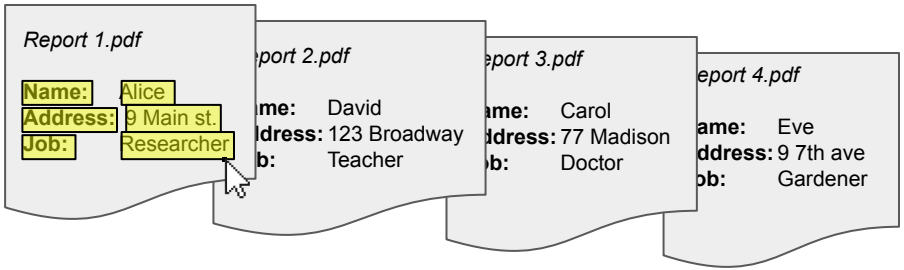
Few-Shot Learning for Structured Information Extraction From Form-Like Documents Using a Diff Algorithm

Nerya Or, Shlomo Urbach

Google

Introduction

Input: a set of similar looking documents, one of which is labeled to highlight key-value fields.



File name	Name	Address	Job
Report 1.pdf	Alice	9 Main st.	Researcher
Report 2.pdf	David	123 Broadway	Teacher
Report 3.pdf	Carol	77 Madison	Doctor
Report 4.pdf	Eve	9 7th ave	Gardener

Output: a trained model, allowing extraction of values of those same fields from additional similar-looking documents.

Document templates

A template is a family of documents, all of which are instances of a single type of form, or printouts of records from a single database.

Different documents in a template may contain certain variations in their layout, such as vertical shifts, optional fields, and sections repeating a variable number of times.

Assumptions about documents in a template

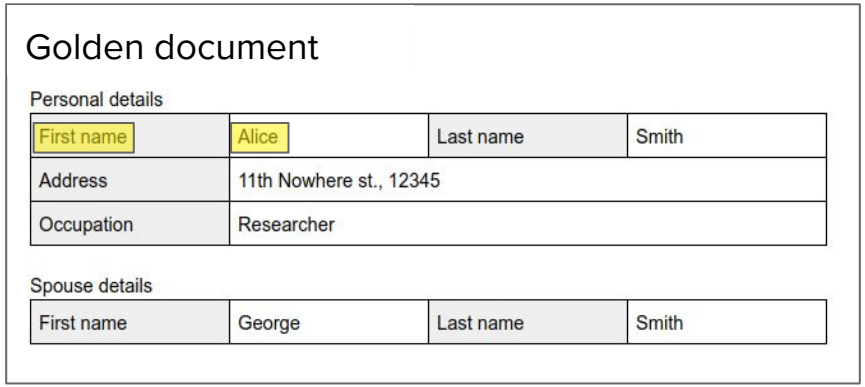
- Bounding boxes of values have a **fixed size, at a fixed offset** from their key.
- The **X coordinate** of boilerplate text tokens is **mostly fixed** (up to some tolerance) across documents in the template.
- **Y coordinates of tokens may vary**. However, if element A appears before element B in the golden document, the same will hold in the input document.
- The OCR engine provides us with a robust **reading order** of the documents.
- Scanned document pages are not significantly rotated, and not shifted or scaled across the image canvas. Correcting any of these transformations (if needed) can be done as a pre-processing step.

Sample run

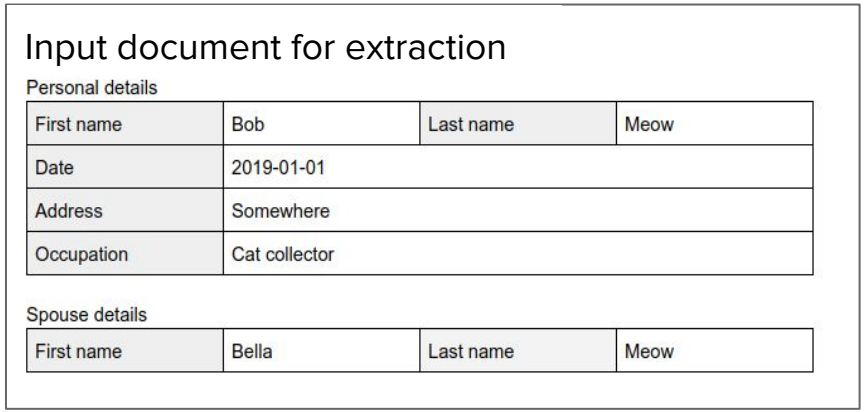
Input:

- Labeled key/values on a *golden* document.
- Training set of additional unlabeled documents.
- Input document(s), for extraction.

All documents should share the same template.



Output: Extracted fields from the input document(s).



Training: text token clustering

- Cluster the text tokens found in the training set.
  - Similarity of two given tokens is a function of their horizontal distance and text edit distance.
- Delete all clusters that are too small (less than some % of documents).
  - This removes most “value-only” tokens.

Thus, each text token is either:

- A member of some token-cluster (i.e., suspected as form label/boilerplate).
- Not a member of a cluster (i.e., suspected as a form value).

Clustering - example

Document 1:

First name: Alice **Last name**: Smith

Document 2:

First name: Bob **Last name**: Meow

⋮

Clustering - example

Document 1:

Document 2:

⋮

OCR text tokens and their X coordinates:
<First, 10>, <name, 150>, <Alice, 350>,
<Last, 600>, <nane, 750>, <Smith, 960>,
<First, 12>, <name, 147>, <Bob, 360>,
<Last, 598>, <name, 751>, <Meow, 955>

OCR error:
name → nane

Clusters:
3 = {<First, 10>, <First, 12>},
56 = {<name, 150>, <name, 147>},
82 = {<Last, 600>, <Last, 598>},
7 = {<nane, 750>, <name, 751>},
{<Alice, 350>},
{<Smith, 960>},
{<Bob, 360>},
{<Meow, 955>}

Unpopular clusters; will be deleted.

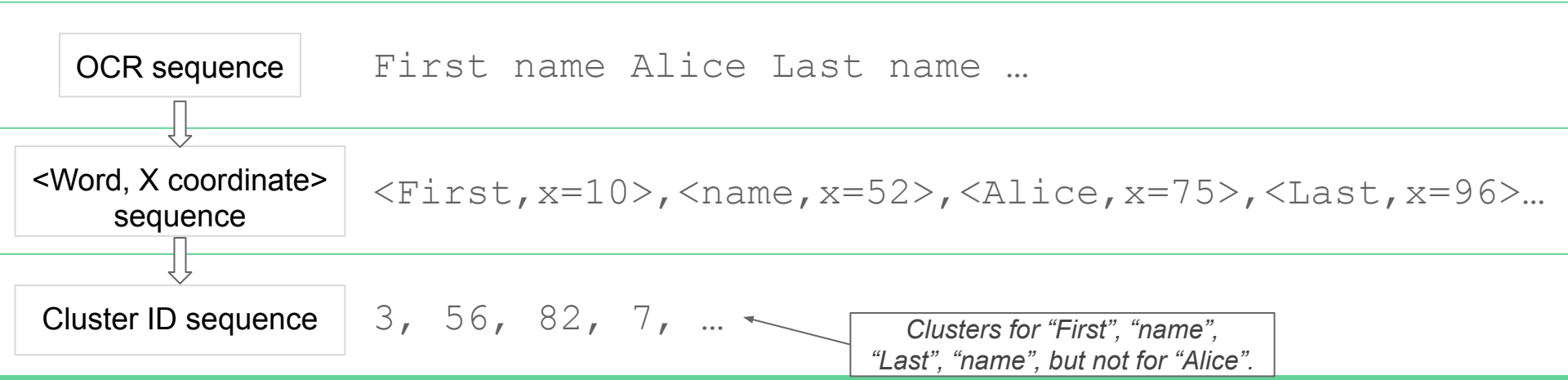
Extraction: cluster sequence generation

OCR words in a given document come in a well-defined reading order.

We generate a **sequence of cluster IDs** for the text tokens in the document, by classifying them using the clusters we found earlier.

Unclassified text tokens are **excluded.**

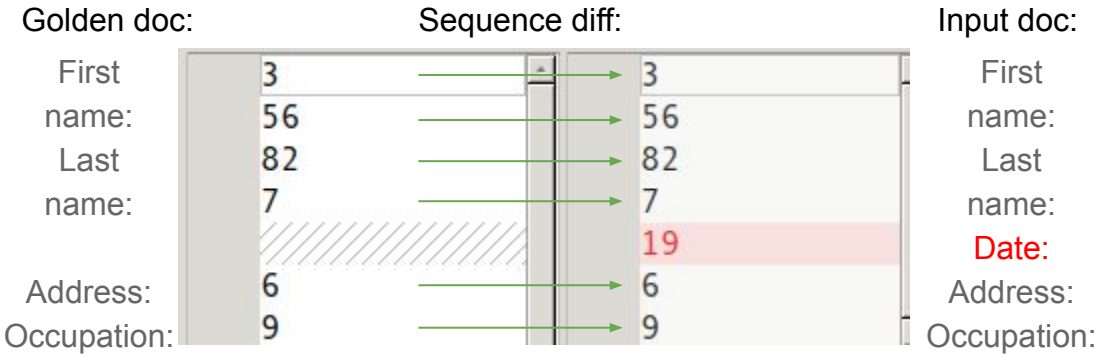
Example:



Extraction: diff between the golden and input documents

Given the golden document and another input document, we look at both of their cluster sequences, and perform a Longest Common Subsequence (a.k.a. “diff”) calculation.

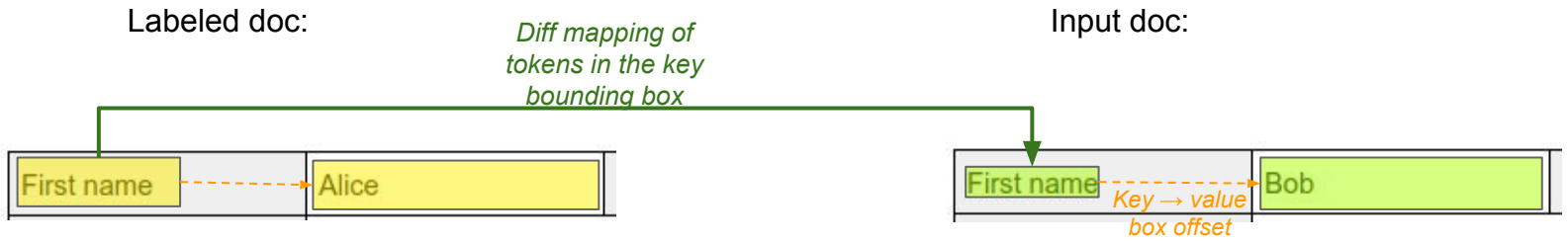
The result is a matching between “clustered” text tokens.



Extraction: finding values

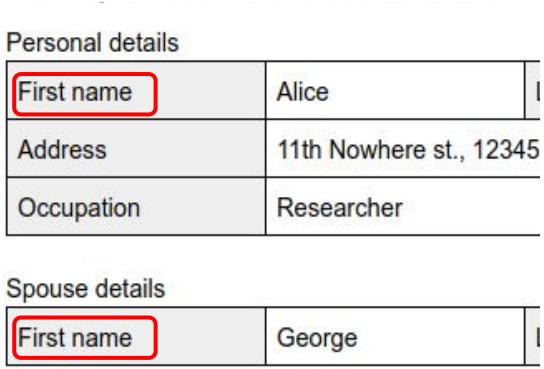
For each annotated key in the golden document:

1. Use the diff result, together with various heuristics, to find the corresponding text tokens in the input document. We now have a “key” bounding box in the input document.
2. Calculate the “value” box in the input document, using the offset between the given key and value boxes in the golden document.
3. Extract OCR words from the value box in the input document.



Diff optimizes across the full document

Words in the key box may repeat in other locations in the document.
In our example, “First name” appears twice:



Personal details

First name	Alice	
Address	11th Nowhere st., 12345	
Occupation	Researcher	

Spouse details

First name	George	

The diff-based approach can overcome this difficulty, by matching word order.

Repeating sections

Sections that repeat a variable number of times in each document, and contain key-value fields.

For example:

Country: USA	Date visited: 1985
Country: England	Date visited: 2002

Given some annotations specifying the repeating section on the golden document, we can run a modified diff algorithm to find a repeating correspondence.

Thus, values from all iterations of the repeating section can be extracted.

Evaluation

- Tested on several hundred documents (proprietary dataset).
  - Results are promising; F1 score of 0.914.

- It would be nice to have a public dataset to use for benchmarking.

Thank you