Naval Research Laboratory
Washington, DC 20375-5320

NRL/7340/MR—2022/3

Synthetic Data Generation Project for a Document Parsing AI

CHARLES NORSWORTHY

Center for Geospatial Sciences Branch
Ocean Sciences Division

December 13, 2022

DISTRIBUTION STATEMENT A: Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
13-12-2022	NRL Memorandum Report	02/02/2022 – 09/15/2022

4. TITLE AND SUBTITLE

Synthetic Data Generation Project for a Document Parsing AI

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Charles Norsworthy

5d. PROJECT NUMBER

5e. TASK NUMBER
BE031-03-42

5f. WORK UNIT NUMBER
1Y90

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
1005 Balch Boulevard
Stennis Space Center, MS 39529

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/7340/MR--2022/3

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Research
875 N Randolph St.
Arlington, VA 22217-1992

10. SPONSOR / MONITOR'S ACRONYM(S)

ONR

11. SPONSOR / MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A: Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This report presents documentation for objects that will be created by a synthetic data generation software. The data created by this software will be the training, validation, and testing data for an Artificial Intelligence system that can convert an image of a document into a JSON representation of that document.

15. SUBJECT TERMS

Document parsing
AI,
Artificial intelligence
Synthetic data generation

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					Charles Norsworthy
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**	U	25	**19b. TELEPHONE NUMBER** *(include area code)*
U	U	U			(228) 688-5534

This page intentionally left blank.

CONTENTS

This page intentionally left blank.

EXECUTIVE SUMMARY

This report presents documentation for objects that will be created by a synthetic data generation software. The data created by this software will be the training, validation, and testing data for an Artificial Intelligence system that can convert an image of a document into a JSON representation of that document.

This page intentionally left blank.

Synthetic Data Generation Project for a Document Parsing AI

1. INTRODUCTION

An invention in development that is part of the "Aero Document Data Extraction" project is the "Synthetic Data Generation Project for a Document Parsing AI". In this invention, NRL is writing software for generating synthetic training data for an AI that will parse a document into JSON. Currently, our software parses FAA forms by using a pre-defined configuration file for each different type of form that it encounters. (For example, 8260-3, 8260-4, etc.) It breaks the document down into a list of objects. However, we will have to create a new configuration file for each different form that exists. On https://www.faa.gov/forms/ [12] under "Export a List of All Forms", there is a download link which downloads an Excel file, this file lists 183 documents. Even if we were to write a configuration file for all these forms, if a new form is added, we will have to add a new configuration file, and if an existing form is significantly changed we will have to edit a configuration file. Thus, it may be a better use of our time to instead design an AI that can parse a wide variety of forms.

When finished, the software for this synthetic data generation project will work by randomly creating Word Documents that consist of a random selection of pre-defined objects, each with random characteristics (such as how many rows does a table have) and random strings. This will create a document that will function like a password, meaning that if an AI correctly parses a document, it probably didn't randomly guess what its structure is. And since the software is creating the document, it knows what the correct JSON parsing is and can record this in a JSON file.

The code for creating synthetic training, validation, and testing data for a document parsing AI will use Java and Python. The Java code reads in a JSON file and writes a .docx file based on the JSON. It uses Apache POI [1] for writing data to a .docx file. The Python code converts this .docx file into a PDF using the "docx2pdf" project [2], and then converts this PDF into an image file using the "pdf2image" project [13].

This documentation and this software are a work in progress.

1.1 Document Objects

A document that can be used in training is constructed by combining pre-defined objects together. There currently exist seven objects which can be used to construct a document: KeyValuePair, LinesOfKeyValuePair, Table, Paragraph, ParagraphRow, Header, and Footer.

1.2 Document Object Attributes

Objects that are written to a document can have many attributes.

1.2.1 Lines Above and Below

These objects have the option of having a line above the object or above the object's label if it exists, or a line below the object, or both. Objects that exist within other objects do not have the option to have these lines. This includes KeyValuePair objects that exist within in a LinesOfKeyValuePair object. The objects that exist within an indented object list also do not have the option to have these lines.

1.2.2 Alignment

Objects have three options for how they are aligned: left, center, and right. The KeyValuePair objects present in a LinesOfKeyValuePair object do not have the option to be aligned left, center, or right.

1.2.3 Indentation

Objects have the option to be placed in the document at a specified left or right indentation level. Each successive indentation level is some previously defined distance away from the previous indentation level. The KeyValuePair objects in a LinesOfKeyValuePair object do not have the option for a left or right indentation level. The KeyValuePair, LinesOfKeyValuePair, Table, and ParagraphRow objects do not have the option to have a right or left indentation level if they are aligned right or center in the

Manuscript approved December 9, 2022.

document. This is because these objects are all tables in a Microsoft Word document. The Paragraph objects also have the option to have either a hanging indent or a first line indent.

1.2.4 Label

All objects have the option for a label to be placed above the object. This label is always in bold, and can be aligned left, center, or right in the document. A label can also have a left or right indentation level in the document. A label can consist of multiple paragraphs.

1.2.5 Indented Object List

The indented object list attribute lets any object have an indented list of objects associated with it. Each object in an indented object list can itself have another indented object list associated with it. Any object except Header or Footer objects can be in an indented object list. In the parsed JSON for an object with an indented object list, the parsed JSON data for each object in an indented object list is placed in its own JSON object, and each of these JSON objects is placed in a JSON array, and this JSON array is included in the parsed JSON for the original object.

1.3 Object Types

1.3.1 KeyValuePair

The KeyValuePair object has different value types. There are two different categories of value types. The "one_string" category is composed of value types that have one string for the value. The "multiple_string" category is composed of value types that have multiple strings for the value.

Checkbox characters are characters that are meant to be interpreted as a boolean value in the parsed JSON. Thus, there are two types of checkbox characters: characters that are meant to be interpreted as false, and characters that are meant to be interpreted as true. An unselected checkbox is associated with false, and a selected checkbox is associated with true. If a KeyValuePair has one string for its value, this string can be a checkbox character. If a KeyValuePair has a list of strings for its value, any string in this list can be a checkbox character.

The checkbox characters associated with the false boolean value are:
1. "BALLOT BOX" (U+2610), "☐". [3]
2. "WHITE SQUARE" (U+25A1), "□". [4]

The checkbox characters associated with the true boolean value are:
1. "BALLOT BOX WITH X" (U+2612), "☒". [5]
2. "BALLOT BOX WITH LIGHT X" (U+2BBD), "☒". [6]
3. "BALLOT BOX WITH CHECK" (U+2611), "☑". [7]
4. "WHITE SQUARE CONTAINING BLACK SMALL SQUARE" (U+25A3), "▣". [8]

1.3.1.1 *Category: "one_string"*

With the "left_offset" value type, the value is one string to the left of the key.

Charles Norsworthy **Name**

With the "right_offset" value type, the value is one string to the right of the key.

Name Charles Norsworthy

With the "left_over" value type, the value is one string placed above the key and aligned to the left of the key.

Charles Norsworthy
Name

With the "center_over" value type, the value is one string centered above the key.

Charles Norsworthy
Name

With the "right_over" value type, the value is one string placed above the key and aligned to the right of the key.

Charles Norsworthy
Name

With the "left_under" value type, the value is one string under the key and aligned to the left of the key.

Name
Charles Norsworthy

With the "center_under" value type, the value is one string centered under the key.

Name
Charles Norsworthy

With the "right_under" value type, the value is one string under the key, and this value is aligned to the right of the key.

Name
Charles Norsworthy

1.3.1.2 *Category: "multiple_strings"*

With the "left_offset_list" value type, the value is a list of strings to the left of the key. Each string in this list of strings is placed under the first string in the list and is aligned to the right of the first string in the list.

Charles **Name**
Norsworthy
□

With the "left_offset_right_under_list" value type, the value is a list of strings. The first string in this list is placed to the left of the key, and the rest of the strings in this list are placed under the key and are aligned to the right of the key.

Charles **Name**
Norsworthy
□

With the "right_offset_list" value type, the value is a list of strings to the right of the key. Each string in this list of strings is placed under the first string in the list and is aligned to the left of the first string in the list.

Name Charles
Norsworthy
□

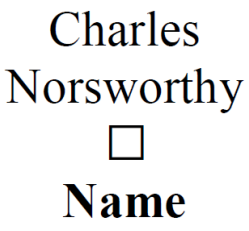
With the "right_offset_left_under_list" value type, the value is a list of strings. The first string in this list is placed to the right of the key, and the rest of the strings in this list are placed under the key and are aligned to the left of the key.

Name Charles
Norsworthy
□

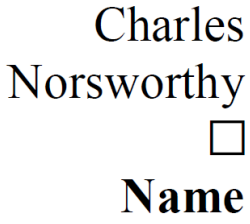
With the "left_over_list" value type, the value is a list of strings above the key. Each string in this list of strings is aligned to the left of the key.

Charles
Norsworthy
□
Name

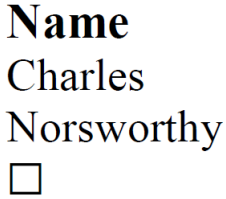
With the "center_over_list" value type, the value is a list of strings centered above the key.



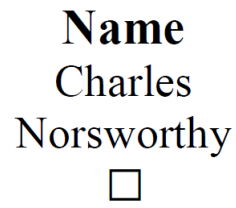
With the "right_over_list" value type, the value is a list of strings above the key. Each string in this list of strings is aligned to the right of the key.



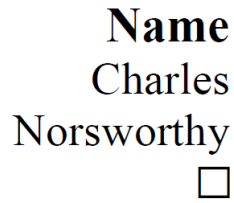
With the "left_under_list" value type, the value is a list of strings aligned left under the key.



With the "center_under_list" value type, the value is a list of strings centered under the key.



With the "right_under_list" value type, the value is a list of strings under the key, and all strings in this list are aligned to the right of the key.



1.3.2 LinesOfKeyValuePair

The LinesOfKeyValuePair object consists of one or more lines of one or more KeyValuePair objects. These KeyValuePair objects may be of any value type. This object's data except for the label is placed in a Microsoft Word document table with its borders removed. The LinesOfKeyValuePair object is inspired by data found in FAA forms.

A LinesOfKeyValuePair object is shown below, where this object has a label.

Example

Name1 Charles Norsworthy **Name2** Charles Norsworthy **Name3** Charles Norsworthy

Name1 Charles Norsworthy **Name2** Charles Norsworthy **Name3** Charles Norsworthy

1.3.3 Paragraph

A Paragraph object is text that is placed in the document. This text can span several lines in the document. Each Paragraph object may be preceded by a delimiter that denotes that a new Paragraph object is present. This delimiter can be many things, including a number, a Roman numeral, a letter, or a character such as "-". Numbers, Roman numerals, and letters may be followed by a period ".". A space is placed between a delimiter and the paragraph text, or between the period after the delimiter and the paragraph text.

A Paragraph object with a label is seen below.

Example

This is a paragraph.

1.3.4 Table

A Table object is a table in a Microsoft Word document.

In the image below, "Table 1" presents a Table object that contains a label.

Table 1

header1	header2	header3
value1_1	value1_2	value1_3
value2_1	value2_2	value2_3

In the image below, we can also see that it can be specified that a table not have any internal vertical borders ("Table 2"), any internal horizontal borders ("Table 3"), or any outside border ("Table 4"). Also, any combination of these three options can be specified.

Table 2

header1	header2	header3
value1_1	value1_2	value1_3
value2_1	value2_2	value2_3

Table 3

header1	header2	header3
value1_1	value1_2	value1_3
value2_1	value2_2	value2_3

Table 4

header1	header2	header3
value1_1	value1_2	value1_3
value2_1	value2_2	value2_3

Tables have the option of not having header rows ("Table 5").

Table 5

value1_1	value1_2	value1_3
value2_1	value2_2	value2_3

In "Table 6" in the image below, headers contain multiple lines of text. In the parsed JSON for this table, these multiple lines of text are combined into one line of text.

Table 6

This is header1	This is header2
value1	value2

In "Table 7" in the image below, certain headers in the header row have different characteristics than what is specified for all headers. Also, certain elements in the value row have different characteristics than what is specified for all values.

Table 7

header1	***header2***	***header3***
value1	value2	*value3*

In "Table 1" in the image below, the Table object contains two header rows instead of one. This table is based on the "Category" table from FAA forms such as [9].

Table 1

header1_1	**header1_2**			**header1_3**		
header2_1	**header2_2**	**header2_3**	**header2_4**	**header2_2**	**header2_3**	**header2_4**
value1_1	value1_2	value1_3	value1_4	value1_5	value1_6	value1_7
value2_1	value2_2	value2_3	value2_4	value2_5	value2_6	value2_7

In "Table 2" in the image below, the Table object contains three header rows. None of the headers in a row are repeated.

Table 2

header1					**header2**				
header1-1			**header1-2**		**header2-1**			**header2-2**	
header1-1-1	**header1-1-2**	**header1-1-3**	**header1-2-1**	**header1-2-2**	**header2-1-1**	**header2-1-2**	**header2-1-3**	**header2-2-1**	**header2-2-2**
value1-1	value1-2	value1-3	value1-4	value1-5	value1-6	value1-7	value1-8	value1-9	value1-10
value2-1	value2-2	value2-3	value2-4	value2-5	value2-6	value2-7	value2-8	value2-9	value2-10

In "Table 3" in the image below, the Table object contains three header rows, but in two of the header rows, headers are repeated.

Table 3

header1					**header2**				
header1-1			**header1-2**		**C**			**C**	
A	**B**	**A**	**A**	**B**	**A**	**B**	**A**	**A**	**A**
value1-1	value1-2	value1-3	value1-4	value1-5	value1-6	value1-7	value1-8	value1-9	value1-10
value2-1	value2-2	value2-3	value2-4	value2-5	value2-6	value2-7	value2-8	value2-9	value2-10

In "Table 4" in the image below, there are two header rows, but the first header row only has one header, and this header is a super header for only two of the headers on the second header row. This table is inspired by "TBL 4-1-1" in information provided by the FAA [10].

Table 4

			header	
A	**B**	**C**	**D**	**E**
value1-1	value1-2	value1-3	value1-4	value1-5
value2-1	value2-2	value2-3	value2-4	value2-5

1.3.4.1 *Table Characteristic: "repeated_headers"*

With the "repeated_headers" characteristic, a table has at least one header that is placed two or more times in a header row. In the parsed JSON for tables with this characteristic, the values in a value row in the table are placed into JSON objects in a JSON array, with each JSON object in this array containing key value pairs such that there is not a duplicate key in that object.

There is an algorithm for computing the headers that will appear in each JSON object in a JSON array representing one value row. This algorithm adds headers to a header group until it encounters a header that will produce a duplicate in that group. Then it starts a new group with that header being the first header of this new group. It continues to process headers like this until there are no more headers to process.

In "Table 1" in the image below, the table has these headers: A, B, C, A, B, C. This table has three unique core headers: A, B, and C. These three headers are placed twice in the header row.

Table 1

A	B	C	A	B	C
row2_column1	row2_column2	row2_column3	row2_column4	row2_column5	row2_column6
row3_column1	row3_column2	row3_column3	row3_column4	row3_column5	row3_column6

In "Table 2" in the image below, the table has these headers: A, B, C, A, D, E. This table has five unique core headers: A, B, C, D, and E. Only the "A" header is repeated.

Table 2

A	B	C	A	D	E
row2_column1	row2_column2	row2_column3	row2_column4	row2_column5	row2_column6
row3_column1	row3_column2	row3_column3	row3_column4	row3_column5	row3_column6

1.3.4.2 *Table Characteristic: "object_values"*

A Table with the "object_values" characteristic is a Table in which one or more of the cells in the value rows contain one or more objects instead of one or more Paragraph strings. These objects can be of any type and have any characteristic, including having a nested object list. In a value cell, there can be several objects. Objects in a table cell do not have the option to have a line above or a line below the object.

In "Table 1" in the image below, the first cell in the first value row contains a list of two objects: a KeyValuePair object, and a Paragraph object. The second cell in the first value row contains a LinesOfKeyValuePair object. The third cell in the first value row contains a Paragraph object. The fourth cell in the first value row contains a Table object with an indented object list. The second value row simply contains one string in each cell.

Table 1

header1	header2	header3	header4
Name Charles Norsworthy This is a paragraph in cell 1.	**Name1** Charles Norsworthy **Name2** Charles Norsworthy **Name1** ☐ **Name2** ☐	This is a paragraph in cell 3.	**Value Table** A | B | C value1_1 | value1_2 | value1_3 value2_1 | value2_2 | value2_3 This is an indented paragraph in cell 4. **Left Indentation Level:** 2
value2-1	value2-2	value2-3	value2-4

1.3.4.3 *Table Characteristic: "combined_value_cells"*

A Table with the "combined_value_cells" characteristic is a Table in which one or more of the cells in the table have been combined with another cell in the table above, or below, or to the left, or to the right of the cell. When cells are combined horizontally, the value in the combined cell is associated with the headers for the cells that have been combined. When cells are combined vertically, the value rows containing the cells that have been combined all share the value in the combined cell. When cells are combined both horizontally and vertically, both of the two above sentences are true.

In "Table 1" in the image below, the first two cells of the first value row have been combined, and the last two cells of the second value row have been combined.

Table 1

header1	header2	header3	header4
value_1-1		value_1-2	value_1-3
value_2-1	value_2-2	value_2-3	

In "Table 2" in the image below, the cells in the first column and the cells in the last column of the value rows have been combined.

Table 2

header1	header2	header3	header4
combined_value_1	value_1-1	value_1-2	combined_value_2
	value_2-1	value_2-2	

In "Table 3" in the image below, the cells from (1, 1) to (2, 2) have been combined diagonally.

Table 3

header1	header2	header3	header4
value_1-1	value_1-2	value_1-3	value_1-4
value_2-1	combined_value		value_2-4
value_3-1			value_3-4
value_4-1	value_4-2	value_4-3	value_4-4

In "Table 4" in the image below, all the value cells in the only value row have been combined.

Table 4

header1	header2	header3	header4
value_1-1			

1.3.4.4 *Table Characteristic: "combined_header_cells"*

A Table with the "combined_header_cells" characteristic is a Table in which one or more of the cells in the header rows have been combined with another cell in the header rows above, or below, or to the left, or to the right of the cell.

In "Table 1" in the image below, the cells in the first two header cells are horizontally combined. This means that this header relates to the two value columns.

Table 1

header1		header2	header3
value1_1	value1_2	value1_3	value1_4
value2_1	value2_2	value2_3	value2_4

In "Table 2" in the image below, the cells in the first header column are vertically combined.

Table 2

header1	header1_2		
	header2_1	header2_2	header2_3
value1_1	value1_2	value1_3	value1_4
value2_1	value2_2	value2_3	value2_4

In "Table 3" in the image below, the table object is the same as the previous "Table 2", except that the last two cells in the second header row have been combined and now have one header for both cells.

Table 3

header1	header1_2		
	header2_1	header2_2	
value1_1	value1_2	value1_3	value1_4
value2_1	value2_2	value2_3	value2_4

In "Table 4" in the image below, the table object has the first header cell diagonally merged.

Table 4

header1		header1_2	
		header2_1	header2_2
value1_1	value1_2	value1_3	value1_4
value2_1	value2_2	value2_3	value2_4

In "Table 5" in the image below, the table object has two header groups in its one header row, and these header groups contain cells that have been combined.

Table 5

A		B	A	B	
value1_1	value1_2	value1_3	value1_4	value1_5	value1_6
value2_1	value2_2	value2_3	value2_4	value2_5	value2_6

In "Table 6" in the image below, the table object has three header rows with headers that are diagonally and vertically combined.

Table 6

header1		header2
header3_1	header3_2	
value1_1	value1_2	value1_3
value2_1	value2_2	value2_3

In "Table 7" in the image below, the table object has three header rows with headers that are diagonally, horizontally, and vertically combined.

Table 7

header1			**header2**
header3_1	**header3_2**		
value1_1	value1_2	value1_3	value1_4
value2_1	value2_2	value2_3	value2_4

1.3.4.5 *Table Characteristic: "sub_labels"*
A Table with the "sub_labels" characteristic is a Table in which there are rows in which all cells in the row are merged horizontally together, and there is one bold header in this combined cell. This header is a sub label. This header acts like a label for the Table values until either the Table values end or until the next sub label.

In "Table 1" in the image below, a Table object is presented that has a single sub label.

Table 1

header1	**header2**
value1	value2
Label A	
value_a_1	value_a_2

In "Table 2" in the image below, a Table object is presented that has two sub labels.

Table 2

header1	**header2**
value1	value2
Label A	
value_a_1	value_a_2
value_a_3	value_a_4
Label B	
value_b_1	value_b_2
value_b_3	value_b_4

1.3.4.6 *Table Subtype: "key_value_pair"*
A Table of the "key_value_pair" subtype is a Table that is composed of a collection of key value pairs inside the table. A key is placed in one cell, and the value for this key is placed in another cell. The keys are in bold, and the values are not in bold. A value can be one string or a list of strings. If key value pair

has one string for its value, this string can be any one of the checkbox characters. If a key value pair has a list of strings for its value, any string in this list can be any one of the checkbox characters.

There are currently four categories of Table objects for the "key_value_pair" Table subtype. The Table category is determined by where the value for a key is placed. With the "right_offset" key value pair table category, the value for a key is placed in the cell to the right of this key. With the "center_under" key value pair table category, the value for a key is placed in the cell below this key. With the "left_offset" key value pair table category, the value for a key is placed in the cell to the left of this key. With the "center_over" key value pair table category, the value for a key is placed in the cell below this key.

1.3.4.6.1 Category: "right_offset"

In "Table 1" in the image below, a Table object of the "key_value_pair" Table subtype and of the "right_offset" key value pair table category is presented without a header row.

Table 1

Key1	row1_column2_value	**Key2**	☐	**Key3**	row1_column6_value1 row1_column6_value2	**Key4**	row1_column8_value1 row1_column8_value2 ☐
Key5	row2_column2_value	**Key6**	☐	**Key7**	row2_column6_value1 row2_column6_value2	**Key8**	row2_column8_value1 row2_column8_value2 ☐

In "Table 2" in the image below, "Table 1" is presented, except with a header row. In this table, every header is unique.

Table 2

A	B	C	D	E	F	G	H
Key1	row1_column2_value	**Key2**	☐	**Key3**	row1_column6_value1 row1_column6_value2	**Key4**	row1_column8_value1 row1_column8_value2 ☐
Key5	row2_column2_value	**Key6**	☐	**Key7**	row2_column6_value1 row2_column6_value2	**Key8**	row2_column8_value1 row2_column8_value2 ☐

In "Table 3" in the image below, "Table 1" is presented, except with a header row. In this table, there are four core headers: A, B, C, and D. Each header is repeated twice in the header row.

Table 3

A	B	C	D	A	B	C	D
Key1	row1_column2_value	**Key2**	☐	**Key3**	row1_column6_value1 row1_column6_value2	**Key4**	row1_column8_value1 row1_column8_value2 ☐
Key5	row2_column2_value	**Key6**	☐	**Key7**	row2_column6_value1 row2_column6_value2	**Key8**	row2_column8_value1 row2_column8_value2 ☐

In "Table 4" in the image below, "Table 1" is presented, except with a header row. In this table, there are two core headers: A and B. These headers are repeated multiple times in the header row.

Table 4

A	B	A	B	B	A	A	B
Key1	row1_column2_value	**Key2**	☐	**Key3**	row1_column6_value1 row1_column6_value2	**Key4**	row1_column8_value1 row1_column8_value2 ☐
Key5	row2_column2_value	**Key6**	☐	**Key7**	row2_column6_value1 row2_column6_value2	**Key8**	row2_column8_value1 row2_column8_value2 ☐

1.3.4.6.2 Category: "center_under"

In "Table 1" in the image below, a Table object of the "key_value_pair" Table subtype and of the "center_under" key value pair table category is presented without a header row.

Table 1

Key1	**Key2**	**Key3**	**Key4**
row2_column1_value	☐	row2_column3_value1 row2_column3_value2	row2_column4_value1 row2_column4_value2 ☐
Key5	**Key6**	**Key7**	**Key8**
row4_column1_value	☐	row4_column3_value1 row4_column3_value2	row4_column4_value1 row4_column4_value2 ☐

In "Table 2" in the image below, "Table 1" is presented, except with a header row. In this table, every header is unique.

Table 2

A	B	C	D
Key1	**Key2**	**Key3**	**Key4**
row2_column1_value	☐	row2_column3_value1 row2_column3_value2	row2_column4_value1 row2_column4_value2 ☐
Key5	**Key6**	**Key7**	**Key8**
row4_column1_value	☐	row4_column3_value1 row4_column3_value2	row4_column4_value1 row4_column4_value2 ☐

In "Table 3" in the image below, "Table 1" is presented, except with a header row. In this table, there are two core headers: A and B. Each header is repeated twice in the header row.

Table 3

A	B	A	B
Key1	**Key2**	**Key3**	**Key4**
row2_column1_value	☐	row2_column3_value1 row2_column3_value2	row2_column4_value1 row2_column4_value2 ☐
Key5	**Key6**	**Key7**	**Key8**
row4_column1_value	☐	row4_column3_value1 row4_column3_value2	row4_column4_value1 row4_column4_value2 ☐

In "Table 4" in the image below, "Table 1" is presented, except with a header row. In this table, there are two core headers: A and B. Only the "A" header is repeated in the header row.

Table 4

A	B	A	A
Key1	**Key2**	**Key3**	**Key4**
row2_column1_value	☐	row2_column3_value1 row2_column3_value2	row2_column4_value1 row2_column4_value2 ☐
Key5	**Key6**	**Key7**	**Key8**
row4_column1_value	☐	row4_column3_value1 row4_column3_value2	row4_column4_value1 row4_column4_value2 ☐

1.3.4.6.3 Category: "left_offset"

In "Table 1" in the image below, a Table object of the "key_value_pair" Table subtype and of the "left_offset" key value pair table category is presented without a header row.

Table 1

row1_column2_value	**Key1**	☐	**Key2**	row1_column6_value1 row1_column6_value2	**Key3**	row1_column8_value1 row1_column8_value2 ☐	**Key4**
row2_column2_value	**Key5**	☐	**Key6**	row2_column6_value1 row2_column6_value2	**Key7**	row2_column8_value1 row2_column8_value2 ☐	**Key8**

In "Table 2" in the image below, "Table 1" is presented, except with a header row. In this table, every header is unique.

Table 2

A	B	C	D	E	F	G	H
row1_column2_value	**Key1**	☐	**Key2**	row1_column6_value1 row1_column6_value2	**Key3**	row1_column8_value1 row1_column8_value2 ☐	**Key4**
row2_column2_value	**Key5**	☐	**Key6**	row2_column6_value1 row2_column6_value2	**Key7**	row2_column8_value1 row2_column8_value2 ☐	**Key8**

In "Table 3" in the image below, "Table 1" is presented, except with a header row. In this table, there are four core headers: A, B, C, and D. Each header is repeated twice in the header row.

Table 3

A	B	C	D	A	B	C	D
row1_column2_value	**Key1**	☐	**Key2**	row1_column6_value1 row1_column6_value2	**Key3**	row1_column8_value1 row1_column8_value2 ☐	**Key4**
row2_column2_value	**Key5**	☐	**Key6**	row2_column6_value1 row2_column6_value2	**Key7**	row2_column8_value1 row2_column8_value2 ☐	**Key8**

In "Table 4" in the image below, "Table 1" is presented, except with a header row. In this table, there are two core headers: A and B. These headers are repeated multiple times in the header row.

Table 4

A	B	A	B	B	A	A	B
row1_column2_value	**Key1**	☐	**Key2**	row1_column6_value1 row1_column6_value2	**Key3**	row1_column8_value1 row1_column8_value2 ☐	**Key4**
row2_column2_value	**Key5**	☐	**Key6**	row2_column6_value1 row2_column6_value2	**Key7**	row2_column8_value1 row2_column8_value2 ☐	**Key8**

1.3.4.6.4 Category: "center_over"

In "Table 1" in the image below, a Table object of the "key_value_pair" Table subtype and of the "center_over" key value pair table category is presented without a header row.

Table 1

row2_column1_value	☐	row2_column3_value1 row2_column3_value2	row2_column4_value1 row2_column4_value2 ☐
Key1	**Key2**	**Key3**	**Key4**
row4_column1_value	☐	row4_column3_value1 row4_column3_value2	row4_column4_value1 row4_column4_value2 ☐
Key5	**Key6**	**Key7**	**Key8**

In "Table 2" in the image below, "Table 1" is presented, except with a header row. In this table, every header is unique.

Table 2

A	B	C	D
row2_column1_value	☐	row2_column3_value1 row2_column3_value2	row2_column4_value1 row2_column4_value2 ☐
Key1	**Key2**	**Key3**	**Key4**
row4_column1_value	☐	row4_column3_value1 row4_column3_value2	row4_column4_value1 row4_column4_value2 ☐
Key5	**Key6**	**Key7**	**Key8**

In "Table 3" in the image below, "Table 1" is presented, except with a header row. In this table, there are two core headers: A and B. Each header is repeated twice in the header row.

Table 3

A	B	A	B
row2_column1_value	□	row2_column3_value1 row2_column3_value2	row2_column4_value1 row2_column4_value2 □
Key1	**Key2**	**Key3**	**Key4**
row4_column1_value	□	row4_column3_value1 row4_column3_value2	row4_column4_value1 row4_column4_value2 □
Key5	**Key6**	**Key7**	**Key8**

In "Table 4" in the image below, "Table 1" is presented, except with a header row. In this table, there are two core headers: A and B. Only the "A" header is repeated in the header row.

Table 4

A	B	A	A
row2_column1_value	□	row2_column3_value1 row2_column3_value2	row2_column4_value1 row2_column4_value2 □
Key1	**Key2**	**Key3**	**Key4**
row4_column1_value	□	row4_column3_value1 row4_column3_value2	row4_column4_value1 row4_column4_value2 □
Key5	**Key6**	**Key7**	**Key8**

1.3.4.7 *Table Subtype: "first_header_cell_absent"*

A Table of the "first_header_cell_absent" subtype is a Table that has one or more header rows, and in each of these header rows, the first cell in a header row is absent. In the parsed JSON for tables of this subtype, the first element in each value row is the key for a JSON object that contains the headers associated with the rest of the values in the table. In the next four Table examples, a Table object of the "first_header_cell_absent" subtype is presented.

In "Table 1" in the image below, every header is unique.

Table 1

	A	B	C	D	E
value1-1	value1-2	value1-3	value1-4	value1-5-1 value1-5-2	value1-6-1 value1-6-2 □
value2-1	value2-2	value2-3	value2-4	value2-5-1 value2-5-2	value2-6-1 value2-6-2 □

In "Table 2" in the image below, there are repeated headers.

Table 2

	A	B	C	A	B
value1-1	value1-2	value1-3	value1-4	value1-5-1 value1-5-2	value1-6-1 value1-6-2 □
value2-1	value2-2	value2-3	value2-4	value2-5-1 value2-5-2	value2-6-1 value2-6-2 □

In "Table 3" in the image below, there are two header rows with no repeated headers.

Table 3

	header1			header2	
	header1-1	header1-2	header1-3	header2-1	header2-2
value1-1	value1-2	value1-3	value1-4	value1-5-1 value1-5-2	value1-6-1 value1-6-2 □
value2-1	value2-2	value2-3	value2-4	value2-5-1 value2-5-2	value2-6-1 value2-6-2 □

In "Table 4" in the image below, there are two header rows in which the "A" header is repeated.

Table 4

	header1			header2	
	A	B	A	A	A
value1-1	value1-2	value1-3	value1-4	value1-5-1 value1-5-2	value1-6-1 value1-6-2 □
value2-1	value2-2	value2-3	value2-4	value2-5-1 value2-5-2	value2-6-1 value2-6-2 □

In "Table 5" in the image below, there are two header rows, but the first header row only has one header, and this header is a super header for only two of the headers on the second header row. This table is inspired by "TBL 4-1-1" in [10].

Table 5

				header	
	A	B	C	D	E
value1-1	value1-2	value1-3	value1-4	value1-5	value1-6
value2-1	value2-2	value2-3	value2-4	value2-5	value2-6

1.3.4.8 *Table Subtype: "changing_headers"*

A Table with the "changing_headers" characteristic is a Table in which the header or headers for values under the header row or header rows change at least once.

In "Table 1" in the image below, there are two headers in one header row in the table, and these headers change after one value row.

Table 1

A	B
value_1-1	value_1-2
C	D
value_2-1	value_2-2
value_2-1	**Value Table** A | B | C value1_1 | value1_2 | value1_3 value2_1 | value2_2 | value2_3 This is an indented paragraph in cell 2. **Left Indentation Level:** 2 This is a paragraph in cell 2.

In "Table 2" in the image below, there are two headers in each header row, but the first header cell in each header row is empty. This table example is technically a combination of two table subtypes: "changing_headers" and "first_header_cell_absent". This example is based on "TBL 5-1-2" in [11].

Table 2

	A	B
value_1-1	value_1-2	value_1-3
	C	D
value_2-1	value_2-2	value_2-3
value_3-1	value_3-2	**Value Table** A | B | C value1_1 | value1_2 | value1_3 value2_1 | value2_2 | value2_3 This is an indented paragraph in cell 3. **Left Indentation Level:** 2 This is a paragraph in cell 3.

1.3.5 ParagraphRow

A ParagraphRow object is an object in which a table with one row and two or three columns is placed in the document. This table should span the width of the document, and should have all its borders removed. Each cell in this invisible table contains a Paragraph object. When there are two cells in a ParagraphRow object, the first cell's Paragraph object is aligned to the left of the cell. The second cell's Paragraph object is aligned to the right of the cell. In "ParagraphRow 1" in the image below, the ParagraphRow object has two cells.

ParagraphRow 1

This is a paragraph in cell 1.

This is paragraph 1 in cell 2.
This is paragraph 2 in cell 2.

When there are three cells in a ParagraphRow object, the first cell's Paragraph object is aligned to the left of the cell. The second cell's Paragraph object is aligned in the center of the cell. The third cell's Paragraph object is aligned to the right of the cell. In "ParagraphRow 2" in the image below, the ParagraphRow object has three cells.

ParagraphRow 2

This is paragraph 1 in cell 3.

This is a paragraph in cell 1.

This is paragraph 2 in cell 3.

This is a paragraph in cell 2.

1.3.6 Header and Footer

The Header and Footer objects are a Microsoft Word header or footer placed at the top or bottom of a page. Currently, the code only supports writing Paragraph and ParagraphRow objects in a Header or a Footer.

REFERENCES

1. https://poi.apache.org/
2. https://pypi.org/project/docx2pdf/
3. https://util.unicode.org/UnicodeJsps/character.jsp?a=2610
4. https://util.unicode.org/UnicodeJsps/character.jsp?a=25A1
5. https://util.unicode.org/UnicodeJsps/character.jsp?a=2612
6. https://util.unicode.org/UnicodeJsps/character.jsp?a=2BBD
7. https://util.unicode.org/UnicodeJsps/character.jsp?a=2611
8. https://util.unicode.org/UnicodeJsps/character.jsp?a=25A3
9. https://www.faa.gov/aero_docs/acifp/NDBR/20190329287062010003-HDC-NDBR/LA_HAMMOND_IL18_HDC.pdf
10. https://www.faa.gov/air_traffic/publications/atpubs/aim_html/chap4_section_1.html
11. https://www.faa.gov/air_traffic/publications/atpubs/aim_html/chap5_section_1.html
12. https://www.faa.gov/forms/
13. https://pypi.org/project/pdf2image/