Applying Machine Learning NLP Algorithm for Reconciliation Geology and Petrophysics in Rock Typing

Conference Paper · October 2023

DOI: 10.2118/216223-MS

7 authors, including:

Alexey Tveritnev
Abu Dhabi National Oil Company
15 PUBLICATIONS **10** CITATIONS

SEE PROFILE

Almaz Ermilov
UiT - The Arctic University of Norway, campus Narvik
1 PUBLICATION **0** CITATIONS

SEE PROFILE

SPE-216223-MS

Applying Machine Learning NLP Algorithm for Reconciliation Geology and Petrophysics in Rock Typing

A. Tveritnev, M. Khanji, S. Abdullah, and L. Rojas, ADNOC, Abu Dhabi, UAE; A. Ermilov, UiT the Arctic University of Norway; F. Al Mansoori and A. Alblooshi, ADNOC, Abu Dhabi, UAE

Oil- and gas-bearing rock deposits have distinct properties that significantly influence fluid distribution in pore spaces and the rock's ability to facilitate fluid flow. Rock typing involves analyzing various subsurface data to understand property relationships, enabling predictions even in data-limited areas. Central to this is understanding porosity, permeability, and saturation, which are crucial for identifying fluid types, volumes, flow rates, and estimating fluid recovery potential. These fundamental properties form the basis for informed decision-making in hydrocarbon reservoir development. While extensive descriptions with significant information exist, the data is frozen in text format and needs integration into analytical solutions like rock typing algorithms.

Applying text analysis, a crucial area in natural language processing, aims to extract meaningful insights and valuable information from unstructured textual data. With the vast amount of text generated every day, automated and efficient text analysis methods are becoming increasingly essential. Machine learning techniques have revolutionized the analysis and understanding of text data. In this paper, we present a comprehensive summary of the available methods for text analysis using machine learning, covering various stages of the process, from data preprocessing to advanced text modeling approaches. The overview explores the strengths and limitations of each method, providing researchers and practitioners with valuable insights for their text analysis endeavors.

A data-driven rock typing scheme is necessary for decision-making and optimization to achieve the best ultimate recovery of hydrocarbons in the most efficient way.

Introduction

Oil- and gas-bearing rock deposits have distinct properties that significantly influence fluid distribution in pore spaces and the rock's ability to facilitate fluid flow. Rock typing involves analyzing various subsurface data to understand property relationships, enabling predictions even in data-limited areas. Central to this is understanding porosity, permeability, and saturation, which are crucial for identifying fluid types, volumes, flow rates, and estimating fluid recovery potential. These fundamental properties form the basis for informed decision-making in hydrocarbon reservoir development. While extensive descriptions with significant

information exist, the data is frozen in text format and needs integration into analytical solutions like rock typing algorithms.

Applying text analysis, a crucial area in natural language processing, aims to extract meaningful insights and valuable information from unstructured textual data. With the vast amount of text generated every day, automated and efficient text analysis methods are becoming increasingly essential. Machine learning techniques have revolutionized the analysis and understanding of text data. In this paper, we present a comprehensive summary of the available methods for text analysis using machine learning, covering various stages of the process, from data preprocessing to advanced text modeling approaches. The overview explores the strengths and limitations of each method, providing researchers and practitioners with valuable insights for their text analysis endeavors.

A data-driven rock typing scheme is necessary for decision-making and optimization to achieve the best ultimate recovery of hydrocarbons in the most efficient way.

Objectives/Scope

In the study of reservoir characterization, statistical methods are increasingly used to build models and maximize the use of information with minimal error. Various machine learning methods are already employed to find correlations between different properties. However, some steps, such as integrating geological descriptions for core descriptions or thin sections, are done manually into the rock typing scheme (Rojas, L., 2020). In this study, we present a multistep approach for generating a rock typing scheme for carbonate reservoirs (Figure 1). The green area represents a statistically driven approach with various data analysis methods, while the red area denotes the deterministic process of geological interpretation. The integration process involved iterative brainstorming sessions and discussion meetings to combine descriptive geological data with numerical petrophysical calculations.

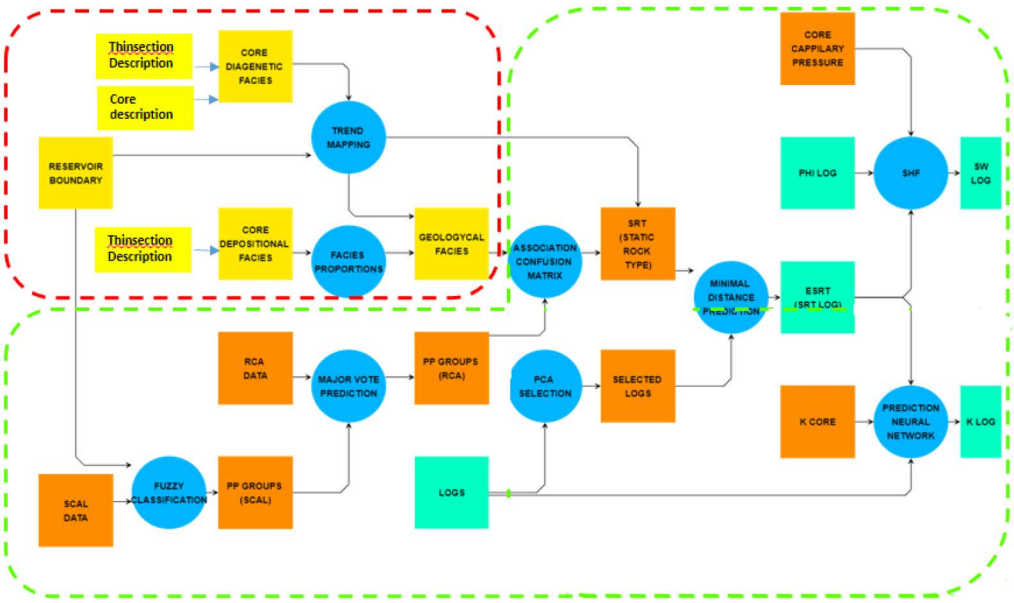


Figure 1—Rock type workflow with machine learning method in green area and manual analysis in red area (Rojas, L. 2020)

The rock type workflow can be fully digitized through the use of image recognition tools for automated thin section description (Figure 2), eliminating the need for manual analysis and incorporating the results into automated rock typing based on porosity, permeability, and mercury injection (Shebl, H.T., 2021) or more complex by accounting NMR data and 3D core measurements (Peesu, R., 2022). However, this is only possible for recent studies, as a large amount of geologic descriptions produced in the past centuries still

exist in reports with descriptive text, analyzed manually by geologists. The manual approach can be time-consuming and prone to errors in routine tasks. In recent years, Natural Language Processing (NLP) has been used to automate some tasks associated with text recognition. This research article presents a method for analyzing textual descriptions of rocks using NLP techniques, providing a comprehensive and efficient approach to transforming deposit characterization reports into a structured quantitative dataset for further integration into statistical methods.

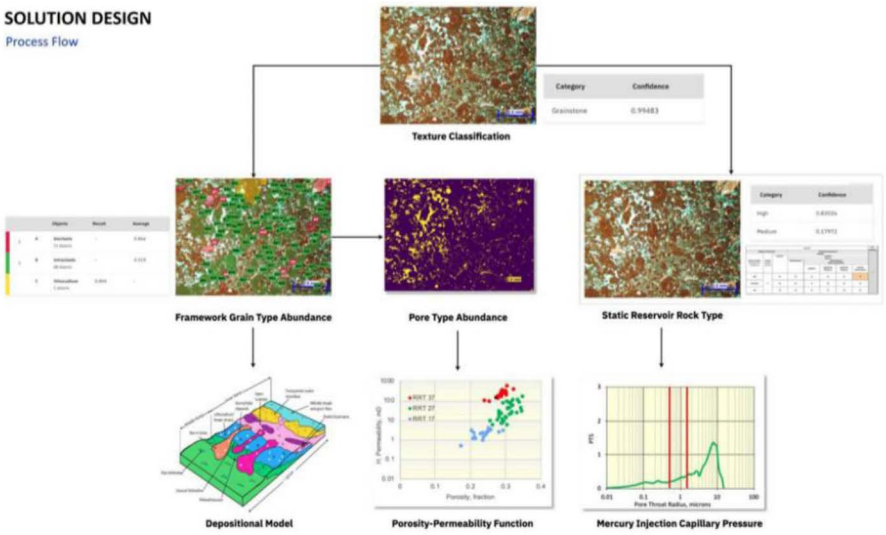


Figure 2—Thin section numerical characterization and including in machine learning method (Shebl, H. T. 2021).

The goal of the study was not only to review available language-based models (LM) but also to develop a tool for analyzing reservoir descriptive texts using NLP techniques. The scope includes finding features in the written geologic reports and defining scales for their quantitative description, as well as presenting the results in a structured format. The analysis aims to provide valuable insight into reservoir characterization by converting textual information into numerical data for use in various statistical methods to predict properties.

Theoretical Methods Overview

Text analysis involves converting raw text data into structured and actionable information, enabling data-driven decision-making in various domains. The emergence of machine learning has led to significant advancements in text analysis methodologies, improving accuracy and efficiency in handling large-scale text datasets (Kumar, P. et al., 2023).

Data Preprocessing: It is a crucial first step in text analysis, as it helps to clean and prepare the raw text data for further analysis. Techniques such as tokenization, stopword removal, stemming, and lemmatization aid in reducing noise and standardizing the text data.

Feature Extraction: It is techniques transform text data into numerical representations suitable for machine learning algorithms. This step is vital in capturing the underlying patterns and semantic information present in the text.

Bag-of-Words (BoW): BoW is a simple and effective technique that represents each document as a vector of word frequencies. It disregards the word order and considers each word's occurrence as a separate feature. BoW can be powerful for text classification tasks, especially when combined with algorithms like Naive Bayes and SVM.

Term Frequency-Inverse Document Frequency (TF-IDF): TF-IDF is a statistical measure that evaluates the importance of a word in a document relative to its occurrence across the entire corpus. It assigns higher

weights to words that are frequent in a specific document but relatively rare in other documents, emphasizing their importance in characterizing the document.

Word Embeddings: Word embeddings aim to capture the semantic meaning of words and their relationships by representing words as dense vectors in a continuous space. Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) are popular methods for generating word embeddings.

Contextual Word Embeddings: Contextual word embeddings, such as BERT (Devlin et al., 2018) and GPT (Vaswani et al., 2017), take into account the surrounding context of each word, resulting in representations that are context-dependent.

Sentiment Analysis: Sentiment analysis focuses on determining the emotional tone of the text, whether it is positive, negative, or neutral. Machine learning models like Naive Bayes, Support Vector Machines (SVM), and Recurrent Neural Networks (RNN) (Hochreiter & Schmidhuber, 1997) are widely used for sentiment classification tasks.

Text Classification: Text classification involves categorizing text documents into predefined classes or categories. Methods like Logistic Regression, Decision Trees, Random Forest, and Convolutional Neural Networks (CNN) have shown success in text classification tasks.

Named Entity Recognition (NER): NER identifies and classifies named entities (e.g., names of persons, organizations, locations) mentioned in the text. Conditional Random Fields (CRF) and BiLSTM-CRF (Hochreiter & Schmidhuber, 1997) are popular approaches for NER.

Topic Modeling: Topic modeling uncovers underlying themes or topics within a collection of documents. Latent Dirichlet Allocation (LDA) and Non-negative Matrix Factorization (NMF) are widely used for topic modeling.

Text Summarization: Text summarization aims to generate concise and coherent summaries of lengthy documents. Extractive methods, which select and combine existing sentences, and abstractive methods, which paraphrase and rephrase content, are two common approaches.

Text Generation: Text generation involves producing new text based on patterns learned from the training data. Recurrent Neural Networks (RNNs) and Transformer-based models like GPT-3 (Vaswani et al., 2017) have demonstrated impressive capabilities in text generation tasks.

Text Clustering: Text clustering groups similar documents together based on their content. K-means, hierarchical clustering, and density-based clustering algorithms are widely used for this purpose.

Text-to-Speech and Speech-to-Text: Machine learning models, such as WaveNet and DeepSpeech, have improved the accuracy and naturalness of text-to-speech and speech-to-text conversions.

For geological information extraction, the most applicable method related to traditional text analysis is the Feature Extraction of pre-processed text.

Applied Methods

To achieve the research objectives, the following methods, procedures, and processes were applied: Initially, a dataset of core plug and thin section rock descriptions was collected from available archived reports and digitized. The thin section and core descriptions usually have annotations with information about the lithofacies, rock fabric, porosity, permeability, energy, allochems, skeletal grain size, and other recognized reservoir properties of the rocks. The trained model was used to extract information from the descriptions, and the model's performance was evaluated using various metrics, including accuracy, recall, and precision score.

There are two main approaches to extract data from unstructured text data. The first is based on programming-based algorithms for learning contextual word representations for various downstream tasks, the best known being BERT (Bidirectional Encoder Representations from Transformers); another method is based on text input prompts such as ChatGPT. Both BERT and ChatGPT use large language models (LLMs), but differ fundamentally in their architecture and training goals.

The analyzed data was presented in a tabular format, showcasing the sample ID, lithofacies index, grain size index, scaled frequency and quality of various properties. The tables provided a clear representation of the results in quantitative form, enabling easy interpretation and data anlysisys algorithms.

Contextual, Programming-based language model (Automated Exploratory Data Analysis)

Programming based text models uses the common programming languages for data analysis, e.g. Phython, R, Jave, etc. The application of workflows with various libraries enables data processing as described below.

Clustering is often done as an initial step in machine learning pipelines. We used Latent Dirichlet allocation (LDA) for clustering text to gain insights into the distribution of words. Figure 3 shows words grouped into clusters.

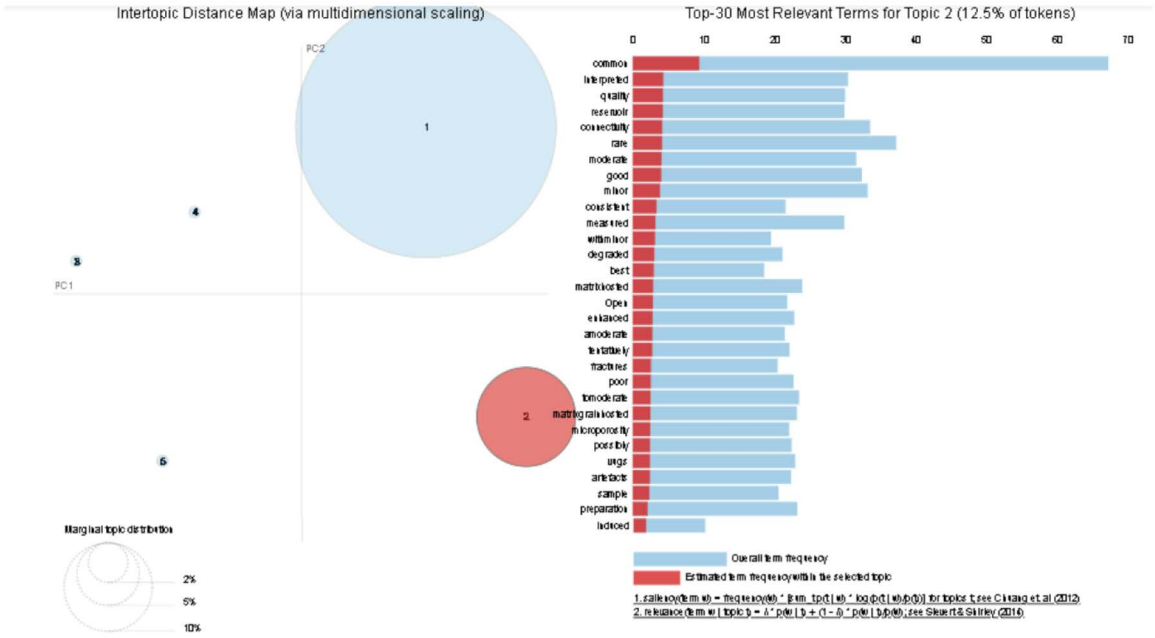


Figure 3—Word clusters using Latent Dirichlet allocation (LDA)

Text Pre-processing / Cleaning. Machine learning models are mathematical models that work with numbers. To infer insights from text, unstructured documents need to be converted into a structured numerical form that can be fed into ML models.

The following are text processing techniques used to clean text and to come up with textual data that can be analyzed properly:

Auto spell-check correction. Word tokenization: Tokenization is the process of representing words as tokens, identifying each word by a numeric ID instead of the word itself.

Sentence tokenization: Sentence tokenization converts a document into a number of sentences, enabling each sentence to be separately fed into the NLP model for better results.

Stop words: Stop word removal reduces dimensionality and avoids overfitting. However, they are important for context since negation is crucial in this problem. For example, in phrases like "not present," removing the stop word "not" can change the meaning completely. Therefore, we want to keep the word order to preserve negations and the actual meaning.

Punctuation removal: Punctuation adds noise to textual data and increases dimensionality, as each word is processed as a token.

Lowercasing: Lowercasing is important because in programming, a camel case word "Reservoir" is not equal to "reservoir." This could lead to increased dimensionality affecting overfitting.

Text normalization: Words can have different variations, increasing the complexity of the data. Approaches like stemming or lemmatization help use a standard, uniform form of the words by fine-graining their linguistic features. For example, Lemmatization removes morphological features that modify the root of words, such as the verb form and the adverb form in "degrading" and "degraded," respectively (Spacy.io, n.d.). Instead, both can be represented by one token, the lemma "degrade."

N-gram analysis: N-gram is the number of words considered as one token. Unigram considers one word, bigram considers two consecutive words (e.g., "reservoir quality"), and trigram considers three words (e.g., "measured permeability value").

Text Representation / Vectorization. Word Embedding is an effective structured numeric representation of text in the form of vectors. Unlike other vector representations such as Bag of Words, TF, and TF-IDF, word embeddings preserve the contextual meaning in long sequences as they are learned by training deep neural networks that consider previous and next words in the sentence. We used TensorFlow to generate embedding vectors for our text. The number of words in the text affects the number of dimensions of the vectors. Embeddings are usually highly dimensional for large datasets so that they can capture relationships and semantics. To visualize our word embeddings, we projected them into 3-dimensional space using Principal Component Analysis (PCA), a technique for dimensionality reduction.

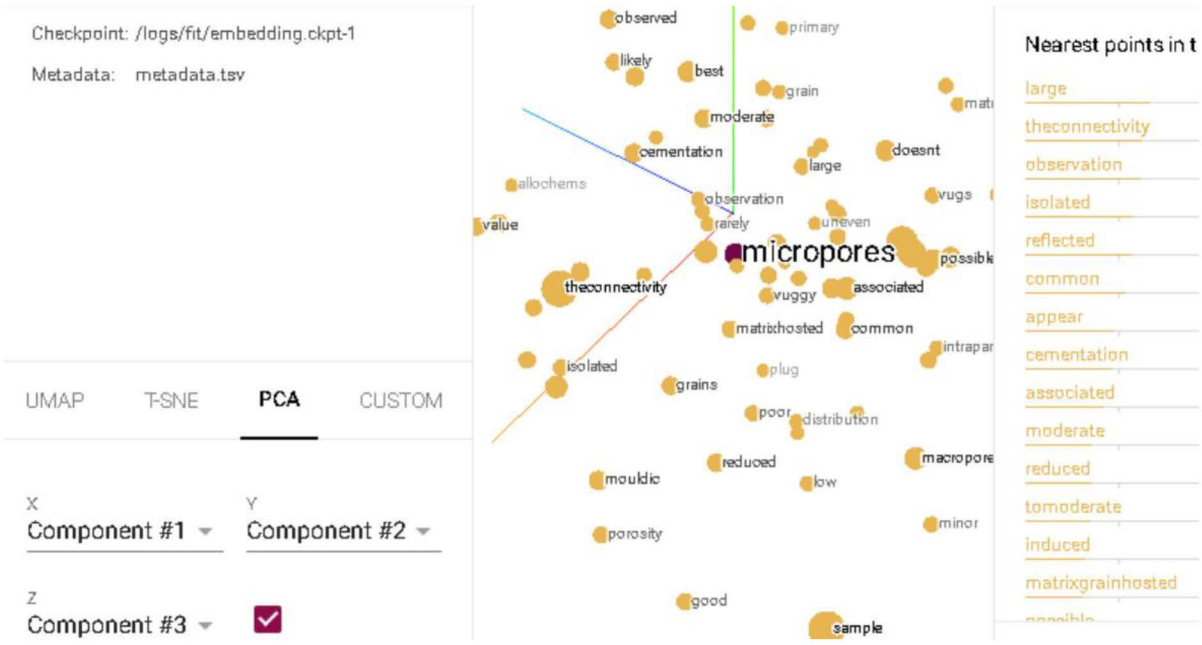


Figure 4—Word Embeddings projected into 3-dimensional space.

Modelling and Parameters/Scales Extraction. Pre-trained NLP models are used to accomplish different downstream tasks. The Natural Language Processing Toolkit (NLTK) Python library was used to break each sample's text into sentences. The sentences are then fed as input to the NLP model. We used a pre-trained NLP model from the Spacy Python library, which is trained on a large corpus of data and uses the Transformers algorithm. Transformers are state-of-the-art deep learning algorithms that many advanced AI models today are based on. Transformers can capture long contextual relationships in the text using the "Attention" mechanism, which learns the importance of words to each other. This is done by having connections to the entire input at each part of the sequence in each layer. The weights of these connections are learned, and they determine the relative importance of parts of the text to a word. This makes it better than other deep learning architectures like RNN and Bi-LSTMs that require a sequence to be processed one

by one for many recurrent steps. Due to its architecture, Transformers can process large amounts of data in parallel, utilizing GPUs.

Figure 5 Visual Abstraction of the Attention mechanism by learning relationships between words shows an abstraction of the attention mechanism (Tamura, 2021).

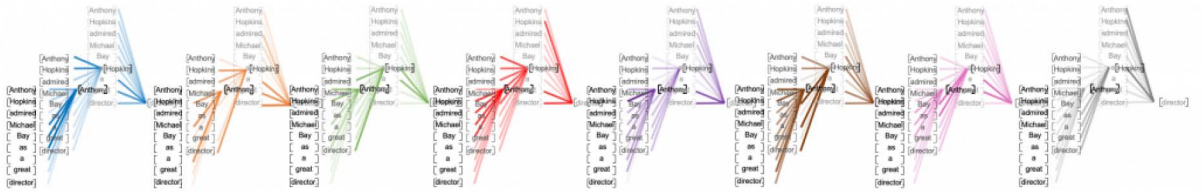


Figure 5—Visual Abstraction of the Attention mechanism by learning relationships between words (Tamura, 2021)

Noun-phrase chunking: Using the trained NLP model, noun phrases are extracted from each sentence.

For each extracted noun phrase, we extract the parameters using two approaches. Parts of Speech Tagging: Parts of speech tagging categorizes words according to their syntactic function in the sentence, such as nouns, adjectives, conjunctions, etc. This is a crucial step in the text mining pipeline as it helps understand the grammatical structure and semantics of the text. The Spacy NLP is trained to identify the parts of speech. Then we use the noun as the parameter and the adjective as the scale (the quantitative value).

Figure 6 shows parts of speech tags for the Nouns, Adjectives, and Adverbs.

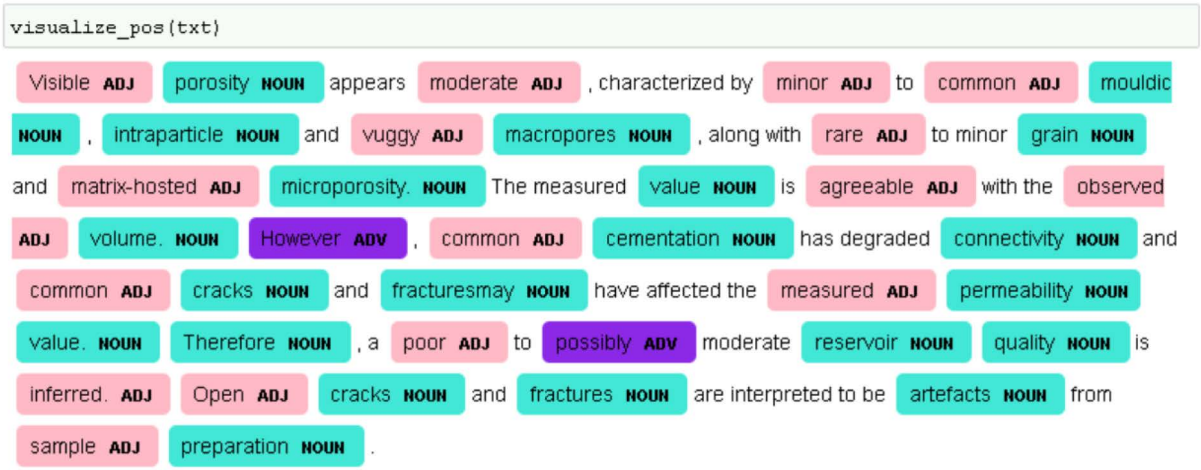


Figure 6—Parts of Speech Tagging for Nouns, Adjectives and Adverbs

Dependency Parsing: "Dependency parsing is the task of analyzing the syntactic dependency structure of a given input sentence" (Lisa Wang, 2019). The used dependency parser also has the capability of sentence boundary detection, which is important to avoid "context fragmentation" (Le, 2019), which may lead to undesired results.

Figure 7 shows the grammatical structure in a sentence, relationships between words in the dependency graph. The labels show the nature of the dependency between words according to the Universal Dependency Relations.

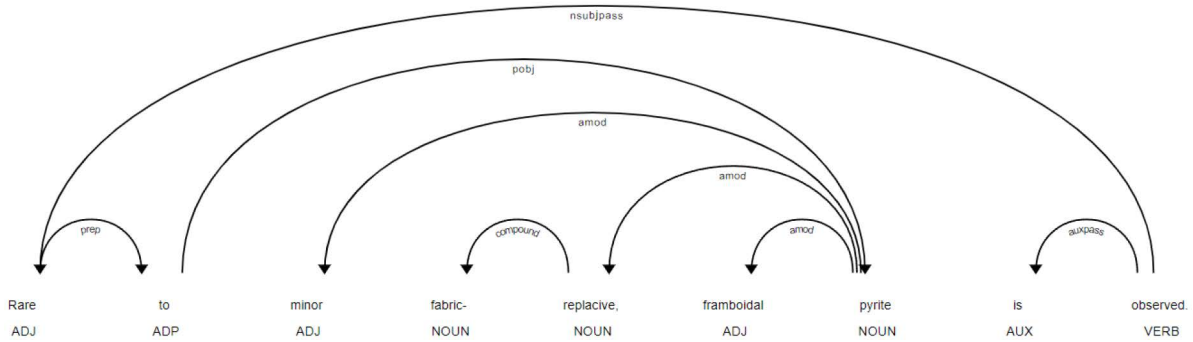


Figure 7—Dependency graph

We used the dependency tree for a few rule-based scenarios to consider cases such as "matrix-hosted microporosity" to meet the conventional parameter names used by geologists like "matrix-hosted microporosity" and "reservoir quality." For example, if a word (matrix) has a dependency as "noun phrase as adverbial modifier", and the head of the word "hosted" has an "adjectival modifier" dependency, then we consider this as one token as shown in Figure 8 and Figure 9.

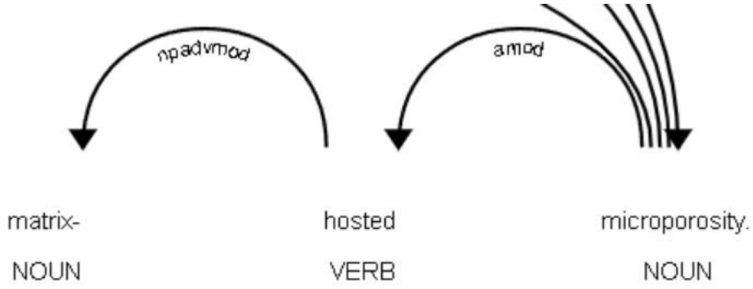


Figure 8—Dependency tree for a rule-based scenarios "noun modifier" and "adjectival modifier"

```
for tok in chunk:
    if tok.dep_=="compound":
        noun = str(doc1[tok.i:tok.head.i+1])
        print(Fore.RED, "compound noun:", noun)
        print(Fore.RESET)

    elif tok.dep_ == 'npadvmod': # noun phrase as adverbial modifier
        if tok.head.dep_=='amod': #adjectival modifier
            noun = doc1[tok.i:tok.head.i+2].text
            print(Fore.RED, "adverbial modifier:", noun)
            print(Fore.RESET)

    else:
        if tok.pos_ == "NOUN":
            noun = tok.text
```

Figure 9—Python script for tree for a rule-based scenarios "noun modifier" and "adjectival modifier"

Question Answering Model. We used a question answering model that considers each geological property (noun) for a question and tries to find the best span in the text that corresponds to the answer. We used the extracted nouns from the previous stage to form questions about them, such as "How is the porosity?" or "Describe the permeability," and the text as the context. A pre-trained BERT question answering model RoBERTa (A Robustly Optimized BERT Pretraining Approach) is used. BERT stands for Bidirectional Encoder Representations from Transformers, which is an architecture that uses the "Encoder" part of the Transformer (Toutanov, 2019). BERT learns the representations of words and their statistical relationships.

To extract the scale value of a geological property, we treat the text of each geological group as context and turn the property (noun) into a question.

The BERT question answering model is composed of three layers. Embedding layer: It feeds the tokens as input into the Attention layer.

Attention layer: It models the contextual relationships among the input sequence.

Output layer: It generates the final results.

Figure 10 shows a high-level architecture of the BERT model.

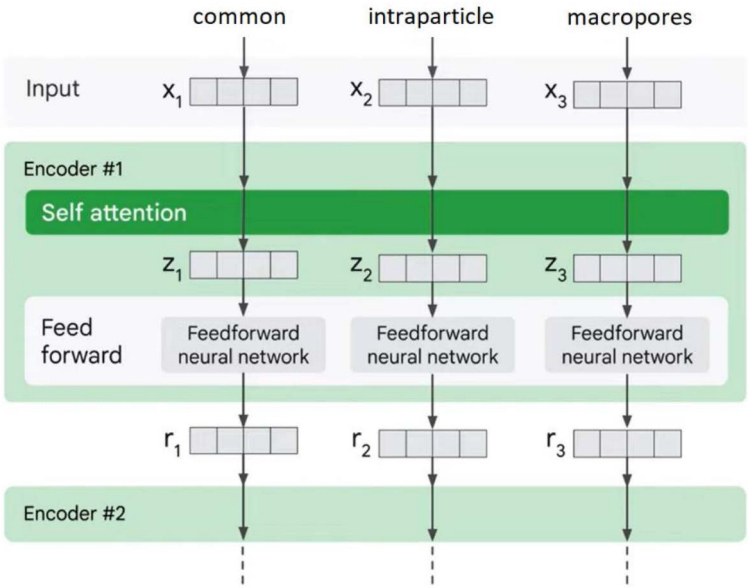


Figure 10—High level architecture of the Encoder in BERT model (figure was modified from the original source tensorflow.org, n.d.)

The following is an example of the outputs using these techniques.

```
 Sentance: Visible porosity is moderate to good and includes common interparticle macropores,
primary and secondary intraparticle and mouldic macropores, along with minor grain and matrix-
hosted micropores.
  1 Noun chunk:  Visible porosity
QA output: porosity :  moderate to good
POS output: {'porosity': 'Visible'}

 2 Noun chunk:  common interparticle macropores
compound noun: interparticle macropores

QA output: interparticle macropores :  common
QA output: macropores :  interparticle
POS output: {'macropores': 'common'}

 3 Noun chunk:  primary and secondary intraparticle and mouldic macropores
POS output: {'macropores': 'primary and secondary intraparticle and mouldic'}
```

The following shows results using RoBERTa model solely without POS and dependency trees.

```
                                        [ Group: Allochems Skeletal Grains  ]
group_params_names: ['preserved benthic foraminifera', 'echinoderm debris', 'bivalve debris', 'thin bivalve
debris', 'thick-shelled debris', 'indeterminate algal debris', 'dasycladacean green algae', 'echinoderm deb
ris ', 'dasyclad green algae', 'algal debris', 'benthic foraminifera', 'thin-shelled bivalve/ostracod debri
s', 'orbitolinids', 'undifferentiated skeletal', 'indistinctalgal debris', 'lithocodium bacinella fragment
s', 'indistinct algal debris', 'coarse bivalve/', 'rudist', 'indeterminate skeletal debris']

 Text: Common thin and thick-shelled bivalve debris, echinoderm debris, indistinct algal debris, Lithocodiu
m bacinella fragments, poorly preserved benthic foraminifera and common indeterminate skeletal debris are o
bserved.
 preserved benthic foraminifera :  poorly    Score: 0.8352
 echinoderm debris :  Common thin and thick-shelled bivalve debris    Score: 0.1185
 bivalve debris :  Common thin and thick-shelled    Score: 0.7227
 thin bivalve debris :  thick-shelled    Score: 0.5498
 thick-shelled debris :  Common thin    Score: 0.2291
 indeterminate algal debris :  indistinct    Score: 0.0401
 dasycladacean green algae :  indistinct algal debris    Score: 0.0045
 echinoderm debris  :  Common thin and thick-shelled bivalve debris    Score: 0.1553

dasyclad green algae :  indistinct algal debris    Score: 0.0492
algal debris :  indistinct    Score: 0.9102
benthic foraminifera :  poorly preserved    Score: 0.9731
thin-shelled bivalve/ostracod debris :  Common thin and thick-shelled bivalve debris    Score: 0.0289
orbitolinids :  Common thin and thick-shelled bivalve debris    Score: 0.0
undifferentiated skeletal :  indeterminate skeletal debris    Score: 0.0722
indistinctalgal debris :  Common thin and thick-shelled bivalve debris    Score: 0.0011
lithocodium bacinella fragments :  common indeterminate skeletal debris are observed    Score: 0.0168
indistinct algal debris :  Common thin and thick-shelled bivalve debris    Score: 0.0037
coarse bivalve/ :  thick-shelled    Score: 0.0683
indeterminate skeletal debris :  common    Score: 0.0452
```

Cosine similarity. In the case the extracted qualitative value (adjective scale) is not within the unified scales names, we used cosine similarity to find the most semantically similar scale from our unified scales list. The following are some examples for most similar text with confidence scores for each.

ID	group	property	value	similar_scales	most_similar_scale
WC3-1	Reservoir Properties	cementation	poor to moderate at best	[('poor to moderate', 298.7), ('low to moderate', 282.25), ('moderately to well', 274.94), ('poor to possibly moderate', 270.56)]	poor to moderate
WC1-1	Rock Fabric	fractures	multiple fractures cutting the fabric locally are also occluded by calcite cement	[('minor to locally common', 47.97), ('multiple', 44.08), ('patchily distributed', 42.32), ('partly occluded', 38.11)]	minor to locally common
45-12A	Rock Fabric	microporous	patchily	[('patchil', 361.42), ('poorly', 263.31), ('weakly', 233.64), ('patchily distributed', 230.73)]	patchil
WC2-1	Rock Fabric	microporous	aweakly microporous	[('weakly', 231.39), ('sparitic to equant', 215.23), ('patchily distributed', 208.7), ('rare to minor', 207.25)]	weakly

Figure 11—Sample output using Noun-chunking, Parts of Speech Tagging and Transformer-based Question Answering models

Semi-supervised Sequence Labelling. In the previous methods, we use pre-trained models to achieve the results. We also use sequence labelling approach by labelling the parameters and their scales and training a custom Named Entity Recognition (NER) model to recognize them in new unseen text. Despite labelling being an expensive approach, better results are anticipated by training the model with our domain-specific content rather than the pre-trained models that trained on generic texts.

Although there are online tools for NER annotation, it is still a time-consuming process. Hence, we used the list of unified parameters names and scales to automate the annotation of the whole data frame. The following figures shows an example annotated automatically.

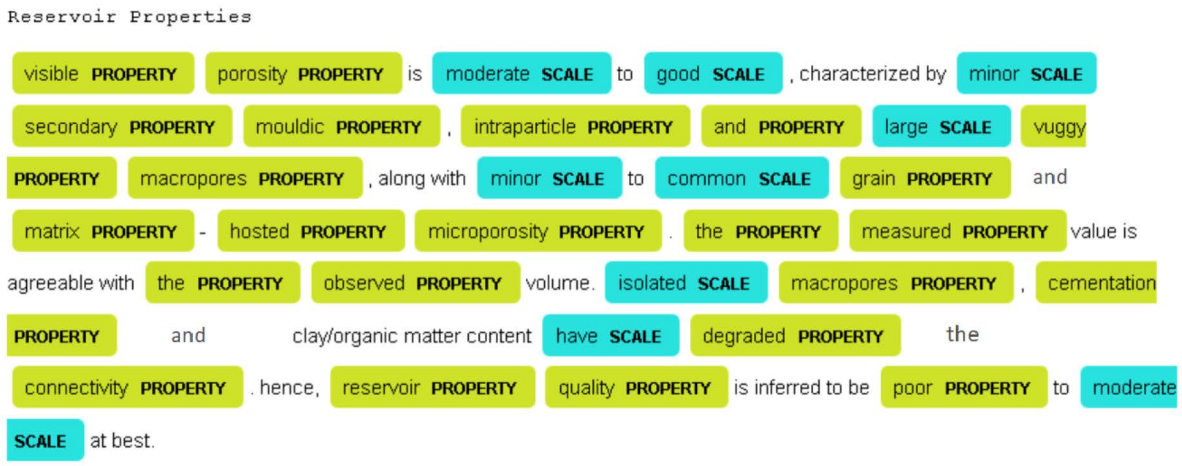
```
('visible porosity appears moderate, characterized by minor to common mouldic, intraparticle and vuggy macrop
ores, along with rare to minor grain and matrix-hosted microporosity. the measured value is agreeable with th
e observed volume. however, common cementation has degraded connectivity and common cracks and fractures may
have affected the measured permeability value. therefore, a poor to possibly moderate reservoir quality is in
ferred.open cracks and fractures are interpreted to be artefacts from sample preparation.',
 {'entities': [(0, 16, 'PROPERTY'),
   (411, 428, 'PROPERTY'),
   (8, 16, 'PROPERTY'),
   (343, 364, 'PROPERTY'),
   (148, 175, 'PROPERTY'),
   (101, 111, 'PROPERTY'),
   (77, 90, 'PROPERTY'),
   (311, 320, 'PROPERTY'),
   (300, 306, 'PROPERTY'),
   (276, 288, 'PROPERTY'),
   (251, 288, 'PROPERTY'),
   (251, 262, 'PROPERTY'),
   (52, 57, 'SCALE'),
   (124, 128, 'SCALE'),
   (52, 67, 'SCALE'),
   (25, 33, 'SCALE'),
   (218, 226, 'SCALE'),
   (61, 67, 'SCALE'),
   (385, 410, 'SCALE'),
   (441, 445, 'SCALE'),
   (199, 208, 'SCALE'),

 Group: Reservoir Properties
sentance: Visible porosity is moderate to good, characterized by common primary and secondary
intraparticle and mouldic macroporosity, along with minor microporosity hosted by the grains a
nd matrix.
 0 16 visible porosity PROPERTY
 8 16 porosity PROPERTY
 84 97 intraparticle PROPERTY
 170 176 grains PROPERTY
136 141 minor SCALE
20 36 moderate to good SCALE
20 28 moderate SCALE
55 61 common SCALE
sentance: The measured porosity value is agreeable with the observed volume.
 13 21 porosity PROPERTY
 4 21 measured porosity PROPERTY
```

Fuzzy Matching. However, since we have text of 24 samples of data only, we combined the statistical named entity recognition with other rule-based components to boost our statistical models. We used fuzzy matching from Spacy which allows matching different variations of the tokens without having to specify every possible variant of the word.

```
from spaczz.matcher import FuzzyMatcher
from spacy.matcher import Matcher
nlp = spacy.blank("en")

ruler = nlp.add_pipe("entity_ruler")

property_pattern = [{"TEXT": {"FUZZY": {"IN": prop_list}}}]
pattern1 = {"pattern": property_pattern, "label":"PROPERTY" }
ruler.add_patterns([pattern1])

scale_pattern = [{"TEXT": {"FUZZY": {"IN": scales_list}}}]
pattern2 = {"pattern": scale_pattern, "label":"SCALE"}
ruler.add_patterns([pattern2])
```

Figures and show the named entity recognition results before and after using Fuzzy matching.
The following figure the patterns using fuzzy matching.
Results for statistical Named Entity Recognition (NER) model

Named entity recognition with Fuzzy matching



Conversational, Prompt based language model

The analysis of recent publications about machine learning shows significant development of methods for analyzing text using machine learning algorithms. While both BERT and GPT are LLMs, they have different training objectives and use cases. While GPT models specialize in generating text based on input prompts, our approach involves using the GPT model to analyze raw text descriptions of core data extracted from wellbores.

These descriptions, often referred to as thin section descriptions, provide detailed information about the geological characteristics of the core samples. However, these descriptions are typically unstructured and can be challenging to interpret and analyze manually due to their complexity and volume. The application is used to process these descriptions and structure the information into a more manageable format.

The process begins by feeding the raw text descriptions into the application. The application, leveraging the power of the GPT-4 model, reads and comprehends the information in the descriptions. It identifies key parameters and their corresponding values in the text, understanding the context and relationships between different pieces of information. Once the application has processed the descriptions, it generates a structured output in the form of a table (see figure 12). Each row in the table corresponds to a specific depth point in the core sample, and each column represents a geological parameter. The values in the table are derived directly from the thin section descriptions. This process effectively transforms the unstructured raw text into a structured format, often referred to as discrete logs.

Let's open xls with this section description:

```
dfThinSectDescr = pd.read_excel('ThinSectDescrExampl.xls')
dfThinSectDescr.head()
```

Python

	Measured depth	ID	Phi, v/v	GD, g/cc	K, mD	Lithofacies	Allochems Non-Skeletal Grains	Allochems Skeletal Grains	Energy	Reservoir Properties	Rock Fabric	Textural
0	1	1	0.123	2.70	0.95	RPF	Common indistinct and distinct peloids are pre...	Common benthic foraminifera (inc. miliolids, r...	Based on the sedimentary features and grain as...	Visible porosity appearsmoderate, characterize...	This is a poorly sorted, indistinctly bioturba...	Rudist, peloidal Floatstone
1	2	2	0.194	2.70	5.60	SPP	Common indistinct and distinct peloids are pre...	Common benthic foraminifera (inc. miliolids, r...	moderate-energy, shoal-flanking depositional e...	Visible porosity is moderateto good and includ...	A poorly sorted, indistinctly bioturbated, ske...	Skeletal, peloidal Packstone
2	3	3	0.197	2.70	44.00	RPF	Common indistinct and distinct peloids are pre...	Common benthic foraminifera (inc. miliolids, r...	The allochem assemblage and sedimentary charac...	Visible porosity is moderate togood and includ...	This is a poorly sorted, indistinctly bioturba...	Rudist, peloidal Floatstone
3	4	4	0.212	2.70	2.80	SPP	Common indistinct and distinct peloids are pre...	Common poorly preserved, micritized benthic fo...	Deposition in a moderate-energy backshoal, or ...	Visible porosity is moderateto good, character...	A poorly sorted, indistinctly bioturbated skel...	Peloidal, skeletal Packstone
4	5	5	0.205	2.69	2.10	SPP	Common indistinct and distinct peloids are pre...	Common echinoderm debris, minor Lithocodium/Ba...	The allochem assemblage and the sedimentary ch...	Visible porosity is moderateto good, character...	This is a poorly sorted, indistinctly bioturba...	Skeletal, peloidal Packstone

Figure 12—Processed the raw text of descriptions with generating a structured output in the form of a table.

The creation of these discrete logs is a significant step in the petrophysical workflow. They provide a clear and concise overview of the geological characteristics at each depth point, significantly simplifying the process of rock typing. These properties, derived from the discrete logs, provide valuable inputs for reservoir simulation and flow modelling.

One of the key advantages of using large language models like GPT-4 is their ability to handle complex and domain-specific language. Geological descriptions often include domain-specific jargon and terminology that may not be well understood by generic language models. However, GPT-4, with its advanced language comprehension capabilities, can accurately interpret and process such language as presented in Figure 13.

```
| Reference Index | Visible Porosity | Macropores | Microporosity | Cementation | Fractures |
| --- | --- | --- | --- | --- | --- |
| 1 | Moderate | Rare to minor mouldic, intraparticle and vuggy | Rare to minor grain and matrix-hosted | Common | Common cracks and fractu
res |
| 2 | Moderate to good | Minor to common secondary mouldic, intraparticle and vuggy | Minor grain and matrix-hosted | Common | Macropores c
ommonly isolated |
| 3 | Moderate to good | Minor to common secondary mouldic, intraparticle and large vuggy | Minor to common grain and matrix-hosted | Commo
n, unevenly distributed | Large, possible artefact vugs |
| 4 | Moderate to good | Minor secondary mouldic, intraparticle and large vuggy | Minor to common grain and matrix-hosted | Isolated macrop
ores, cementation and clay/organic matter content | N/A |
| 5 | Moderate to good | Rare mouldic macropores, minor secondary intraparticle | Common grain and matrix-hosted | Neomorphism of the micri
te matrix, compaction and cementation | N/A |
```

Figure 13—The result table with structured reservoir properties and qualitative descriptions.

Additionally, the application of GPT-4 in this context demonstrates its scalability and efficiency in processing large volumes of geological data. Analyzing and interpreting core descriptions manually can be time-consuming and labor-intensive, especially when dealing with extensive wellbore datasets. GPT-4, with its parallel processing capabilities, can handle multiple descriptions simultaneously, significantly reducing the analysis time and human effort required.

Moreover, the output of the GPT-4 model can be easily integrated into existing reservoir characterization workflows. The structured data generated by the application can be readily used as input for reservoir modeling and simulation studies, enhancing the accuracy and reliability of the predictions.

Results and Evaluation

The performance of the applied methods was evaluated using a test dataset consisting of core descriptions not used during the model training phase. Several evaluation metrics were employed, including accuracy, recall, and precision score. The results demonstrated a high level of accuracy and reliability in extracting geological parameters and their corresponding scales from the raw text descriptions. The application achieved an accuracy score of over 90% in identifying and categorizing key geological properties.

Figure 14—The workflow for manual preparation a benchmark data set

The evaluation also highlighted the effectiveness of the question-answering model in extracting scale values for different geological parameters. The model achieved a recall rate of 85% in accurately identifying scale values, making it a valuable tool for quantifying geological characteristics.

The logs of the NER model show very high accuracy during the training phase.

```
[i] Using CPU

=========================== Initializing pipeline ===========================
[+] Initialized pipeline

============================= Training pipeline =============================
[i] Pipeline: ['tok2vec', 'ner']
[i] Initial learn rate: 0.001
E    #       LOSS TOK2VEC  LOSS NER  ENTS_F  ENTS_P  ENTS_R  SCORE
---  ------  ------------  --------  ------  ------  ------  ------
  0       0          0.00     46.50    0.00    0.00    0.00    0.00
  1     200         11.97   1118.77  100.00  100.00  100.00    1.00
  3     400          0.00      0.00  100.00  100.00  100.00    1.00
  6     600          0.00      0.00  100.00  100.00  100.00    1.00
  9     800          0.00      0.00  100.00  100.00  100.00    1.00
 13    1000          0.00      0.00  100.00  100.00  100.00    1.00
 18    1200          0.00      0.00  100.00  100.00  100.00    1.00
 25    1400          0.00      0.00  100.00  100.00  100.00    1.00
 32    1600          0.00      0.00  100.00  100.00  100.00    1.00
 41    1800          0.00      0.00  100.00  100.00  100.00    1.00
[+] Saved pipeline to output directory
NER\model-last
```

Furthermore, the comparison between the semi-supervised sequence labeling and the large language model (LLM) approach showed that the LLM approach, specifically using GPT-4, outperformed the traditional sequence labeling method in terms of accuracy and efficiency. The LLM approach's ability to handle domain-specific language and understand complex geological descriptions contributed to its superior performance.

Results show that using combination of parts-of-speech tagging, the pre-trained Question Answering BERT model along with cosine similarity exceeds the other approaches.

Limitations and Challenges

While NLP techniques offer significant advantages in analyzing geological descriptions, there are some limitations and challenges to consider:

Data Quality and Availability: The accuracy and effectiveness of NLP models heavily depend on the quality and quantity of training data available. In some cases, geological descriptions may be incomplete, ambiguous, or contain errors, leading to challenges in accurate information extraction.

Domain-specific Language: Geological descriptions often contain domain-specific language and terminology that may not be present in generic language models. Fine-tuning NLP models on domain-specific data can address this issue, but it requires sufficient annotated data for training.

Contextual Understanding: While contextual word embeddings like BERT can capture relationships between words, the model's performance may be limited by the length of the text it can process. Very long descriptions or documents may require further segmentation or summarization.

Bias and Fairness: NLP models can inherit biases present in the training data, leading to biased outputs. Efforts should be made to ensure fairness and mitigate bias in the analysis, especially in critical decision-making processes.

Interpretability: NLP models, particularly large language models, are often considered black boxes due to their complexity. Ensuring transparency and interpretability of the model's decisions is essential for gaining trust and understanding potential limitations.

Ethical Considerations

The application of NLP techniques in geological data analysis raises ethical considerations, especially when the analysis contributes to decision-making in hydrocarbon reservoir development. Some key ethical considerations include:

Transparency and Accountability: It is crucial to be transparent about the data sources, methods, and algorithms used in the NLP analysis. Providing explanations for the model's decisions and being accountable for the results are essential for building trust.

Bias and Fairness: As mentioned earlier, NLP models can inherit biases from the training data. Efforts should be made to detect and mitigate bias to ensure fair and unbiased analysis.

Privacy and Data Security: Geological descriptions may contain sensitive information about oil and gas reserves, which must be handled with utmost care to protect confidentiality and comply with data privacy regulations.

Environmental Impact: The use of NLP and machine learning technologies, while beneficial, requires computational resources that may have environmental implications. Employing energy-efficient hardware and optimizing algorithms can help mitigate environmental impacts.

Conclusion

In conclusion, this study showcases the effectiveness of machine learning and natural language processing techniques, particularly the application of the GPT-4 large language model, in analyzing geological text descriptions. The developed application successfully extracts valuable geological parameters and their scales from unstructured text, facilitating rock typing and permeability prediction in subsurface reservoirs.

The combination of NLP techniques, question answering models, and semi-supervised sequence labeling offers a comprehensive and efficient solution for geological data analysis. The application's scalability and ability to handle complex domain-specific language make it a valuable asset in the petrophysical workflow.

As machine learning models continue to evolve and the availability of domain-specific training data increases, the potential for data-driven insights in the oil and gas industry grows exponentially. Continued research and development in this field hold promise for enhancing reservoir characterization, optimizing production, and improving overall decision-making processes in the energy sector.

Acknowledgments

The authors wish to thank and acknowledge the management of Abu Dhabi Company for Onshore Oil Operations (ADNOC Onshore) and Abu Dhabi National Oil Company (ADNOC), Abu Dhabi, for permission to publish this paper.

References

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, **9**(8), 1735–1780. https://ai.googleblog.com/2019/01/transformer-xl-unleashing-potential-of.html

Kumar, P., Tveritnev, A., Jan, S. A., & Iqbal, R. (2023, March 13). Challenges to Opportunity: Getting Value Out of Unstructured Data Management. Day 2 Tue, March 14, 2023. https://doi.org/10.2118/214251-MS

Lisa Wang, J. N. (2019). stanford.edu. Retrieved from web.stanford.edu: https://web.stanford.edu/class/cs224n/readings/cs224n-2019-notes04-dependencyparsing.pdf

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

OpenAI. (2022). OpenAI API. https://platform.openai.com/docs/

Peesu, R. R., Voleti, D. K., Dutta, A., Vanam, P. R., & Reddicharla, N. (2022). Automated Image Processing of Petrographic Thin Sections for Digital Reservoir Description: A Bridge to Correlate with Core and NMR Data. Society of Petroleum Engineers - ADIPEC 2022. https://doi.org/10.2118/211691-MS

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532–1543).

Rauf Iqbal, Pranav Kumar, Muhammad Aamir, Shaikha Obaid Al-Naqbi, Salahuddin Abdullah Jan and Pravin Kumar, (2022) "IBTIKAR Digital Lab – A Collaborative approach towards research and Development Challenges in Oil and Gas Upstream", SPE-211115-MS, ADIPEC, 31-Oct to 3 Nov 2022

Rojas, L., Tveritnev, A., & Pinillos, C. (2020, November 9). Rock Type Characterization Methodology for Dynamic Reservoir Modelling of a Highly Heterogeneous Carbonate Reservoir in Abu Dhabi, UAE. Day 2 Tue, November 10, 2020. https://doi.org/10.2118/203414-MS

Shebl, H. T., al Tamimi, M. A., Boyd, D. A., & Nehaid, H. A. (2021, December 9). Automation of Carbonate Rock Thin Section Description Using Cognitive Image Recognition. Day 3 Wed, November 17, 2021. https://doi.org/10.2118/208149-MS

Spacy.io. (n.d.). Linguistic Features. Retrieved from spacy.io: https://spacy.io/usage/linguistic-features

Tamura, Y. (2021). Multi-head attention mechanism: "queries", "keys", and "values," over and over again. Retrieved from data-science-blog.com: https://data-science-blog.com/blog/2021/04/07/multi-head-attention-mechanism/

Tensorflow.Org. (n.d.). Neural machine translation with a Transformer and Keras. Retrieved from tensorflow.org: https://www.tensorflow.org/text/tutorials/transformer

Toutanov, J. D.-W. (2019). BERT: Pre-training of Deep Bidirectional Transformers for. Google AI Language.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998–6008).