What’s a Pinpoint Document Template?

Latest Update: 12-June-2023

Introduction

Journalist Studio/Pinpoint is introducing a new tool for extracting data from form-like documents, named “**Extract structured data**”. It enables you to import form-based data, and export the processed data as a spreadsheet. For example, if you have ten thousand scanned auto accident reports that use a similar form, you can import the scans and export a spreadsheet that enables you to group, sort, or filter accidents by date, automobile manufacture, or any other fields provided in the source documents.

Goal

To use “**Extract Structured Data**” to process and export data, the document collection should consist of documents that share the same template (unlike a typical Pinpoint collection which can hold any documents). The goal of this guide is to define what constitutes a valid template for this purpose.

So, What’s a “template”?

Intuitively, documents that share the same template look identical, except for the values of some fields (which typically are what the user is interested in extracting). In other words, they all look like they were printed by the same clerk. Minor differences due to scanning (e.g. slight rotation) are also allowed.

More accurately, a template is a rendering of structured information with considerable boilerplate (common) text that maintains a consistent reading order and a consistent horizontal position. These concepts are explained below.

A collection of documents that merely have the same title, or that provide similar information but in vastly different organization, are not a template. For example, a collection of invoices may contain similar structured information, but they might be formatted in so many different ways that they are not considered a template for the purpose of this tool.

Reading Order

A template has a natural reading order. The tool expects this to be top-to-bottom, left-to-right (which means you read the text row by row)[1].

Some examples of templates that aren't supported:

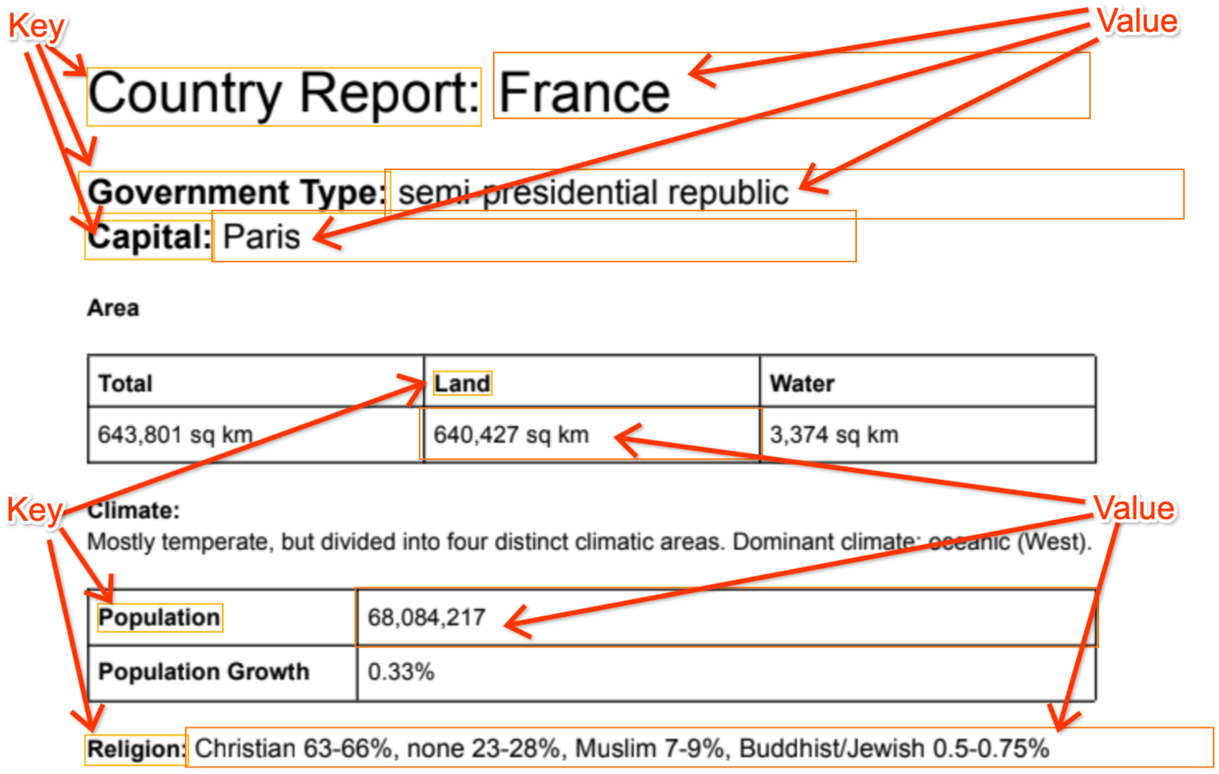
- Top-to-bottom text (for example, traditional Japanese).
- Column format (as in some newspapers) where reading order flows from the bottom of a column to the top of the next column.
- Cards (text boxes) that are laid out side by side (even if each card may have a valid reading order) - example.

Contents of a template

A template can have boilerplate text, key/value pairs (for example, name and address sections in a form), and tables.

Key-Value Pairs

In many form-like documents, each document contains many key-value pairs (e.g. the key, or label might be "Address", and the value might be "123 Main St, SmallTown, XY 12345"). Typically, the key is the same across all documents while the value differs between documents.



Country Report: France

Government Type: semi-presidential republic
Capital: Paris

Area

Total	Land	Water
643,801 sq km	640,427 sq km	3,374 sq km

Climate:
Mostly temperate, but divided into four distinct climatic areas. Dominant climate: oceanic (West).

Population	68,084,217
Population Growth	0.33%

Religion: Christian 63-66%, none 23-28%, Muslim 7-9%, Buddhist/Jewish 0.5-0.75%

[1] It actually works well also for right-to-left if the document is written in a right-to-left language.

The current release supports two types of values:

- String, inc. numerical values which are automatically converted to strings (so 1234 is extracted as "1234")).
- Checkbox (with value extracted as either **true** or **false**).

Fixed Forms

The simplest example of a template is a fixed form: Each document has a fixed number of pages, with the exact same layout across documents. For example, if the form has a “Residence Address” field with an adjacent box where the address is expected to be filled in, then both the key and the value box reside in almost exactly the same location in each of the documents. The only difference between the documents is the values that are written in these value boxes.

A typical form includes boilerplate text (e.g. instructions, headers) along with a number of key-value pairs, organized in some meaningful way. The boilerplate text and the keys are expected to be in the same place in all documents associated with this template.

Fixed forms are usually associated with paper. They are typically printable on paper in their naked form: before any values are printed on them. Many government forms are fixed (e.g. W-8BEN).

Non Fixed Forms

Unlike fixed forms, many form-like documents are printed from some structured data by a computer system, and may include, besides the boilerplate and fixed key-value pairs, also:

- Optional fields or sections.
- A variable number of fields or sections.
- Variable-length tables.

In a valid template, all boilerplate text and key-value pairs are expected to keep the same reading order. For example, if a template has a “Name” key followed by an “Address” key, then in all documents “Name” must appear before “Address”. They are also assumed to have a fixed horizontal position. So, if “Name” appears at the start of the line in one document, it appears at the start of the line in all documents (in which it is found).

Boilerplate or key-value pairs may be optional - appearing in only some of the documents, but if they exist, they must appear in the same order.

Repeated Sections

Sometimes templates allow a variable number of key-value fields, to describe an unknown number of similar entities. Each “entity” is described by a full-width horizontal section, containing one or more key-value fields describing this entity. The section repeats multiple times, each time using the same boilerplate and keys, but typically different values. The repeated section can be viewed as a mini-template having a variable number of consecutive instances within the full template.

A simple example:

Involved Persons:

First Name: John	Last Name: Smith	Badge #: 12345	Gender: M
First Name: Mary	Last Name: Jones	Badge #: 98765	Gender: F
First Name: Grace	Last Name: Kelly	Badge #: 01298	Gender: F

This demonstrates 3 repeating sections, each containing the fields “First Name”, “Last Name”, “Badge #” and “Gender”. Other documents in the collection may have a different number of persons.

Note: the grid isn’t needed, and each section may occupy more than one line, as in:

```
First Name: John           Badge #: 12345             Gender: M
Last Name: Smith

First Name: Mary           Badge #: 98765             Gender: F
Last Name: Jones

First Name: Grace          Badge #: 01298             Gender: F
Last Name: Kelly
```

As with the whole document, a repeated section in a template requires that:

- Reading order is fixed (that is, it is not allowed for one instance to swXitch the rows between “First Name” and “Last Name”).
- Horizontal position of each boilerplate/key is fixed (that is, it is not allowed for one instance to switch the position between “Badge #” and “Gender”).
- The value’s allocated area (“value box”) is in a fixed position relative to its key.

Nesting of repeated sections isn’t currently supported.

Tables

Tables are structured containers of data, which are organized in rows and columns. A template may contain tables which have the same structure across all documents. The structure is defined by the column layout: their horizontal positions and their column headers (if any).

For example, the above data may be formatted as a table:

First Name	Last Name	Badge #	Gender
John	Smith	12345	Male
Mary	Jones	98765	Female
Grace	Kelly	01298	Female

Despite their intuitive similarity, there is a fine line differentiating tables from repeated sections:

- Repeated sections contain Key-value pairs, whereas tables contain values.
- Tables may (but aren't required to) contain column headers, which represent the "key" for all values in that column.
- Note that while grid lines are more common in tables, they don't necessarily rule out repeated sections (as seen in the first example).

In rare cases, it may become difficult to tell whether a template component is a table or a repeated section, but in most cases following the above guidelines makes the choice clear.

Pages

The documents are allowed to spill over from one page to another. The extraction process is mostly insensitive to page headers or footers which may be inserted in the flow of text. Hence, both repeated sections and tables are allowed to span multiple pages.

Scanning Issues

When a document is scanned, some scaling, shifting or rotation of each of its pages may occur. A collection of documents that are originally of the same template are still considered to be of that template even if they are slightly distorted in this way.

Minor Changes

The tool is able to overlook small differences in documents, including:

- Font/style changes.

- Slight movements of text position.
- Slight changes to boilerplate text.
- Addition of new boilerplate text or new key-value pairs.

Strictly speaking, such a collection is NOT of a single template, but of a number of close templates (e.g. yearly versions of the same form). If the differences are small, the tool can overcome these differences, but if the differences are large (especially changes to text or order of key-value pairs), then results might be disappointing. If your documents have large differences, you might want to partition them into similar collections, and “Extract structured data” from each of these subset collections separately, and then combine your downloaded results.

When in doubt...

Just give it a try. Sometimes, if documents aren’t really of the same template, it will give garbage. But sometimes you may be surprised to get some good results. This is especially true if there’s a segment of the documents (e.g the first page) that adheres to these rules while the rest doesn’t. You may be able to get good extraction results from the “good” segment.

Examples (from across the web)

- Country factbook dataset: Toy dataset built for demos. Includes key-value pairs, tables and repeated-section. Great for learning how to use the tool.
- Medical Examination Report Form: Despite its complexity, if the form is fixed then it’s a valid template. Some sections (e.g. “Testing”) may have a column layout, but as long as it is fixed across documents, that’s still valid. Note that support for extracting checkboxes is not yet available in th.com/pinpoint-extract/resources/country_factbook_sample_collection.zipe Alpha release.
- Hearing Aid Compatibility Status Certifications (FCC Form 855): Demonstrates how key-value pairs may be subtle, how optional sections are allowed, and how the document flows across pages.
- Charlotte-Mecklenburg Police Department Incident Report: A great example of a non-fixed form template. Note the many key-value pairs, and the repeated sections (“Property”).
- Bill Receipt: Somewhat problematic, since the top key-value pairs are split into two distinct columns. If they are always consistent (e.g. “Phone” on the left is always on the same row as “Date” on the right) then it’s ok. Otherwise, attempting to extract from the headers might be flaky.