Knowledge Graphs Using Cloud Services

Pruthvi Raj Venkatesh[a], Chaitanya Kanchibhotla[b], DVLN Somayajulu[c], and Radhakrishna P[d]
[a] Ambedkar Institute of Technology. pruthviraj_v@hotmail.com
[b,c,d] NIT Warangal. ckanchibhotla@gmail.com, soma@nitw.ac.in, prkrishna@nitw.ac.in

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online:28April 2021

Abstract: Typically, industries store varied types of data in both structured and unstructured formats. This data is very vast and valuable as it is collected over many years. This data is present in multiple data sources, instances and is generated using expensive business processes. Though some of this data is old, it is still relevant in the present context as it provides valuable leads for the current ongoing study. One of the potential problems in any industry is to identify and extract knowledge from these varied data sources as they are: (1) geologically spread, (2) extracted from diverse systems categorized as structured or unstructured data (3) incomplete knowledge of data present and (4) high retrieval time and cost. Even though there are RDBMS databases for storing structured data and document management systems like SharePoint to organize and search unstructured data, there is a need for an efficient system that can link relational databases along with the knowledge present in unstructured data to produce a single knowledge repository and show in the form of a knowledge graph(KG). The knowledge graph should also be supplemented with additional functionalities like search, knowledge extraction, storage, and maintenance. This paper proposes a novel cloud-based approach to generate a knowledge graph by indexing structured and unstructured data and creating a single knowledge graph. We have provided details about implementation approaches in two popular cloud providers, namely Azure and AWS. We have implemented the approach on the dataset provided by the Bureau of Safety and Environmental Enforcement (BSEE) [4], which belongs to the oil and gas domain. The concepts detailed in the paper can be implemented using other cloud providers and can be extended to other industries.

Keywords: Well Data, Oil and Gas, Cloud, Azure, Azure Cognitive Search, Custom Vision Service, Forms Recognizer Service

1. Introduction

The Exploration team in the oil industry collects a lot of information in huge volumes and variety. Collecting this data and finding the information of importance is difficult as data is spread across multiple databases, file servers, content management systems, and various digitized subsurface documents. Because of the diverse nature of the data, exploration users such as geologists must refer to multiple systems separately to understand and arrive at a decision. As this procedure needs a significant analysis time, oil industries rely on analytic vendors for insights as they do not have the required technical expertise. Moreover, it is difficult to procure the base infrastructure to process diversified data and provide possible insights to drive the business decision. Many oil industries spend millions of dollars to buy this data from various vendors.

Data generated by Exploration team in the oil industry is valuable as this information is used for critical decisions such as finding oil well locations for drilling, location of interest for seismic study and so on. Structured Exploration data is typically stored in RDMS databases such as SQL and Oracle. These databases follow a standardized data model that has evolved over many years of experience and is efficiently modeled to store diverse structured data generated in the subsurface space. Oil industries also customize the default standard data model to suit their business. Though RDBMS databases are efficient in storing structured data, few critical pain points limit its implementation and ease of usage.

1. In Oil and Gas domain, relational database schemas will be huge (typically spreading over 1000 tables). Understanding this data requires skilled resources with domain knowledge.
2. As the relational database is developed to handle multiple scenarios, tables in the database contain multiple columns, all of which may not be useable in implementation.
3. As the schema is highly normalized, the addition or removal of columns requires domain knowledge and a complete understanding of the entire relational database.
4. Schema change is tedious due to the highly normalized structure and interdependency between multiple consuming applications.
5. RDBMS database is not designed for unstructured data.

Another vital source of information in the oil industry is unstructured data. Unstructured data usually comes from Microsoft Office documents, PDF files, images, and text information. The two most common formats in which the data is organized in these documents are:

1. Paragraphs text: Critical information is present in sentences of a paragraph; for example, in the sentence "crude was found in multiple wells in the California River basin." California River basin becomes critical information that conveys information related to Well existence.
2. Table format data: Information in the form of tables with or without column headers and line separators with value series specified against the column.

Information extraction from unstructured data have multiple challenges like

1. Tool Procurement & Setup: Oil industries invest in procurement and setup of tools in their environment. This activity is time-consuming and expensive, depending on an organization's procurement process and infrastructure setup complexity.
2. Domain Knowledge Requirement: To extract knowledge from text data, ontology setup is a mandatory step, which requires the involvement of highly skilled domain experts.
3. Table data extraction from Unstructured Data: Extracting structured information from tables/forms has challenges since the existing tools are not capable of handling all the scenarios:

- o Value of data extracted: The value of information extracted from structured data in table format can only be seen if the extracted data is populated into the RDMS database for query and analytics.
- o Diverse formats: Table information can be in various formats and varied header layouts.
- o Image Quality: Table information in images poses challenges due to low image quality.

This paper aims to present a cost-effective cloud-native approach using the graph database to solve the above problems. The paper discusses techniques to provide a comprehensive view of data from the various data sources, batch clusters for parallel processing of unstructured data, Search and cognitive skills for the entity, and key-value extraction. Following are the highlights of the paper:

1. Propose a knowledge extraction framework that uses Platform as a Service(PaaS) Batch services as a high-performance computing platform to process and extract information from documents and significantly reduce the overall processing time and cost.
2. Introduced an approach for using cloud search services and cognitive skills to index the extract data and create value from non-standard document stored in complex folder structures.
3. The paper proposes PaaS services to minimize setup time and support cost, thus reducing the overall initial setup time and cost.
4. Discusses techniques to enable business users to use machine learning(ML) and artificial intelligence (AI) data extraction rules using cognitive skills to extract information rather than relying on domain teams to define an ontology for data extraction.
5. Introduces a novel approach to extract knowledge and store it in a graph database to facilitate relationships and search and provide a unified view to business users from various data sources.
6. Suggests a technique of using search service incremental crawling to update and maintain knowledge graphs so that the most recent information is available to the users with very minimal effort.

The rest of the paper is organized as follows. Section II describes the concepts used in this paper, such as Azure and its relevant services. Section III details all the steps in the proposed method. Section IV presents the proposed method's performance and results, and section V concludes the paper.

2 Literature Survey

Researchers have done a considerable amount of work in the area of knowledge extraction. Nawroth et al. [5]cloud-based services to create a taxonomy and extracted named entities to form the knowledge sources. They classified the documents using SVM. Mittal et.al.[Mittal] used Hadoop-based infrastructure to extract knowledge from unstructured legal text documents. They also used NLP techniques in the cloud to speed up the processing. Fan et.al.[8]extracted knowledge from large-scale documents using a two-stage approach. In the first stage, they extracted knowledge from a large collection of documents using shallow knowledge. In the second step, they applied semantics for knowledge extraction. Nitya Kumari et al. [11] presented a cloud-based framework for extracting knowledge using text mining techniques from PubMed literature.

3 Basic Concepts

This section introduces all the basic concepts used in this paper, along with a brief explanation:

3.1 Subsurface

Subsurface is a part of the oil industry primarily related to the study of contents below the earth's surface and exploring natural hydrocarbon reserves. The subsurface team primarily consists of Geologists and Geophysicists who refer to various exploration study documents such as seismic sections, seismic reports, well logs, and well reports to find potential natural carbon reserves. The exploration study would determine locations where the oil industries can set up the necessary hydrocarbon exploration infrastructure.

3.2 Custom Vision

Custom Vision Service (CV) is an AI service provided by cloud service providers (CSP). It is a platform for identifying image characteristics and object identification to a business problem. Most cloud-hosted CV platform provides a user-friendly interface that allows users to create, train, and publish custom computer vision models and expose them as an application programming interface(API) in the cloud platform.

3.3 Forms Recognizer

Forms Recognizer Service is a document extraction service that is part of the AI platform provided by CSP. This service can be used as a document extraction service to automate information extraction from forms. Form Recognizer uses machine learning to accurately extract text, key/value pairs, and tables from documents. The

service provides labeling tools to facilitate the creation of form extraction rules and models. The models will be published as REST service and consumed in applications.

3.4 Graph Database

A Graph database is mainly designed to store the relationships between data elements. A graph database can be represented as a collection of nodes and edges where nodes represent entities and edges represent relationships. Graph databases are advantageous in use cases such as social networking, recommendation engines, and fraud detection. This paper shows a novel approach on how cloud services can be used to populate a graph database to store relations between entities such as well, basin.

3.5 Search Service and cognitive skills

Search service provided by CSP enables searching and indexing of documents uploaded to cloud storage. The default capabilities of search indexers can be further enhanced using cognitive skills for knowledge extraction from documents. Examples of cognitive skills associated with the default search pipeline include custom skills such as OCR, entity extraction, forms extraction, and so on.

4 Proposed Approach

This section presents the proposed framework for knowledge extraction from digital media and generates a knowledge graph using the extracted information.

4.1 Knowledge Mining Framework

Figure 1 shows the process overview diagram describing the sequence of steps involved in the knowledge extraction framework. The steps are numbered in sequence for easy understanding.

1. Document Store: Data from diversified data sources are consolidated in the cloud blob storage service. Every cloud provider offers this service for customers to store documents and images of varying sizes. Multiple transfer options are also provided to move the data to the cloud, depending on the total data size to be transferred into the cloud.

2. Prepare & Train: This step involves executing a series of prerequisite activities for the knowledge extraction process. This step consists of the following.

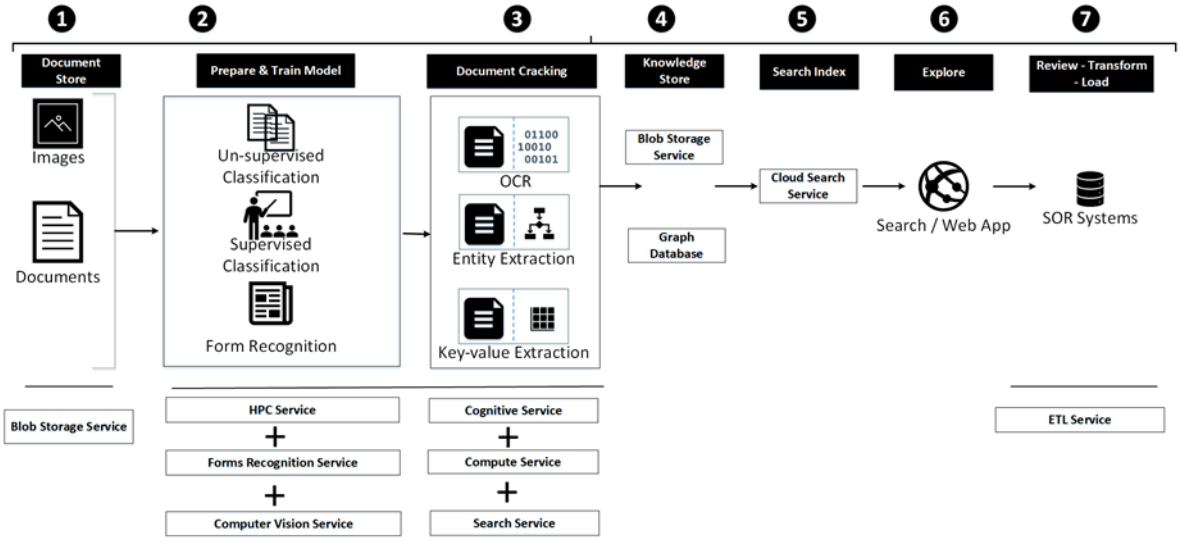

Figure 1 : Knowledge Extraction Framework

o Unsupervised Clustering: The activity in this step involves the extraction of images from documents. The extracted images and the documents are further classified using an unsupervised clustering model based on the features. The output of the model serves as the first-level indication of information present in the document store. It is used as an input to geologists for identifying the critical information from the vast amount of data loaded. This filtering step significantly reduces the overall time consumed in the process. This activity uses batch service provided in the cloud for parallel processing and reduces the overall execution time.

o Supervised Classification: This activity involves configuring custom vision service instances in the cloud for creating a supervised model for image classification. The classification model is trained to identify critical images from the vast number of images broadly classified in the previous step using the unsupervised clustering. The classification model is exposed as a REST service and used during the search indexing process.

o Form Recognition: This activity involves using the forms recognizer service to train a model for extracting key-value pairs from the classified images. The model creation involves loading every critical image and marking fields of interest using annotation tools provided by the CSP. Once the annotation is completed, the model is trained with multiple images of the same category to facilitate the extraction of key-value pairs. The model is then published as a REST service for consumption from the search service during the indexing process.

3. Document Cracking: This activity involves extracting knowledge from the document using the search service indexing process. The default search service capability can be enhanced by associating cognitive skills with the search pipeline. Following skills are used for knowledge extraction.

o OCR: This cognitive skill is used for the extraction of text from embedded images. Most CSP provide OCR capabilities through cognitive skills.

o Entity Extraction: This activity involves extracting entities from documents and OCR output using Entity Recognition cognitive skills and Custom Entity Lookup cognitive skills. Entity Recognition skills help extract names of entities such as a person, location, and organization. Custom Entity Lookup cognitive skills are used for extracting key terms from a custom user-defined list such as Well Names, Basin Names.

o Key-Value Extraction: This activity involves extracting key-value pairs from text or tables present in images and storing them into the Graph database. The implementation involves 1) Triggering custom vision REST service to identify the type of document (2) Trigger forms recognizer REST service to extract the key-value pairs (3) Load the extracted values into the search index and graph database.

4. Knowledge Store: Information extracted from the search service indexing pipeline is loaded into the search index and graph database. Search index will hold the indexed data from diverse data sources, and the graph database holds the entities and relationships between the entities derived from the search index. This novel framework uses the concept of multiple data source indexing to interlink the relation between multiple entities in a single search index. In this paper, we have indexed the following 4 data sources into a single index:

o Master data of the documents extracted from excel sheets from the BSEE website [BSSE]

o Entire text content extracted from the document.

o Entities extracted from the document.

o Key-Value or table data extracted from images. This approach can be extended to many data sources, giving a single search index to query and get all the related entities in a single integrated search result.

5. Search Web App: A web application is used to display the search results in a custom web interface.

6. System of Record(SOR) Data Load: The consolidated search index is exposed as REST services. An external application such as System Of Record can query the search index using search API and load it into a record system.

4.2 Solution Overview

In this subsection, we present the technical components along with the flow of events in the framework. Figure 2 displays the logical layers used in the proposed solution.

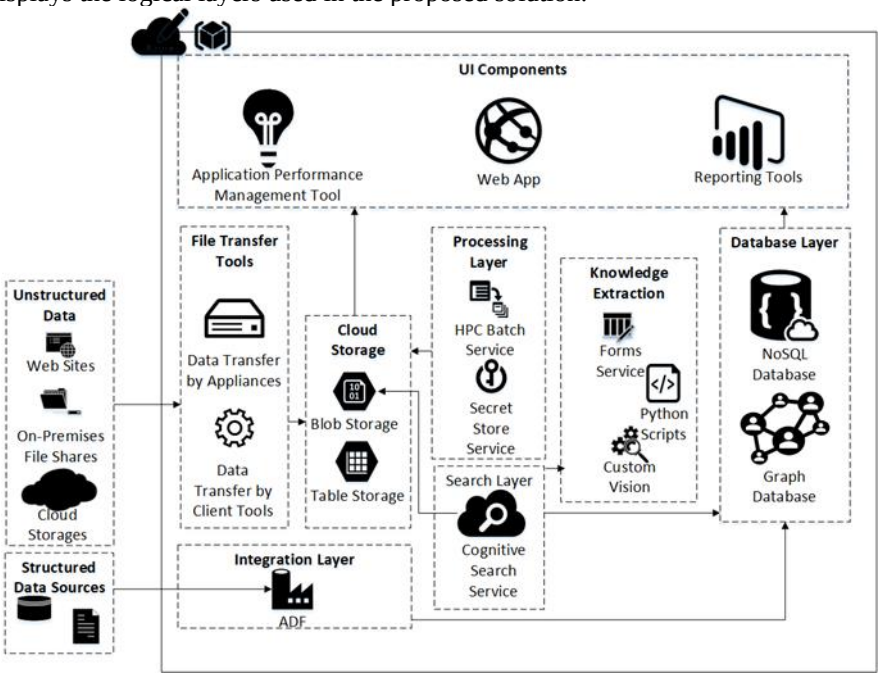

Figure 2: Solution Overview

1. File Transfer Tools: Data from various content management systems such as OneDrive, shared drives, SharePoint are transferred to the Blob storage service using data transfer tools depending on the CSP and the total content transferred.

2. Blob Storage: Blobs storage service will store the content transferred using Blob transfer tools and become the data source for the knowledge extraction framework. The table storage service is used for storing solution configuration, giving the flexibility for dynamic changes in the configuration such as source data location, data access secrets

3. Processing Layer: Contains PaaS components like web hobs, high process compute(HPC) batch service to provide the computational setup for clustering.

4. Knowledge Extraction Layer: This layer contains python scripts triggered through the HPC batch service for clustering, custom vision service for image classification, and forms recognizer service for keyword extraction.

5. Search Layer: This consists of CSP's cognitive search service to index the data sources and triggers cognitive skills in the search pipeline(refer to Section 4.3) to enhance the search index.

6. Database Layer: Database Layer is composed of the NoSQL database and Graph database. NoSQL Database is used as a database for storing metadata related to the documents and the output of key-value extraction modules. A graph database is used for storing entities and entity relations ships.

4. UI Components: This layer consists of reporting tools such as Power BI, Tableau [10] for statistical reporting, Web Application to display table extraction results, performance management tools for statistical analysis of application performance such as web response, search crawl time, search service response time. The application team will use this statistical reporting on performance numbers to remediate performance issues.

5. Integration Components: The integration layer consist of Extract, Transform, and Load (ETL) services to import metadata related to the documents or domain master data into the cloud database system. This imported data is used during the indexing process to associate additional metadata with the documents being indexed.

4.3 Knowledge Mining Workflow

In this subsection, we present our approach to generate a knowledge graph using cognitive search capabilities. Figure 3 shows the overall workflow. The workflow consists of four steps.

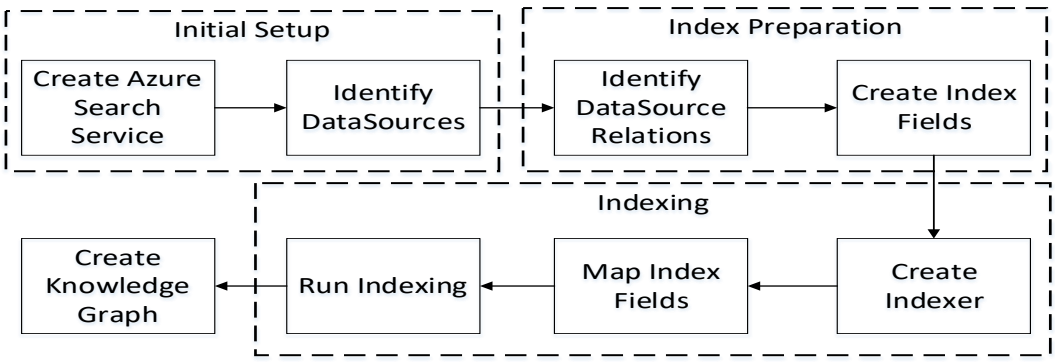

Figure 1: Knowledge Mining Workflow

1. Initial Setup: This step involves creating a cloud search service to configure the indexer and search index. We identify all the external metadata sources that have an association with the documents uploaded in the blob. The external data could include data sources such as relational databases, form fields embedded as structured data or images inside the document, and named entity fields present within the document text.

2. Index Preparation: This setup involves the creation of the search index and index fields. The various attributes that must be extracted from the documents and external data sources are identified. A search index field with an appropriate data type to hold the identified data attribute is created in the search index.

3. Indexing: This step involves creating search indexers to index the various data sources and create a single search index consisting of all the related fields from the diverse data sources. Input to the indexer is a mapping schema that maps the source data fields to the destination index field. The extraction of source data may involve a cognitive search pipeline with complex extraction rules to extract the intended data. The complex extraction rules may involve querying the relational database on foreign key relations, extracting images from the document, and executing custom vision API to classify the image and extract fields from images through forms extraction API.

4. Create a Knowledge Graph: The extracted fields populated in the search Index will also be populated into the Graph database through the custom cognitive skill pipeline. The cognitive skill pipeline would involve triggering Cypher[6] or Gremlin queries[3] to create a knowledge graph. The search index fields will be created as nodes and edges based on their relation. The later section details the advantage of the knowledge graph over search in entities relationship search.

5 Oil Industry Well & Basin Indexing Illustration

5.1 Index Definition

This section describes a setup done for an Oil industry where the search index was set up to hold attributes from diverse data sources. The indexing pipeline was set up to populate the extracted fields into a search index and knowledge graph. The scenario involved a digitization exercise where historical paper documents generated many decades back were scanned to reduce the risk of physical data loss due to decay. The digitization vendor scanned the physical paper documents to generated the digitized copy and a metadata file holding the scanned file attributes. The requirement was to load the scans and the corresponding metadata file into the cloud and complement them with additional metadata from diverse data sources. We have tried to illustrate this setup in Figure 4, where a simple pictorial representation of the multiple content sources is represented. We have restricted the number of columns for simplicity. The black boxes in the diagram capture the process involved that generate the data attributes, and the grey box capture the data repository where the attributes will be present, and the grey shaded box represent the primary key in the data repository:

1. Load Document Catalogue: The metadata file generated was loaded into the No-SQL database using a data transformation service. The metadata file contained four fields called Data Type, Country, Document Name, and Document Location. Loading of data into the No-SQL database automatically generates an "ID" field that uniquely identified the record. We have used this field to bind the data from multiple data sources into a single index file.

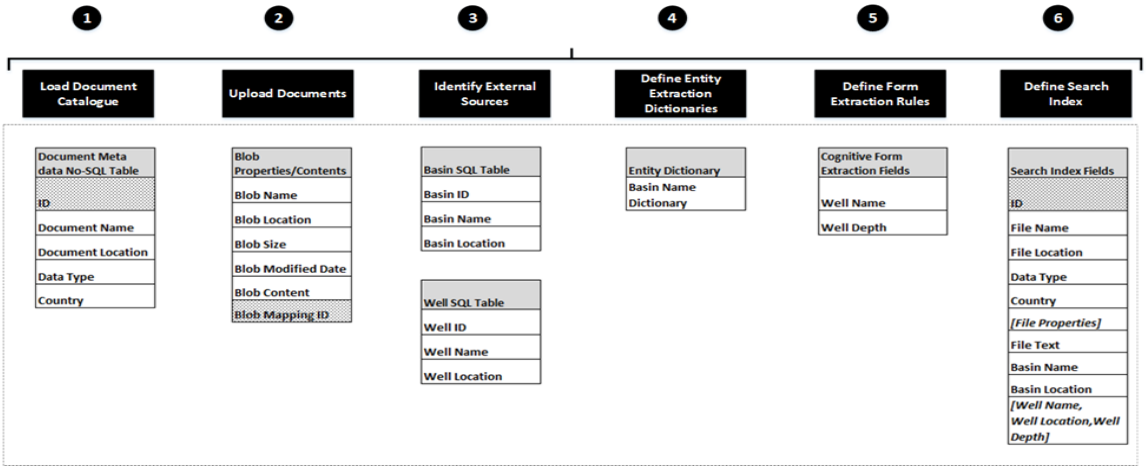

Figure 4: Index Definition

2. Upload Documents: The scanned files provided by the digitization vendor were uploaded to the blob, which automatically generated the blob attributes. The custom loading process added an attribute called "Blob Mapping ID," which will hold the "ID" field value from the corresponding No-SQL record.
3. Identify External Data Sources: External data involved, populating metadata from two data sources. Scanned documents will have a mention of Well or Basin names. The Well location and Basin location will be fetched and associated with the search Index and knowledge graph database.

o "Well" SQL table: Contained details about Well such as Well Location of an oil well.

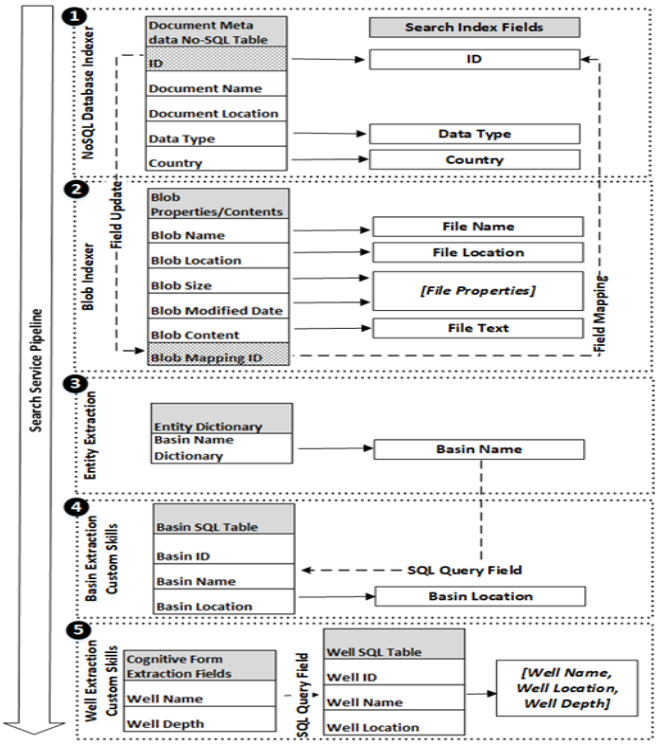

Figure 5: Search Indexing Pipeline

o "Basin" SQL table. Contained details about Basin. Basin is a physical location on the earth and will hold multiple wells. Basin can also span multiple countries.

4. Define Entity Extraction Dictionary: Basin Names were loaded into a dictionary and used as input for entity extraction. Entity extraction skills in the indexing process involve finding occurrences of Basin names in the scanned documents and associating the Basin details with the document.

5. Define Form Extraction Dictionary: Cognitive form extraction rules are defined to extract Well Name and Well Depth from scanned Well diagrams.

6. Define Search Index: A Index is defined that contains all the fields from the various data sources.

5.2 Index Pipeline Configuration

The following four indexers will be created to crawl and index the data sources, as indicated in Figure 5. The diagram shows a novel approach to create a knowledge graph using the cloud's PaaS services.

1. No SQL Database Indexer: Indexes No SQL database and stores the Id, Data Type, and Country field into the search index.

2. Blob Indexer: Indexes blob data that contains all the documents and populates the following into the index file:

o Blob properties such as Blob Name, Location, Size, and Modified Date

o Blob contents extracted as text into the File Text index field.

o Blob Mapping ID populated during the upload process will be used as a mapping field to merge the index file created during the No SQL database indexing with the index created out of Blob Indexer.

3. Entity Extraction Skill: Reads the File Text field extracted from the Blob Indexer and feeds it to the Entity Extraction cognitive skill that extracts Basin Name. Basin Names will be embedded with the text, and the Entity Extraction skill uses the dictionary of basin names configured with the skill to looks for Basin Names and store them into the Basin Name Index Field.

4. Basin Extraction Custom Skill: Uses the Basin Names extracted by the Entity Extraction skill to query the Basin SQL data table based on Basin Name and fetch Basin details such as Basin Location and populate the queried value into the search index.

5. Well Extraction Custom Skill: Performs the following steps

o Extract images embedded within the document using inbuilt image extraction skills.

o Triggers the custom vision REST APIs to classify the image.

o Triggers forms extraction API to extract Well Name and Well Depth from images.

o Uses the Well Name extracted in the previous step to query the Forms SQL table and populate the queried fields into the Index Field.

6. Graph Database Custom Skill: Custom skill associated with the blob indexer that triggers graph database insert queries to create a knowledge graph from the index fields. Fig 6 shows an example of the generated graph.

5.3 Search & Graph Database Queries

This section discusses the graph database and search index advantages, where the entity relation is captured.

1. One of the key benefits is the ability to query a single data source where data from multiple data sources is consolidated. The users need not worry about access to multiple data sources to get the information needed.
2. Flexibility to add additional fields from data sources.
3. Most Graph database provides a pictorial view of the data and the relation used to understand clustering details.

We have used the example of the graph generated in Fig 6 to illustrate some of the advantages of using graph databases. The advantages are detailed in Table 1. The Oil industry's specific requirements are captured, and details on how this information can be retrieved using Search Queries and Graph Database Cypher Query[6].

Table 1 - Sample Query from Graph and Search Index

Scenario	Search Query	Cypher Query

List all documents which have a mention of Well Name "ACADIA" in the document.	POST /IndexURL?api-version=verNo
	{
	"search": "*",
	"facets": ["WellName,values: ACADIA "]
	}
		MATCH (D:DOCUMENT)-[:has_Well]->(W:Well) WHERE W.WellName = "ACADIA" RETURN D.FileName, D.FileLocation

List all documents which have mention of Country "US" and are associated with Basin "California River"	POST /IndexURL?api-version=verNo
	{
	"search": "*",
	"facets": ["Country, Basin, values: US , California River"]
	}
		MATCH (D:DOCUMENT)-[:has_Country]->(C:COUNTRY) WHERE C.CountryName = "US" WITH D,C MATCH (D)-[:has_Basin]->(B:BASIN) WHERE B.BasinName = "California River" RETURN D.FileName,D.FileLocation

List all documents associated with Wells where the Well Depth is between "450" and "800" units.	POST /IndexURL?api-version=verNo
	{
	"search": "WellDepth ge 450 and Rating le 800"
	}
		MATCH (D:DOCUMENT)-[:has_Well]->(W:Well) WHERE W.WellDepth >= "450" AND W.WellDepth <= "800" RETURN D.FileName, D.FileLocation

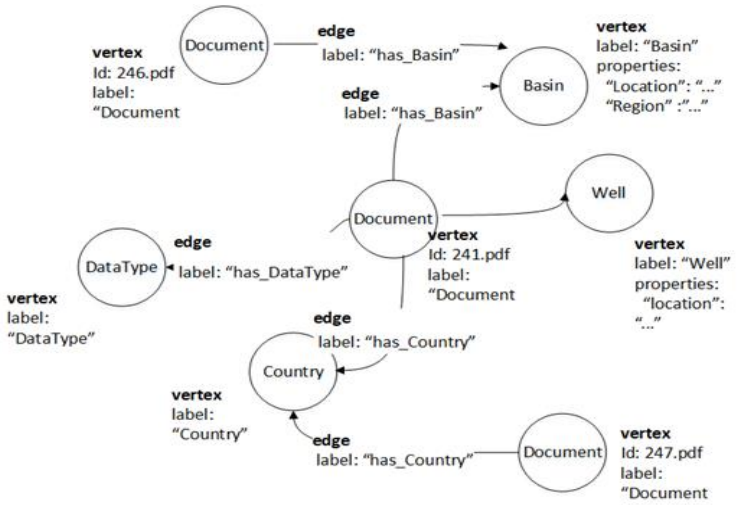

Figure 6 : Knowledge Graph

5.4 Azure and AWS Services

This section details the various cloud services used in the implementation and their corresponding offerings by the two most popular cloud services, Azure and AWS.

Table 2 - Cloud Services

No	Cloud Service	Details	Azure Offering[7]	AWS Offering[2]
1.	Blob Storage	Blob storage is a scalable storage offering provided in the cloud for object storage. This service can be used for storing files, media, and images	Azure Blob Storage	Amazon Simple Storage Service(Amazon S3)
2.	HPC Batch Service	Batch service is an offering provided by the cloud to run batch computing	Azure Batch Service	AWS Batch

		workloads. Batch computing provides resources on a need basis and manages the creation and maintenance of underlying infrastructure.		
3.	Graph Database	A graph database is an optimized database service used for storing relationships.	Azure Cosmos Database	Amazon Neptune
4.	NoSQL Database	NoSQL database service is a scalable document database service.	Azure Cosmos Database	Amazon DynamoDB
5.	Cognitive Search Service	Cognitive Search Service is a fully managed service in the cloud that provides features for setting up a search solution for websites and mobiles.	Azure Cognitive Search	Amazon CloudSearch
6.	Secret store	The secret store is a fully managed secrets management service that helps securely encrypt, store, and retrieve credentials used in applications.	Azure KeyVault	AWS Secrets Manager
7.	Custom Vision Service	Custom Vision is a managed service that facilitates the building of learning models to identify images.	Azure Custom Vision Service	Amazon Rekognition
8.	Forms Service	Forms service provides features to build machine learning models to detect key-value pairs in document images and PDF documents.	Azure Forms Service	Amazon Textract

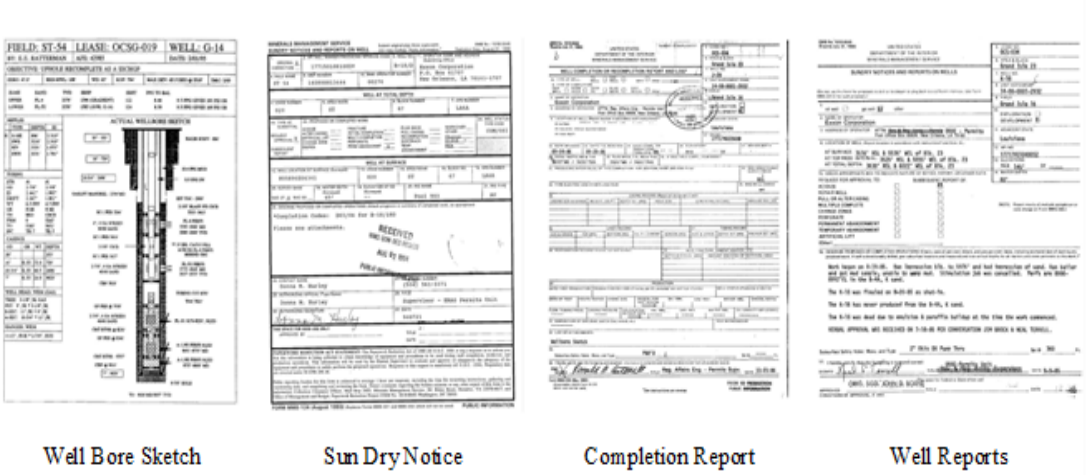

Figure 1 – Image Categories of Test Data Set

6 Results & Discussion

In this section, we present the results generated at each stage of the knowledge extraction framework. For the execution, we have downloaded data from BSSE [4]. The dataset consists of 122 PDF files, and the total number of pages present in all the PDF documents is 2134. Figure 7 shows the four important categories, namely sundry notice, wellbore sketch, completion report, and Well report, that were considered for the implementation. The majority of these images have the following five different formats. (i) images containing only text (ii) images with text and only one table (iii) images with multiple tables (iv) images with wellbore (v) images with wiggly lines.

6.1 Unsupervised Clustering

For Unsupervised Clustering, we have used the pre-trained model called VGG16 present in Keras. The model is implemented in python.VGG16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman [9]. The model is trained with the "imagenet" dataset, a visual database that is publicly available. We have resized the image size to 64*64 for speeding the process and extracted features from each image in the testing phase. The features are stored in the feature list in python. We have used these features to classify the

images. For classification, we have used the k-means algorithm with a cluster size of four as we have five formats of data. Figure 8 shows the results of the Unsupervised Clustering model. The image shows the output generated from a PDF document. The number before the underscore in the file name is the cluster number. It can be observed that the model has returned a decent performance on the test data and the clusters that are formed with images of a similar appearance. The classification results obtained by the unsupervised model were beneficial to train the cloud-native image classification model with training data to facilitate accurate classification. Only the key images from the clustered results selected by the domain expert were used to train the custom vision model in the subsequent step.

6.2 Custom vision classification results

The essential images classified as part of the Clustering model were used to train the Custom Vision model. The custom model was also fed with negative images to get a classification accuracy of more than 90%. Figure 9 shows the Custom vision service interface used in Azure to check the prediction results with a Well Bore Sketch. The results indicate that the model was able to predict wellbore sketch with a probability of 99.9 %. The model was similarly able to accurately predict the other types of images fed into the model. The REST services exposed by the model automatically classified the documents when called from the cognitive search skills. We have only shown results from Azure in this paper.

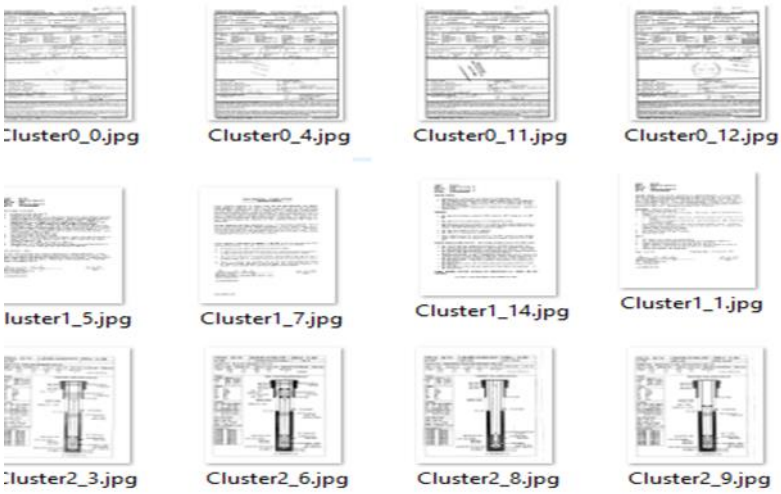

Figure 8 : Unsupervised Clustering Results

6.3 Training Results of Form Recognizer Model

The training set used in Section 6.1 was used to train the form recognizer service to extract key-value pairs from the four different types of images. Figure 10 shows the field extraction accuracy for a Well Bore Sketch diagram using the fott website[1] hosted by Microsoft. The figure indicates that the fields such as Lease, Well, Field, TopDepth, BoreDepth, and BottomDepth are extracted with an accuracy of 100%

6.4 Results of other metrics

Figure 11 shows the average classification accuracy of Azure custom vision service for the input dataset. It can be observed from the figure that the custom vision service has given better results for the images containing diagrams and comparatively less accuracy values for the complete report. One possible reason might be because of text in paragraphs, images, and tables in the report.

Figure 9: Custom Vision Test Ouput(Well Bore Sketch)

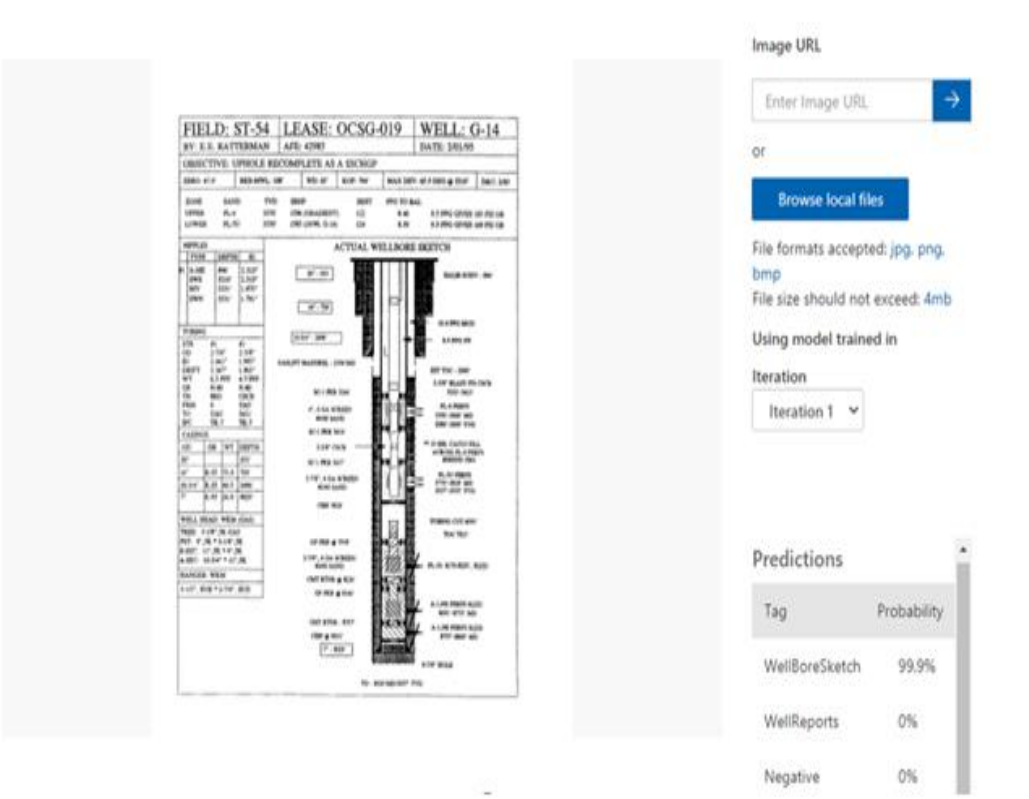

Figure 9: Custom Vision Test Ouput(Well Bore Sketch)

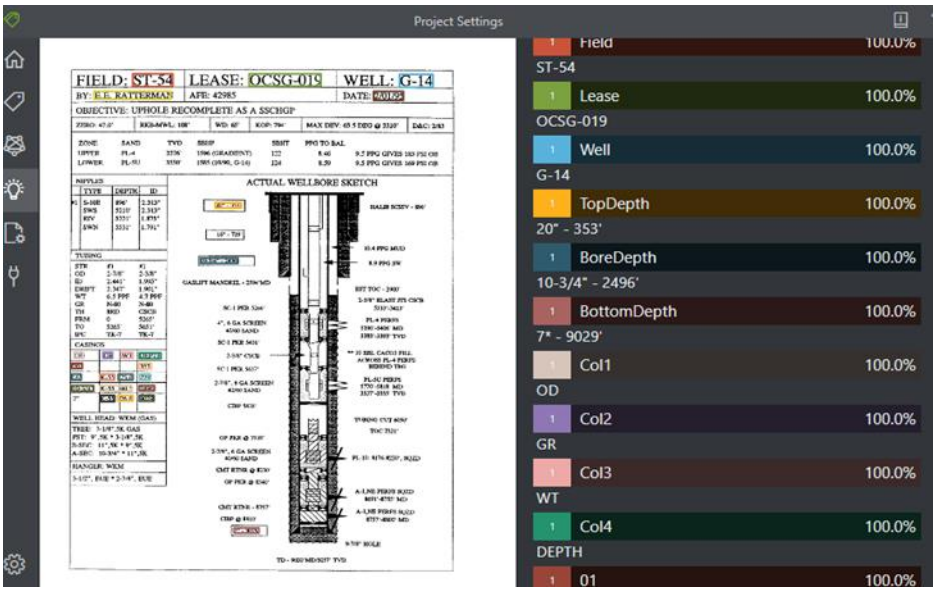

Figure 10: Form Recognizer Test Output(Well Bore Sketch)

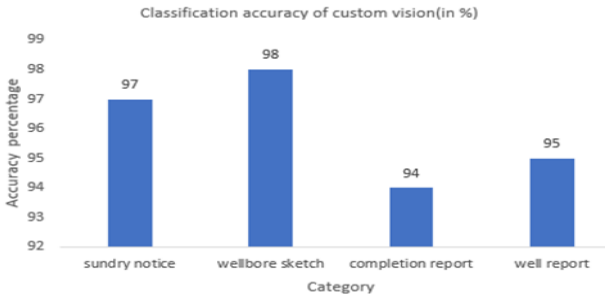

Figure 11: Classification accuracy of forms recognizer for the dataset.

Figure 12: Azure Batch Metrics

Table 2: Table showing the average execution time for all steps

Phase	**Step**	**Average time taken (in minutes)**
Document Store	Upload data to Azure	6
Model execution	Unsupervised Clusterting	7
	Creation of model	60
	Training the model	1
	Supervised classification	2
	Forms recognizer	2
Document Cracking	Entity extraction	5
	Key Value pairs extraction	5
knowledge Store	Creating search indexers	10
	using search indexers	4
Explore	Data display using Azure Web App	2
	Total	**53**

Table 2 shows the average execution time in minutes for all the steps in the process. This time was calculated for 25 images present in the dataset. It can be observed that the majority of the time is taken for two steps, which are marked in grey, namely creating the supervised classification model and creating search indexers. The reason is because of manual operations in both steps. We have manually uploaded the images and provided tags for each image while creating the model. We manually provided the indexer's name for creating the search index and defined the index field by selecting the column type. Kindly note that this is a one-time activity. Figure 12 shows the results from the azure batch. It can be noted that the entire operation took 238.9 minutes for 2134 images from 122 PDF documents. Moreover, we had also observed that 15 tasks had failed because of improper embedding of images in the PDF document.

7 Conclusion

Industries store data in both structured and unstructured formats. This data is present in multiple data sources and instances. This paper proposed a novel cloud-based framework using cloud services to generate an index for storing knowledge from multiple data sources such as blob, NoSQL database, relational database. Cloud AI services such as forms recognizer and custom vision were used in the framework. The data is stored in a search index and knowledge graph(graph database) to quickly make an informed decision. The procedure comprises five stages, starting with loading the data into the blob, executing supervised and unsupervised models, extracting knowledge from the documents in the form of key-value pairs, creating search indexers, and showing the data in a web application. We have done experiments on the dataset from BSEE belonging to the oil and gas domain.

We also presented the measured metrics results such as classification efficiency, total execution time in all the steps, and batch service. It is observed that the proposed framework gave good results in all the steps, and the quality of the data that is extracted is also satisfactory. It is observed that forms recognizer gave comparatively less efficiency for reports containing text in paragraphs, images, and tables in the report. This framework can be implemented using other clouds such as AWS, Google, and their native components.

References

1. Analyze - Form OCR Testing Tool. https://fott.azurewebsites.net/
2. AWS Documentation. https://docs.aws.amazon.com/index.html
3. Azure documentation | Microsoft Docs. https://docs.microsoft.com/en-us/azure/
4. Bureau of Safety and Environmental Enforcement https://www.data.bsee.gov/Other/DiscMediaStore/ScanWellFiles.aspx/
5. Christian Nawroth, Matthäus Schmedding, Holger Brocks, Michael Kaufmann, Michael Fuchs, Matthias Hemmje, Towards Cloud-Based Knowledge Capturing Based on Natural Language Processing,Procedia Computer Science,Volume 68,2015, 206-216.
6. Cypher Query Language - Developer Guides. https://neo4j.com/developer/cypher/
7. Directory of Azure Cloud Services | Microsoft Azure. https://azure.microsoft.com/en-in/services/
8. Fan, J. & Kalyanpur, Aditya & Gondek, D.C. & Ferrucci, D.A.. (2012). Automatic knowledge extraction from documents. IBM Journal of Research and Development. 56. 5:1-5:10. 10.1147/JRD.2012.2186519.
9. Karen Simonyan and Andrew Zisserman, Very Deep Convolutional Networks for Large-Scale Image, Computer Vision and Pattern Recognition,2014
10. List of reporting software - Wikipedia. https://en.wikipedia.org/wiki/List_of_reporting_software
11. Niyati Kumari Behera, GS Mahalakshmi,A cloud based knowledge discovery framework, for medicinal plants from PubMed literature,Informatics in Medicine unlocked,Volume 16,2019,100105,ISSN 2352-9148,https://doi.org/10.1016/j.imu.2018.04.006.
12. S. Mittal, K. P. Joshi, C. Pearce and A. Joshi, "Parallelizing natural language techniques for knowledge extraction from cloud service level agreements," 2015 IEEE International Conference on Big Data (Big Data), Santa Clara, CA, 2015, pp. 2831-2833, doi: 10.1109/BigData.2015.7364092.
13. Search Documents (Azure Cognitive Search REST API) | Microsoft Docs. https://docs.microsoft.com/en-us/rest/api/searchservice/search-documents