Torch.AI™ Prism Document Analysis Solution

TORCH

Torch.AI Prism Document Analysis Solution

This white paper describes Prism, an intelligent document data extraction solution built on the Torch platform. Prism is a unique AI/ML-enabled system that extracts intelligence from structured, semi-structured, and unstructured data in documents, automating business processes with higher speed and accuracy than otherwise possible.

Introduction

Current document management systems help organizations process and store documents, but untold value is locked away in those documents as unstructured data and free text. Document analysis and data extraction becomes more and more challenging as the mountain of documents continues to grow. Challenges include:

- High operational costs. Manual processing is expensive, time consuming, and prone to errors.
- Existing tools are optimized for structured data. Conventional IT tools and solutions are limited in their ability to extract information from unstructured data and process that information at speed and at scale.
- Complex variable data elements. Unstructured content contains large variations in data dimensions, thus complicating the implementation of specific rules for extraction.

Document data extraction is the process of electronically retrieving data out of data sources for further processing. There are many existing solutions that leverage machine learning and Natural Language Processing (NLP) to extract text from documents, however, each has shortcomings and technical limitations. For example, one solution may be effective at one type of document structure but incompatible with another. Torch.AI has developed a novel approach to extracting data from documents that, we believe, overcomes the shortcomings of competing technologies and represents the state-of-the-art document data extraction system.

Technical Challenges

Today's document data extraction solutions are typically designed for strictly defined use cases and lack the adaptability and flexibility required for real-world applications. In response to this challenge, Torch.AI applied its core data intelligence platform to the document data extraction domain. To be successful in industrial-grade applications, the solution must address the following challenges:

- Ability to define the type of information to extract. A commercial solution must be easily customized to the unique customer use case.
- Ability to accept a variety of document and file types. Organizations are awash in documents and untold value is locked away in unstructured, unpredictable data.
- Ability to easily connect to downstream processes or systems. The solution must enhance, not displace, existing investments in data infrastructure.
- Ability to continuously fine-tune the machine learning models. Organizations should leverage an efficient data cycle for operations and maintenance.

In addition to these broad challenges, there are some specific technical hurdles faced by all document data extraction systems. While most can effectively extract raw text, many issues exist in the interpretation of that text. Specifically, recognizing information in fields and tables.

A field is more than just a chunk of text—it is a piece of information represented as a key value pair such as "Name: Jon Doe". Fields can be difficult to identify due to the variety of ways they are formatted in a document. For example, a field may be horizontally aligned and separated by a colon, vertically aligned in a form, embedded within a string of text, or inside a table. Groups of fields may be confused as a table. A document data extraction system must be able to recognize and parse fields, regardless of how they're represented in the document.

Text formatted in tabular format has proven particularly problematic for data extraction systems, first in detecting the table and then in extracting information from its rows and columns. There is notable prior work on identifying and extracting tabular data (S. Paliwal et al, 2020), however, each piece of research focuses on a specific problem domain. For example, some solutions require that the tables be input as text-based grids rather than scanned image files. No one solution handles the wide variety of document or tabular formats that appear in a typical real-life scenario.

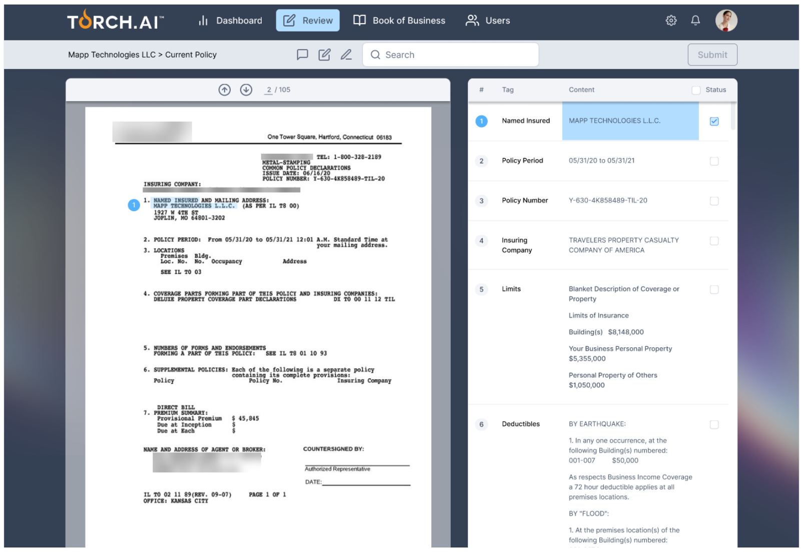
The remainder of this document describes how Prism approaches and overcomes these challenges.

Our Approach

After evaluating current state-of-the-art data extraction solutions including TableNet (S. Paliwal et al, 2020) and Detectron2 from Facebook Research (https://ai.facebook.com/tools/detectron2), we decided on the segmentation-based approach. Segmentation breaks down the content in the document to its basic elements and then extracts the information. We developed a collection of proprietary machine learning models that work in concert to identify the different types of data objects in a document, including fields and tables, and extract the information within. We determined that the input should be a scanned image of the document, rather than a text-based solution, to enable the system to input any type of document regardless of its source file type.

Training Optimization

Training and fine-tuning machine learning models is both an art and science. Prism streamlines the model-tuning process through its innovative review and feedback loop. Users provide feedback on the accuracy of the models through a simple check box in the Prism user interface. This feedback is then used to refine the training data and continuously improve the models. This continual learning process enables organizations to optimize the models for their business use case.



```
[redacted]
                                   One Tower Square, Hartford, Connecticut 06183

                                   [redacted] TEL: 1-800-328-2189
                                   METAL-STAMPING
                                   COMMON POLICY DECLARATIONS
                                   ISSUE DATE: 06/16/20
                                   POLICY NUMBER: Y-630-4K858489-TIL-20
INSURING COMPANY:
[redacted]
1. NAMED INSURED AND MAILING ADDRESS:
   MAPP TECHNOLOGIES L.L.C.  (AS PER IL T8 00)
   1927 W 4TH ST
   JOPLIN, MO 64801-3202

2. POLICY PERIOD:  From 05/31/20 to 05/31/21 12:01 A.M. Standard Time at
                                                  your mailing address.
3. LOCATIONS
     Premises  Bldg.
     Loc. No.  No.  Occupancy            Address

     SEE IL TO 03

4. COVERAGE PARTS FORMING PART OF THIS POLICY AND INSURING COMPANIES:
   DELUXE PROPERTY COVERAGE PART DECLARATIONS           DX TO 00 11 12 TIL

5. NUMBERS OF FORMS AND ENDORSEMENTS
   FORMING A PART OF THIS POLICY:   SEE IL T8 01 10 93

6. SUPPLEMENTAL POLICIES: Each of the following is a separate policy
                          containing its complete provisions:
   Policy                      Policy No.                Insuring Company

    DIRECT BILL
7. PREMIUM SUMMARY:
    Provisional Premium    $ 45,845
    Due at Inception       $
    Due at Each            $

NAME AND ADDRESS OF AGENT OR BROKER:            COUNTERSIGNED BY:
   [redacted]
                                                ______________________
                                                Authorized Representative

                                                DATE:_________________

IL TO 02 11 89(REV. 09-07)     PAGE 1 OF 1
OFFICE: KANSAS CITY
```

#	Tag	Content	Status
1	Named Insured	MAPP TECHNOLOGIES L.L.C.	✓
2	Policy Period	05/31/20 to 05/31/21	☐
3	Policy Number	Y-630-4K858489-TIL-20	☐
4	Insuring Company	TRAVELERS PROPERTY CASUALTY COMPANY OF AMERICA	☐
5	Limits	Blanket Description of Coverage or Property Limits of Insurance Building(s) $8,148,000 Your Business Personal Property $5,355,000 Personal Property of Others $1,050,000	☐
6	Deductibles	BY EARTHQUAKE: 1. In any one occurrence, at the following Building(s) numbered: 001-007 $50,000 As respects Business Income Coverage a 72 hour deductible applies at all premises locations. BY "FLOOD": 1. At the premises location(s) of the following Building(s) numbered:	☐

Figure: Prism Document Review

The Document Analysis Process

This process for analyzing a document consists of the following steps:

- **Target Mapping.** Defines the types of information that the system should identify in the document.
- **Pruning.** Inputs the document and removes unnecessary, extraneous content.
- **Segmentation.** Breaks down the content in the document to its most basic, granular elements.
- **Extraction.** Convert the elements to machine-readable text.
- **Interpretation.** Analyzes the elements for the desired information.

The following sections describe these steps in more detail.

Target Mapping

Fundamental to the segmentation-based approach of Prism is the process of breaking down a document to its most basic, granular elements, and then understanding the contextual meaning of those elements. This is a non-trivial task that leverages a novel collection of custom AI/ML-enabled algorithms.

The element types that the system recognizes within a document are:

- **Text** in sentence form containing a noun and a verb.
- **Fields** in key-value pairs (for example, “Policy Number: WVFH508749”).
- **Titles** such as section headings and page headings.
- **Address Blocks** which are addresses structured in a block of lines.
- **Tables** or tabular data.

As part of the business problem statement, the user must define the types of information to obtain from the input documents. Each type of information is referred to as a “target data point,” and they are defined in the target map. The target map is essential for both the Pruning and Interpretation steps.

The target map provides the following information for each target data point:

- **Name and ID** of the target data point.
- **Expected element types**, for example, a policy number is typically a field.

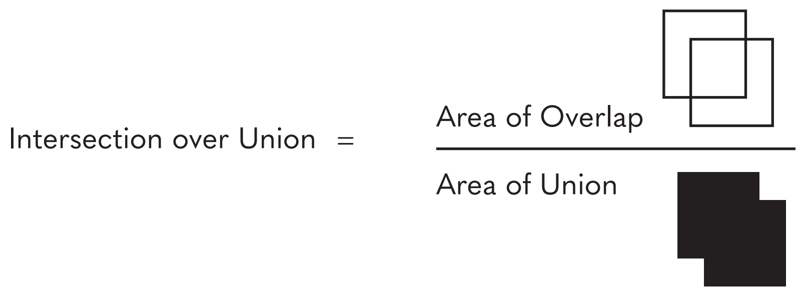
- **Queries** to identify the target data point, for example, “What is the policy number?”
- **Tokens** to identify the target data point, for example, “policy number”, “policy #”, and “policy num”.
- **Scoring** cutoff threshold (scoring is described next).

Scoring for each element is calculated as follows:

- **Text.** Calculated using the distance between the dense vector representation of the query and the vector representation of the result.
- **Title.** Calculated by looking for a partial fuzzy string match in the entire table. If the string is in the title, it is scored a 100% match.
- **Field.** Calculated by measuring the cosine distance between the target match token and the key in the key-value pair.
- **Address Block.** Calculated by looking for a partial fuzzy string match on each line of the address block. If the string is in any of the lines, it is scored a 100% match.
- **Table.** Calculated by looking for a partial fuzzy string match in the entire table. If the string is in the table, it is scored a 100% match.

Object Detection Metrics

This section describes performance metrics used to evaluate the object detection machine learning algorithms. Object detection is essential for detecting tables in a document. The most popular performance metric for object detection is **Intersection over Union (IoU)**. In object detection or segmentation problems, the ground truth labels are masks of a portion or a bounding box where the object is present. The IoU metric finds the difference between the prediction bounding box and the ground truth bounding box.



IoU is a value between zero and one. Perfectly overlapping boxes score one, and non-overlapping predictions score zero. Generally, IoU should be above 0.5 for an effective object detection model.

Mean Average Precision (mAP) is often used in object detection using IoU. To determine mAP, set an IoU Threshold value, for example, 0.5. Then we can classify an IoU prediction of 0.8 as True Positive. A prediction less than 0.5, such as 0.4, is a False Positive. Varying the thresholds provides different metrics.

Average Precision (AP) is obtained by finding the area under the precision-recall curve. The mAP for object detection is the average of the AP calculated for all the classes to determine the accuracy of a set of object detections from a model when compared to ground-truth object annotations of a dataset.

The mean average precision is calculated by taking the mean of AP over all classes and/or overall IoU thresholds. Many object detection algorithms, including Faster R-CNN and MobileNet, use this metric. This metric provides numerical value making it easier to compare with other models.

Pruning

The Pruning step inputs the source document and optimizes it for downstream processing. It identifies the meaningful content in the document and removes, or prunes, extraneous content. Pruning uses optical character recognition (OCR), information from the target map, and proprietary algorithms to determine whether a page contains relevant content. It then removes the pages that do not. This step effectively removes unwanted “noise” from the document, which prevents the waste of valuable computing resources.

Segmentation

The Segmentation step inputs the pruned document and breaks down the data to its basic elements (for a description of the elements, see the “Target Mapping” section earlier in this document). This step uses image segmentation models based on the Mask Region-Based Convolutional Neural Network (R-CNN) framework. Each model contains over 500 hyper-parameters and over 50,000,000 parameters. The Field Detection model detects key-value pairs in the document, and the Selection Object Detection model identifies document features such as check boxes and radio buttons.

The Tabular Data Extraction model is optimized for the unique challenges of extracting data from tables. It identifies the structure of tabular data, decomposes it into its smallest elements, and extracts each data point. It then reassembles the tabular elements to maintain structural context. It was base-trained on the TableBank dataset of 130,463 samples, and it was further fine-tuned on a custom dataset of 8,466 annotations created by Torch.AI.

Image Segmentation Metrics

This section describes performance metrics used to evaluate the image segmentation machine learning algorithms. The first metric we review is Pixel Accuracy, which is the ratio of the pixels that are classified to the total number of pixels in the image. Suppose that there are K+1 classes in an image where K is the number of all the object classes. A class refers to type of object, such as, in our case, a table within a document.

$$PixelAccuracy = \frac{\sum_{i=0}^{K} Pii}{\sum_{i=0}^{K}\sum_{j=0}^{K} Pij}$$

$P_{i,j}$ denotes the total number of pixels which belong to class i and are predicted to belong to class j.

Mean Pixel Accuracy is defined as ratio of the correct pixels as computed in a per-class manner. The value is then averaged over the total number of classes.

$$MeanPixelAccuracy = \frac{1}{K+1}\sum_{i=0}^{K}\frac{Pii}{\sum_{j=0}^{K} Pij}$$

The next metric we consider is **mean IoU** defined as the average of IoU's over all the classes.

Dice Coefficient is defined as the ratio of the twice the intersection of the predicted and ground truth segmentation maps to the total area of both the segmentation maps.

$$Dice = \frac{2|A \cap B|}{|A| + |B|}$$

Dice Loss is a popular performance evaluation metric that can also be used as the loss function while training the algorithm.

$$DiceLoss = 1 - \frac{2|A \cap B| + Smooth}{|A| + |B| + Smooth}$$

Extraction

Once the data has been segmented into its most basic elements, the Extraction step uses optical character recognition (OCR) to convert the elements to machine-readable text. The elements are now ready for the Interpretation step.

Interpretation

The Interpretation step applies advanced techniques to capture the semantic meaning of the data and identify the desired information. Techniques include question answering (QA), knowledge synthesis, summarization, and table querying.

The QA process requires each element and query to be converted to a vector. It consists of two steps: retrieval and read.

- The retrieval process uses the Dense Passage Retrieval (DPR) model to measure the mathematical distance between the element vectors and the query vectors. The element vectors with the closest distance to the query vectors are chosen for the read process.

- The read process performs reading comprehension on the chosen element vectors to extract the desired information. Reading comprehension uses the RoBERTa base transformer trained on the Stanford Question Answering Dataset (SQuAD) dataset consisting of 16GB of uncompressed text. Model training required 500,000 steps.

Classification and Information Retrieval Metrics

This section describes performance metrics used to evaluate the machine learning algorithms used in classification and information retrieval tasks. Classification tasks are measured by accuracy and error rate. Accuracy refers to proportion of examples which the model produces correct output. Error rate is the proportion of examples for which the model produces incorrect output.

Given a prediction p, and level y, a loss function measures the discrepancy between the algorithm's prediction and the desired output. The table below provides a list of various loss functions used in classification.

Loss	Function	Minimizer	Example Usage
Squared	$(p-y)^2$	Expectation (mean)	Regression •Expected return on stock
Quantile	$r(y-p)\mathbb{I}(y \geq p)+(1-r)(p-y)\mathbb{I}(y \leq p)$	Median	Regression •What is a typical price for a house?
Logistic	$log(1+exp(-yp))$	Probability	Classification •Probability of click on ad
Hinge	$max(0,1-yp)$	0-1 Approximation	Classification •Is the digit a 7?
Poisson		Counts (Log Mean)	Regression •Number of call events to call center
Classic	Squared loss without importance weight aware updates	Expectation (mean)	Regression •Squared loss often performs better than classic

The classification algorithms determine the model to minimize the loss function. The classifier label is either T (True) or F (False).

	Correct Label=True	Correct Label=False
Classifier Label=True	True Positive	FP Type 1 error
Classifier Label=False	FN Type 2 error	True Negative

The performance metrics are then defined as:

$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$
$Precision = \frac{TP}{TP+FP}$
$Recall = \frac{TP}{TP+FN}$
$F-measure = \frac{2}{\frac{1}{P}+\frac{1}{R}} = \frac{2PR}{P+R}$

Precision and **Recall** are useful when the true class is rare, such as a rare disease. Same holds true in information retrieval when only a few of a large number of documents are relevant.

Receiver Operator Characteristic (ROC) is created by plotting the True Positive Rate against False Positive Rate for various thresholds.

Generally, if a prediction has a value above 0.5, we classify it into positive class, otherwise, it's a negative class. In the diagram below, the decision boundary 0.5 is denoted as the threshold. It is not always necessary to use 0.5 as the threshold, sometimes other values might give the best results.

Dissemination of Results

In addition to providing a convenient user interface out-of-the-box, Prism makes the extracted information available in the client's preferred format. It can create a report for human consumption or automatically send the data to a downstream system. The Torch platform communicates through virtually any protocol, including REST, SOAP, File IQ, EDO, and SQL. It can connect to any data source, including relational warehouse, graph, data lake, or NoSQL databases.

Additional Capabilities of the Torch Platform

The Torch platform is a commercial-grade data intelligence platform with additional capabilities beyond document data extraction. It is an extensible data "mesh" built with a suite of AI-powered models, algorithms, and features that facilitate the analysis, transformation, and enrichment of data in real time. As an enterprise data platform, Torch can:

- Access federated, transformed, and enriched data, in real time, directly from the authoritative data sources.
- Interact with enterprise data through our Experience APIs, geared towards the business needs of the consuming systems and users.
- Take advantage of context-based data discovery and cataloging capabilities driven by the Torch Halo knowledge graph.
- Unlock the real value of a data ecosystem to systems and users with Semantic Stitching, a proprietary data retrieval system based on business terminology. Semantic Stitching does not require knowledge of SQL queries or programming skills.

The data extraction features are only one application of the Torch platform; it is capable of solving a virtually unlimited number of real-world use-cases that require large-scale ingestion, processing, enhancement, and analysis of data.

Conclusion

The Prism document data extraction solution is a novel approach to extracting data from documents. It is designed specifically for real-world use cases and overcomes the shortcomings of competing technologies. The solution utilizes a collection of proprietary machine learning models orchestrated to identify the different types of data objects in a document and extract the information within. The Torch platform inputs scanned images of documents regardless of the source file type. Our optimized OCR engine performs 12% faster than other available engines.

The solution uses a segmentation-based approach that breaks down the content in the document to its basic elements and then extracts the information. It leverages Torch.AI's extensive experience extracting information from unstructured data, including state-of-the-art techniques for extracting data in fields and tables. The segmentation predictions use detectron2 on an optimized serving platform, reducing processing time by 15% per sample.

Prism is not locked into one specific use case and can be customized and adapted as the needs of the client evolve. The Target Mapping functionality enables clients to define the type of information to extract, a necessary requirement for commercial-grade solutions. The user review and feedback loop streamlines the model-tuning process enabling organizations to continuously improve their models for their business user case. The Torch platform's advanced model training techniques reduce new model training and deployment time by 40%. Models can be generated automatically with optimized architectures to reduce processing time per GPU by 25%. Results are shared instantly either as reports for human consumption or electronically to downstream processes or systems.

This application of Prism, combined with the additional features available in the Torch platform, represent a ground-breaking leap in the art and science of document data extraction.

References

1. S. Paliwal, D. Vishwanath D, R. Rahul, M. Sharma, L. Vig. “TableNet: Deep Learning model for end-to-end Table detection and Tabular data extraction from Scanned Document Images”, 2020. https://arxiv.org/abs/2001.01469

2. Meta AI. “Detectron2”, 2022. https://ai.facebook.com/tools/detectron2