
AWS
re:Invent

AIM310

Intelligent document processing for the insurance industry

Raj Copparapu
Senior Project Manager
AWS

Reddi Gudla
Staff Vice President
Anthem, Inc.

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Benefits of Amazon Textract

How Anthem automates insurance claims

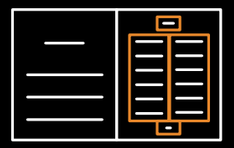
Four steps to implement Amazon Textract at scale



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Benefits of Amazon Textract

Extract data quickly and accurately; flexible extraction
(extract from variable document types)





No ML experience required

Eliminate manual effort
(including template development)





Lower document processing costs
(average of 10x savings for OCR workloads)

Textract customers



How Anthem automates insurance claims



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Problem Statement



Manual process to read claim attachments of up to 30 pages and classify documents for downstream functions



Identification and classification of the documents for assignment to various teams



Manual process takes ~20 minutes per claim attachment to classify with an average of 2,000+ claim attachments

Why Amazon Textract?



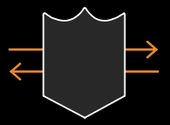
Image preprocessing capability



Ability to detect table and forms



Handwriting detection

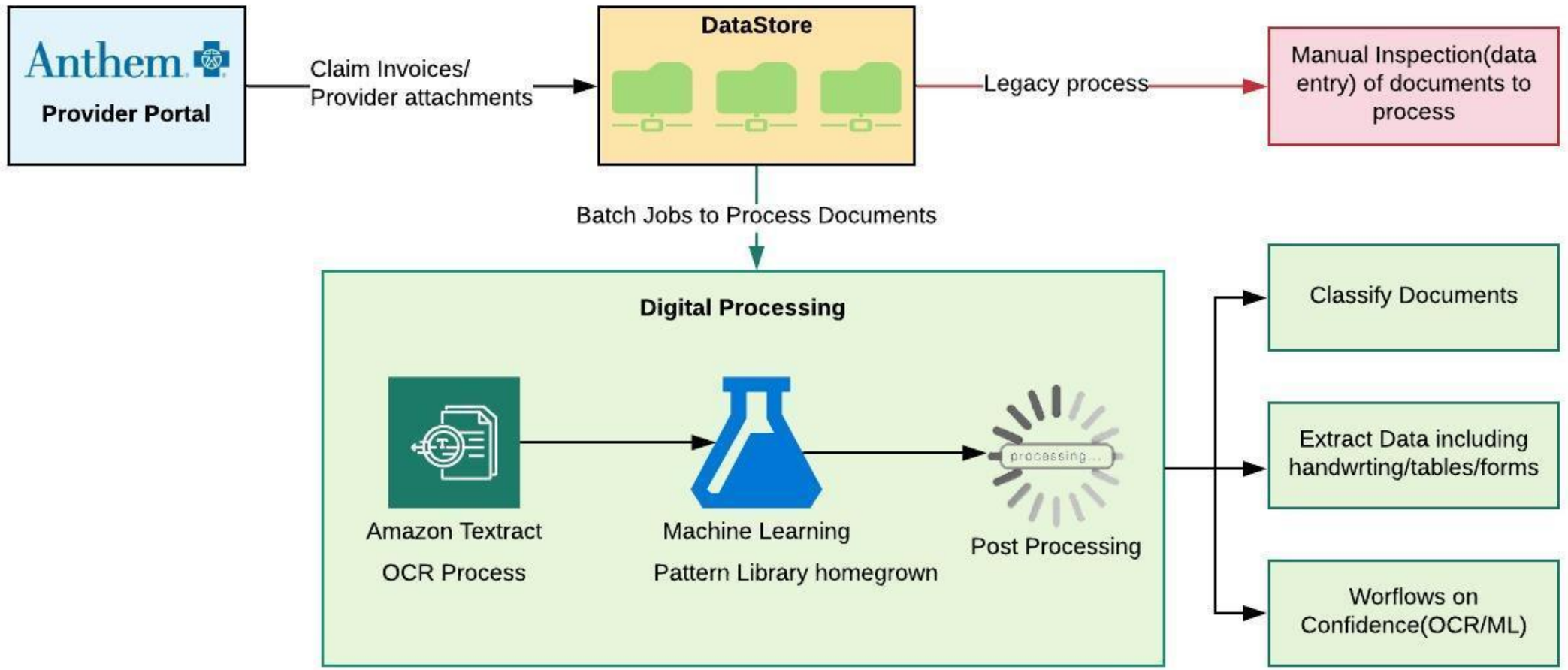


Security and compliance to handle sensitive information and alignment with Anthem standards



Performance and accuracy of OCR extraction

Claims digitization workflow



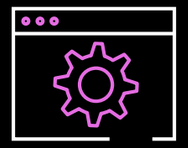
Four steps to implement Amazon Textract at scale



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

4 steps to production with Amazon Textract



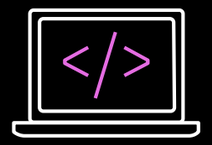
Console / CLI

Upload a sample document into the Amazon Textract console



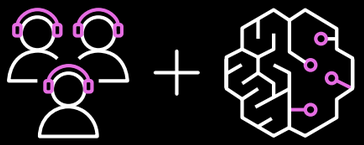
Sample set

Test a sample set of documents to gauge accuracy



Post process

Use code samples to tweak output



Augmented AI

Get sensitive data reviewed by humans

Console / AWS CLI: Upload a document

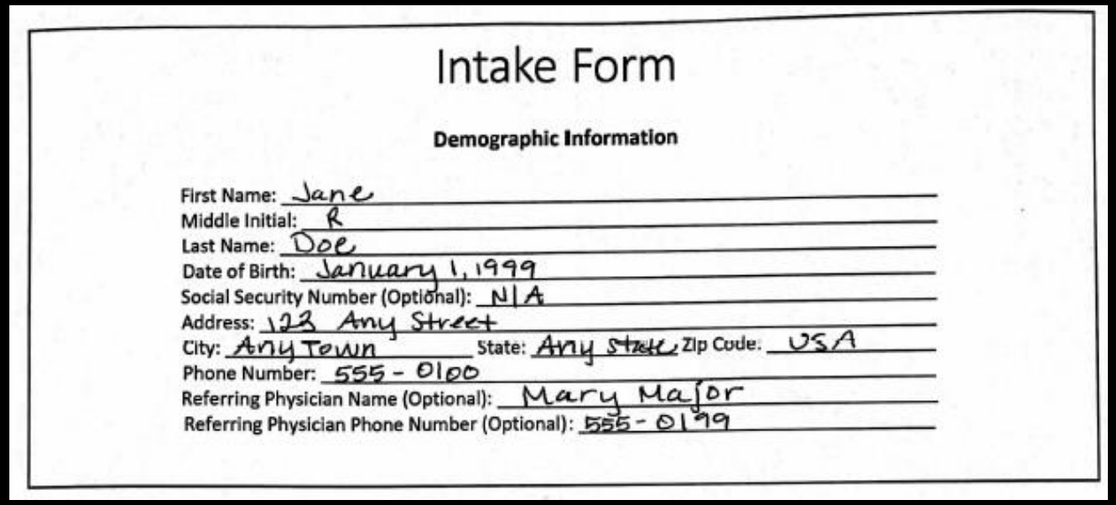


AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Upload a document into the console

Input

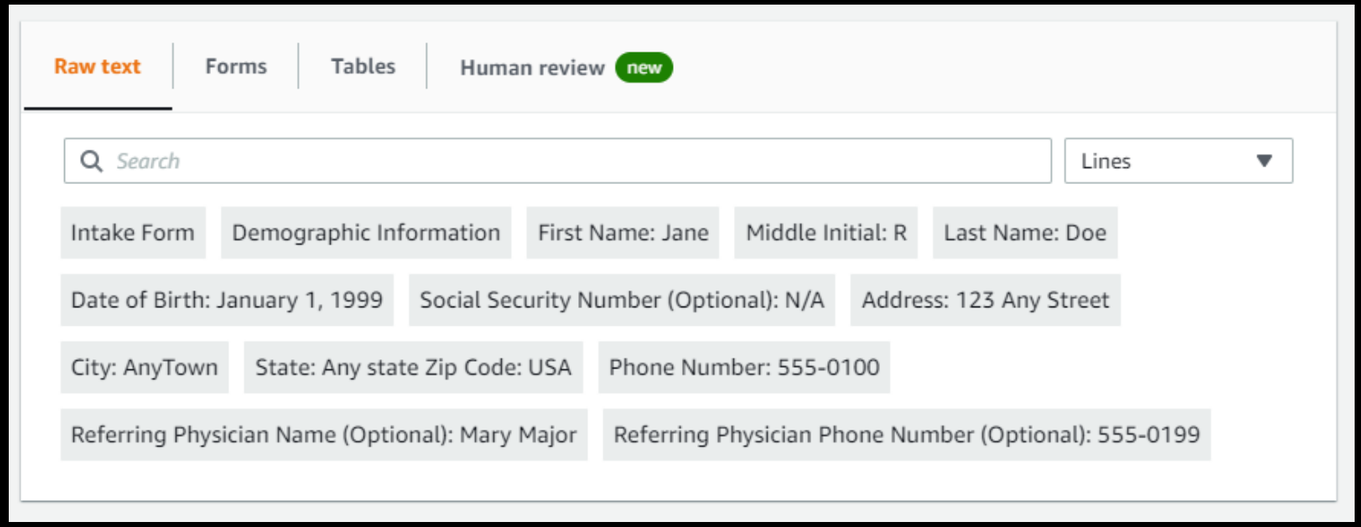


Intake Form

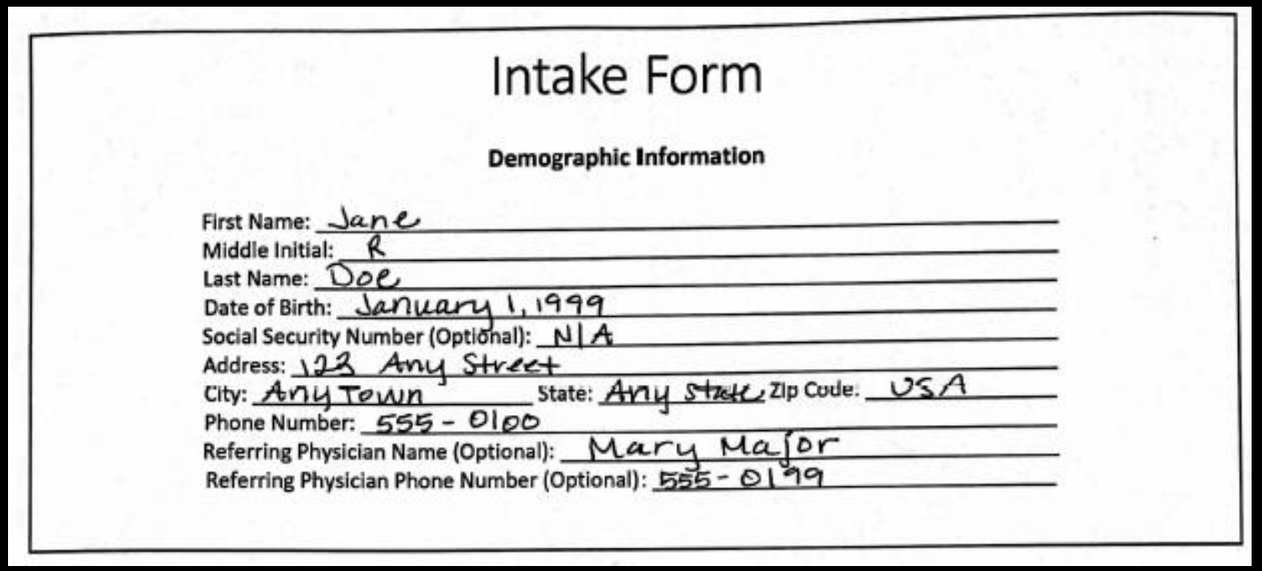
Demographic Information

First Name: Jane
Middle Initial: R
Last Name: Doe
Date of Birth: January 1, 1999
Social Security Number (Optional): N/A
Address: 123 Any Street
City: AnyTown State: Any state Zip Code: USA
Phone Number: 555-0100
Referring Physician Name (Optional): Mary Major
Referring Physician Phone Number (Optional): 555-0199

Output



Or, use the AWS CLI



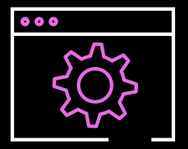
Intake Form

Demographic Information

First Name: Jane
Middle Initial: R
Last Name: Doe
Date of Birth: January 1, 1999
Social Security Number (Optional): N/A
Address: 123 Any Street
City: Any Town State: Any State Zip Code: USA
Phone Number: 555-0100
Referring Physician Name (Optional): Mary Major
Referring Physician Phone Number (Optional): 555-0199


```
{"BlockType": "LINE",
"Confidence": 99.97123718261719,
"Text": "City: Any Town",
"Geometry":
{"BoundingBox":
{"Width": 0.17882174253463745,
"Height": 0.06146746873855591,
"Left": 0.149515300989151,
"Top": 0.681251585483551},
"Polygon":
[{"X": 0.149515300989151, "Y": 0.681251585483551},
{"X": 0.32833704352378845, "Y": 0.681251585483551},
{"X": 0.32833704352378845, "Y": 0.7427190542221069},
{"X": 0.149515300989151, "Y": 0.7427190542221069}]},
"Id": "2f33a0dd-c59d-4c44-b801-23c58515b465",
"Relationships": [{"Type": "CHILD",
"Ids":
["9c779742-f71f-4e7a-91c9-d0d2662d826d",
"92650c62-1395-43b6-8f55-619ed63ae72a",
"9e844ceb-c290-462a-a454-a089ab8a692c"]}]}
```

Even better – use a code sample



```
python call_textract.py --s3-url rajkumar1/Picture3.png|python pretty_print_textract_result.py
```

		Previous Employment History		
Start Date	**End Date**	**Employer Name**	**Position Held**	**Reason for leaving**
1/15/2009	6/30/2011	AnyCompany	Assistant Baker	Family relocated
7/1/2011	8/10/2013	AnyCompany Gas	Baker	Better opportunity
8/15/2013	present	Example Corp.	Head Baker	N/A, current employer

```
|------------|----------|----------------------|----------------|----------------------|
|            |          | Previous Employment  | History        |                      |
| Start Date | End Date | Employer Name        | Position Held  | Reason for leaving   |
| 1/15/2009  | 6/30/2011| AnyCompany           | Assistant Baker| Family relocated     |
| 7/1/2011   | 8/10/2013| AnyCompany Gas       | Baker          | Better opportunity   |
| 8/15/2013  | present  | Example Corp.        | Head Baker     | N/A, current employer|
```

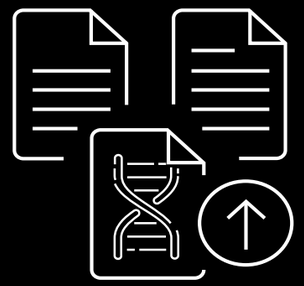
Process a sample set



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

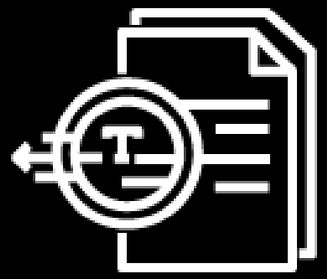
Textractor

```
amazon-textract-textractor-master\src\output>python ..\textractor.py --documents
..\documents\document_10.jpg --text --forms --tables
```



Input

Sample documents like medical claims, insurance forms, etc.



Textractor

Calls Amazon Textract



Output

Raw text, forms, tables, JSON output

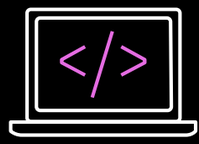
https://github.com/aws-samples/amazon-textract-textractor

Optional: Post processing



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

JSON Response Parser Library



```python
# Print detected text
for item in response["Blocks"]:
    if item["BlockType"] == "LINE":
        print (item["Text"])
```



```python
# Print detected text
for page in doc.pages:
    for line in page.lines:
        print(line.text)
```

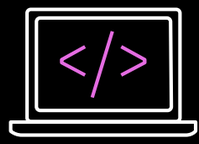
```python
# Print fields
for field in page.form.fields:
    print("Field: Key: {}, Value: {}".format(field.key.text, field.value.text))

# Get field by key
key = "Phone Number:"
field = page.form.getFieldByKey(key)
if(field):
    print("Field: Key: {}, Value: {}".format(field.key, field.value))

# Search fields by key
key = "address"
fields = page.form.searchFieldsByKey(key)
for field in fields:
    print("Field: Key: {}, Value: {}".format(field.key, field.value))
```

https://github.com/aws-samples/amazon-textract-response-parser

Find right key with poor document quality



```
import difflib

extractedKeys = [“Patient's nam and addres", “Gender", “Insurance"]

keyToFind = “Patient's name and address"

matchedList =difflib.get_close_matches(keyToFind,extractedKeys, n =
1,cutoff = 0.6)

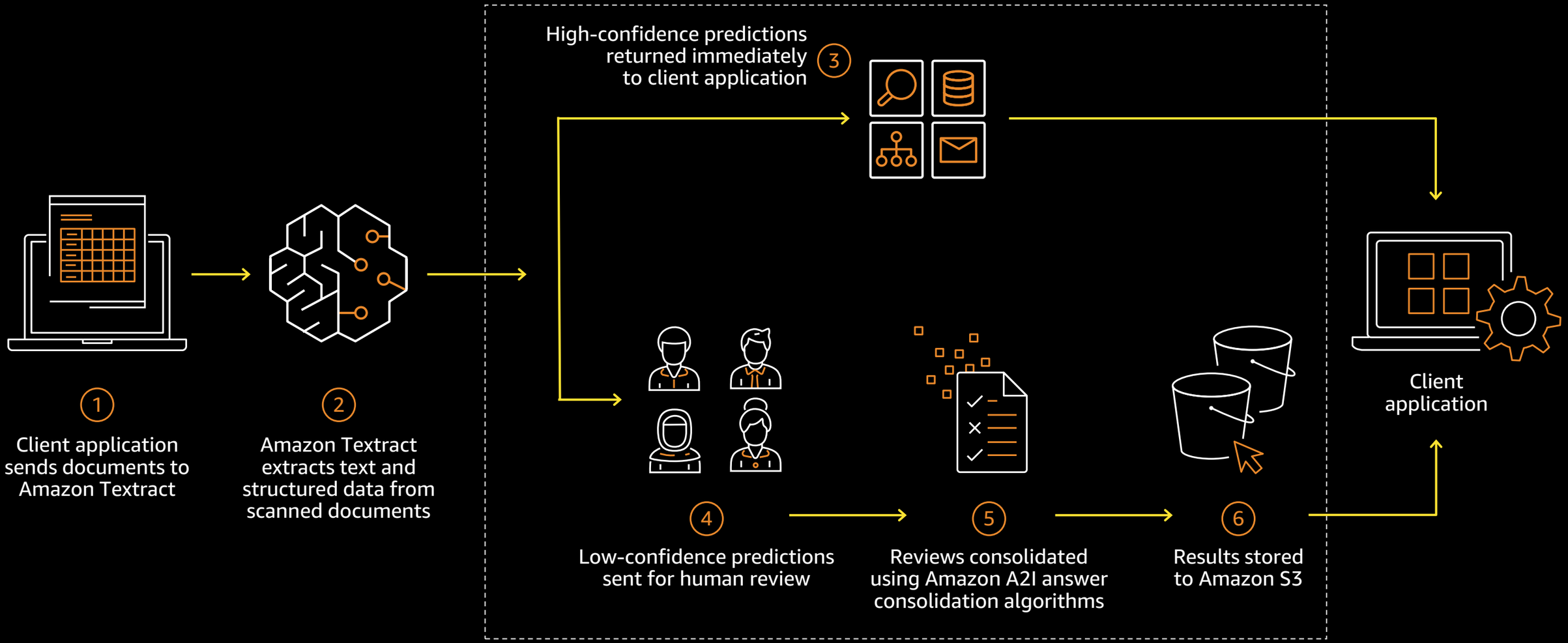
print(matchedList)
```

Optional: Augmented AI

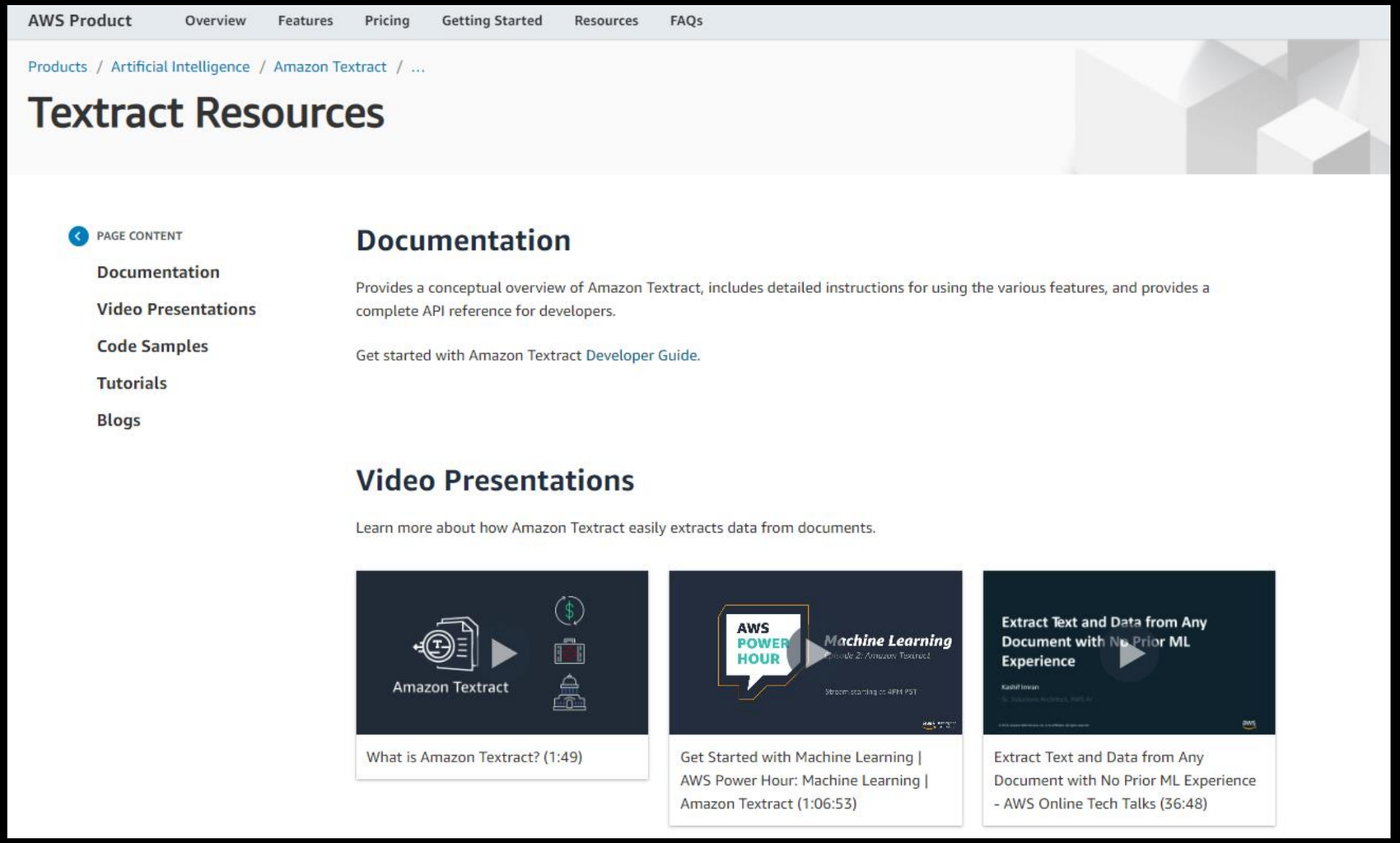


© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Native integration with Amazon A2I



Amazon Textract Resources



www.aws.amazon.com/textract/resources

Thank you!



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Please complete the session survey

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.