July 11, 2018

Structured Data from Multiple Scanned Forms

Shlomo Urbach

STRUCTURED DATA FROM MULTIPLE SCANNED FORMS

Introduction

The present disclosure provides systems and methods to obtain structured data from multiple scanned forms and/or other scanned documents. It may be desirable to process a large number of similar files (e.g., PDF files, etc.) generated from scanned documents such as forms reports, and/or the like, to extract data sets comprising some desired information (e.g., to provide for analysis, summarize results, determine statistics, etc.). However, processing scanned documents in an unsupervised manner (e.g., without requiring user input) can prove challenging, for example in matching values to form labels, due to fuzziness in locations, misinterpreted text, lost styles, lost separators, and/or the like resulting from document scanning and optical character recognition (OCR). The systems and methods of the present disclosure can provide for inputting multiple similar scanned forms, reports, and/or other documents (e.g., a document collection), determining schema for the document collection, and providing as output structured data extracted from the document collection.

Summary

According to an aspect of the present disclosure, a collection of scanned documents from which data is to be extracted can be provided as input. A schema can be generated for the document collection and the schema with values can be identified for each input document in the collection. Output can be generated comprising the resulting schema and a structured data set of all the extracted values for the document collection.

Detailed Description

According to an aspect of the present disclosure, systems and methods discussed herein can provide for obtaining structured data (e.g., tables, spreadsheets, etc.) from multiple scanned

forms and/or other scanned documents (e.g., portable document format (PDF) documents resulting from scanning and optical character recognition (OCR) of similar forms, reports, etc.). The systems and methods allow for processing the documents in an unsupervised manner (e.g., requiring little or no user input) to extract values for each of the fields of the scanned form, report, etc. and collect all the values into a single structured data set (e.g., a spreadsheet where rows equal documents and columns equal fields, etc.). In particular, the systems and methods of the present disclosure allow for providing, as input, multiple document files based on some form or structured printout and receiving, as output, a combined schema (e.g., label hierarchy) with values for each input document and a structured data set comprising the extracted values for all the input documents (e.g., a comma separated values file, other delimited text file, etc.).

According to an aspect of the present disclosure, a scanned document such as a form, report, and/or the like can be defined as a collection of fields. Each field can include a label (e.g., text tag) which may originally be the same across many or all of the set of documents to be processed and a value which may typically differ across the set of documents. Each field can have some consistent "label-value relation" (e.g., where the value is located relative to the label). For example, a value may be to the right of the field label or just below the field label. In some instances, a document (form, report, etc.) may also include some additional text that may be irrelevant to the desired data to be extracted from the document set. Since the documents may be the result of scanning and OCR, there may be some fuzziness to the location and OCR text. In addition, helpful cues to label/value associations, such as text styles, separators, etc., may be lost in the OCR process. In some cases, the scanned documents may not be instances of a fully static template, but rather may be reports generated (e.g., printed) by some type of software. In such

cases, some sections/labels may be omitted in some documents and/or the vertical position of fields might differ across the documents.

According to an example implementation of the present disclosure, a set of scanned documents (e.g., PDF files, etc.) from which data is desired to be extracted may be obtained. The scanned documents can each represent some instance of a form or other structured printout (report, etc.) having a label-value structure and having a strong similarity between the documents. In some instances, the scanned documents may have a hierarchical label-value structure and may have some cardinality (e.g., many instances of some level). The set of scanned documents (e.g., PDF documents) can be processed to convert each scanned document into text spans (e.g., generating a text span for each line in a document), for example, converting the PDF documents to Hypertext Markup Language (HTML) files comprising text spans. The HTML files can be parsed to extract texts which may represent labels thereby creating a label pool. An optimization process can be performed to identify the most probable labels for the document set (e.g., the schema for the document set) from the label pool. Output can then be generated comprising the resulting schema, the schema with values for each input document, and a structured data set of all the extracted values for the document set (e.g., a CSV file, etc.).

According to another aspect of the present disclosure, a schema for a set of scanned documents can comprise a collection of labels (e.g., selected from the label pool). The schema can include a name for each label (e.g., extracted text from the scanned documents), a position for each label (e.g., horizontal position in a document, etc.), and a value matching heuristic to allow for extracting a value for a label, for example, a top label (e.g. where the value is expected to be found below the label), a left label (e.g., where the value is expected to be found to the right

of the label), and/or a mixed label (e.g., where the position of the value relative to the label may be unknown).

According to another aspect of the present disclosure, a search can be performed on the set of scanned documents to identify a set of labels (e.g., each with a horizontal position and a label-value relation) which defines the schema. This set of labels (e.g., schema) can be used to extract a value (if possible) for each label for each document in the set. In some implementations, this search can be defined as an optimization problem, attempting to maximize a utility function which depends mainly on the average number of values found across all documents, while keeping the number of labels as low as possible.

In general, a combinatorial optimization problem has a definition of state, a method for scoring a state, a method for generating neighboring states (e.g., by changing something in the current state), and an optimization algorithm.

According to example implementation of the present disclosure, schema identification can be defined as an optimization problem where the state is defined as the current schema. For example, in some implementations, the text in all the documents can be parsed to identify a set of all candidate labels (e.g., the label pool). In some implementations, for example, a text value can be identified as a candidate label if it appears at a nearby column (e.g., up to some allowed edit-distance) across some defined percentage of all the documents. Each such candidate label is thus associated with an estimated horizontal position. The search space for the optimization problem is then to find the actual set of labels (which is assumed to be a subset of the set of candidate labels) and for each included label, decide on its label-value relation. A state (within this search space) is thus a collection of <label, label-value-relation> (where "label" includes the estimated horizontal position). The search process begins with some initial state (e.g. none of the

candidates, or all of them, or some random selection). It proceeds by examining all possible steps, scoring them, and deciding on a move to the next state based on these scores. For example, in some embodiments, the move selected can be that of the highest score. The search process concludes when no further step can be taken.

According to another aspect of the present disclosure, in some implementations, scoring of states can be done by attempting to match values from each document to the labels in the current state. The score can be based on a selected utility function. In some implementations, a utility function could be the average number of matching values across all documents in the set minus some small factor times the number of labels in the current state. For example, in some implementations, the scoring of a state can proceed in the following manner. Each document can have a number of tags (text spans). The tags in a document are processed to determine tags that match to the schema labels in the current state (e.g., using a similar label heuristic - horizontal position, edit distance, etc.). Each of these labels is used to match remaining tags that represent values for the labels using the matching heuristic associated with each label. The document score can then be determined based on the number of matched values. The score for the state (e.g., schema) can be determined as mean(document score) - $\lambda|\text{state}|$. In some implementations, matching can be viewed as an assignment problem where a heuristic gives each edge of a graph a score, with a zero score not being allowed. A match can then be found that provides the maximum score. In some implementations, a scoring function could include the use of semantic information to improve label selection.

In some implementations, the generation of neighboring states can comprise small changes to the current state (e.g., schema). For example, the current state can be changed by

removing a label in the state, adding a label from the label pool to the current state, or changing a label-value matching heuristic (e.g., the label-value relation – top label, left label, etc.).

According to another aspect of the present disclosure, in some implementations, the optimization algorithm to move between neighboring states can be a steepest ascent (hill climbing) algorithm and/or the like. In some implementations, the optimization algorithm can be steepest ascent with restarts.

Figure 1 depicts an example system 100 according to an example implementation of the present disclosure. Figure 1 illustrates one example computing system that can be used to implement the present disclosure. Other computing systems can be used as well. The system 100 may comprise one or more computing devices, such as computing device 102 and one or more remote computing devices (e.g., sever computing systems, etc.), such as remote computing device 140, coupled over one or more networks, such as network 180.

The computing device 102 can include one or more processors 104 and one or more memories 106. The one or more processors 104 can be any suitable processing device and can be one processor or a plurality of processors that are operatively connected. The memory 106 can include one or more non-transitory computer-readable storage mediums, such as RAM, ROM, EEPROM, EPROM, flash memory devices, magnetic disks, etc., and combinations thereof. The memory 106 can store data 108 and instructions 110 which are executed by the processor 104 to cause the computing device 102 to perform operations, including one or more of the operations disclosed herein.

According to an aspect of the present disclosure, the computing device 102 can include a schema generation/value extraction system 112 that can implement features of the present disclosure. For example, the computing device 102 can obtain data comprising a collection of

scanned documents, for example, from remote computing device 140, and the schema generation/value extraction system 112 can generate a collection schema based on the collection of scanned documents. The schema generation/value extraction system 112 can extract values from the collection of scanned documents based on the collection schema and generate structured data comprising the extracted values.

The computing device 102 can also include one or more input/output interface(s) 116. One or more input/output interface(s) 116 can include, for example, devices for receiving information from or providing information to a user, such as through a display device, touch screen, touch pad, mouse, data entry keys, an audio output device such as one or more speakers, a microphone, haptic feedback device, etc. The computing device 102 can also include one or more communication/network interface(s) 118 used to communicate with one or more systems or devices, including systems or devices that are remotely located from the computing device 102.

The remote computing device 140 can include one or more processors 142 and one or more memories 144. The one or more processors 142 can be any suitable processing device and can be one processor or a plurality of processors that are operatively connected. The memory 144 can include one or more non-transitory computer-readable storage mediums, such as RAM, ROM, EEPROM, EPROM, flash memory devices, magnetic disks, etc., and combinations thereof. The memory 144 can store data 146 and instructions 148 which are executed by the processor 142 to cause the remote computing device 140 to perform operations, for example, such as to implement operations as discussed herein. The remote computing device 140 may generate, store, process, and/or the like data comprising forms, reports, and/or the like which can be processed according to features of the present disclosure, for example, by providing such data to the computing device 102.

The remote computing device 140 can also include one or more communication/network interface(s) 152 used to communicate with one or more systems or devices, including systems or devices that are remotely located from the remote computing device 140, such as computing device 102, for example. The remote computing device 140 can also include one or more input/output interface(s) 154.

Figure 2 depicts a flowchart illustrating example operations 200 for providing structured data from multiple scanned documents in accordance with aspects of the present disclosure. Although operations 200 are shown and described in a particular order for purposes of illustration and discussion, the operations are not limited to the particularly illustrated order or arrangement and certain operations can be performed in different orders or simultaneously.

The operations begin at block 202 where multiple scanned documents to be processed are obtained by a computing device. For example, a collection of scanned documents (e.g., PDF files) containing data to be extracted can be obtained from one or more sources (e.g., a remote computing system, scanned and processed with OCR on the computing device, etc.). The scanned documents can represent forms, reports, or other structured documents having a label-value structure and with a strong similarity between the documents.

At block 204, each of the multiple scanned documents (e.g., PDF files) can be converted to a file comprising a series of text spans, for example, in HTML files. At block 206, the HTML files can be parsed to extract text segments which may represent labels in the scanned documents which can be used to generate a candidate label pool. For example, in some implementations, a text segment can be identified as a candidate label if it appears at a nearby column (e.g., up to some allowed edit-distance) across some defined percentage of all the documents.

At block 208, optimization can be performed to identify the most probable set of labels for the scanned documents (e.g., the schema for the document set) from the candidate label pool. For example, in some implementations, an optimization process can be performed to find the actual set of labels and for each included label, decide on its label-value relation. This set of labels (e.g., schema) is thus a collection of <label, label-value-relation> where "label" includes the estimated horizontal position of the label. The optimization process can begin with some initial state (e.g. none of the candidates, or all of them, or some random selection) and proceed by examining all possible steps, scoring them, and deciding on a move to the next state based on these scores. For example, in some embodiments, the move selected can be that of the highest score. The optimization process can conclude when no further step can be taken.

At block 210, the schema for the set of scanned documents can be used to extract a value (if possible) for each label from each document in the set.

At block 212, output can be generated comprising the resulting schema, the schema with values for each input document, and a structured data set of all the extracted values for the document set (e.g., a CSV file, etc.).

Figure 3 depicts an example page of a scanned document 300 in accordance with aspects of the present disclosure. As illustrated in Figure 3, the scanned document page 300 comprises a plurality of fields comprising a label and an associated value. Each label associated with a field of the scanned document page 300 can comprise a horizontal position and a label-value relation (e.g., top label, left label, etc.), as described herein.

Figures

Figure 1

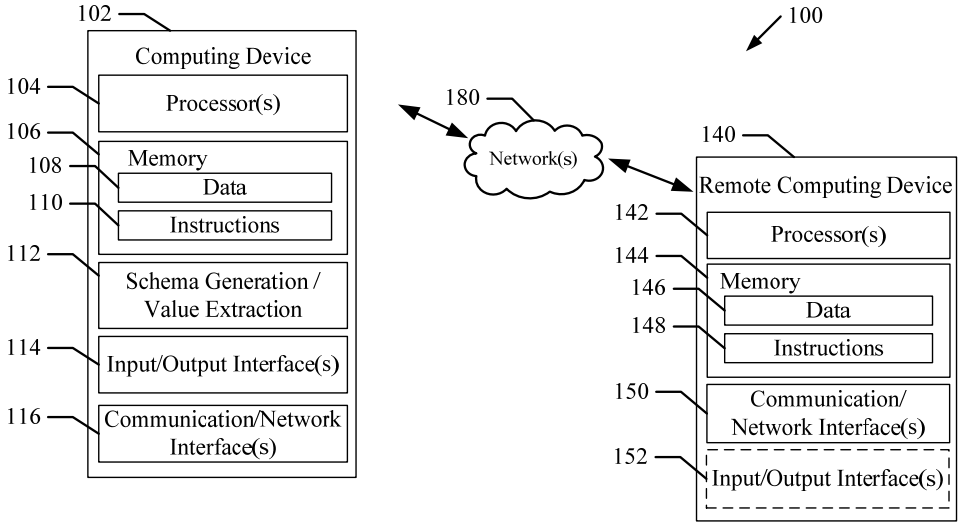

Figure 2

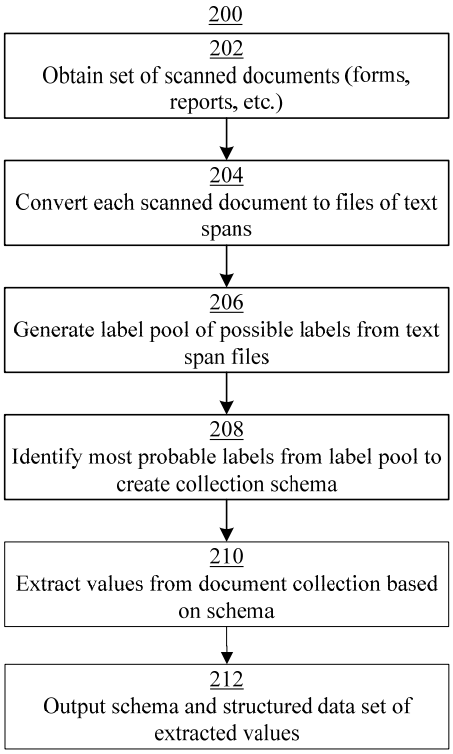

Figure 3

300

Sample Information System Report — Complaint

Case Information

Control ID : 1234567890	**IC Number :** 11-AB123456	**Submission Method :** Web	**Status :** Pending Analyst Review
Level One :	**Level Two :**	**Level Three :**	**Assigned Code Acronym :**
Owner Rep Number : MNO89	**Creator :** cgb.475	**Problem Submitted Time :** 5/15/2018 10:22:55 AM	**Date Closed :**

Contact Information

Consumer Personal Information Omitted

City/State/ZipCode : City, ST 12345

Complaint Type

Date of Problem :	5/11/2018	**Channel :**	QRS
Time of Problem :	9pm	**City/State :**	National
System :	XYZ	**Type :**	

Name of Problem : Thaodskjadfhasd

Problem Description

Thiodsfhasid asdfgsogi asfjasiop asdofhoasdjh asdhfgasdogh sadfgispgj asdpioaspdoj asdigpasdogj aspdogjps adog

Abstract

The present disclosure describes systems and methods that provide to obtain structured data from multiple scanned forms, scanned reports, and/or other scanned documents. More particularly, the systems and methods of the present disclosure allow for providing, as input, multiple scanned document files based on some form, report, or other structured printout and receiving, as output, a combined schema with values for each input document and a structured data set comprising the extracted values for all the input documents.