Content Insights & Discovery Accelerator (CIDA)
The Atlantic Article Segmentation

Introduction

Content Insights & Discovery Accelerator (CIDA) grew out of a need for newsrooms to analyze massive reams of data, from investigative reports examined under deadline pressure to story archives for research and republishing. An intelligent search solution, CIDA employs Microsoft Azure cloud-based cognitive search, object vision and video indexing. The technology goes leagues beyond manual tagging and allows users to "peek" inside multiple content types—print, video or audio clips—to enable users to find exactly what they are searching for.

CIDA deploys artificial intelligence (AI) to track and understand the big-data overload of our own making: By 2020, human beings will create 44 zettabytes of data every day.[1] In other words, thanks to traditional and digital publishing, social media conversations and Internet of Things (IoT) analyses, we've far outstripped our ability to process the information we create.

Solving problems in the modern age requires building custom algorithms to target specific problems. The walking, talking, calculating robots of science fiction are still very long ways away. Consider Google's AlphaGo[2]: The bot can beat masters at the Chinese strategy board game but not checkers nor chess champions. This same principle exists with self-driving cars, which require unique algorithms to recognize street signs, determine proximity to other vehicles and the million other variables that make up driving. The autonomous system combines these inputs but can't necessarily extrapolate a solution to a new problem.

Created through a partnership with Unify and Microsoft News Labs, CIDA faced a similar challenge with analyzing the library of *The Atlantic* magazine, founded in 1857. Its library of literary and cultural commentaries existed largely as print. The media organization wanted to reignite works of the past for research and republication. For both archiving and republishing purposes, however, CIDA had to take into account different typefaces and identify page segments as metadata. As *The Atlantic* underwent generational changes in print media, CIDA needed to integrate separate algorithms for various layouts.

For instance, page segmentation requires understanding and identifying distinct segments on a document page, extract those segments (e.g. text, illustrations, photographs) and categorize them appropriately. In the case of *The Atlantic,* those segments included title/headline, author, columns, quotes and so forth.

[1] "A Day in Data," Technology: Future of Data 2019, Raconteur
[2] 'AlphaGo: Mastering the ancient game of Go with Machine Learning,' Google AI Blog, Jan. 27, 2016

Training a model to recognize these segments not only the varying publication styles but also the different types of pages within *The Atlantic* proved a complex task.

The core issue in using custom AI is that it is by design inherently narrow; specialization rigidly binds what the algorithm can identify. Moreover, how an algorithm "learns" differs from the way humans learn and incorporate new information. A human applies prior knowledge differently than the way knowledge is transferred from one algorithm to the next. Understanding this difference is still an open problem in AI research today. While we can apply the learning algorithm to different layouts, the model still needs to be retrained with each new data change.

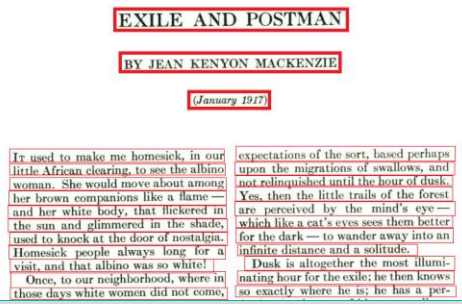
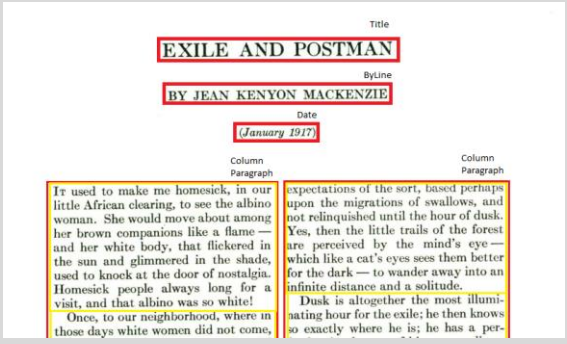
To solve *The Atlantic*'s document segmentation problem with high accuracy scores, CIDA split the problem into multiple models and built a higher degree of accuracy by training them on more consistent, smaller, generational subset of its archives.

Document layout analysis

Humans and machines "see" images differently. Whereas human beings can call upon a lifetime of experience and knowledge, a machine is limited not only to data provided in training but also the structural limitations of the underlying algorithm.

Consider teaching a child to identify a cat. A child may begin by referring to any random object as a cat, then refine the identification to four-legged creatures such as dogs and horses, before truly understanding what a cat looks like and more important, understand what a cat is. Those segregation tasks and processes are reinforced in that child as he or she grows and learns, transferring this knowledge to other problems.

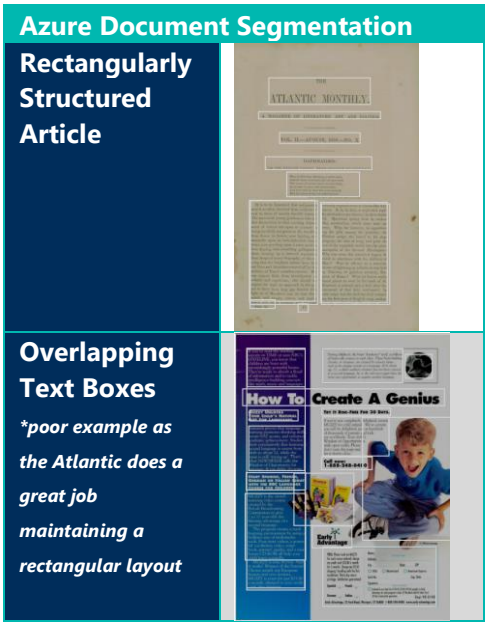
Similarly, we train a machine learning algorithm to identify specific segments by providing labeled consistent data. At the end of training, we have an algorithm that can identify a cat with one significant difference to the human learning model—the algorithm has no concept of what a cat is. It should be noted that while humans can take a few pieces of data to create a set of baseline assumptions, AI requires thousands of examples to begin to reach the level of sophistication that one would expect from a human labeler.

Labeling Method	
Line by Line: **Pros: Easy to implement** **Cons: Doesn't combine segments into contiguous segments**	
Segmentation Classes: **Pros: Class labels** **Cons: Lower Scoring metrics due to intricate labeling strategies**	

For the document problem, the goal was to identify constituent segments consistently across a large set of documents. A logical structure of a document image is a mapping from the regions in the document to their class labels; for example, identifying and labeling a header, a title, a by-line, or a page-number. State-of-the-art solutions in document layout analysis currently favors deep learning approaches and computer vision algorithms. In using these techniques, the question of the label level and hierarchy becomes imperative; a poorly specified or suboptimal labeling pattern or algorithm selection will lead to lost accuracy and model recall. This can be due to insufficient examples in the training data or not being able to distinguish one labeled segment from another similarly labeled segment on the page.

While one part of the solution involved building a model to detect the segments of the document image, an equally important part was analyzing the document images to build a consistent, logical and identifiable label set.

To solve the object detection model, CIDA leveraged Microsoft Azure's Cognitive Service Custom Vision client. The labels was selected as a combination of *The Atlantic*'s wish list of extracted segments and empirical testing with respect to a set of accuracy metrics. To increase the accuracy of document segments, complex solution entailed classifying articles by generational layouts, then building custom algorithms for each generational subset.



Using Microsoft Azure's Custom Vision client, we defined bounding boxes for document segments to concatenate them across multiple pages. Using the Custom Vision application, we then segmented the document using these bounding boxes, which worked well for rectangularly structured content. However, in alternative document structures outside of articles or other standardized document formats, the Custom Vision client was less significant. Instead of defining bounding boxes, a more adaptive algorithm which allows non-rectangle bounds, known as masks, would be more appropriate.

Considerations in Applying AI

When applying an AI solution to a problem, issues range from identifying version of the research question that can be solved by applying a machine learning model, to picking a sample size and identifying class labels..We Each step requires evaluating trade-offs between comprehensively solving the stated business question and returning a result that can be interpreted by the end user. This is due in a large part to using a black box model, with which we lose the ability to interpret the effect features have on the outcome due to the complex interactions of deep learning models. The AI that supports CIDA strikes a balance between being able to explain the results and the model with high performance in application.

The questions commonly asked when building a data science solution are: "Do we have enough data?", "Do we have the right data?" and "How much data do I need to achieve a specified accuracy score?"

Let's talk about how much data we need. There is not a straight answer to this question, but it comes down to how specific the model needs to be at distinguishing between similar items. Harder-to-distinguish labels require significantly more data than more differentiable labels; coupled with this is that

the model is learning how to identify these objects relative to the images given for training. In an object detection problem, we don't have a way to say, 'focus on this object within a bounding box and ignore the rest,' as there could be a consistent pattern within a label box that may give rise to false correlations. A model could then learn the wrong set of traits to describe an object.

A classic example of this challenge was a model trained to identify dogs vs. cats. The training images for dogs had been taken outdoors whereas the cat images had been taken indoors. As a result, when the model was validated on images of cats on grass, it consistently identified the felines as canines. As one might expect, the reverse was true: images of dogs indoors identified them as cats. Thus, an important consideration beyond just dataset size is how representative is your data.

A general guideline in computer vision is that if a human cannot easily distinguish between classes, a machine will likely have a difficult time and require significantly more examples to learn distinguishing characteristics. When selecting a sample size in a deep learning application such as this document segmentation and segment extraction, we must consider distinguishing the class labels at extreme granularity. For instance, what distinguishes a paragraph of text in an article as opposed to a paragraph of text within an advertisement? How many example images are needed to give the model to successfully and consistently differentiate between these different types of text?

Azure Custom Vision likes to see a minimum of 50 examples for each label as a soft threshold; however, as the labeling gets more complex, the minimum count comes down to empirical testing. In the interests of time, many researchers build an initial database, test the model performance, add more example images, retrain and retest, and continue this iterative threshold until the desired performance metric threshold is met. Our work on CIDA showed that we can attain relatively high accuracy, in the upper 80 percent, with approximately 500 example documents; however, the model's recall performance metric dropped significantly as we added a more diverse set of eras into the training dataset.

The second question, "Do I have the right data?" is a challenge for any specific narrow AI task. When training a deep learning model, the algorithm learns the features of the given dataset. The model then typically performs very well on data that is exceedingly like the data used in the training set. However, when applied to data or problem spaces that differ drastically, those models perform terribly. Thus, when training a model for solving narrow AI problems, a prime consideration is the specificity of a training dataset. If a very specific dataset is given to the algorithm to train, the model will typically perform very well on that data; if a more general dataset is provided to the algorithm to train, the model will typically drop in performance but perform better when the problem domain crosses into those more general spaces. This is a tradeoff that must be considered when training a model.

Confusion Matrix

		Predicted	
		Negative	**Positive**
Actual	**Negative**	True Negative	False Positive
	Positive	False Negative	True Positive

This leads us to the third question: "How much data is needed to have to attain some arbitrary level of performance for the model under some metric?" A common metric is "accuracy." Precision is a score that measures how often, when the model identifies a label, that the stated label is correct. This may seem like a great metric for which to

optimize; however, we can set the threshold for how confident the model needs to be to return a label. In practice, what this means is that as we increase the precision threshold, the model's recall begins to drop. Recall is a metric that identifies how likely the algorithm is to identify all the possible labels in an inputted file. Therefore, as we optimize the algorithm for precision, we may be unintentionally reducing the model's ability to recall all the labels on a document. One way to address that issue is to increase the sample size, or more specifically increase the examples for either minority labels or labels that aren't discriminating well between each other.

Metrics	
Precision: Measure of Exactness	$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$
Recall: Measure of Completeness	$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$

The Atlantic Case Study

Working with *The Atlantic* provided an opportunity to explore the complexities of building a robust document segment algorithm on a diverse set of similar documents. Through its 150+ year existence covering literature, arts and politics, *The Atlantic* has adapted to changing audiences as well as the changes in the print medium itself. While the mission has remained constant, the layout has evolved—the crux of the document segmentation we are working on solving today.

We first collected sample magazine issues from *The Atlantic*, compiled a wish list of features to be extracted, then analyze generational changes within the archives. CIDA labeled approximately 500 article images, randomly sampled from each generational age to determine the preliminary discriminating characteristics of the selected document segment labels. The documents and bounding box labels were then uploaded to Azure's Custom Vision application and a model was trained to identify document segments. The results of that segmentation procedure were integrated into the optical character recognition (OCR) process to improve the faceted search features of CIDA and accurately join articles together.

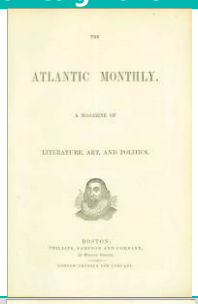
Phase One: Exploratory Analysis

The Atlantic granted CIDA access to its EBSCO database which contains a copy of every *The Atlantic* issue printed from 1857 to 2014. Structural differences throughout the storied history of *The Atlantic* created challenges for document segmentation and segment identification. Based on an empirical examination of *The Atlantic* database over time, we denoted three major shifts in the layout of *The Atlantic*: 1857-1906, 1907-1948, 1949-2014. While there were shifts within each "era," there were overall distinctive and significant layout changes that negatively affected the performance of the models when we include data across these generations.

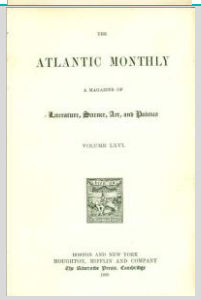
Layout Generations	
Early	1857-1906
Mid	1907-1948
Modern	1949-2014

Atlantic Design Over Time

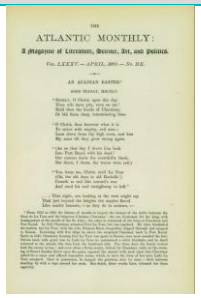
1857



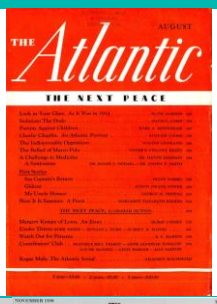
1890



1900

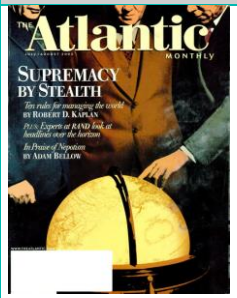


1939



1990

2003



In the 1800s, The Atlantic primarily used a 2-column layout. By the 1900s, the magazine switched to a consistent 3-column layout that in turn is further split by more artifacts, images, and symbols.

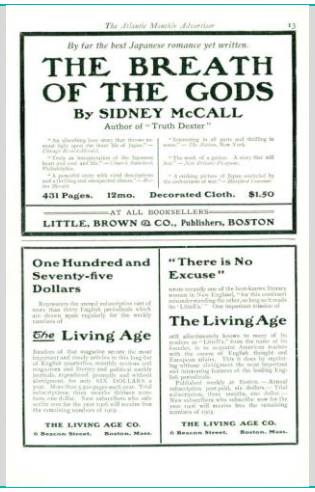
Advertisements did not appear within *The Atlantic* until the 1900s. Full-page ads of the early 1900s and embedded article-page ads beginning in the mid-1900s introduced complex layout changes that required different models.

Evolution of Atlantic Advertisements

Late 1800's

Mid 1900's

Early 1900's



Early 2000's

Phase Two: *The Atlantic* Labeling Strategy

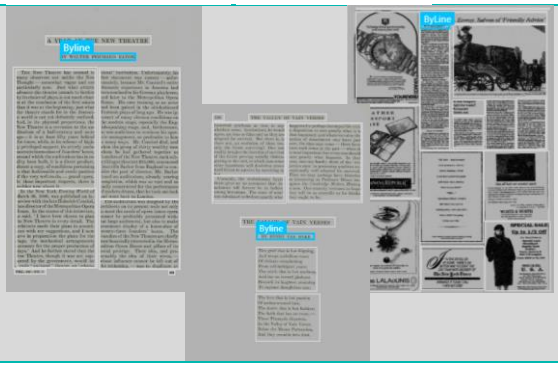
The Atlantic's archive team generated a prioritized wish list of attributes to be extracted from the uploaded documents. These segments included "Author," "Topic," "Related Topics," "Keywords," "Date," "Volume/Issue," "Headline," "Byline," "first paragraph," "articles," "Image Extraction," "Image Subject," "Image Topic," "Image Credit," "Issue Collection" and "Issue Topic" as defined by cover page. All these individual segments further required different strategies and machine learning algorithms.

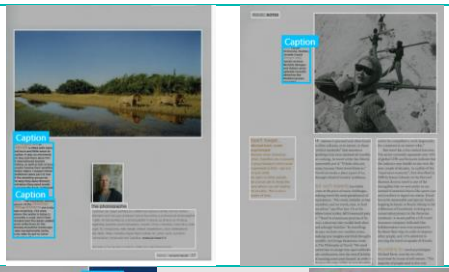
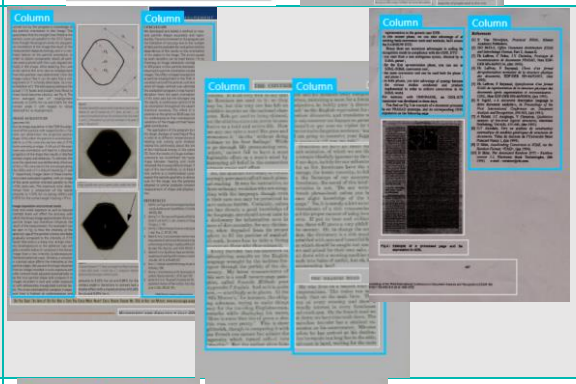
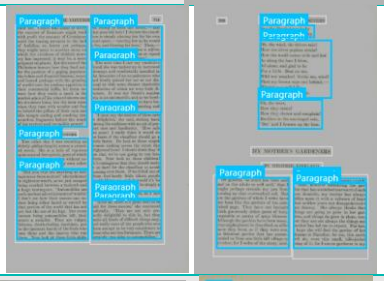
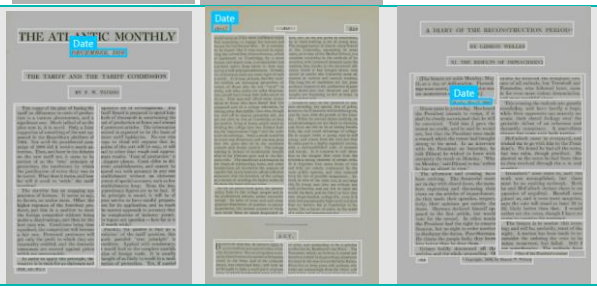
CIDA focused on segments that could be extracted using a computer vision algorithm to label prominent document segments. This direction demonstrated the technology available through Azure and illustrated the power of this preliminary segment extraction.

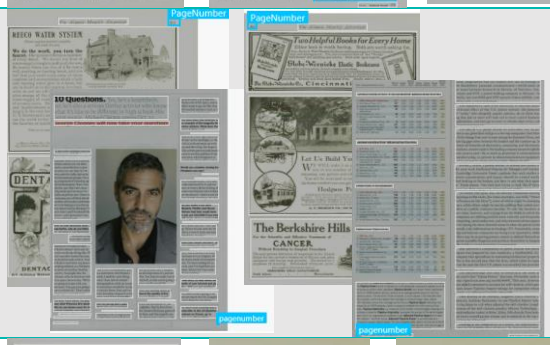
For phase two of *The Atlantic* Segmentation CIDA identified the following document segments to assist with the OCR processing:

- Advertisement
- Byline
- Caption
- Column
- Paragraph
- Date
- Float Text
- Header
- Image
- Page Number
- Section Header
- Title

The resulting labeling strategy was a compromise between the wish list presented by *The* Atlantic and the time, data, and resources allotted for presenting an acceptable solution. A primary consideration when selecting labels was: What value does extracting this as a separate segment have in improving the searchability or research potential of text within the identified label? This consideration needs to be balanced with the data's ability to support finding enough examples of minority labels to train the algorithm.

***The Atlantic* Labels**		
Advertisement	Bounds should incorporate each full advertisement separately	
Byline	Bounds should contain only the author's name and any associated reference text. For example, "by Jane Doe" should all be incorporated within the box, not just "Jane Doe"	

Caption	Bounds should include photo or segment captions	
Column	Any column of text that is structured as a continuous set of the same article. This contrasts with float text in that it is interruptions or non-standard columns	
Paragraph	Paragraph of text that is a sub-label of a column	
Date	Any date that identifies the date of the article, issue, or page.	
Float Text	Any text that interrupts the flow of a column or exists external to the primary text of the article. Accounts for non-sequitur sections.	

Header	The header text.	
Image	Any image as part of an article that is not an advertisement.	
Page Number	Any page number on page that is not part of a contiguous header or footer.	
Section Header	Section breaks in columns or as breaks within the same article.	
Title	Title of an article.	

These selected labels were only a start to the analytics process for extracting more precise features and were generated based upon visual data. These labels contributed to piecing together articles for republication but didn't fully automate the extraction of key facets or research points. Locations of where

a segment was placed provided additional information when extracting. Additional natural language processing techniques augmented the index and allowed more precise extractions. The primary focus of this model set however was to identify document structure and accelerate improvements in the OCR process.

Phase Three: Model Building and Scoring

While CIDA was still conducting experiments and calibrating parameters for building successful models, it was decided based upon related research on document segmentation to take an empirical approach to scoring the model. Using the label definitions above, the team aimed to label approximately 500 pages from each of the three generations of layouts as a basis to understand not only initial model performance but also label distributions, since some label features occurred significantly more often than others. In building a model for each of the first two generations of *The* Atlantic's articles, the CIDA team labeled over 3000 images as training data.

Scoring 1857-1906	
Precision	93%
Recall	70%
Threshold	50%
Sample Size	1290
Train Time	18 Hours

Current training on the first two generations of *The* Atlantic's magazine archive show that we can achieve a precision score of at least 90% on each of the generations. However, the model is only able to recall about 70% of the labeled inputs which is a figure that leaves room for improvement.

Scoring 1907-1948	
Precision	91%
Recall	73%
Threshold	50%
Sample Size	1873
Train Time	24 Hours

Phase Four: Integration into the OCR Process

The final phase of the document segmentation process for *The Atlantic* case study was to utilize the results from the document segmentation model. The model broke down the page's constituent segments. Turning those segments back into continuous stories required combining results from the document segmentation model, OCR output, and strategic programming that coalesced columns, titles, and other segments into a story in the proper order. This is important as Azure's Computer Vision Recognize Text v2.0 API does not have a way to identify that the text is within a column or separate the title, first paragraph, or columns from each other, let alone separate advertisements from the article text. Due to the nature of the labels and the bounding boxes, we could solve both problems simultaneously.

The OCR process entails the following steps

1. segment the document
2. OCR each segment
3. string the document together into a single text file
4. extract key facets
5. integrate results into the search index
6. provide full documents back to end user.

Concluding Thoughts For *The Atlantic*

CIDA's OCR process is designed with customers in mind. By building separate models to identify the evolving nature of *The Atlantic*'s layouts over time, we attained higher accuracy in identifying and correctly labeling document segments. Once those custom models were built, we integrated human perception into selecting the appropriate document segmentation algorithm to use on a set of uploaded documents to attain state-of-the-art results for a variety of document types.

Contact

Unifyconsulting.com

hello@unifyconsulting.com

8259 122nd Ave NE, Kirkland, WA 98033

206.395.2600