Automated data collection: tools, methods, and current efficacy

Originally published at https://www.itransition.com/blog/automated-data-collection
Published: June 23, 2021 | By Martin Anderson Independent AI expert

In 2019, Datamation estimated that unstructured information grows at 55-65% per year, and that 80% of all enterprise data falls into the 'unstructured' category. Hidden in these growing corporate data lakes are patterns and trends that offer insight and exploitable business advantages—if some way can be found to sift, quantify and analyze that data.

Even companies unconvinced that such investigations would be profitable are often obliged to maintain and at least minimally curate these high data volumes, since some or all of the information in them may fall under regulatory oversight and might eventually be needed to prove compliance—or to refute litigation, among other possible galvanizing influences.

In truth, therefore, there isn't much point in putting off the task of disentangling the apparently 'opaque' data mountain that your company may be holding onto, since you may need to find out exactly what's in there even if only to establish whether or not it's legal (or advisable) to delete it.

On a more positive note, that same process of investigation is an opportunity to bring order and structure to your unstructured data; to establish machine learning solutions that can help build and promote the company; to jettison the ever-growing expense of SaaS contracts, together with the potential exposure of private information to a third party service; and to leverage data on the company's own terms.

In this article, we'll take a look at some of the types of data that can be extracted through automated data collection, and some of the advantages of establishing local workflows to analyze and leverage one's own business intelligence resources.

Structured and unstructured data

There are three types of data that a company is likely to be holding onto: structured, unstructured, and semi-structured.

Structured data comes in a form that is to some extent 'pre-indexed', such as:

- Geolocation data.
- Tabular CSV worksheets with financial or known statistical data points, POS information or other types of data. Any program that can read the data can usually perform *some* type of analysis on it, and the data can also be used in more powerful analytical systems, including machine learning systems.
- Legacy database formats that are susceptible to a one-to-one mapping into a new data storage system.

Unstructured data has no such accompanying data model. It might come in a number of forms:

- A final, rasterized PDF, such as a contract signed by multiple parties in an email exchange—effectively just an opaque bunch of pixels in a PDF wrapper.
- Unscanned faxes, equally lacking in machine-interpretable content.
- Images in standard formats such as JPEG and PNG, with minimal or no metadata.
- Legacy communications formats that have been allowed to accrue due to a company's technical debt, which either never had a useful data model or where a viable analytical platform is no longer available.
- Call center recordings, which have been retained for manual oversight and compliance, but which represent opaque and unstructured audio data.
- Video content, including recorded VoIP conversations.

Semi-structured data has a schema, but not necessarily a data model. Data of this type can include:

- XML-formatted documents, which are rigidly structured but semantically 'meaningless'.

Email archives, which conform to a definite schema and which contain many searchable points of information (such as sender, date, and sender data), but which in themselves are 'open ended', undefined and unquantified, more suitable for targeted forensic investigation than casual data exploration.

Extracting information from PDFs

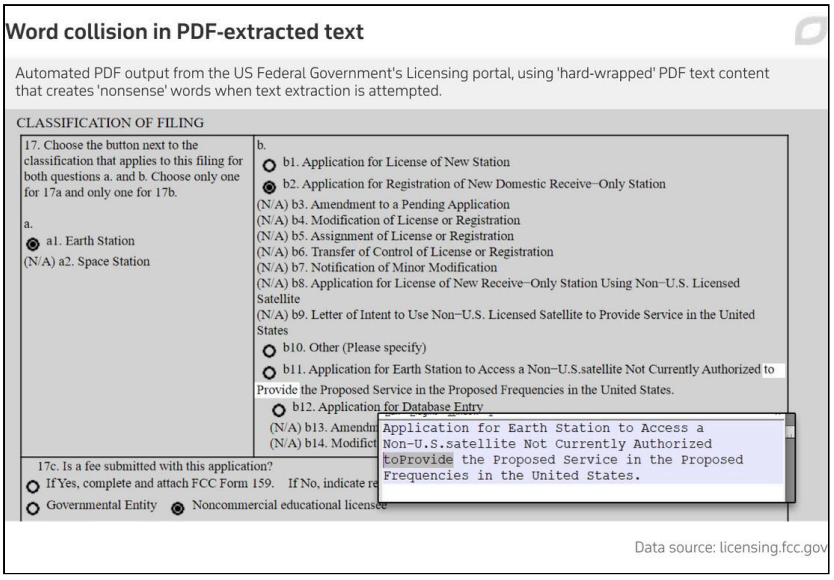
Since its underlying PostScript language gained an early foothold in desktop publishing in the late 1980s, Adobe's Portable Document Format (PDF) has stubbornly held on to its prized place in bureaucracy—despite its many shortcomings and the relative difficulty of extracting content for use in datasets and other more modern types of information storage.

PDF content in a corporate data lake will likely be one of three types:

- **Rasterized** image PDFs, including signed contracts, scanned faxes, and other types of content. These are unlikely to contain any useful intrinsic metadata, except for file permissions, at least a default user-name for the creator and creation dates. Text is rasterized (turned into an image).
- **Optimized** PDFs, where text-formatting has been 'baked' into paragraphs and columns to save on file size. Text is real rather than rasterized, but words at sentence start/endpoints are 'glued' together (see image below), requiring tedious and often expensive manual correction.
- **Natural-text** PDFs with 'reflowing' accessible text content that's more suitable for automatic conversion to HTML and other formats.

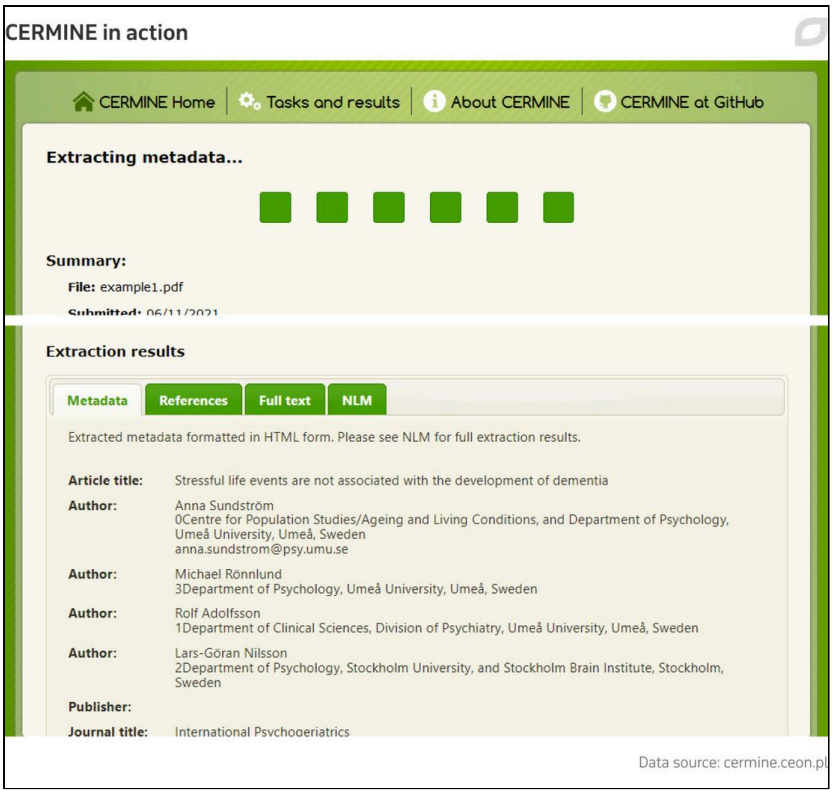
The content of reflowed PDFs is similar to HTML content and easier to parse into new systems, but the first two types present a challenge to data migration strategies, and can be addressed by automated data collection systems.

Many automated information systems, including US Federal systems, still output rasterized or hard-wrapped PDFs in order to save bandwidth and storage space, or simply because the originating systems suffer from technical debt and are unlikely to be reviewed or upgraded.

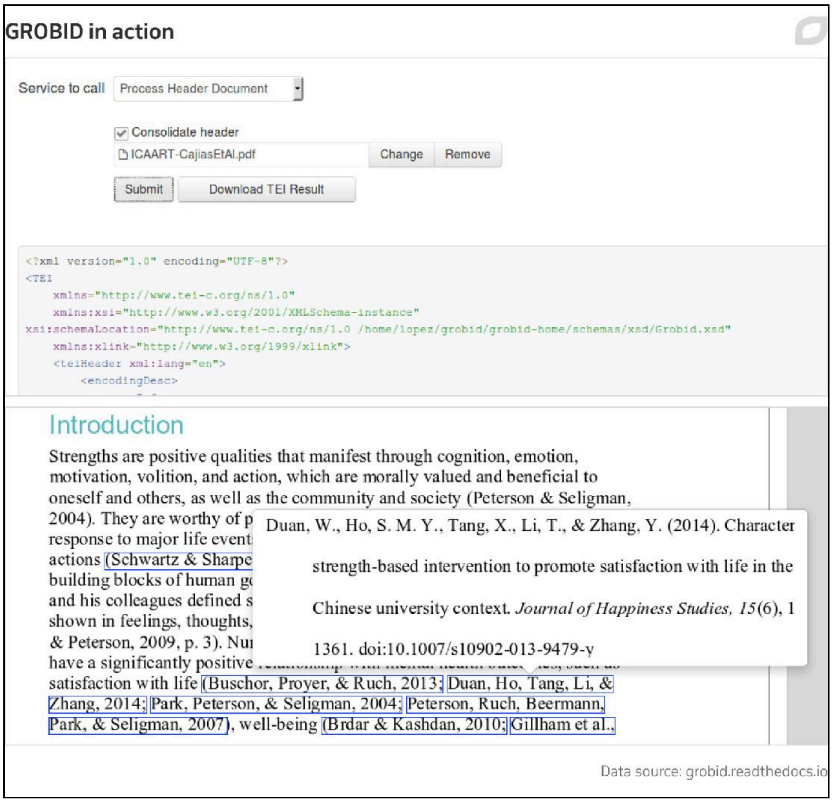


Converting baked text-formatting in PDFs

A number of proprietary and open-source projects have addressed the problem of extracting structured information from scientific research PDFs. Content ExtRactor and MINEr (CERMINE) is an open-source Java library, also made available as an online web service, that has been designed to extract structured elements such as article title, journal and bibliographic information, authors and keywords, among others.

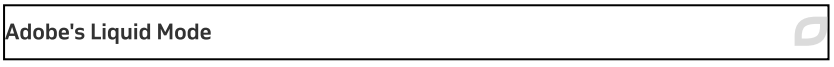


An alternative FOSS solution is GROBID (GeneRation Of Bibliographic Data), a machine learning library for the extraction, parsing and restructuring of PDF scientific and technical documents into XML or TEI-encoded document formats. GROBID can derive a very wide range of facets by default, including patent references and even information from 'interactive' PDFs.



Adobe, originator of the PDF format and itself a leading machine learning research leader, has begun to address PDF interpretability, releasing a new 'Liquid Mode' that can convert certain 'baked' PDFs into re-flowed PDFs.

The service is powered by Adobe's Sensei platform, and is initially being made available for Adobe Reader on Android and iOS devices, with plans to roll out the technology to desktop and other platforms.



Data source: youtube.com—Liquid Mode in Adobe Acrobat Reader | Adobe Acrobat, 2020

Though a business-level API for Liquid Mode has been available since late summer of 2020, neither commercial roll-out nor likely pricing have yet come to light.

Optical character recognition for rasterized PDFs

Optical character recognition (OCR) involves the algorithmic recognition of text that appears in images and video. It's an old technology, first pioneered in the 1920s and actively developed from the 1970s onward.

In recent years, intense interest in OCR from the machine learning community has consolidated the complex processing pipelines of older OCR technologies into data-driven frameworks that can be easily leveraged in an automated data collection system.

Consequently, there are a number of commercial APIs that can quickly solve this aspect of a custom framework, albeit with the precarity of relying on SaaS, including possible changes to pricing and terms of service over time.

Economic alternatives can be found in the various available FOSS solutions. These include:

- **Kraken**, a dedicated open-source turnkey system developed from the less production-ready Ocropus project. Kraken runs exclusively on Linux and MacOS, is managed via Conda, and supports CUDA acceleration libraries.
- **Calamari** OCR engine, which leverages TensorFlow's neural network feature-set and is written exclusively in Python 3.

Tesseract, a FOSS command line framework developed by Hewlett Packard in the 1980s, and maintained by Google since 2006.

Identifying a project's status with contribution analysis

Contributor activity is a useful indication that a community or internal company project is making progress, or at least is not floundering or in current need of review.

A project may have a variety of central channels through which its contributors communicate and report activity, including email, proprietary centralized platforms such as Slack, internal ticketing systems, and even messaging systems such as WhatsApp groups.

Analytical systems can be trained on some or all of these channels and offer a potentially valuable statistical insight into the state of a project. This may present an eye-opening alternate view to the individual reports and summaries of contributors, or from time-constrained managers who are trying to glean project trends from diverse inputs across a number of channels.

Identifying keys and features

Analytical systems are likely to key on centrally stored metadata from transactions, or from JSON or SQLite instances that define a record of activity, though many other protocols are possible.

Analyzing transactions of this type requires a data analytics strategy with a clear set of definitions about the significance of the data points that are gathered. Inactivity, for instance, could as easily signify an intense period of work (on the project itself, or, by sheer demand, on competing projects), or reflect the contributions of new staff; and an increase in ticket-based support requests can actually indicate a breakthrough phase, as formerly intractable problems yield to continued effort.

Therefore either a period of prior study will be necessary in order to develop an interpretive framework layer, or the system should replicate FOSS or otherwise usable prior frameworks that have already established methods to meaningfully interpret data points in a central locus of activity, such as a group project.

Employee status evaluation

Speaking of FOSS, one such initiative was undertaken recently by an Italian research group, who sought to understand the causes for the abandonment or decline of open-source projects. The researchers developed a preliminary state model based on interviews with researchers and used these insights to develop a machine learning-based analysis framework for FOSS initiatives.

The research established non-obvious correlations between contribution frequency and the long-term health of developers' relationships with a project. For instance, the results showed that founding or central contributors take frequent breaks of various lengths from contributing (with all developers taking at least one break), and that this 'inactivity' is likely an indicator of continued long-term effort.

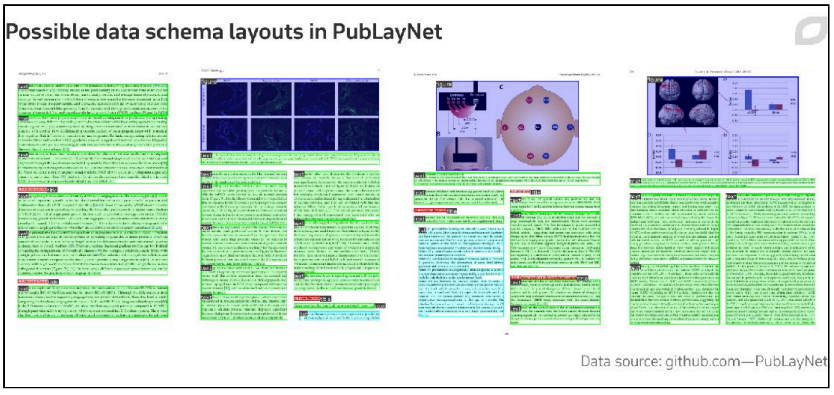
Data extraction from complex legacy documents

Though PDF presents some specific challenges in terms of extracting actionable data from legacy documents, the more generic challenge for a data extraction pipeline is to create agile machine systems capable of interpreting varied styles of layout and formatting across a range of document types.

Where the data in the lake is text-based—a format that may include everything from plain text files (a relative rarity in a business context) to JSON and email, among other plain-text formats—the content is inevitably linear, since the schema supports nothing else and the entire challenge involves segmenting and leveraging components in that conveniently straightforward stream of data.

However, analyzing rich text document types requires an understanding of where the information is likely to reside inside the document. Where formatting varies a lot, it's possible to develop a template to catch all likely scenarios so that the information can be reliably exfiltrated.

Among many companies that routinely create such definitions prior to document extraction, IBM has created its own large dataset of annotated document images, called PubLayNet.



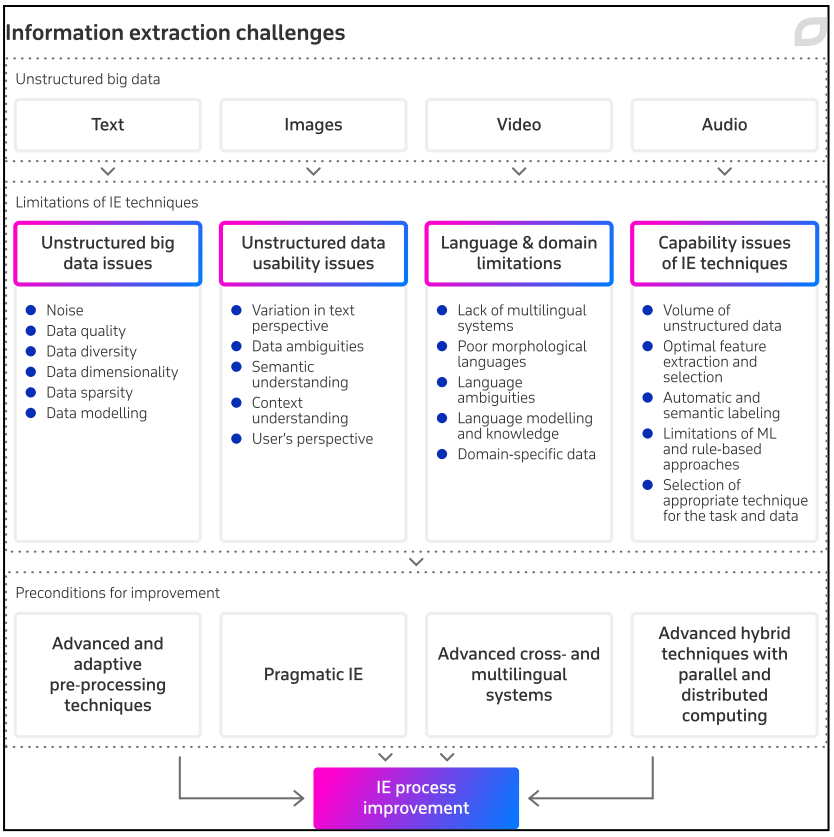
The open-source dataset contains over 360,000 labeled and annotated document images, and can serve as a useful starting point for a proprietary extraction framework. IBM itself has used the work as a basis for its own Corpus Conversion service, designed to improve the discoverability of PDF-based data by extracting and converting PDFs to the JSON format.

Extracting actionable information from audio and video

It's possible to derive a variety of types of data from video and audio content as well as from the audio component of stored videos.

Hyperplexed content

Since video content covers three potential and distinct domains (audio, visual, and semantically derived textual content), it represents a significant challenge to information extraction (IE) systems. In 2019, the *International Journal of Engineering Business Management* classified a number of challenges in this sector:



Nonetheless, there are effective open-source and commercial solutions to address the extraction of all components, which can be integrated into bespoke on-premises or cloud-based data collection pipelines, as necessary. Since most of these have arisen out of specific industry and/or governmental need, it's worth considering the type of information you'll need to derive from your content, and whether a FOSS component can be adapted economically to the task.

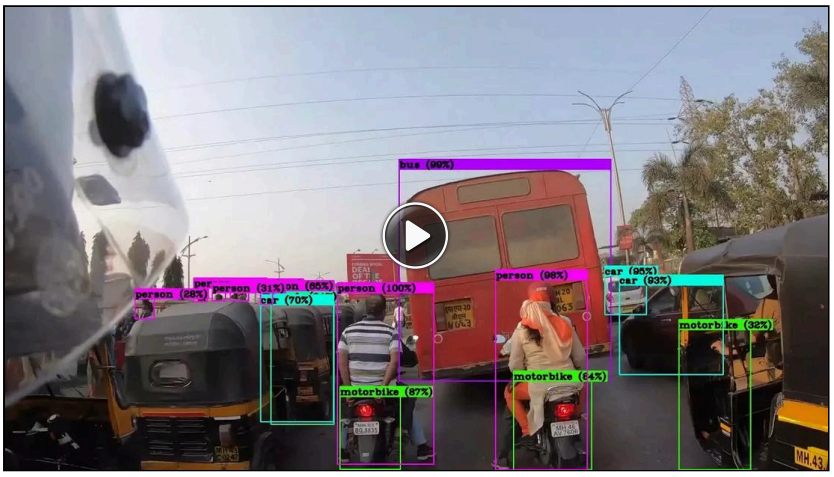
Reasons to analyze and deconstruct video and audio content include:

- Extracting **subtitles and effective content summaries** from video, to allow existing videos to become more discoverable.
- Extracting **biometric data** from archived image content, in order to develop effective security recognition systems.
- Extracting semantic interpretations of **actions** occurring in the content, to develop datasets capable of training security systems and for other purposes, such as in-store customer behavior analysis that makes part of computer vision applications in retail.
- Creating image-based **search functionality**, which requires object recognition tokens for video content and semantic segmentation routines to filter out individual facets, such as people and types of objects.
- To perform **emotion recognition** and/or **semantic analysis** on archived call center or sales center audio/video recordings, in order to develop a deeper understanding of the most effective communications methods and policies for staff, and to identify other actionable data from staff behavior.

Semantic segmentation

Among a healthy variety of FOSS systems available for identifying individuals in video, You Only Look Once (YOLO) has become a favorite among researchers and industry engineers in recent years.

Currently at version 4, YOLO offers an open-source CUDA-optimized object detection framework that's now capable of running effective semantic segmentation at 65fps, depending on the host hardware.



YOLO applies a single neural network to analysis of the entire image frame, creating bounding boxes around identified entities. The creators, Joseph Redmon and Ali Farhadi, Ali, claim that it runs up to 1,000 times faster than the next best CNN-based solution.

YOLO, written in C and native CUDA, runs under the Darknet open-source neural network framework, which offers pretrained models and a Discord-based support group. However, if you need a model that's more adapted to PyTorch or some other common framework language, there are hundreds of other actively-developed projects currently available.

Speech recognition: audio to text

Besides the impressive SaaS offeringsof the FAANG API giants, each of which has a vested interest in unlocking video and audio content, the FOSS speech recognition scene is equally vibrant:

- **Vosk** is a Python-based offline speech recognition toolkit that supports 18 languages, with continuous large vocabulary transcription, speaker identity recognition, and a zero-latency streaming API.
- **Julius** is a C-based speech recognition engine with academic roots hailing back to the early 1990s.
- **Flashlight** is Facebook's successor to the older Wav2Letter FOSS library, and rolls speech recognition into a wider set of capabilities, including object detection and image classification.

There are besides these a great many applicable open-source speech recognition projects suitable for inclusion in a dedicated information extraction framework, including Baidu's Python-based DeepSpeech2 (now incorporated into NVIDIA's OpenSeq2Seq), Facebook's Python-based Fairseq, and Kaldi's Speech Recognition Toolkit, which also features native (non-abstracted) support for neural networks.

Conclusion

Though it falls to each company to decide whether long-term maintenance of a dedicated data extraction framework compares well to the economics of cloud-based SaaS infrastructure, the extraordinary abundance of highly effective open-source information extraction projects represents a 'golden era' for creating custom data gathering pipelines—particularly when one considers that many of the SaaS solutions, from Amazon to RunwayML, are often only adding GUIs and infrastructure to deployments of the very same FOSS repositories.