural networks, have led to rapid innovations in a variety of industries. Political science research using these models has evolved down two paths. Large language models have been used on textual data for tasks such as sentiment analysis and topic modeling (Ornstein, Blasingame, and Truscott n.d.; Wang 2024). Alternatively, work with visual data studies media appearances with facial recognition (Girbau *et al.* 2024), and analyzes media depictions of political topics using visual frames (Torres 2024).

Nevertheless, political science has yet to benefit from the use of these models in the data collection process. While political scientists using these tools have dealt with data in readable-text format, there is a significant portion of textual data on poorly formatted PDFs that receive less attention. These documents commonly consist of nested subsections. Researchers are often interested in creating datasets of these subsections. However, there is no efficient way of extracting these subsections. Instead, political science research relies on optical character recognition (OCR) software. While OCR can extract complete texts, it doesn't help in differentiating between subsections of text. Therefore, scholars have relied on overly precise regular expressions. When regular expressions fail, researchers must extract the text by hand.

In this letter, I propose a workflow that simplifies the extraction of text subsections using advances in artificial intelligence. I discuss how to use computer vision models for structured text extraction using three software tools.[1] I walk through the use of Label Studio, a data labeling platform, to create a training set of annotated documents. Then, I use LayoutParser, a toolkit for document image analysis, to train an object detection model to identify visual formatting patterns. Finally, Tesseract OCR extracts the subsections of a document required for dataset generation.

To validate its accuracy, I use the method to extract agenda items from city council meeting records. Comparing the extracted agenda items to two ground-truth datasets, I find that the method is extremely accurate. Finally, I find that this method is over 30 times faster than the alternative method of hand-copying each agenda item.

By demonstrating how a simple workflow based on easily accessible tools can be a powerful data collection process, this research note encourages researchers to reconsider the various uses of object detection models. While previously reserved for those working with image data, this note demonstrates how these models can interact with textual data. By using object detection models with text data, we can better parse documents to create new datasets. Even as a simplistic demonstration of these tools, this project opens new doors for researchers struggling with difficult documents.

There are numerous areas within political science that could benefit from this method. Meetings of interest occur at the subnational, national, and international level. While textual data surrounding national legislatures have largely been processed, data collection at other levels of government is limited (Mortensen, Loftis, and Seeberg 2022; Shannon 2022). Numerous annual reviews have identified data availability as an impediment to research on subnational governments, while in 2020, researchers studying counties referred to them as "forgotten governments" (De Benedictis-Kessner and Warshaw 2020; Lim and Snyder 2021; Warshaw 2019). Beyond sub-national governments, corporate, intergovernmental organization, and union meetings are all meeting types that this method makes more accessible. Beyond the study of meetings, researchers extracting executive orders or working with transcripts could use this method (Jost *et al.* 2024).

2. Methodology

The goal of structured text extraction is to identify a set of visual layout principles that identify segments of text within a document. If these segments are obvious when looking at the document, then this method should accurately identify those segments. Once these visual layout principles are identified, trained object detection models identify similarly formatted segments of text. Finally, the model extracts the text from each segment as a separate row of data.

This method requires three steps, each of which relies on separate but accessible software. In the first step, a subset of the text corpus is converted into images of each page and hand annotated using Label Studio. Label Studio is an open-source data labeling platform (Tkachenko *et al.* 2020-2022). Images of each page are uploaded to Label Studio, and the researcher draws annotation boxes around the segments of text.[2] Finally, the data is exported in the Common Object in Context (COCO) format.

The second step uses a Fast R-CNN R50-FPN object detection model, a variation of the regions within convolutional neural networks model (R-CNN) (Wu *et al.* 2019). An R-CNN model takes an input image and proposes a large number of potential object regions of different sizes and aspect ratios. For each region, thousands of features are extracted to determine the object within that region. While these general models are focused on images, here I fine tune the model to look for patterns of pixels that indicate the start and end of a text segment. To fine tune the model I split the annotated data 70–30 into a training and validation set. The validation set is used to evaluate the model performance.[3]

Finally, I use the fine-tuned model to draw boxes around similarly formatted subsections on the remaining pages of text. Once identified, LayoutParser, a python toolkit for deep-learning-based

[1] The steps described here are adapted from a presentation by Label Studio (Label Studio 2022) on extracting citations from scholarly documents.

[2] For items that extend across pages, append pages together into single taller images so that the items can be captured in full.

[3] Find more details on this step in Appendix B of the Supplementary Material.

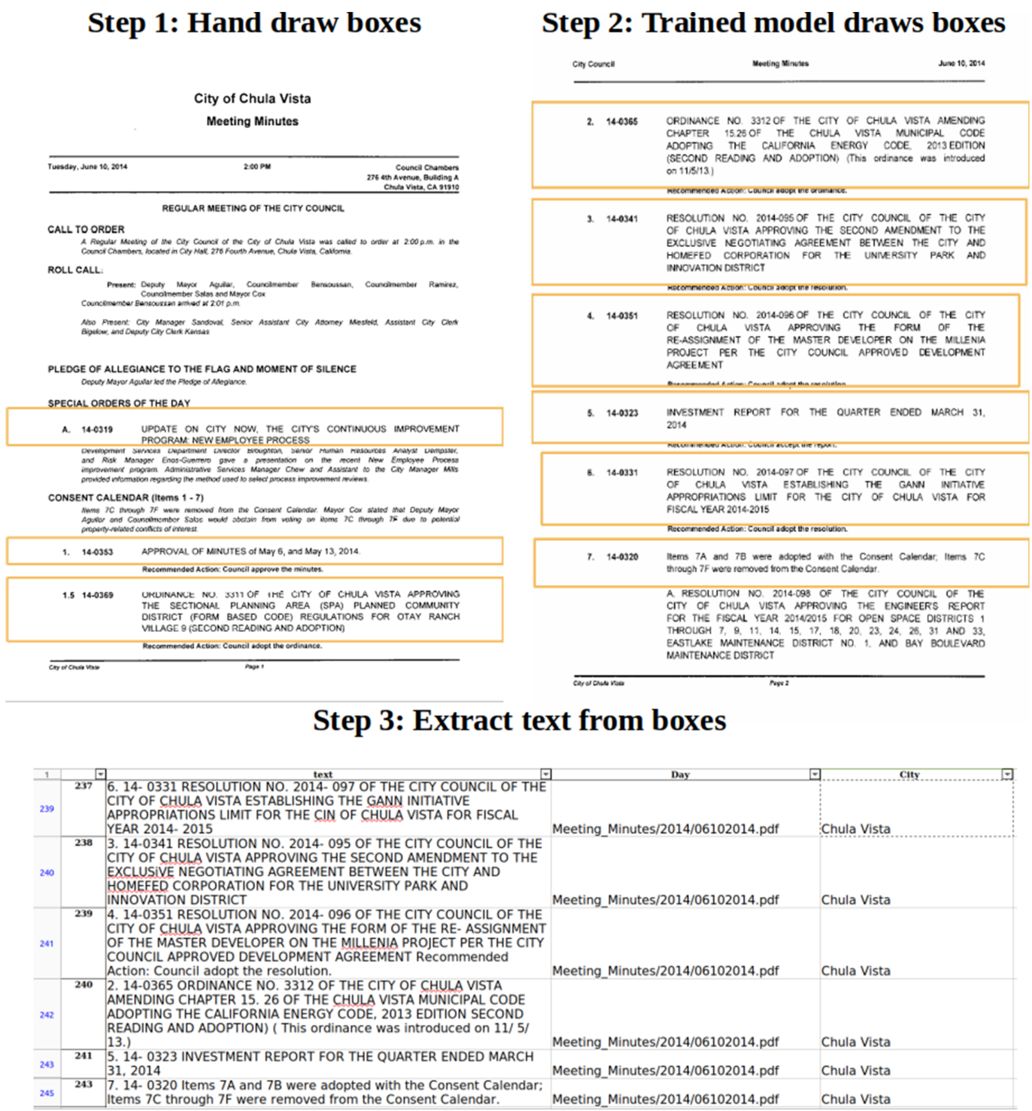

Figure 1. Shows the workflow used to identify and extract agenda items from city council meeting records. In step 1, I use Label Studio to annotate a training set of meeting minutes specifying segments of the page with agenda items. In step 2, I train an object detection model that identifies segments on a different page that matches the formatting. In step 3, I use LayoutParser to extract text from those segments with OCR. In this example, both pages are from the city of Chula Vista's June 10, 2014 meeting.

document image analysis, extracts the text within those boxes using Tesseract OCR, a free OCR engine (Shen *et al.* 2021). In Figure 1, I show a visual representation of this workflow using Chula Vista's June 10, 2014 meeting records.

3. Examining Municipal Meeting Records

To demonstrate this method, I focus on the extraction of agenda items from city council meeting records. Agenda items offer an ideal test case for several reasons. First, agenda items make up a subsection of a meeting record, but contain the most important information. Additionally, while agenda items are easily identifiable to the human eye due to being indented, starting with a number, or some

https://doi.org/10.1017/pan.2025.10006

combination of formatting patterns, the text is not so structured that regular expressions would be effective.

While the process could be applied to any city, I focus on five cities in California: Chula Vista, South San Francisco, Visalia, Santa Rosa, and Temecula. These cities were chosen due to the existence of a ground-truth dataset to compare my method to. Because this method requires a consistent formatting pattern, the process is done separately for each city. For each city, two meeting records per year were selected as the training set. Each page was converted into an image, uploaded to Label Studio, and annotated with boxes around each agenda item. Then, the Fast R-CNN model was trained using the training set and evaluated using the evaluation set. Finally, each model was used to collect agenda items for the remaining meeting records. Before checking the accuracy of the model output, I evaluate the model's performance as measured by its average precision. For each city, the model's average precision is similar to the scores reached by general, state-of-the-art object detection models.[4]

Given there are no benefits to hand-annotating more than two meetings per year,[5] we can use two meetings as a benchmark to compare the time this method takes to the hand-collected alternative. Overall, it takes approximately an hour to carry out this method for one city. Hand coding the agenda items into an excel sheet would take approximately 31 hours.[6] Thus, this method offers significant time savings.

4. Validating Method Performance and Accuracy

I assess the accuracy of this method by comparing the agenda items identified to two separate ground-truth datasets. The first comes from Legistar, a legislative management software.[7] The second is a random

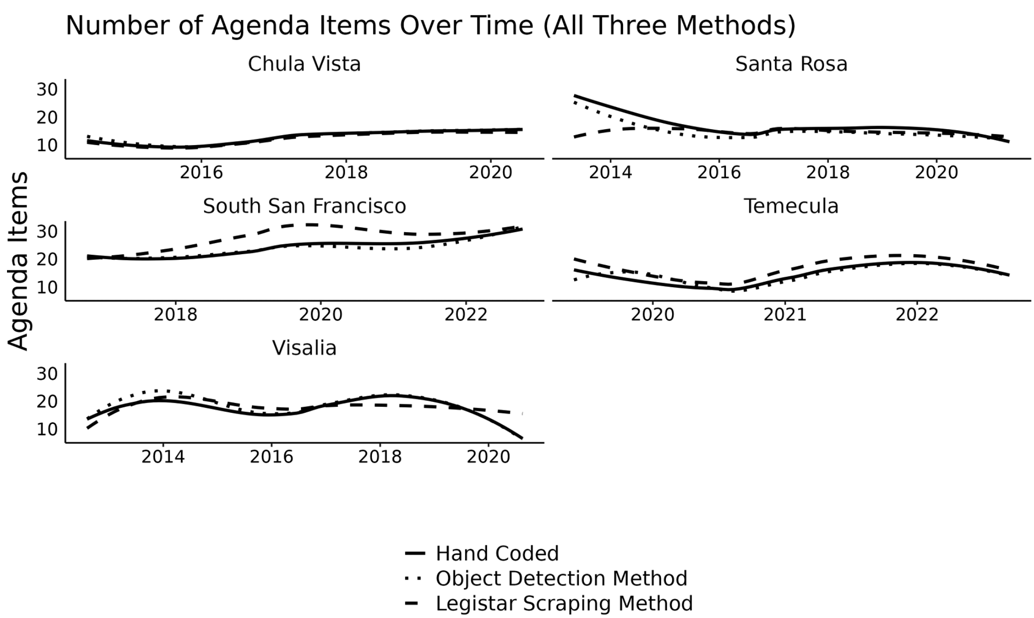

Figure 2. Compares the number of agenda items identified in meeting minutes to the number of agenda items listed on Legistar for that same date to the number of items identified by hand coding agenda items from the meeting records.

[4]Find a discussion of average precision in Appendix A of the Supplementary Material.
[5]A discussion on the number of meetings to annotate is in Appendix D of the Supplementary Material.
[6]Find in Appendix E of the Supplementary Material a description of how these times were calculated.
[7]In Appendix F of the Supplementary Material, I briefly discuss Legistar, the information it has on each agenda item, and how the data was collected.

https://doi.org/10.1017/pan.2025.10006

Table 1. Similarity between matched agenda items.

	Precision	Jaccard similarity	Cosine similarity	Levenshtein distance
Chula Vista	0.94	0.81	0.96	0.74
Temecula	0.82	0.91	0.99	0.89
South San Francisco	0.98	0.90	0.96	0.84
Visalia	0.78	0.97	1.00	0.96
Santa Rosa	0.77	0.81	0.91	0.64

The precision score is a measure of the number of matched agenda items divided by the total number of agenda items identified using my method. Using only the matched agenda items, I then calculate the Levenshtein distance, Jaccard similarity, and Cosine similarity scores between the texts of those matched items.

sample of 20 meetings from each city that I hand-extract the agenda items from. I use these ground-truth datasets to assess the method on several measures of accuracy.[8]

In Figure 2, I show the number of agenda items identified by my method and the two ground-truth datasets over time. The overall count of agenda items over time across the methods are closely matched, though my method more closely tracks the number of hand-coded agenda items. Next, I examine the lexical similarity between matched pairs of agenda items from my method and from Legistar. As shown in Table 1, the lexical similarity between the pairs of agenda items is high regardless of the metric.

Across each validation of my method, I find that my method both accurately identifies the segments of interest on the page of text and effectively captures the text within that segment.

5. When to Use This Method

This method may not be useful for all researchers. Specifically, a researcher should determine whether the data meets four conditions. First, the text must be in formatted document form rather than already processed into a dataset. For example, research looking at the topics discussed in state and national legislatures or focusing on national news coverage would not need this method, as the text has already been processed (Quinn *et al.* 2010; Young and Soroka 2012).

Second, this method will not be useful if interested in analyzing the full text of a document. Research interested in extracting the sentiment or policy focus at the document level, for example, will want the full text of the document (Crow, Albright, and Koebele 2020; Grimmer 2010). Instead, this method is for researchers interested in extracting individual subsections of text.

Third, the researcher should consider what distinguishes the subsections of interest. If researchers are interested in pieces of information that are distinguishable by the text content, this method will not be useful. This would include extracting individual names from a text, or breaking the text into two-sentence segments (Incerti 2024; Merz, Regel, and Lewandowski 2016). Because the method is not focused on the text itself, the language used in the document should have no impact on the training of the object detection model. Similarly, the method should work for tables or other unique formatting structures.

Finally, the researcher should consider how many subsections are extracted from each page, how long each document is, and how many documents are being studied. A 2012 paper analyzing U.S. treaties with American Indians is a good example; the number of documents being analyzed is under 600 and the number of sections in each document is one (Spirling 2012). Given the low number of total subsections to be extracted, hand coding the data is likely a better approach.

In this research note, I demonstrate how to use Label Studio, LayoutParser, and Tesseract-OCR to carry out structured text extraction. This method is ideal for difficult records that contain text segments of interest formatted in a visually distinct way from other text in the document but not capturable using

[8] Additional details on how these measures are calculated can be found in Appendix G of the Supplementary Material.

regular expressions. Using the extraction of agenda items from meeting minutes as a test case, I show that the method is both accurate and quicker than hand-coding. This letter takes one of the first steps to show how we can use available tools to collect segments of similarly formatted text from documents when collecting the data by hand would be infeasible.

Acknowledgments. The author thanks Christopher Hare, Scott MacKenzie, Ryan Hübert, Hanno Hilbig, and Sam Fuller for their helpful comments and feedback during the preparation of this draft.

Author Contributions. J.C.: Data Curation, Funding Acquisition, Methodology, Validation, Visualization, and Writing.

Funding Statement. This material is based upon work supported by the National Science Foundation SBE Postdoctoral Research Fellowship under Grant No. 2403505.

Competing Interests. The author declares no competing interests exist.

Ethical Standards. Not applicable.

Author Biographies. Jonathan Colner, PhD is an Assistant Professor of Data Science/Faculty Fellow at New York University's Center for Data Science and a visiting scholar at City University of New York, Graduate Center's Center for Urban Research. He received his PhD in Political Science at the University of California, Davis in 2024. His research focus is in the area of local politics, with a special interest in municipal records and electoral institutions.

Data Availability Statement. Replication code for this article is available at Colner (2025). A preservation copy of the same code and data can also be accessed via Dataverse at https://doi.org/10.7910/DVN/8BE6M9.

Supplementary Material. The supplementary material for this article can be found at https://doi.org/10.1017/pan.2025.10006.

References

Colner, J. 2025. "Replication Data for: Detecting Formatted Text: Data Collection Using Computer Vision." Harvard Dataverse. https://doi.org/doi:10.7910/DVN/8BE6M9.

Crow, D. A., E. A. Albright, and E. Koebele. 2020. "Evaluating Stakeholder Participation and Influence on State-Level Rulemaking." *Policy Studies Journal* 48 (4): 953–981.

De Benedictis-Kessner, J., and C. Warshaw. 2020. "Politics in Forgotten Governments: The Partisan Composition of County Legislatures and County Fiscal Policies." *The Journal of Politics* 82 (2): 460–475.

Girbau, A., T. Kobayashi, B. Renoust, Y. Matsui, and S. Satoh. 2024. "Face Detection, Tracking, and Classification from Large-Scale News Archives for Analysis of Key Political Figures." *Political Analysis* 32 (2): 221–39.

Grimmer, J. 2010. "A Bayesian Hierarchical Topic Model for Political Texts: Measuring Expressed Agendas in Senate Press Releases." *Political Analysis* 18 (1): 1–35.

Incerti, T. 2024. "Countering Capture in Local Politics: Evidence from Eight Field Experiments." *The Journal of Politics* 86 (4): 1603–1607.

Jost, T., J. D. Kertzer, E. Min, and R. Schub. 2024. "Advisers and Aggregation in Foreign Policy Decision Making." *International Organization* 78 (1): 1–37.

Shen, S, B. Lee, and M. Malyuk. "Customized Layout Detection for Scientific PDFs with LayoutParser and Label Studio." partner webinars, streamed live on February 9, 2022. https://labelstud.io/videos/customized-layout-detection-for-scientific-pdfs-with-layoutparser-and-label-studio/

Lim, C. S. H., and J. M. Snyder. 2021. "What Shapes the Quality and Behavior of Government Officials? Institutional Variation in Selection and Retention Methods." *Annual Review of Economics* 13: 87–109.

Merz, N., S. Regel, and J. Lewandowski. 2016. "The Manifesto Corpus: A New Resource for Research on Political Parties and Quantitative Text Analysis." *Research & Politics* 3 (2): 2053168016643346.

Mortensen, H. B., M. W. Loftis, and H. B. Seeberg. 2022. "Explaining Local Policy Agendas: Institutions, Problems, Elections and Actors." In *Comparative Studies of Political Agendas*, edited by C. Green-Pederson, L. C. Bonafont, A. Tiommermans, F. Varone, and F. R. Baumgartner. Cham: Springer International Publishing.

Ornstein, J. T., E. N. Blasingame, and J. S. Truscott. 2025. "How to Train Your Stochastic Parrot: Large Language Models for Political Texts." *Political Science Research and Methods* 13 (2): 264–81.

Quinn, K. M., B. L. Monroe, M. Colaresi, M. H. Crespin, and D. R. Radev. 2010. "How to Analyze Political Attention with Minimal Assumptions and Costs." *American Journal of Political Science* 54 (1): 209–228.

Shannon, B. N. 2022. "Can Institutional Reform Have a Lasting Impact on the Policy Agenda? Evidence From the 10-1 in Austin, TX." *Urban Affairs Review* 58 (6): 1689–1718.

Shen, Z., R. Zhang, M. Dell, B. C. G. Lee, J. Carlson, W. Li. 2021. "LayoutParser: A Unified Toolkit for Deep Learning Based Document Image Analysis." https://layout-parser.github.io/.

Spirling, A. 2012. "U.S. Treaty Making with American Indians: Institutional Change and Relative Power, 1784–1911." *American Journal of Political Science* 56 (1): 84–97.

Tkachenko, M., M. Malyuk, A. Holmanyuk, and N. Liubimov. 2020. "Label Studio: Data labeling software." https://github.com/HumanSignal/label-studio.

Torres, M. 2024. "A Framework for the Unsupervised and Semi-Supervised Analysis of Visual Frames." *Political Analysis* 32 (2): 199–220.

Wang, Y. 2024. "On Finetuning Large Language Models." *Political Analysis* 32 (3): 379–83.

Warshaw, C. 2019. "Local Elections and Representation in the United States." *Annual Review of Political Science* 22: 461–479.

Wu, Y., A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. 2019. "Detectron2." https://github.com/facebookresearch/detectron2.

Young, L., and S. Soroka. 2012. "Affective News: The Automated Coding of Sentiment in Political Texts." *Political Communication* 29 (2): 205–231.

Cite this article: Colner, J. (2025). Detecting Formatted Text: Data Collection Using Computer Vision. *Political Analysis*, 1–7. https://doi.org/10.1017/pan.2025.10006