TEXT EMBEDDING BASED TOPIC MODELING ON NOISY HISTORICAL DRILLING DATA

by

Goutham Reddy Narravula

Submitted in partial fulfillment of the requirements
for the degree of Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
December 2021

Dedicated to my beloved friends and family.

Table of Contents

List of Tables

List of Figures

Abstract

In the oil and gas industry, well reports (daily drilling reports) play a vital role in documenting critical events that take place on a drilling rig. Well activities are frequently associated with a drilling process, as wells are drilled to obtain, store, inject, and extract crude oil and natural gas. Reports contain findings, unfavourable events and summaries for every phase of the project. This information will help foresee drilling risks and mitigate unwanted surprises beforehand, significantly reducing development costs and saving time for future projects. Manually going through thousands of reports to find relevant information can be a time-consuming and laborious process. This thesis proposes an approach for extracting human interpretable groups of words that can best summarize a cluster of well reports using state-of-the-art topic modeling and text embedding techniques. Generated topics are used to optimize the existing information retrieval system for documents in this area by incorporating document classification and query expansion applications.

Most of these reports date back to the late 1970s and are handwritten by multiple project engineers. Data extraction from these reports was done using an Optical Character Recognition engine, which lead to disorderly and noisy text. Due to the complexity of data, conventional data preprocessing techniques and traditional topic modeling algorithms could not produce desired results. Hence, we propose an approach that uses distributed representations to capture text semantic and syntactic context from a small, domain-specific dataset. Oil and gas subject matter experts reviewed generated topics to examine topic diversity and assign appropriate labels. Detailed analysis shows that our results are more coherent and diverse than traditional methods.

Acknowledgements

Throughout my thesis journey, I have received tremendous support and assistance. I would first like to thank my supervisor, Dr. Vlado Keselj, for his valuable guidance during various stages of the thesis. Your insightful feedbacks on thesis structure truly helped me with the writing process. I would also like to thank my co-supervisor, Dr. Dijana Kosmajac, for introducing this project and helping me with valuable suggestions during experimentation.

I would like to thank Dr. Keselj, industry partner and Mitacs, for providing me with funding through grants and mitacs accelerate program. I would also like to thank Mr. Blair MacDougall and other project engineers at WESI for critiquing the outputs and providing constructive feedback. I would like to acknowledge the data science team at my industry partner organization, WESI, for providing me required datasets and helping me understand the industry aspects of this project. I would also like to thank Dr. Nur Zincir-Heywood and Dr. Evangelos Milios for reviewing my thesis.

Finally, I would like to thank my parents and my sister for their wise counsel and encouragement. I would also like to extend my gratitude to my friends who helped me through these difficult times of pandemic.

List of Abbreviations Used

BERT	- Bidirectional Encoder Representations from Transformers
BoW	- Bag of Words
DBOW	- Distributed Bag of Words
DDR	- Daily Drilling Report
DL	- Deep Learning
DSM	- Distributional Semantic Model
E&P	- Exploration and Production
HDBSCAN	- Hierarchical Density-Based Spatial Clustering of Applications with Noise
IR	- Information Retrieval
LDA	- Latent Dirichlet Allocation
LSA	- Latent Semantic Analysis
ML	- Machine Learning
MLM	- Masked Language Model
NLP	- Natural Language Processing
OCR	- Optical Character Recognition
O&G	- Oil and Gas
PLSA	- Probabilistic Latent Semantic Analysis
PTM	- Pretrained Model
PVDM	- Distributed Memory Model of Paragraph Vectors
RBO	- Ranked Biased Overlap
SME	- Subject Matter Expert
SPE	- Society of Petroleum Engineers
SVD	- Singular Value Decomposition
TF-IDF	- Term Frequency–Inverse Document Frequency
UMAP	- Uniform Manifold Approximation and Projection

Chapter 1

Introduction

The global oil and gas (O&G) sector is a multi-million dollar industry. According to a 2021 market research by a leading business intelligence firm called IBISWorld [29], O&G exploration and production (E&P) is in the top 10 fastest-growing sectors in terms of dollar value and has an estimated revenue growth of 15.3%. Oil plays a crucial role in global economic structure, especially for its largest producers like United States, Canada, Saudi Arabia and Russia. Canada ranks as the fifth-largest oil and natural gas producer and is home to vast deposits of both resources. In 2019, this industry was responsible for more than 500,000 jobs across Canada [12] and provided $10 billion in average annual revenue to the government from 2017 to 2019. This revenue played a vital role during the COVID-19 economic crisis and helped government provide financial support to healthcare systems across Canada.

The impact of the COVID-19 pandemic has drastically changed our lives, from how we function daily to the health of the global economy. Every industry throughout the world has been affected, for better or worse, and was forced to modify its long-established business plans. On March 11th, 2020 World Health Organization has declared COVID-19 as a pandemic [23], which led governments across the globe to initiate lockdowns and border restrictions. After the infection became a global health crisis, economies crumbled, which led to business closures and trade and manufacturing disruptions. Canada's unemployment rate rose to 13.7% in June 2020, the highest ever seen in the last four decades [53]. According to a 2021 analysis by S&P Global Market Intelligence, O&G drilling is in the five most impacted industries by COVID-19 [24].

Amid all this crisis, the natural oil sector is heading towards a technological direction by restoring and managing information found in old reports. Industry 4.0 [71] is slowly revolutionizing how companies function by integrating traditional practices with smart technologies like the Internet of Things, Machine learning and Artificial

Intelligence, which motivated the creation of oil and gas 4.0 [40]. Its main objective is the creation of a data-driven intelligence system with digitalization and file management. With the advent of unconditional oil drilling, the upstream O&G industry has dramatically evolved in the past few decades. It is estimated that 40–60% of the O&G industry workforce will retire in the next five years, so it is essential to preserve the knowledge stored in well reports. The industry partner (Waterford Energy Services Inc. [1]) has been developing a document management system to support this objective.

1.1 Motivation

Daily drilling reports are a crucial part of the documentation in well-related projects, and a large portion of the information found in them is represented as free text. Computationally, it can be very challenging to comprehend the narrative due to many factors. One of them is that multiple authors often write the reports, so there is a change in style throughout the documents. The authors hand-wrote most of these well reports. So the conversion of free text to machine-encoded text led to the presence of unwanted noise. Another factor is that the narrative is domain-specific, and semantics are different from everyday language. Nevertheless, these documents can provide valuable knowledge about former projects and guide engineers to learn from past experiences.

The motivation behind this research project lies in the improvement of a document management system of historical well reports with state-of-the-art text mining techniques. Traditionally, domain experts go through numerous pages of largely paper-based well reports to find relevant information, which can be time-consuming and laborious. Hence, the industry is encouraged to develop best management practices. DDRs carry valuable information like unfavourable events, lessons learned, any detected anomalies and best practices. Due to the high volumes of documentation, it is important to create a digital system where engineers can quickly and efficiently access required documentation. Organizing well reports based on generated word patterns will help better understand the available information. Due to the unlabeled nature of our dataset, supervised machine learning and deep learning algorithms are

[1] `https://wesi.ca/`

not suited, and conventional NLP methods like parsing do not work because of the noisy ungrammatical nature of the text. Hence, topic modeling and word embedding methods are considered to find meaningful patterns in the textual corpus.

The generated topics can help divide the entire document collection into various groups, where documents of a group primarily talk about a specific subject in the oil industry. A few examples of subjects are reservoir characterization, casing and cementing, drilling operations etc. Engineers can select a subject based on these topics and check for the previously mentioned insights like lessons learned. Finalized topics can also be integrated with the existing query expansion system. Basic queries on document indices usually do not work well if the user is not elaborate enough when typing in the query. An average user will not provide more than a few words, and he/she may not know or remember the best set of query words to get the best results. Because of these shortcomings, the system needs to make the best of the few words it has been provided with. Topics generated can be used to expand these queries. In addition to generation of meaning groups of words, topic models also provide information about the documents that can be categorized by topics. For each topic, the most dominant documents can be viewed with respective semantic similarity scores. Documents that are represented by the same topic have similar content. Users can also perform semantic search on documents, using terms of their choice.

The best way to advance in any industry is to stay up-to-date with ongoing research and constantly learn from the academic community. Finding the relevant journal papers can be significant to better understand and improve upon the drilling projects described in well reports. Classifying drilling reports based on their relevance with the ongoing research in exploration and production (E&P) segment can help project engineers quickly refer to the relevant journals and gain a better understanding.

1.2 Research Question

Acknowledging the noisy nature of the text, how to identify meaningful hidden patterns and bring structure to limited, domain-specific, ungrammatical data? What evaluation measures can be used to systematically analyze various characteristics of the generated output while handling an unlabeled dataset?

1.3 Contributions

To the best of our knowledge, there is little to no research in O&G that deals with discovering meaningful semantic structures from historical drilling reports. Our main contributions are outlined below.

1. The proposed data preprocessing architecture handles unstructured noisy textual data that can exploit external resources for domain-specific terminology. It also supports filtering out unwanted documents that adversely affect the topic model training.

2. A thorough qualitative and quantitative analysis of various topic models was conducted to determine the best model for discovering meaningful topics in the text corpus.

3. This thesis proposes an approach of associating produced topics with upstream O&G journal articles. Trend analysis was also conducted on the research articles to estimate uncertain events in the industry.

1.4 Thesis Structure

A brief explanation of the structure of the thesis is as follows. Chapter 2 discusses the background and related work of various technologies used for this research. Chapter 3 provides information regarding different oil and gas-related datasets used throughout the project. Chapter 4 gives a detailed description of the proposed architecture for topic extraction. An in-depth explanation of data preparation, domain-expert evaluations and topic labeling is also provided. Topic modeling experiments and produced results are studied in detail in Chapter 5. Visualizations of the evaluation measures and finalized topics are also illustrated. Finally, chapter 6 provides a conclusion with some recommendations for future work.

Chapter 2

Background and Related Work

This chapter discusses the background of text mining and topic modeling in detail, mainly in the oil and gas industry area. It also provides insights into related work done in the respective field. Furthermore, this chapter also explains various topic modeling techniques practised in this thesis.

2.1 Text Mining

Text mining [28], also referred to as text data mining [25], is the process of deriving valuable high-quality information from a collection of unstructured textual data. From the business point of view, one could argue that text mining has more value than data mining, as an estimation of 80% of an organization's data is present in textual format [63, 33]. Text mining is essential for the digitalization of historical reports. It is an umbrella branch for various text analysis techniques like Information Retrieval, Document Summarization, Sentiment Analysis, Topic Modeling, etc. Throughout this thesis, multiple text mining concepts are implemented and discussed.

The recent technological improvements in the energy industry gave rise to massive amounts of data being created and circulated in the form of printed and computer-generated text. Accordingly, text mining has gained an immense reputation in making sense of terabytes of datasets and attracted a wide range of clientele in retrieving, digitizing, organizing, navigating and storing data. Several studies [70, 43] have been conducted on how to extract data and discover knowledge using various text mining methods on textual databases. Recently, Noshi [52] conducted a survey on various text mining methods, challenges and applications for oil and gas industry. Daily drilling reports (DDRs) are recognized as valuable documents as they contain substantial information and recordings of activities that can educate on costs and risks for future projects. Zhang et al. [75] conducted text classification on a large labelled DRR dataset using various traditional ML techniques like the random forest, LSTM and convolutional neural network for anomaly detection. This paper demonstrates that semantic information is best obtained by the vector representations of words. Hoffimann et al. [26] proposed an automatic classification of sentences in DDRs into three predefined labels using three deep learning methods. These papers either perform supervised learning on labelled reports or perform unsupervised clustering on large and noiseless datasets.

2.2 Information Retrieval

Information retrieval (IR) is a process of searching, identifying, locating and retrieving relevant information from massive quantities of recorded data. Early IR systems date back to the late 1940s [61], way before the era of the internet began. In 1959, Hans Peter Luhn published a paper [41] on IR systems whose research helped in the development of the internet for navigating through massive volumes of data. IR systems laid the foundations for designing complex web-based search engine frameworks. IR systems have also been practiced in the O&G industry since the late 1900s [17, 65]. Topics produced in this project will improve the quality of query expansion and enhance the existing information retrieval system.

Query expansion is one of the essential techniques in the information retrieval field that provides a series of methods to reconstruct a user's query to increase retrieval performance and precision. Basic queries on document indices usually do not work well if the user is not elaborate enough when typing in the query. An average user will not provide more than few words, and he/she may not know or remember the best set of query words to get the best results. Because of these shortcomings, the system needs to make the best of the few words provided. Word groups generated by topic models can be used to expand these queries. Jian et al. [31] considered term-based and semantic information as two features of query terms and presented an efficient IR system using topic modeling. Here, latent semantic information extracted with topic modeling is combined with term-based information, improving a typical IR system. Qing et al. [74] proposed a method of identifying topics containing query terms and utilizing the weight associated with terms to rank the matching topics. The ten highest-ranking were then used in the query expansion.

2.3 Word Embedding

Language has always been the human's greatest invention and made the evolution of cultures possible. Our ability to interpret complex sentences of the language we speak is truly outstanding. Understanding the meanings of terms is vital to understand any language. For a machine to understand and comprehend the nuances of a human language, it is essential to convert it to machine-readable format. Many of the existing ML and DL architectures are not equipped for directly handling plain text, let alone understand the context of the text. Word embedding is a process of encoding words or phrases into numbered vectors by capturing semantic and syntactic context. It is basically a vector representation of given words. The introduction of word embedding has revolutionized many fields like Natural Language Processing, Deep Learning and Computational Linguistics. Word embeddings capture contextual, semantic and syntactic information of each word in corpus based on the position. Words with similar semantics have similar representations, where each word has its unique vector representation. Firth [20] famously summarized this concept by saying, *a word is characterized by the company it keeps.*

2.3.1 Distributional Semantics

Distributional semantics is an approach to word representation that is based on frequency of term occurrences in the corpus. The easiest way to represent text is count vectorization (one-hot encoding). Each unique word in the corpus will be represented as a sparse vector whose dimensionality is the same as the vocabulary size, resulting in large vectors. One-hot encoding only captures the count of the respective term and does not regard the semantic and relational information. Unlike One-hot, TF-IDF [69] considers the word occurrences across the documents.

$$tf(t,d) = \frac{\text{count of } t \text{ in } d}{\text{number of words in } d} \tag{2.1}$$

$$df(t) = \text{occurrence of } t \text{ in documents} \tag{2.2}$$

$$idf(t) = \log \frac{N}{df + 1} \tag{2.3}$$

$$\textit{tf-idf}(t,d) = \textit{tf}(t,d) \cdot \textit{idf}(t)) \tag{2.4}$$

It is a product of two frequencies; term frequency, a measure of the number of times a term appeared in a document. And inverse document frequency is a measure to determine how common or rare a word is. Words that occur throughout the corpus are given lesser significance to ones that occur more in limited documents. The frequency of word belonging to a document is given by Eq. 2.1 and the inverse document frequency is given by Eq. 2.3. Then TF-IDF is calculated as shown in Eq. 2.4.

On the other hand, the Co-occurrence matrix captures the count of words that co-occur throughout the corpus. It works with the principle that words that occur together tend to have a similar meaning. Advantages of using distributional semantics are as follows. (1) It is the simplest form of word embeddings. Easy to interpret and generate respective vector representation. (2) Generated vector representations entirely depend on the input data i.e, always produce the same results. Two major disadvantages of using distributional semantics are as follows.(1) It is computationally challenging because of high dimensionality. Vectors are sparse with most of entities as 0s.(2) Words with lower frequency are not provided with better representations, which may hold crucial information. On the other hand, stop words get better representations.

2.3.2 Neural Network-based Embeddings

Distributional Semantic Models (DSMs) can be seen as count-based models as they are deterministic in nature and depend solely on the count of words. DSMs can capture limited information when it comes to word semantics and word similarities. However, Neural-based word embedding models can be viewed as predict models, as they assign probabilities to terms and try to predict surrounding words. In 2013, a Google team lead by Mikolov et al. created a prediction-based toolkit, word2vec [48], which revolutionalized the NLP community. It was the first one to accomplish tasks like *king-man+women = queen.* Word2vec model is comparatively faster than its forerunners and less computationally expensive. This toolkit is a combination of two prediction architectures called Continuous Bag-of-Words (CBOW) and Skip-grams.

Skip-grams

When an input of a list of sentences is provided, the Skip-gram model slides a window of defined length through the text and tries to predict the neighbouring words $w(t-2)$, $w(t-1)$, $w(t+1)$, $w(t+2)$ based on the current word $w(t)$. In the sliding window, closer context words are given more importance than further ones. If the vector representation of the word cannot predict the context words, components of the vector are adjusted accordingly. Skip-gram is known to produce better representations for less frequent words and smaller datasets.

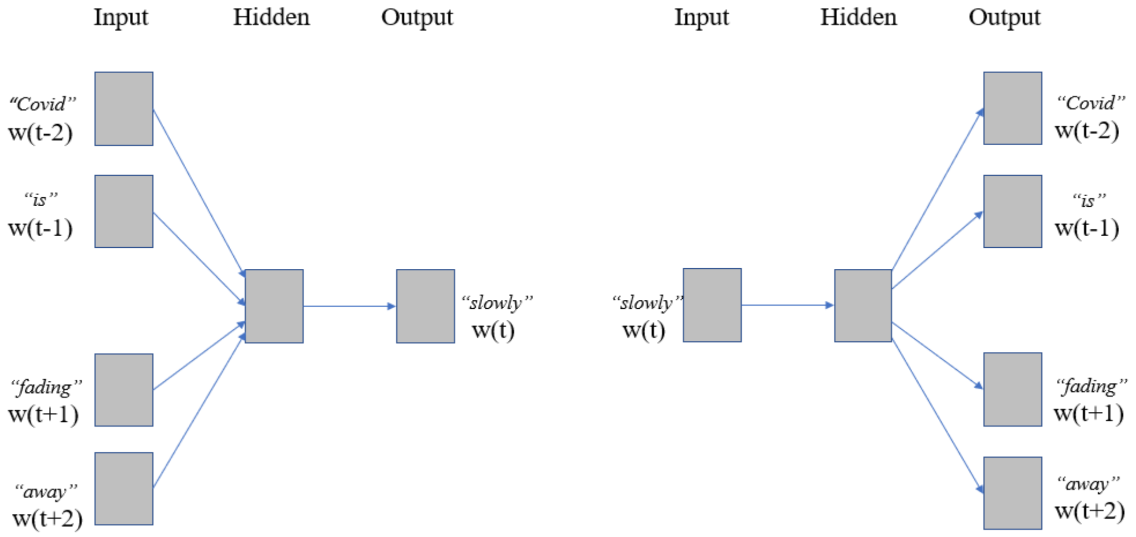


Figure 2.1: Architectures of CBOW and Skip-gram

Continuous Bag-of-Words

In Continuous Bag-of-Words, the distributed representations of context words $w(t-2), w(t-1), w(t+1), w(t+2)$ in the sliding window are combined to predict the target word $w(t)$. The order of word occurrences does not affect the prediction. CBOW is faster to train and does not have huge memory requirements. Both architectures can in seen in Fig 2.1.

2.3.3 Sentence Embedding

Despite the advances in state-of-the-art word embedding techniques, they fail to capture semantic relatedness across large documents and understand the connection of words that are far from each other. These shortcomings were addressed by the doc2vec algorithm. A year after introducing word2vec, Mikolov and his team created an unsupervised algorithm called doc2vec [37] to represent entire sentences in dense vectors.

Doc2Vec

Doc2vec [37] or paragraph2vec is an extension to word2vec that produces fixed-length distributed representations of variable length texts ranging from sentences to documents. The paragraph vectors are inferred with word vectors to predict the primary word, where word vectors are shared. Initially, word2vec randomly assigns word vectors and learns better representations. Vectors are passed through a neural network, and with the help of stochastic gradient descent algorithm, weights are slowly altered by maximizing the likelihood. This results in better predictions and overall better vector representations. It is proven that using jointly embedded space improves the quality of learned document vectors [36]. Paragraph vectors are produced in a similar manner. Doc2vec model extends word2vec by averaging or concatenating the document vector and word vectors to predict the next word in text window. It basically acts as a memory that remembers details missed by word vectors. This model is referred to as Distributed Memory Model of Paragraph Vectors (PVDM).

DBOW is similar to the Skip-gram model, where it randomly samples words from a paragraph and forces the model to predict them, with a paragraph vector as an input. In comparison, PVDM vectors usually excel in most tasks, but authors recommend the combination of both models for better consistency. Due to its flexibility in taking input text with variable length, doc2vec has various applications like text classification, sentiment analysis, sentence similarity, etc.

SentenceBERT

Throughout the history of NLP, eight significant milestones shaped the very perspective of how to make a machine understand natural language. Leveraging some of these breakthroughs like attention and pretrained models, Google introduced a transformer model in 2018 called BERT [16], which stands for Bidirectional Encoder Representations from Transformers.

SentenceBERT [57] is a modification to BERT model [16] that uses siamese and triplet network structures to fine-tune BERT and generate semantically meaningful sentence embeddings. Semantic textual similarity (STS) task can be performed on the generated embedding vectors using measures like Euclidean distance, cosine similarity, Manhattan distance, etc. SBERT architecture practices three main concepts: BERT, Siamese network [68] and pooling layer.

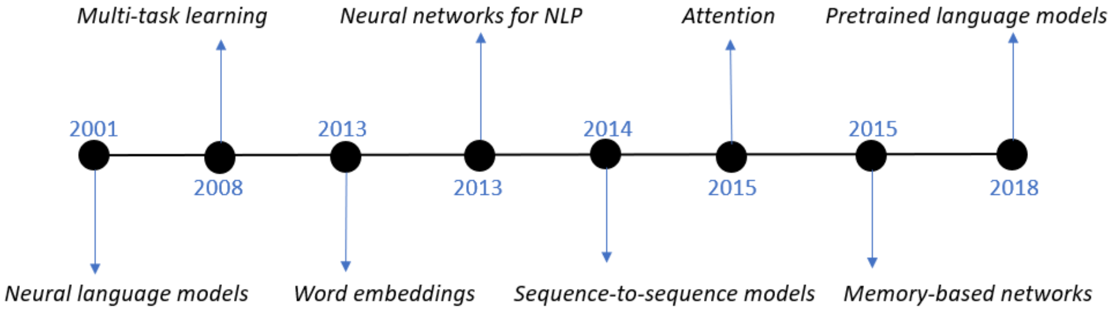


Figure 2.2: Eight milestones of NLP

By the end of the year 2018, the introduction of BERT was the highlight in the NLP community. BERT is a language representation model that captures the semantic meaning and the contextual meaning of the word. For example, the word ‘like’ has different semantic meanings in the sentences “I like playing tennis” and “It’s like when we were kids.” BERT assigns a unique vector representation to each token based on the context.

BERT improves upon existing fine-tuning approaches using a pre-training scheme called Masked Language Model (MLM), which typically masks a percentage of tokens in a sentence and predicts them based on its context. In contrast to traditional left-to-right models, MLM predicts tokens tackling it from both directions, making BERT a deep bidirectional Transformer. BERT, along with its later improvements like RoBERTa [38] and DistilBERT [62], have been trained on large corpora and can be used as pretrained models for a wide variety of language tasks. Sentence BERT optimizes the original BERT with a Siamese/Triplet network structure where the generated weights are shared between both encoders. Sentences pass through pooling layers to generate fixed-sized sentence embeddings.

2.4 Topic Modeling

Topic modeling is an unsupervised machine learning algorithm used to find word and phrase patterns in a set of documents [8]. Topic models are used for various text mining tasks for identifying co-occurring keywords to summarize large collections of textual data. The model takes unstructured and unlabeled text data as input and clusters the word groups that best characterize the documents, with minimum human interaction. For example, topic models can help a librarian organize thousands of digital ebooks based on their genre. In this age of information, manually going through large volumes of textual data is beyond human capabilities. Topics generated can be used to analyze, organize and summarize vast corpus. Thus topic modeling has a wide variety of academic and industrial applications [8] like visualization [32, 14], multilingual modeling [7, 72], query expansion [73] and many more.

The history of topic modeling first began in 1988, with the introduction of an information retrieval technique called Latent Semantic Analysis (LSA) [18]. LSA was the first technique to take advantage of the semantic structure of textual data. The core idea is to create a document-term matrix and decompose it into two matrices: document-topic and topic-term, with a factorization technique called Singular Value Decomposition (SVD). Later, Hofmann proposed an extension to LSA that uses a probabilistic method instead of SVD. Latent Dirichlet Allocation (LDA) [5] is a generative model that assumes each document as a distribution of topics and each topic as a distribution of words. Introduced in 2002, LDA gained a lot of popularity over the years and became a backbone for research and development in the field of topic modeling.

Topic modeling is still an emerging field in the oil industry, and most of the existing research only investigates baseline models like LDA without in-depth study on other models. Due to its simplicity, most of the research recognizes LDA as a preferred method for topic modelling. Dr. Sethupathi Arumugam et al. [2] performed LDA for report classification on a labelled drilling report dataset. This paper also focuses on domain-based ontology to identify risks encountered by offset wells. This proposed framework only works well with labelled datasets and does not discuss the effect of noise. Satyam Priyadarshy et al. [54] trained and compared three different models: SVM, Naïve Bayes and LDA on drilling reports. LDA was proved to be the least efficient due to the small size of the text corpus.

2.4.1 Latent Semantic Indexing

Latent Semantic Indexing (LSI) is one of the fundamental topic modeling techniques. LSI was initially created for automatic indexing and retrieval of documents based on the user's query, by introducing the importance of semantic structure for IR, rather than literally matching the terms in queries with terms in documents. Subsequently, the same idea has been practiced for a broader range of problems in the NLP community. A more general terminology LSA [3] has been used by Jerome Bellegarda in 2005 to describe the approach for topic modeling.

The underlying idea behind LSA is that words that appear in similar contexts will have a similar meaning. LSA constructs a document-term matrix using bag-of-words or TF-IDF [69] and performs matrix decomposition using Singular Value Decomposition (SVD). The document-term matrix contains information on observed term usage across all documents. With rows represented as terms in vocabulary and columns as documents, the matrix stores term occurrences for each document. SVD is a factorization technique that is closely related to a statistical process called factor analysis. The arbitrary rectangular matrix of terms occurrences has different entities on the rows and columns, making SVD a two-mode factor analysis technique. After essential preprocessing and document-term matrix generation, SVD is performed on the matrix to generate three matrices that capture term-term, term-passage and passage-passage correlations.

$$X = T_0 S_0 D_0' \tag{2.5}$$

As shown in Eq. 2.5, T_0 and D_0 are orthogonal matrices with left and right singular values, respectively. Rows of T_0 are vector representations of documents expressed in terms of the topics, where the number of topics t is a hyperparameter passed along with the term-document matrix. Similarly, rows of D_0 are vector representations of terms. S_0 is a diagonal matrix that contains singular values. Truncated SVD reduces matrix dimensionality by considering only the t most significant singular values and keeping the first t columns of T_0 and D_0.

Applications of LSI

LSI technique was originally designed for automatic indexing to overcome the shortcomings of its predecessors. Since then, LSI has gained its importance in numerous applications that go beyond the traditional IR systems. The low-dimensional vector representations of documents can be used for automated classification and clustering using measures like cosine similarity. Due to its ability to interpret large collections of disorganized text on a conceptual basis, LSI is also used for various electronic document discovery (eDiscovery) tasks like Text Smmarization and Information Discovery. The dimensionality reduction property of LSI helps to visualize information in an easily understandable way for a human.

2.4.2 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is the most popular text modeling approach currently in use. Introduced in 2003, LDA is a generative statistical model that can efficiently process text corpus and extract hidden topics. Due to its high modularity and generative nature, the LDA model makes it easier to use in more complex architectures across various applications. LDA is a Bayesian version of probabilistic LSA, and both models require the user to provide the number of topics as model input.

LDA works on the ideology that words contain strong semantic information, and documents handling similar topics have a similar collection of words. LDA assumes that documents are probability distributions over latent topics, and topics are probability distributions over words. Unlike LSA, this model focuses more on probability distributions rather than word frequencies. Along with the vocabulary list and document-word matrix, this model requires two Dirichlet priors as hyperparameter inputs, *alpha* and *beta*. *Alpha* controls the distribution of topics per document, where a smaller value results in documents with less distribution of topics and vice versa. Similarly, *beta* controls the per-topic word distribution, where a smaller value results in topics that will likely have fewer words.

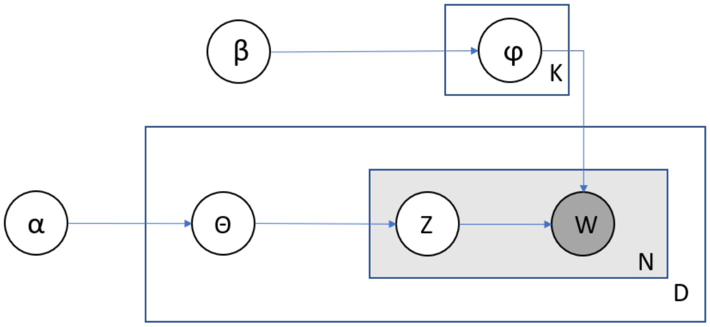


Figure 2.3: LDA architecture

Simply put, LDA is a machine that produces documents and performs reverse engineering to increase the probability of generating original documents. LDA assumes that the corpus contains underlying hidden topics from which documents are generated. Topics can be represented as multinomial distributions over the words of the vocabulary. Document D is produced by sampling a mixture of topics Z and then sampling words W from that topic mixture.

The plate notation in Fig 2.3 captures the variables and their dependencies accurately. M indicates the number of documents in the corpus $D = \{d_0, d_1, d_2, ...d_{\mathrm{M}}\}$, each with N number of words $\{w_0, w_1, w_2, ...w_{\mathrm{M}}\}$. LDA practices Dirichlet prior that can be described as a distribution of multinomial distributions. From the Dirichlet distribution α, we get a random sample of multinomial distribution θ that represents a topic mixture of a particular document. From θ, we select a specific topic Z, based on the distribution. Next, from another Dirichlet distribution β we get a group of multinomial distributions that represent the word distributions ϕ across topics. From ϕ, we choose the word w. Next, from theta, we select a list of topics based on the distribution. Topics combined with ϕ, we get a list of words, one word per topic. We can concatenate these words to obtain a document. The generative process of documents is simplified below.

1. Choose number of words $N \sim \mathrm{Poisson}(\alpha)$,

2. Choose $\theta_i \sim \mathrm{Dir}(\alpha), i \in \{1, \ldots, M\}$

3. Choose $\varphi_k \sim \mathrm{Dir}(\beta), k \in \{1, \ldots, K\}$

4. Choose a topic $z_{i,j} \sim \mathrm{Multinomial}(\theta_i)$.

5. Choose a word $w_{i,j} \sim \text{Multinomial}(\varphi_{z_{i,j}})$.

Gibbs Sampling

Now, the generated documents have topic mixtures where each topic has associated word probabilities. LDA uses inference techniques to increase the probability of producing the original document by assigning suitable topics. The most common inference technique used for LDA is collapsed Gibbs sampling [22]. Initially, the algorithm randomly assigns a topic to each word of all documents. Later it goes through each word and updates it by assuming that every other word-topic association is correct except the one in question. By repeating this process multiple times, the model attempts to make each document as monochromatic as possible. The model also attempts to assign a similar topic to the same words spread across the entire corpus.

2.4.3 Top2Vec Model

Over the last 20 years, traditional modeling techniques like LDA [5], LSA [18] and PLSA [27] have been the cornerstones in the research & development of Topic Modeling. But, due to their several weaknesses, these models are slowly becoming out-of-date in the text mining field. The most critical flaw of probabilistic models is ignoring the ordering and semantics of words in any given text, as these models or operated on a document-term matrix. In contrast, distributional representations of text can capture the semantics and word associations. Models like LDA and PLSA are generative by nature; they design topics that can best recreate the original documents with minimal loss. By doing so, they prioritize words with more frequency over rare keywords that are more informative and add value to the document's theme. The resulting topics are often vague and incoherent. In 2020, Angelov [1] created a word embedding-based topic modeling technique called top2vec, which finds topics that are remarkably more diverse and representative of the text than traditional generative models.

Top2vec is an unsupervised learning algorithm that produces topic vectors by jointly embedding all the word and document vectors in a common vector space. Top2vec leverages several state-of-the-art machine learning techniques like doc2vec, UMAP [46] and HDBSCAN [11] to produce meaningful topics from unlabelled textual data.

Semantic Space

In the top2vec architecture, extraction of meaningful topics begins with creating vector representations for all words and documents and jointly embedding them in a common vector space. This shared space can be referred to as semantic space as projected vectors capture the syntactic and semantic information of the words. Additionally, words that best describe documents are closely positioned to respective document vectors in the semantic space. Top2vec supports two types of embedding techniques; firstly, using doc2vec to train a model from scratch. This method is recommended for large datasets, where it can learn representations by exploring vast data. It is also recommended for domain-specific datasets since pretrained models are trained on general-purpose text corpora and suffer from inherited bias that can negatively affect downstream applications. Secondly, top2vec also supports the use of language models like Universal Sentence Encoder [13] as pretrained embedding models. Few of these models are even suggested for multilingual datasets.

Dimensionality Reduction

In semantic space, documents with semantic similarity are positioned closer to each other, forming a dense cluster. Since the document vectors are accurate representations of their respective underlying topics, it is evident that a dense cluster of documents is indicative of documents having a common theme. Due to its sparsity, finding density clusters in high dimensional space can be needlessly challenging and computationally complex. Top2vec performs dimensionality reduction using Uniform Manifold Approximation and Projection (UMAP) [46, 47] for better interpretation and visualization of document vectors.

Density Based Clustering

After obtaining UMAP reduced low-dimensional representations, the next step is to recognize dense areas of document vectors. A well-recognized clustering technique called Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) [11, 44, 45] is used to achieve this purpose. Unlike K-means, HDBSCAN is an unsupervised technique, as it finds dense clusters without specifying the number of clusters. It is known for distinguishing noisy points in the semantic space from variable density clusters. Identified dense clusters are used to calculate topic vectors, and noisy documents are overlooked by labeling them as outliers.

Topic Vector Generation

Topic vectors are computed for each dense cluster by calculating the arithmetic mean of document vectors belonging to the respective cluster. The number of identified clusters ultimately leads to the number of prominent topics generated for a given corpus. Word vectors that are closest to the centroid of the group, i.e. topic vector, are most representative of the respective topic. The contribution of each word to a particular topic can be determined by the distance between them, where closer words are an indication that they are more semantically similar to the topic. In this manner, top2vec identifies document clusters and generates the most informative topics that are represented by associated word vectors.

To summarize, this chapter primarily talks about the background and related work of various technologies used throughout this thesis. Next chapter introduces three domain-specific datasets handled during various stages of experimentation.

Chapter 3

Datasets

This chapter will give detailed descriptions of the three oil and gas (O&G) datasets used for this project, 1) historical drilling reports data, 2) oilfield glossary, and 3) OnePetro article data. This chapter also describes various challenges faced while handling these datasets.

3.1 Historical Drilling Reports

In the O&G industry, well reports (daily drilling reports) play a vital role in documentation of well-related projects. Well activities are frequently associated with drilling, as wells are drilled to obtain, develop, store, inject, and produce O&G. Drillings happen with the help of a mechanical power rig configured to bore a hole into the Earth's subsurface. To ensure that the activities are performed correctly and that the industry respects and meets all the regulations and standards, operators must create and maintain well reports. Due to their sensitive nature, DDRs are considered confidential documents in many organizations.

DDRs contain detailed analyses on various drilling operations and provide valuable insights for later well-planning exercises. In general, well reports carry information about drilling machinery, geological operations, oil and water analysis, rig activities and logs from tool pusher and company man [54]. Reports contain findings, unfavourable events and summaries for every phase of the project. This information can help foresee drilling risks and mitigate unwanted surprises beforehand, significantly reducing development costs and saving time for future projects [55].

3.1.1 Data Description

The data extracted from historical well reports will be the primary dataset in this project. Meaningful word patterns are extracted from this dataset and submitted to the partner organization to further enhance the existing Information Retrieval system.

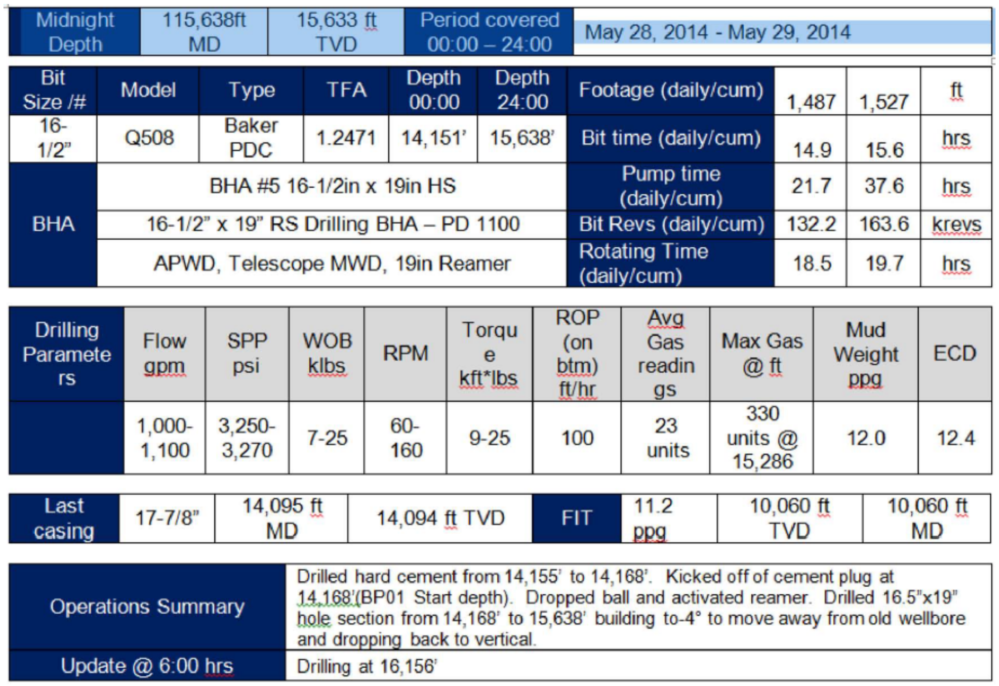
Topic modeling techniques are applied on text extracted from 3,533 historical well reports created between 1970 and 2016. Optical Character Recognition (OCR) is a text recognition technology used to convert printed and handwritten text to machine-readable text. An open-source text recognition engine called Tesseract [58] was used to extract textual data from well reports. About 5% of the documents in the corpus are only comprised of numbers and special characters, indicating that these reports only contained graphical and/or numeric information and can not be used for text analysis. A detailed analysis of document counts can be seen in Table 3.1. We can also see the vast variations in document lengths, ranging from single digits to hundreds of thousands. A snapshot of a standard drilling report can be seen in Fig 3.1 [54]. Most of the information present in these reports is in numerical format with limited textual information. Unfortunately, OCR could not distinguish between valuable text and unwanted characters and extracted all information together, resulting in a messy and noisy dataset. The main challenges with the extracted text are as follows.

No characters	196
100 or less characters	491
100 to 100,000 characters	2742
100,000 or more characters	104
Total Documents	**3533**

Table 3.1: Document counts in the DDR dataset

1. The majority of the historical well reports date back to the late 1970s and are in handwritten format. On top of it, reports also suffer from poor quality due to the deterioration of physical documents over time. Reports are documented by various authors resulting in a vast variability and ambiguity in handwriting styles. As a result, applying OCR on reports did not yield satisfying results, giving rise to unwanted noise.

2. The O&G field is a highly technical and data-intensive field. The narrative in well reports is domain-specific, and semantics are different from everyday language. The extracted text contains numerous abbreviations and named entities that can be challenging for a machine to differentiate from actual noise. Besides, to the best of our knowledge, there are no word embedding models pretrained on an O&G dataset.



Midnight Depth	115,638ft MD	15,633 ft TVD	Period covered 00:00 – 24:00	May 28, 2014 - May 29, 2014

Bit Size /#	Model	Type	TFA	Depth 00:00	Depth 24:00	Footage (daily/cum)	1,487	1,527	ft
16-1/2”	Q508	Baker PDC	1.2471	14,151’	15,638’	Bit time (daily/cum)	14.9	15.6	hrs
BHA	BHA #5 16-1/2in x 19in HS					Pump time (daily/cum)	21.7	37.6	hrs
	16-1/2” x 19” RS Drilling BHA – PD 1100					Bit Revs (daily/cum)	132.2	163.6	krevs
	APWD, Telescope MWD, 19in Reamer					Rotating Time (daily/cum)	18.5	19.7	hrs

Drilling Parameters	Flow gpm	SPP psi	WOB klbs	RPM	Torque kft*lbs	ROP (on btm) ft/hr	Avg Gas readings	Max Gas @ ft	Mud Weight ppg	ECD
	1,000-1,100	3,250-3,270	7-25	60-160	9-25	100	23 units	330 units @ 15,286	12.0	12.4

Last casing	17-7/8”	14,095 ft MD	14,094 ft TVD	FIT	11.2 ppg	10,060 ft TVD	10,060 ft MD

Operations Summary	Drilled hard cement from 14,155’ to 14,168’. Kicked off of cement plug at 14,168’(BP01 Start depth). Dropped ball and activated reamer. Drilled 16.5”x19” hole section from 14,168’ to 15,638’ building to-4° to move away from old wellbore and dropping back to vertical.
Update @ 6:00 hrs	Drilling at 16,156’

Figure 3.1: Sample Drilling Report

3. Well reports are comprised of various graphical and numerical data like pressure gauge recordings, seismic & sonic data and geochemical surveys, with minimum description. Additionally, tables and fill-in-the-blanks are widespread throughout the reports. Units and measurements of various findings are also recorded at every step of the drilling operation. All of this has adversely affected the quality of the extracted text.

4. Text obtained from well reports is the only data provided for this project, which is unlabeled and unstructured in nature. Due to the above-discussed factors, a lot of valuable information like report titles, release dates, geographical locations, onsite well numbers and most importantly, table of content is either lost during the extraction process or is hidden within the rest of the textual data. Based on the nature of the text, programmatically identifying this information is impossible and manually cataloging is not practical.

5. Handling small datasets has always been challenging in NLP and ML disciplines, especially for unsupervised learning like topic modeling. Even though there are various text augmentation techniques like Back Translation [19] and Synonym

Replacement [42], the extracted text does not have a grammatical structure, which is a prime requirement for augmentation. Collecting similar data from other sources is also not possible as historical well reports are considered confidential and legal documents and are not shared publicly or across rival O&G corporations.

3.2 Oilfield Glossary

Every industry across the world has its unique terminology that can be hard for a non-expert to understand. For this reason, major corporations create and manage glossaries for instant reference. These glossaries are frequently updated by technical experts and are made easily accessible to the public. Oilfield terminology consists of terms related to the upstream segment of the O&G industry. It includes words and acronyms relating to professions, companies, equipment, and procedures specific to the O&G domain. It may also include general slang terms commonly used within the industry.

Schlumberger Limited[1] is a renowned oilfield services company with branches in more than 140 countries. Founded in 1926 as an Electric Prospecting Company, Schlumberger quickly expanded its operations in the various petroleum industry sectors like seismic acquisition & processing, formation evaluation, well testing & directional drilling, well cementing and stimulation, software & information management. Since 1998, Schlumberger has been helping specialists and amateurs understand various aspects of oilfield activities with a glossary of oilfield jargon. Schlumberger technical experts often review and update the glossary to ensure efficiency by staying up-to-date with industry trends. Glossary comprises over 4800 definitions, with new information being added as relevant events and technologies unfold in the industry.

Text extracted from historical well reports contains an unusual amount of unwanted noise. As the dataset is O&G domain-specific, programmatically differentiating industry terms from actual noise is not possible. Schlumberger glossary dataset can help identify these terms and, to a large extent, mitigate the risks of losing them. Technical experts at Schlumberger Limited frequently update this glossary on their

[1] https://www.slb.com/

website so that users can re-collect the data for a more updated list. This dataset played an essential role during the data preprocessing stage of the project.

3.3 OnePetro Journal Articles

Topics generated by models are expressed by a list of top k-words that contribute best in representing the respective topic. Assigning a proper label that can best summarize the general theme or subject of topics is essential in topic modeling. Due to the unsupervised nature of models, labeling with minimum human intervention has been challenging in the topic modeling domain. Topic labeling helps end-users better interpret generated topics and also helps in identifying the quality of topics. Generating appropriate labels from the same topic-modeled document collection is questionable when the dataset is extremely noisy. Besides, it is more practical to have a second pair of eyes to determine a topic's quality. For this purpose, an O&G journal dataset is used in this project for topic labeling.

The Society of Petroleum Engineers (SPE)[2] is a non-profit global organization that provides a worldwide forum for professionals in the O&G industry. SPE manages various platforms for easy exchange of technical knowledge that are accessible throughout the world. SPE also organizes various workshops, conferences and teaching courses for educating enthusiasts and identifying industry champions. In March 2007, SPE launched a multi-society library called OnePetro, an online platform for technical literature related to the oil and gas E&P industry. OnePetro is recognized by major O&G organizations and governments and contains thousands of industry-related articles, including journal and conference papers.

3.3.1 Data Description

OnePetro digital library[3] hosts journals that embrace a wide range of subjects in the oil and gas E&P industry. For this thesis, data from about 5000 upstream journal articles was collected and handled for topic labeling. For each article, four features were collected,

- Title of article

[2] `https://www.spe.org/en/`

[3] `https://onepetro.org/`

- Date of publication
- Content — Abstract, introduction and/or summary.
- Subjects — Article categories.

Titles concatenated with corresponding article contents were compared with topic terms in a common vector space, and respective subjects were accepted as label candidates. The publication dates were used to calculate trend analysis to better understand the ongoing research in oil field. OnePetro dataset is authentic and industry-specific, which makes it accurate and reliable for this project.

To conclude, due to the domain-specific nature of our primary dataset (drilling reports text), the oilfield glossary was added to the spell checker dictionary to preserve oil terminology, and the OnePetro article dataset was used to create label candidates for inferred topics. Specifications of both data preparation and topic labeling are explained in detail in the next chapter.

Chapter 4

Methodology

This chapter introduces the techniques used for data preparation for extracting human interpretable topics from an unlabelled, unstructured, noisy, domain-specific dataset. The rest of the chapter is designed to discuss the human evaluation of generated topics and labeling the finalized topics. The proposed architecture can be seen in Fig 4.1.

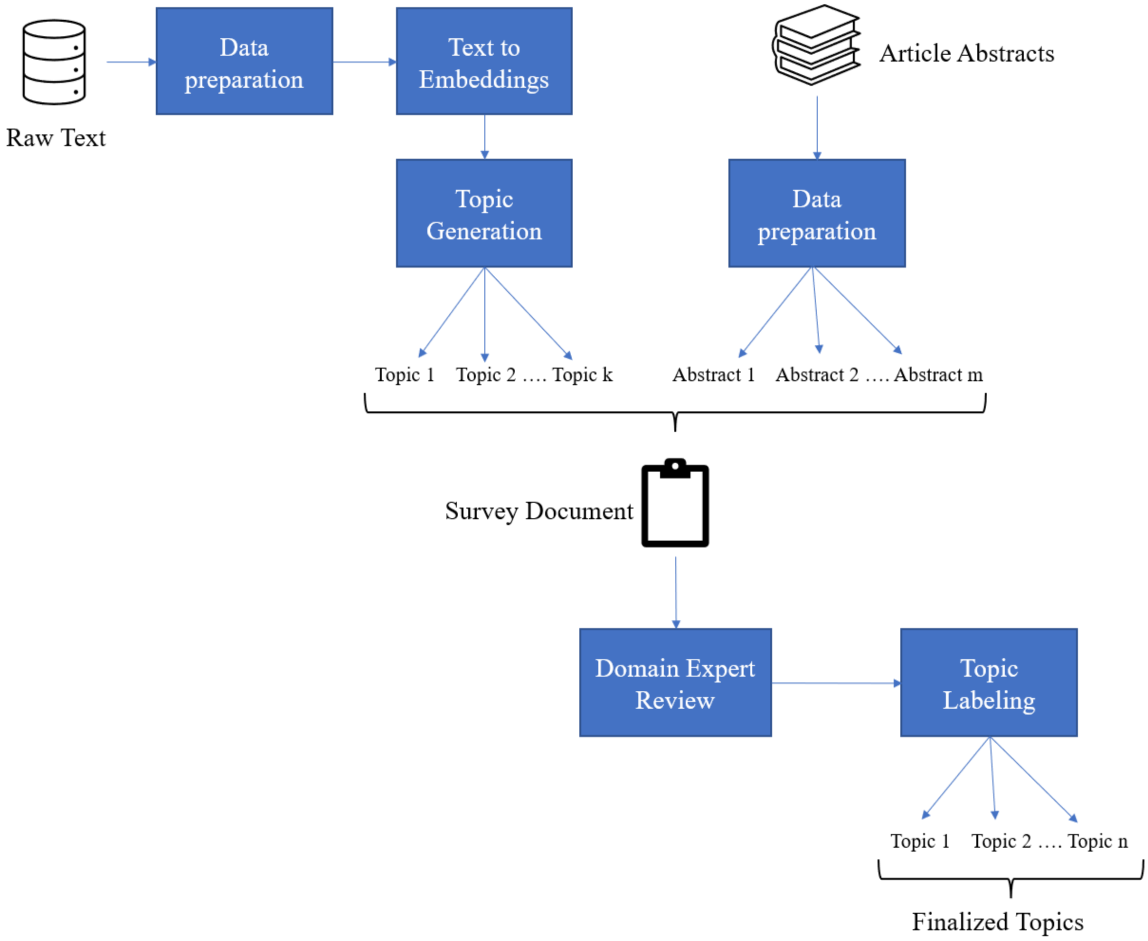


Figure 4.1: Proposed Architecture for Finalizing topics based on expert evaluation

4.1 Data Preparation

Converting raw data to machine-understandable format is crucial across various domains of Machine Learning and Natural Language Processing. As the famous saying goes, "garbage in, garbage out," poor quality input leads to faulty, uninterpretable output, especially in an unsupervised environment. To avoid this, preprocessing techniques are applied to remove or modify unwanted text components and prepare the text for topic modeling.

4.1.1 Daily Drilling Reports

Text preprocessing is traditionally considered a primary and essential step for various NLP tasks. Due to multiple challenges discussed in section 3.1.1, preprocessing plays an important part in this thesis. The entire architecture of preprocessing of DDRs text can be seen in Fig 4.2.

Character Replacements

Well reports contain units of measurement describing various aspects of well construction, like wellbore location, drilling equipment dimensions, logging measurements like natural gamma ray, electrical, acoustic, electromagnetic and pressure. Even though numbers can not contribute much and are eliminated during the preprocessing stage, units, on the other hand, carry essential information regarding the general theme of reports. For example, well location-centric reports contain units in miles and kilometres to pinpoint its geographical location. Since well reports are handwritten or typed by on-site engineers, most measures are expressed in abbreviations like kPa for kilo-pascal and $lb/f3$ for pound per cubic feet. After removing numerical and special characters, these stand-alone abbreviations may be considered as noise by the later implemented spell checker. So, converting these units will help better understand the general theme of the document and also prevent the generation of additional noise. A Python library called Quantulum3 [1] was used to identify the measurements and their units in the text corpus. This library supports more than 290 units and 75 entities.

[1] `https://github.com/marcolagi/quantulum`

Stop Word Removal

Stop words are the most common and general words in any natural language. Stop words do not contribute much to the meaning of documents for analyzing text data and building NLP models. In English language, articles and prepositions are usually considered non-important and removed during preprocessing. Stop word removal is essential for many NLP problems like text classification and topic modeling, but not advisable for tasks like text summarization and machine translation.

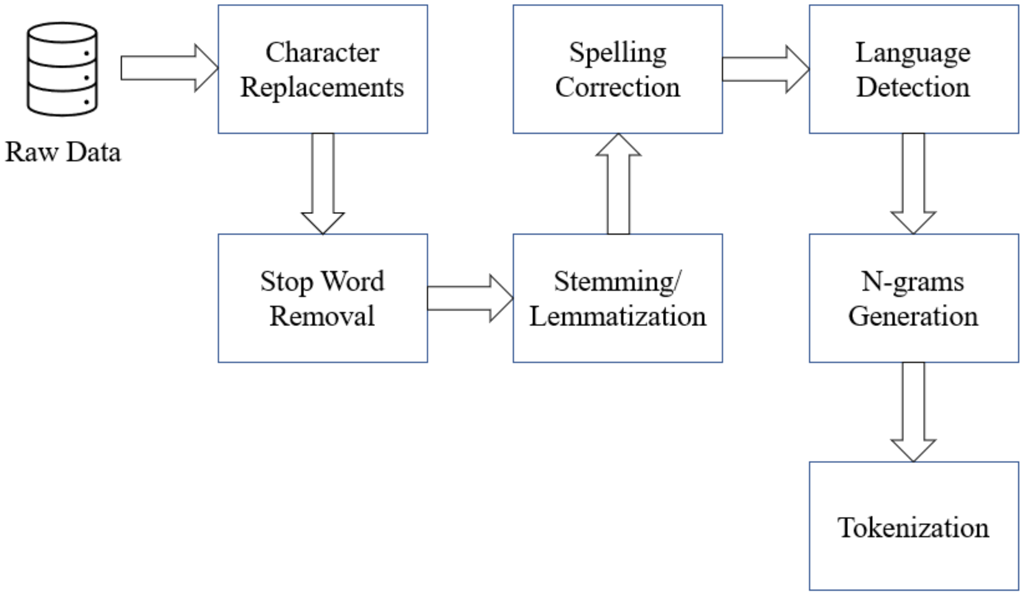


Figure 4.2: Drilling reports data preparation

Lemmatization

Text normalization is a process of transforming a word to its canonical form by removing extensions and only considering the root form. Unlike humans, a machine can not identify the relation among words like *eat, ate, eaten* and treats them separately during text representation like BoW. Normalization can be done by two processes, stemming [39] and lemmatization [50]. Stemming is a rudimentary rule-based approach that refers to a predefined list of prefixes and suffixes and removes them from a given word. On the other hand, lemmatization reduces words to their base word, ensuring that the root word belongs to the language. The root word is called lemma, and it is the dictionary form of a set of words.

Topic modeling relies on the frequency and arrangement of content words. Having multiple variants of a single word across in the corpus will negatively affect the topic generation by grouping variants together and calling it a unique topic. For that matter, any model that exercises BoW, TF-IDF, or any other vector representation benefits from lemmatization.

Spell Correction

Noisy textual data leads to poor model quality. Even though OCR systems have gotten better and more precise over the years, handwritten text recognition remains a challenge. Hence spell correction plays a vital role in OCR post-processing. Spell checkers help identify and modify spelling errors to increase the quality of the text. Generally, spell correction algorithms parse through the entire text, check each word if available in the customized dictionary, and find the most relevant match for non-vocabulary words. Spell checkers perform various operations like insert, delete, transpose and replace to match input term with dictionary term, based on the Damerau–Levenshtein edit distance between the strings. Symspell spelling correction [21] algorithm reduces the complexity of edit candidate generation and dictionary lookup by substituting all operations with only deletion, applied on both input and dictionary terms.

For most spelling checkers, the user is provided with multiple alternative suggestions for a single non-vocabulary word from which a choice can be made. This process requires human involvement and can be time-consuming, especially for large documents. Fully-automatic spelling correction techniques selects the best choice from the list of suggestions. After tokenizing the input text, individual tokens and dictionary tokens are modified with a limit of maximum edit distance to find the perfect suggestion. Symspell supports four different comparison pairs for individual tokens. $e1$ and $e2$ are edit distances and $(e1 + e2) \leq \textit{Maximum edit distance}$.

- input term == dictionary term
- delete(input term, $e1$) == dictionary term
- input term == delete(dictionary term, $e2$)

- delete(input term, $e1$) == delete(dictionary term, $e2$)

Symspell also generates suggestions for combined tokens. The algorithm takes multi-word input and checks for mistakenly missed space between two correct words and wrongly added space in a correct term leading to two incorrect terms. The algorithm prioritizes finding suggestions for individual tokens over combined and split tokens.

Dictionary quality is paramount in spell corrections. Symspell uses a combination of two data sources for dictionary creation; Google books Ngram[2] data and Spell Checker Oriented Word Lists (SCOWL)[3]. Words that only occur in both sources are considered, and the finalized list has approximately 80,000 most frequent words. Google Ngram data also provide information about word frequencies, which helps determine the importance of candidate words with the same edit distance. Words that occur more often are prioritized over the ones that occur less often. A big disadvantage of other spell correctors is that they solely depend on their dedicated dictionaries for recognizing domain-specific terms. Like any other industry, O&G has a unique terminology that includes acronyms and named entities, which will be considered non-vocabulary words and altered during the correction. Symspell allows creating a personalized dictionary by adding terms to the existing vocabulary list. The oilfield glossary discussed in section 3.2 contains petroleum terms and definitions that industry experts frequently update. About 21.3% of the words in our data were neither in Symspell dictionary nor in the oilfield glossary. This glossary is added to the dictionary before running a spell checker through the extracted text to preserve petroleum terms from being altered.

Spell Correction example

Fpr the fou rth timein a row, Finland has be6n namedthe happyest coun try

for the fourth time in a row, finland has been named the happiest country

[2] http://storage.googleapis.com/books/ngrams/books/datasetsv2.html

[3] http://wordlist.aspell.net/

Language Detection

Topic modeling helps in recognizing the general theme occurring across documents. To understand the general theme, it is important to disregard the documents that will negatively affect the outcome of topic models. After running a spell checker, a language detection algorithm is performed to filter out noisy documents from the dataset. A language detection library called Langdetect [64] is used to identify these documents, which is a direct port of Google's library from Java to Python. Langdetect is popular for having over 99% precision for 50+ languages and outputs the two-letter ISO 639-1 code of primary language. Latin script is the most commonly used language system that is used by languages like English, French, Italian and Danish.

The DDR dataset contains reports that specifically discuss laboratory results like pressure-volume relations and boiling point ranges of elements. Text extracted from these reports is comprised of more numerical and special characters than alphabets. Initial preprocessing eliminated non-alphabets and significantly reduced the text size. Random alphabets that were previously associated with non-vocabulary characters are messy, and documents that are mostly comprised of these alphabets negatively affect model performance. Langdetect identifies the probabilities of the languages that the text belongs to. Since this algorithm is non-deterministic, it detects a different incorrect primary language whenever a noisy English text is given as an input. These outputs were used to separate noisy unidentifiable texts from proper English texts. Identifying language for small documents is challenging, and most of the documents that did not have primary language as English had comparatively fewer characters. These documents were manually compared with the original reports and filtered out during preprocessing stage.

N-gram Generation

In the field of computational linguistics, N-grams are a sequence of items in any given sentence. Items can be both characters or words but are usually referred to for words. N-grams are extensively used in Statistical Natural Language Processing and ML fields because they capture the co-existing nature of tokens. Unigrams are one-word sequences, which are basically tokens, and bigrams are two-word sequences, and so on. Word n-grams reflect information about context and content. Certain combinations of words hold more meaning when tokenized unitedly.

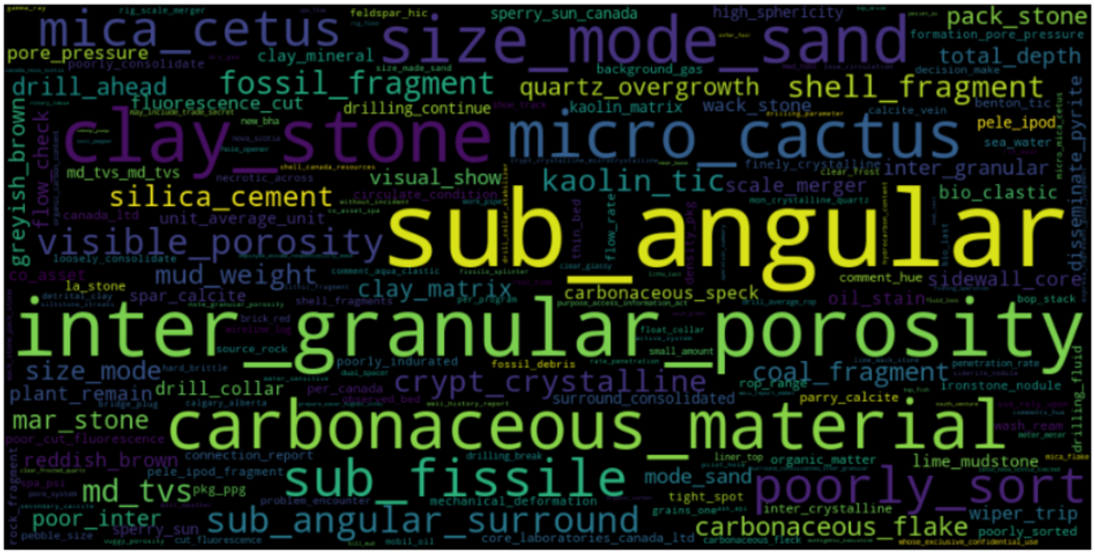


Figure 4.3: WordCloud of bi/trigrams

Bigrams and trigrams are generated and tokenized along with unigrams. A powerful NLP visualization tool called *Wordcloud* [10, 66] was used to exhibit the importance or frequency of the tokens, which can be seen in Fig 4.3. The top 20 most frequent n-grams were listed and presented to oil domain experts to learn the quality of generated n-grams. Experts approved that all n-grams are authentic in terms of their relation to O&G, with minor and recognizable spelling errors.

4.1.2 OnePetro Articles

OnePetro digital library is a trustworthy and definitive resource in the oil and gas E&P industry. The industry is often divided into upstream, midstream and downstream, where upstream refers to locating and extracting crude oil and natural gas from the ground. Well reports are considered an essential part of the upstream sector as they discuss in detail the wellbore projects. Journals available in OnePetro cover a broad range of O&G sectors, including upstream. Therefore, the OnePetro article dataset is used for labeling the topics generated from the DDR dataset.

Unlike text extracted from the well reports, OnePetro article abstracts have grammatical structure and are free from noise. Along with article article, other details like title, date of publication and subjects are also obtained. For every article, title was concatenated with the corresponding abstract and passed through below mentioned preprocessing pipeline:

1. “Lower-casing”
2. Stop word removal
3. Removal of numbers and special characters
4. Lemmatization
5. Tokenization

4.2 Topic Modeling

After data preprocessing, the text is tokenized into smaller units. A unit can be a word or an n-gram. Topic modeling is a process of extracting meaningful word patterns, allowing for a better understanding of documents. For this project, a word embedding-based topic model called top2vec [1] is trained to generate meaningful topics from preprocessed text. Top2vec is unsupervised in nature and produces topic vectors by jointly embedding all the word and document vectors in a common vector space. Before finalizing top2vec, various topic modeling approaches were trained and compared, and these details are explained in chapter 5.

The history of topic models began with the introduction of Latent semantic indexing in distributional semantics. Since then, LSA and Latent Dirichlet Allocation [5], a generalization of probabilistic LSA, became wildly popular and remained the most commonly used models to this date. Traditional models are often considered as baseline models as they are easy to implement. But, despite their popularity, they have several weaknesses and do not work well with complicated datasets. One of the most prominent weaknesses is not considering the semantic and syntactic relationships in a given text. Top2vec is a state-of-the-art word embedding technique that generates vector representations based on the context. The top2vec model trained for this project outperformed traditional models across various evaluation measures.

Top2vec generates vector representations in two ways: training a doc2vec model from scratch and using transfer learning with a pretrained language model. Training a model from scratch can be recommended for large datasets with domain-specific vocabulary. Pretrained models are known for their efficiency and good performance for various low-level tasks. In this project, both types of models are trained and compared in chapter 5. The model trained with doc2vec produced more coherent and diverse topics than the one that used the pretrained embeddings. Top2vec also supports *distiluse-base-multilingual-cased* [57] pretrained sentence transformer for multilingual datasets.

4.3 Topic Labeling

Topics generated by models are conventionally expressed as the topmost representative words *(space, andromeda, moon, tesla, star)*, sorted according to their importance. Since topic models operate on unlabelled datasets, the generated clusters of words are not explicitly labelled *(Space Exploration)* by the models and require human intervention. Understanding every topic based on terms alone can be challenging for end-users, especially if models are trained by non-domain experts. Labels assigned manually can also be subjective, as it entirely depends on the individual knowledge of the reviewer. In this project, we propose an approach for generating a label candidate list for every topic using text embedding and sentence similarity techniques. The proposed approach consists of three steps: (1) topic label generation based on OnePetro journal articles, (2) expert evaluations of shortlisted label candidates, and (3) finalizing labels and filtering topics based on survey responses.

4.3.1 Candidate Generation

Wikipedia encyclopedia[4] contains is a vast collection of knowledge and can be used for general purpose or multilingual use cases. An approach for querying Wikipedia articles using topic terms was proposed by Lau et al. [35], to generate the most relevant candidate labels and rank them. The problem with that approach is that it requires two external Wikipedia resources, among which one is no longer publicly available. As

[4] `https://en.wikipedia.org/wiki/Main_Page`

an improvement, Lau et al. [4] in 2018 proposed using neural embeddings to generate distributed representations of article titles and topic terms and measure relevance between them. In this project, OnePetro journal corpus is used for generating label candidates.

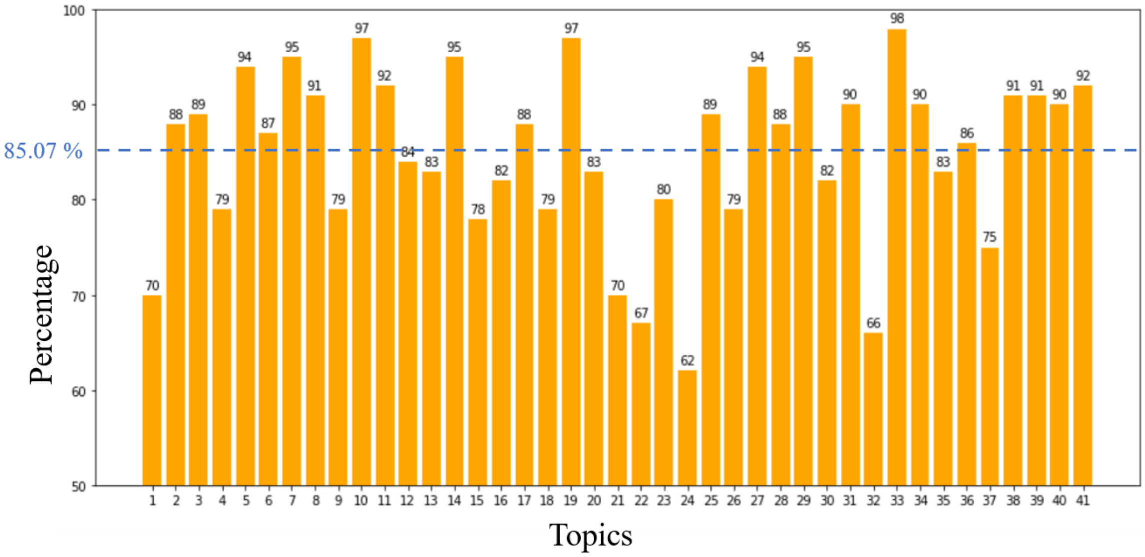


Figure 4.4: Percentage of topic words in Doc2vec vocabulary

The reason behind using the OnePetro dataset is its resemblance to drilling reports, as they both talk about upstream oil and gas. In *onepetro.org*, each article is labelled with a unique set of subjects that represents the theme of the journal's content. These subjects are used to label the topics modelled from the drilling report dataset. To examine the reliability of the OnePetro dataset, a corpus vocabulary was built after preprocessing and tokenizing the articles, and the percentage of topic terms present in vocabulary was calculated. Fig 4.4 shows that an average of 85.07% of topic terms are present in the created dictionary across 41 topics generated by the top2vec model.

Text embedding is a process of converting words, phrases or sentences into numbered vectors by capturing semantic and syntactic context. A document embedding model called doc2vec is trained on OnePetro dataset to create distributed representations in a common vector space. Along with document embeddings for each article, doc2vec internally runs word2vec, learning representations for words as well. As discussed earlier, the topic terms present in OnePetro vocabulary are considered to find related journals from which topic labels are obtained. Given the top-N terms of each topic, the topic embeddings are the word embeddings of these terms. Every article is represented by respective document embedding. Relevant articles are found by performing pairwise cosine similarity with the word embeddings of top-N topic terms and aggregated by taking the arithmetic mean:

$$Similarity(A, T) = \frac{1}{|N|} \sum_{r=1}^{T} \cos(Emb(w_r), Emb(a)) \tag{4.1}$$

The cosine similarity of an article with each topic term is calculated and averaged, where $Emb(w_r)$ is the topic term's word embedding and $Emb(a)$ is the OnePetro article's document embedding. $Similarity(A, T)$ is the final similarity measure between a topic T and an article A. Based on the cosine similarity, top 20 most relevant articles are considered for each topic, and the five most common subjects of these articles are finalized for the domain expert survey.

4.3.2 Domain Expert Review

Discovering topics is an unsupervised process, and despite the model's good performance, generated topics may not match the user's interests. Besides, there are no gold standard topics for drilling reports to compare with our topics. Hence, topic labeling can be challenging, especially when the person who trains the models is not a subject matter expert (SME). In this project, we created a survey document where O&G domain experts can assess the generated topics using labels extracted from the OnePetro dataset. This survey helps determine the quality of topics and label them based on the user responses.

The review document was shared with three project engineers of the partner organization, whose expertise lies in the exploration and development sector. The document consists of a total of 41 topics, represented by the top 10 most semantically similar terms. Each topic is presented with a series of labels that can best summarize the topic, as shown in Fig 4.5. Survey takers were asked to score each of those labels based on their relevance to the topic words, using the following ordinal scale:

1. *perfect label*, a label that accurately describes the topic.

2. *reasonable label*, a label that does not fully capture the nature of the topic.

3. *slightly related label*, that is somewhat related to what topic is addressing.

4. *completely unrelated label*, where the label is truly unrelated, or topic words are extremely noisy and do not make sense.

The finalized SME review document comprises 41 topics, each represented by ten terms and a list of five labels with a corresponding rating dropdown. Along with the list of labels, a text field is also provided for each topic where SMEs can manually enter a more appropriate label. This option allows us to differentiate good interpretable topics with a poor set of labels from noisy "garbage" topics. Clear instructions on how to take the survey were provided in the document, along with a sample example for reference. Engineers were also notified about the nature of topic terms like the possibility of n-grams, spelling errors, abbreviations and named entities. The document was made user-friendly and shared with engineers in editable pdf formats.

4.3.3 Label Evaluation

After the survey results came back from project engineers, ratings were compared and analyzed to determine the interpretability of topics. The first step is to organize topics based on ratings. The percentages and frequencies of each rating for three review documents can be seen in Fig 4.6. Two main observations were made from the survey:

1. The rating *very good* dominates in all surveys, indicating that most topics have at least one label that perfectly summarizes the topic terms. Furthermore, it also indicates that the majority of these topics are comprehensible by experts. Approximately 50% of topics across three surveys have one or more labels that perfectly summarize topic terms.

2. In a total of 41 topics, only 4% of topics have all inappropriate labels, i.e. three topics in SME 1 and a topic each in SME 2 and SME 3. As mentioned earlier, survey takers were presented with an option to manually enter a more suitable label, which was not utilized for these inappropriate topics. This only indicates that these topic terms collectively do not make any sense and are difficult to follow.

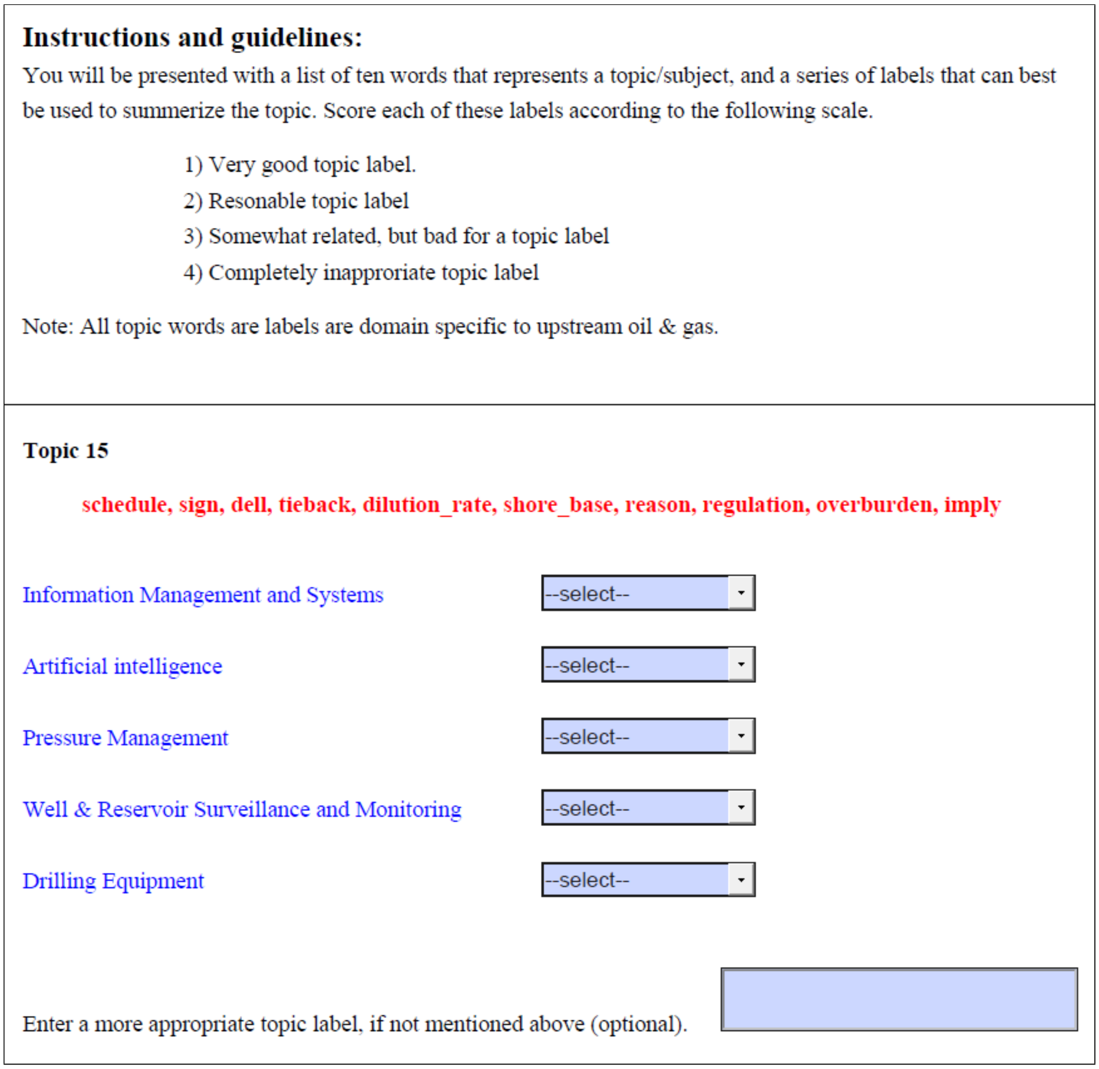
Topics with *very good* or *reasonable* labels in a minimum of two survey documents were only considered for further evaluation. Out of 41 topics, 30 came under the before mentioned scenario. After discarding trash topics, assigning accurate labels was the final task at hand. Each topic is assigned two types of labels where each serves a specific purpose: (1) After ranking the labels for each topic, based on the user ratings, the label with the highest score is assigned as the primary label for the respective topic. (2) As topics are modelled from documents confined to a specific industry, and as all label candidates are selected from a shared pool of article subjects, few sets of topics have the same labels with the highest scores. To uniquely distinguish these topics, a secondary label is also assigned to each topic, which satisfies two requirements: the label should have *very good* or *reasonable* ratings in two or more surveys, and the label should be unique and not associated with other topics. The final output can be seen in Fig 4.7, with finalized topics and corresponding labels.

4.4 Trend Analysis

The Society of Petroleum Engineers (SPE)[5] collaborates with top global oil and gas industry experts and has published various peer-reviewed journals. As SPE manages OnePetro, the articles dataset used for this thesis perfectly portrays the ongoing research in the oil industry. Research trends are most likely affected by the industry's new challenges and innovations, and understanding them will help experts better plan their future projects. Hence, trend analysis was performed using the subjects and publication date of each article. As these subjects are also used as labels for generated topics, domain experts can select well reports based on the research trends. For each finalized label, the percentage of published papers that talk about that subject was calculated per year.

We can notice some interesting patterns in the Fig 4.8. Popularity for *Improved and enhanced recovery* subject was high in 2012 and gradually declined until 2019, and *Unconventional and complex reservoirs* was trending in 2013 from which it slowly declined. The most noticeable pattern is a sudden uptrend of many subjects in 2020, possibly due to the global pandemic's impact on the industry. Our proposed methodology helps domain experts to examine these trends and find relevant reports.

[5] `https://www.spe.org/en/`



Instructions and guidelines:

You will be presented with a list of ten words that represents a topic/subject, and a series of labels that can best be used to summerize the topic. Score each of these labels according to the following scale.

1) Very good topic label.
2) Resonable topic label
3) Somewhat related, but bad for a topic label
4) Completely inapproriate topic label

Note: All topic words are labels are domain specific to upstream oil & gas.

Topic 15

schedule, sign, dell, tieback, dilution_rate, shore_base, reason, regulation, overburden, imply

Information Management and Systems	--select--
Artificial intelligence	--select--
Pressure Management	--select--
Well & Reservoir Surveillance and Monitoring	--select--
Drilling Equipment	--select--

Enter a more appropriate topic label, if not mentioned above (optional).

Figure 4.5: A snapshot of domain expert survey document. Topic terms are represented in red, and labels are represented in blue. Each label is associated with a rating dropdown alongside.

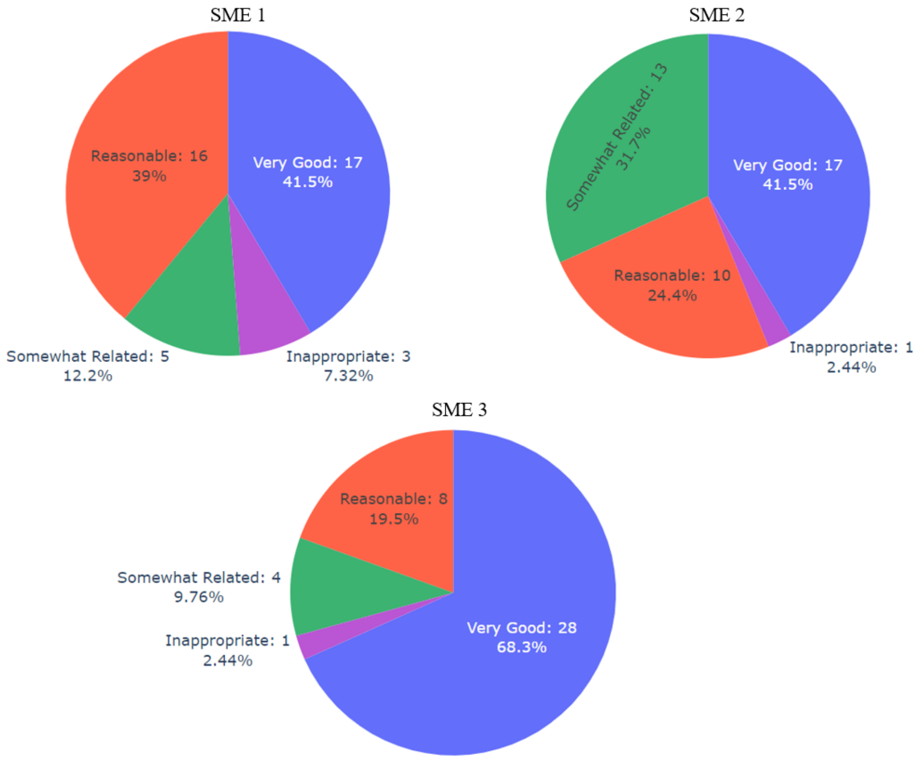
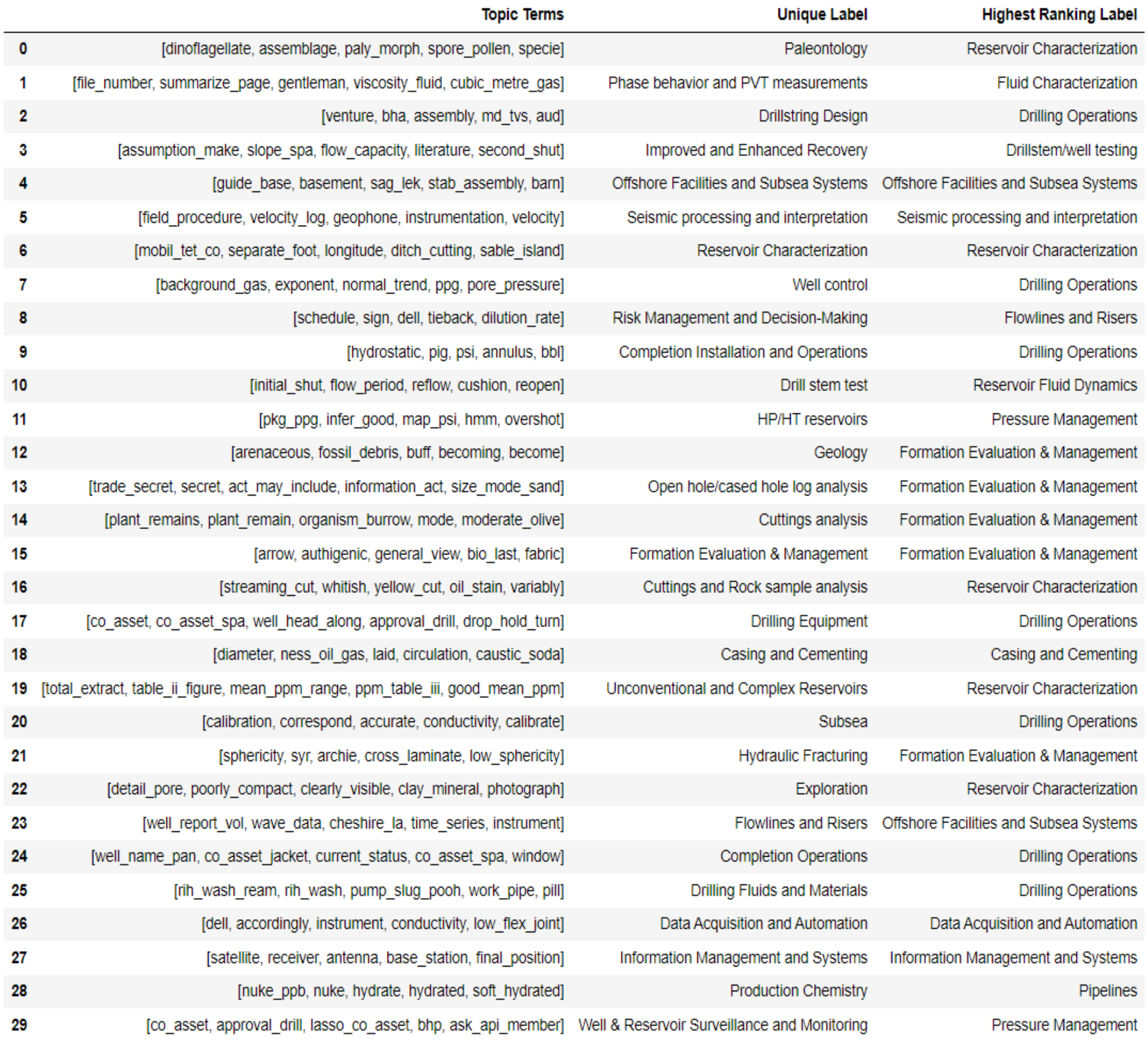


Figure 4.6: Pie Charts displaying survey results from three annotators. Each pie slice refers to the percentage of label rating for all 41 topics.



	Topic Terms	Unique Label	Highest Ranking Label
0	[dinoflagellate, assemblage, paly_morph, spore_pollen, specie]	Paleontology	Reservoir Characterization
1	[file_number, summarize_page, gentleman, viscosity_fluid, cubic_metre_gas]	Phase behavior and PVT measurements	Fluid Characterization
2	[venture, bha, assembly, md_tvs, aud]	Drillstring Design	Drilling Operations
3	[assumption_make, slope_spa, flow_capacity, literature, second_shut]	Improved and Enhanced Recovery	Drillstem/well testing
4	[guide_base, basement, sag_lek, stab_assembly, barn]	Offshore Facilities and Subsea Systems	Offshore Facilities and Subsea Systems
5	[field_procedure, velocity_log, geophone, instrumentation, velocity]	Seismic processing and interpretation	Seismic processing and interpretation
6	[mobil_tet_co, separate_foot, longitude, ditch_cutting, sable_island]	Reservoir Characterization	Reservoir Characterization
7	[background_gas, exponent, normal_trend, ppg, pore_pressure]	Well control	Drilling Operations
8	[schedule, sign, dell, tieback, dilution_rate]	Risk Management and Decision-Making	Flowlines and Risers
9	[hydrostatic, pig, psi, annulus, bbl]	Completion Installation and Operations	Drilling Operations
10	[initial_shut, flow_period, reflow, cushion, reopen]	Drill stem test	Reservoir Fluid Dynamics
11	[pkg_ppg, infer_good, map_psi, hmm, overshot]	HP/HT reservoirs	Pressure Management
12	[arenaceous, fossil_debris, buff, becoming, become]	Geology	Formation Evaluation & Management
13	[trade_secret, secret, act_may_include, information_act, size_mode_sand]	Open hole/cased hole log analysis	Formation Evaluation & Management
14	[plant_remains, plant_remain, organism_burrow, mode, moderate_olive]	Cuttings analysis	Formation Evaluation & Management
15	[arrow, authigenic, general_view, bio_last, fabric]	Formation Evaluation & Management	Formation Evaluation & Management
16	[streaming_cut, whitish, yellow_cut, oil_stain, variably]	Cuttings and Rock sample analysis	Reservoir Characterization
17	[co_asset, co_asset_spa, well_head_along, approval_drill, drop_hold_turn]	Drilling Equipment	Drilling Operations
18	[diameter, ness_oil_gas, laid, circulation, caustic_soda]	Casing and Cementing	Casing and Cementing
19	[total_extract, table_ii_figure, mean_ppm_range, ppm_table_iii, good_mean_ppm]	Unconventional and Complex Reservoirs	Reservoir Characterization
20	[calibration, correspond, accurate, conductivity, calibrate]	Subsea	Drilling Operations
21	[sphericity, syr, archie, cross_laminate, low_sphericity]	Hydraulic Fracturing	Formation Evaluation & Management
22	[detail_pore, poorly_compact, clearly_visible, clay_mineral, photograph]	Exploration	Reservoir Characterization
23	[well_report_vol, wave_data, cheshire_la, time_series, instrument]	Flowlines and Risers	Offshore Facilities and Subsea Systems
24	[well_name_pan, co_asset_jacket, current_status, co_asset_spa, window]	Completion Operations	Drilling Operations
25	[rih_wash_ream, rih_wash, pump_slug_pooh, work_pipe, pill]	Drilling Fluids and Materials	Drilling Operations
26	[dell, accordingly, instrument, conductivity, low_flex_joint]	Data Acquisition and Automation	Data Acquisition and Automation
27	[satellite, receiver, antenna, base_station, final_position]	Information Management and Systems	Information Management and Systems
28	[nuke_ppb, nuke, hydrate, hydrated, soft_hydrated]	Production Chemistry	Pipelines
29	[co_asset, approval_drill, lasso_co_asset, bhp, ask_api_member]	Well & Reservoir Surveillance and Monitoring	Pressure Management

Figure 4.7: List of finalized topics represented by top-5 terms and associated labels

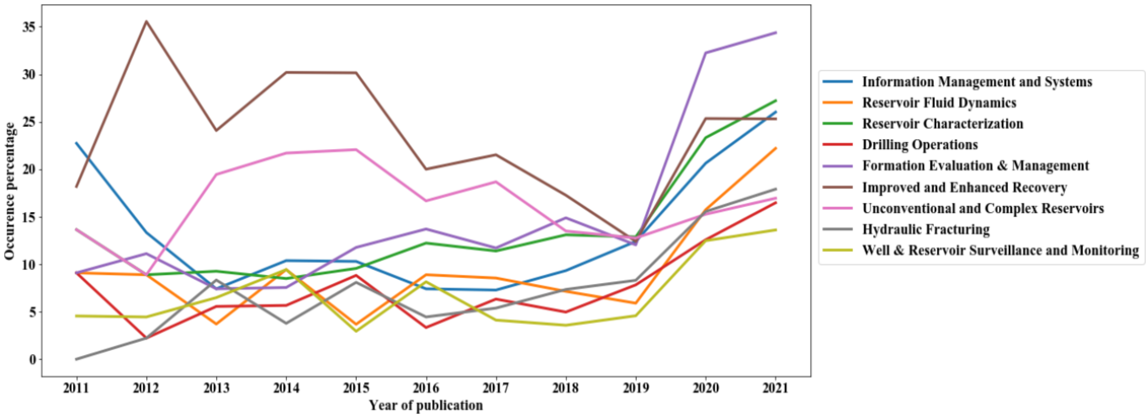


Figure 4.8: Trend analysis of labels over time

Chapter 5

Experiments and Results

This chapter introduces various evaluation methods used to examine a specific feature of topics for choosing the best topic model. The rest of the chapter explains the experimental setup of four distinct topic models and compares the end results.

5.1 Evaluation Methods

When evaluating a topic model, we are primarily concerned with how well a human can understand the representative words of each topic. Unlike many other ML techniques, topic models do not provide a gold standard evaluation metric to determine the quality of produced topics. Topic models only generate the most probable or semantically similar terms and cluster them together to represent a group of documents that possess these terms. Perplexity or likelihood is a commonly used statistical metric in topic modeling for determining how adequately the model performs on unseen test documents. Perplexity fails to capture the semantics in text and is inconsistent with human judgement [15]. Besides, perplexity is only helpful for predictive models and splitting small datasets into train and test sets is not recommended since topic modeling is unsupervised and requires more data for training.

Due to the complexity of outputs, human involvement has always been a decisive factor in evaluating topic models and labeling the topics. These techniques vary from simple human eye-balling methods like manually reviewing all topics to more advanced, fully interactive user-centred systems like human-in-loop topic modeling [34]. Advanced models practice different methods like word and topic intrusions [15], where user studies are conducted to check whether users can identify these out-of-the-place words (or topics). These techniques require a large pool of human subjects and are not feasible for our project since the corpus is domain-specific and, most importantly, reports are confidential.

Text extracted from drilling reports was preprocessed and trained with four topic modeling algorithms: LDA, LSA, top2vec using a pre-trained embedding model and top2vec using doc2vec. Multiple evaluation measures were used to understand the various properties of generated topics. The model that outperformed in most of the observations was finalized for topic labeling and domain-expert survey. The following properties are considered to distinguish the best model from others:

- Topics inferred by the model should be distinctive and not have a significant overlap with each other.
- Topic terms should be more specific, as most of the terminology present in training data is comprised of technical oil and gas words.
- Document distribution amongst topics should be reasonably uniform, rather than one topic representing most of the documents.
- Topic terms should be semantically similar and co-occur in a majority of the documents they represent.
- Inferred topics should match with human judgment and should be easily interpretable by domain experts.

Each of the above characteristics of topics was investigated and compared using an evaluation measure.

5.1.1 Topic Diversity

Irrespective of the type of topic model, each topic is represented by a list of terms arranged in the descending order of their importance. For generative models, topics words are ranked based on their weights, and for embedding-based models, topic words are ranked based on their cosine similarity to the topic embedding. So basically, each topic can be considered as a group of a ranked list of topic terms. To understand the stability of a topic model, it is essential to calculate how similar or distinct the topics are to each other. When a majority of topics have the same words as high-ranking topic terms, it indicates that generated topics are not unique. Two list similarity measures are used to calculate this distinctness of topics: Jaccard Index and Ranked-biased overlap.

Jaccard Index

Given two finite sample sets, the Jaccard index or Jaccard coefficient [30] is used to find similarities between them. As shown in Eq. 5.1 Jaccard index can simply be defined as size of the intersection over size of the union. If both the topics have identical top words, then the Jaccard similarity value will be 1, and vice versa. The undesirable properties of this measure are it does not take the order of the sample set into consideration, and topics are ranked lists where words appearing earlier in list hold more value.

$$J(T_1, T_2) = \frac{T_1 \cap T_2}{T_1 \cup T_2} \tag{5.1}$$

Ranked-Biased Overlap

Ranked biased overlap (RBO) [67] is a more robust measure to estimate the similarity of indefinite ranking lists like topics that are not necessarily conjoint, meaning same words in both topics. In contrast to Jaccard Index, RBO gives preference to high-ranking words by assigning more weight to them. RBO can also handle lists with different lengths. The equation that can be used to calculate the RBO score of two lists T_1 and T_2 can be seen below 5.2.

$$RBO(T_1, T_2, p) = (1 - p) \sum p^{d-1} A_d \tag{5.2}$$

where, d indicates the depth unto which words are considered. A_d is calculated as follows,

$$X_d = S_{:d} \cap T_{:d}$$
$$A_d = \frac{X_d}{d}$$

X_d is the the size of overlap of T_1 and T_2 upto depth d and A_d is called Agreement of T_1 and T_2, given by size of overlap divided by the depth d. Hyperparameter is tunable and ranges from 0 to 1, which is used to determine the contribution of high-ranking words to final RBO similarity value. When $p = 0$, only the top-ranked word is considered, and as p approaches 1, the importance spreads flat through the list.

By default, the p value is set to 0.9, as the authors of RBO recommend it. The final RBO score also ranges from 0 to 1, where 0 means disjoint and 1 means identical sets.

5.1.2 Vocabulary Analysis

The corpus used for this project is text extracted from historical well reports, most of which date back to the late 1970s. All these reports contain highly technical information regarding various drilling-related services like mechanical rig activities and landscape analysis. Hence, after the removal of stop words during the data preparation stage, a lot of the terminology contains technical words, like names of drilling apparatus and soil compositions. The topic models trained on this dataset should also capture these unique words to represent the respective documents. If generated topics are vague and not explicit, these groups of words cannot accurately distinguish between various topic-document collections. To test that, we calculated the percentages of topic terms present solely in the oil and glossary dataset, which was used for spell corrections. This evaluation helps in understanding the unambiguous nature of generated topics.

5.1.3 Document-Topic Distribution

Another necessary factor to consider for identifying the most suitable topic model is the distribution of documents per topic. If a small number of topics represent most of the documents, it indicates that the model fails to produce distinctive topics. In generative models like LDA and LSA, each document is treated as a probability distribution over all topics, and each topic is treated as a probability distribution over all words; meaning that each document can be expressed using every single topic with a different topic weight. So, we considered only the most dominant topic in each document for evaluation. After acquiring document counts for each topic, standard deviation was used to calculate how spread out data is from the mean. Standard deviation for observed values $x_1, x_2, x_3, \ldots, x_n$ can be calculated using formula 5.3.

$$s = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2} \tag{5.3}$$

where N is the size of the number of observations and $\overline{x}$ is the mean value of observations.

5.1.4 Coherence Measure

Alongside perplexity, Topic coherence is one of the widely used evaluation measures for topic modeling and is supported by various open-source libraries like Gensim [56]. Coherence measures [59] gained popularity in the text mining field, as unsupervised learning techniques like topic modeling do not guarantee the interpretability of their output. Coherence is a measure of the degree of semantic similarity among high-ranking terms in every topic. It calculates co-occurrence scores of words within documents being modelled. The concept behind coherence calculation is closely related to embedding representations of text as coherence also deals with syntactic information with the help of a sliding window that moves over the corpus and checks occurrences. Coherence can be calculated using various approaches. The most commonly used ones are C_v, C_umass, C_uci and C_npmi.

1. C_v uses one-set segmentation of top words to calculate word pair probabilities by passing a boolean sliding window through the trained text. Then, a confirmation measure uses normalized pointwise mutual information (NPMI) and cosine similarity to find similarities between topic terms.

2. C_uci [51] uses pointwise mutual information to calculate word co-occurrence counts by passing a sliding window through the corpus.

3. C_umass [49] uses a logarithmic conditional confirmation measure between the top pair of words.

4. C_npmi is an improved version of the C_uci measure that uses normalized pointwise mutual information.

A systematic study by Michael et al. [60] explored a multitude of coherence measures and their relations to human labelled data. The best performing coherence score is the one most related to human judgements, which is C_v. This coherence measure is also the default value for the Gensim library [56], which indicates its popularity. Hence, C_v coherence measure was used in this thesis to calculate the semantic similarity between topic top-ranking terms.

5.2 Training Models

Four different topic models, two classical models, a embedding-based state-of-the-art model with two different settings, were trained for this research. Output topics were evaluated and compared using the above-mentioned evaluation measures.

5.2.1 Latent Dirichlet Allocation

Latent Dirichlet allocation is considered one of the traditional topic modeling algorithms that have been studied and practised by multiple researchers throughout the years. LDA model not only requires the number of topics to be specified by the user but also needs two other hyperparameters that decide the topic distribution over documents and word distribution over topics. Multiple LDA models were trained by iterating over a predefined hyperparameter space to determine the optimum values, and coherence metric was used to compare and learn the best model.

Most of the high-ranking keywords are imprecise and repeated across various topics, as we can see in Table 5.1. The word *grain* is present in the top 10 terms of topics 4 through 7, and the words $fine$, *grey*, *well* and *hole* appear in most of the topics. Even after filtering out high-frequency terms from tokenized data before passing it through the LDA model, the vagueness in topics did not improve. Experimenting with a wide range of hyper-parameters did affect the scores of evaluation measures but did not improve the quality of topics. Top words like *shale*, *pct* and *sandstone* are domain-specific in nature, but the overall structure of topics is hard to interpret.

For a more thorough understanding of how much helpful information LDA managed to capture, we calculated the percentage of topic words that are only present in the oilfield glossary used for spelling correction during data preparation. Interestingly, only 0.7% of the high-ranking words are present solely in the oil glossary, and the rest 99.3%, can be seen in a general-purpose dictionary. As discussed in data preparation section, word n-grams were generated before tokenization to capture the corpus's contextual information, and only 2.14% of the generated topic terms are n-grams. All these characteristics of topics demonstrate that LDA is not a suited model for this project.

Topic Number	Top 10 words
Topic 1	gas, hydrocarbon, sample, zone, well, interval, oil, section, one, meter
Topic 2	cd, al, us, old, st, salt, la, mobil_oil, pm, foot
Topic 3	pressure, well, test, gas, use, report, psi, sand, oil, time
Topic 4	grain, sample, porosity, quartz, clay, cement, sandstone, fine, authigenic, calcite
Topic 5	grey, sandstone, calcareous, fine, shale, trace, medium, light, grain, brown
Topic 6	pct, fine, trace, grain, material, size_mode_sand, comment, calcareous, silt, range
Topic 7	grain, fine, sandstone, medium, trace, shale, minor, part, cement, grey
Topic 8	trace, minor, grey, fine, crystalline, white, brown, show, porosity, argillaceous
Topic 9	grain, porosity, quartz, cement, locally, sample, clay_mineral, well, moderately, siderite
Topic 10	well, hole, drill, run, meter, section, depth, case, mud, spa
Topic 11	interval, sample, well, indicate, occur, low, assemblage, section, age, present
Topic 12	fine, grain, medium, sandstone, grey, trace, coarse, argillaceous, friable, shale
Topic 13	hole, drill, gray, run, bit, shale, well, calcareous, sandstone, light
Topic 14	cement, hole, drill, run, well, pump, pkg, case, mud, section

Table 5.1: Top 14 Topics learned by LDA Model

5.2.2 Latent Semantic Analysis

Latent semantic analysis is also considered as one of the classical topic models in the NLP community. LSA initially creates a document-term matrix using TF-IDF [69] and then performs a dimensionality reduction using various techniques like Principal Component Analysis or Singular Value Decomposition. Like training an LDA model, LSA also requires the number of topics to be specified as a hyperparameter. After experimenting with a wide range of hyperparameters, an LSA model with the best coherence score was finalized for further evaluations and comparisons.

Compared to topics generated by LDA, LSA top words are much more ambiguous and repetitive. Topic 1 is the most dominant topic for nearly half of the total number of documents, and is expressed by the top terms *gray*, *grain*, *sandstone*, *grey*, *minor*, *part*, *trace*, *drill*, *hole* and *crystalline*. These terms can neither individually nor collectively be used to identify a general theme in the corpus and most of these words reappeared in many other topics. Furthermore, the top-ranking terms in all topics are present in a common words database, indicating that the model failed to capture valuable knowledge from our complex training data.

Topic Number	Top 10 words
Topic 1	gray, grain, sandstone, grey, minor, part, trace, drill, hole, crystalline
Topic 2	pct, fine, material, size_mode_sand, sandstone, comment, range, brown, gray, carbonaceous_material
Topic 3	grey, gray, hole, cement, trace, run, drill, well, brown, pump
Topic 4	fine, trace, grain, grey, sandstone, calcareous, shale, medium, pct, light
Topic 5	trace, grey, gray, minor, crystalline, slightly, hard, medium, porosity, sandstone
Topic 6	grain, brown, gray, slightly, clay_stone, minor, sandstone, calcareous, firm, part
Topic 7	moderately, minor, fine, porosity, sandstone, brown, part, inter_granular_porosity, silty, coarse
Topic 8	moderately, grey, part, fine, trace, siltstone, cement, shale, minor, hard
Topic 9	trace, minor, moderately, grain, shale, sandstone, pct, porosity, hard, slightly
Topic 10	grain, sample, shale, porosity, hard, sandstone, quartz, trace, light, grey
Topic 11	part, calcareous, siliceous, brown, silty, shale, moderately, limestone, sandstone, porosity
Topic 12	medium, gray, sample, brown, fine, silt, light, range, hard, minor
Topic 13	calcareous, pyrite, siltstone, brown, minor, hard, sand, md_tvs, argillaceous, friable
Topic 14	fine, pct, shale, siltstone, porosity, grain, minor, glauconite, firm, clay_stone
Topic 15	fine, moderately, hole, cement, grain, hard, gas, glauconite, well, dark

Table 5.2: Top 15 Topics learned by LSA Model

5.2.3 Top2Vec with pretrained model

Top2Vec is an embedding-based topic modeling technique that has gained tremendous popularity since its publication in 2020. It generates embedding representations for all words and documents in the corpus and clusters semantically similar documents together in a common vector space. As discussed in previous chapters, it practices various ML techniques like dimensionality reduction, spatial clustering and transfer learning. Top2Vec creates joint embeddings either by training a model from scratch or by using a pretrained model (PTM) that was trained on a massive dataset. We used a PTM called Universal Sentence Encoder [13], which was created by Google in 2018. Unlike classical models, we do not need to specify the number of topics to top2vec as it clusters the semantically similar documents together and learns topics in this process.

This model formed 12 dense clusters of documents, meaning that there are only 12 topics in total. Firstly, an eyeballing approach of manually reviewing the top-N words suggested that these topics are comprised of plenty of domain-specific and characteristic terms compared to classical models. For example, the top 10 words of topic 10 are *diameter, pi, spherical, sphere, cylinder, circular, case, perforation, thickness* and *casing,* which are clearly talking about the units and measurements of probably a drilling hole or machinery dimensions. But, the modelled topics have an unusual amount of words forms like topic 2 contains *geologist*, *geology*, *geological* and topic 6 contains *calibration*, *calibrate*, *accuracy*, *accurate*, *accurately* and many more. This outcome can be a side effect of using a PTM that was trained on massive domain-general datasets. Overall, top2vec topics look much promising and meaningful when compared with other models.

5.2.4 Top2Vec

Next, we trained a top2vec model from scratch using the doc2vec embedding technique that the author recommends for a unique vocabulary dataset. As discussed earlier, top2vec automatically finds the number of topics and does not require complicated tuning to determine the optimum number. Top2vec operates with the help of UMAP and a neural network, which doc2vec uses for creating embeddings. Since both these algorithms are stochastic and have built-in randomness to them, we ran this model

Topic Number	Top 10 words
Topic 1	misfire, finer, lag, sands, fracturing, ripple, geological, bbl, coarser, metamorphism
Topic 2	geologist, geology, geological, borehole, sediment, gneiss, shale, fracturing, schist, drilling
Topic 3	pressure, frac, gas, petroleum, geologist, exxon_mobil, analysis, gaseous, analyse, geology
Topic 4	misfire, finer, lag, sands, fracturing, ripple, geological, bbl, coarser, metamorphism
Topic 5	schedule, mottling, interpret, discrepancy, kaolin_tic, contingency, pooh_lay, interpretation, washout, deviate
Topic 6	ci, hmm, err, fig, actually, way, upon, one, several_attempt rih_wash_ream
Topic 7	calibration, calibrate, accuracy, histogram, datum, time_series, accurate, resume, summary, accurately
Topic 8	analyse, analysis, samples, sampling, sample, distillation, analytical, analyses, specimen, extract
Topic 9	bbl, misfire, shoulder, sands, ley, pct, lag, pine, haiti, bluish, leu
Topic 10	diameter, pi, spherical, sphere, cylinder, circular, case, perforation, thickness, casing
Topic 11	displacement, bottom, hole, downwards, lateral, calculated, displaced, upward, directional, compute
Topic 12	samples, sampling, sample, specimen, extract, analyse, sampler, study, chromatography, baseline
Topic 13	gas, gaseous, alert, indicate, state, indicator, solid, tell, bottom, exxon_mobil

Table 5.3: Topics learned by Top2Vec Model using a pretrained model

multiple times to check for consistency. Model produced 40 topics on average with minimum variation between topics.

Most of the topic words generated by this model are very specific to the oil industry and cannot be interpreted by a non-expert, which indicates that this model successfully recognized the important information. Unlike the top2vec model trained using pretrained embeddings, this model did not cluster word forms together. Also, compared to previously discussed models, these topics contain the most percentage of word n-grams and oilfield dictionary words. Since this model aced in all the evaluation measures discussed in the following section, top2vec using doc2vec was finalized and sent to subject matter experts for further review.

5.3 Comparisons

Further quantitative analysis was conducted to examine the coherence and diversity of learned topics. Results can be seen in the table 5.4. Top2vec has the highest C_v coherence score of 0.65%, indicating that a majority of topic terms occur together in most of the documents they represent. On the other hand, topics learned by top2vec using a pretrained model have the least coherence score because around 50% of the documents in training data are clustered together and represented by topic 1. Even though the high-ranking words of LDA and LSA topics are vague and repetitive, they have comparatively better coherence scores as these terms are consistently occurring in respective documents.

Model	Coherence Score	Jaccard Similarity	RBO Score
LDA	0.6003	0.0928	0.1221
LSA	0.5851	0.1716	0.2124
Top2Vec (pretrained)	0.4107	0.0196	0.0220
Top2Vec	0.6513	0.0061	0.0086

Table 5.4: Topic coherence and similarity scores of four models

Basic visual evaluation (eyeballing) of the learned topics can tell us that embedding models produced much more valuable and informative results than the classical models. To back that up, Jaccard similarity and ranked-based overlap metrics were used to prove that quantitatively. The top2vec model has the least scores for both metrics, which indicates that learned topics are diverse. In contrast, LDA and LSA have the higher scores.

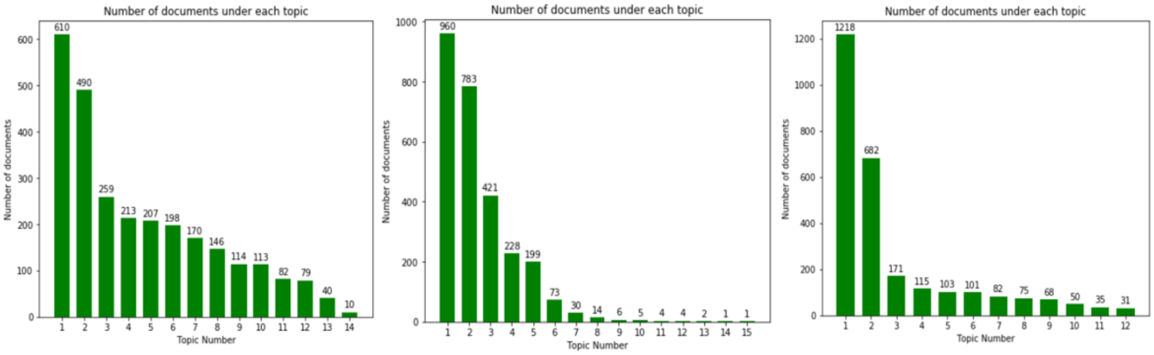


Figure 5.1: Document distribution over topics for models LDA(left), LSA(center) and Top2vec with PTM(right)

Model performance can also be determined by the document distribution across topics. Standard deviation was calculated to check the spread of documents across topics. For LDA, LSA and top2vec (pretrained) models, the first two topics represent more than 50% of documents and have high SD values of 161.37, 295.83 and 343.03. Top2vec model has a low SD value of 51.47, indicating that documents per topic values are much closer to the mean, and the distribution is uniform.

Major drawbacks of LDA

1. Even though LDA is considered the backbone of topic modeling, it has various limitations compared to state-of-the-art models. LDA takes Bag of words as input which does not acknowledge the syntactic information of the text.

2. LDA treats each document as a mixture of topics and each topic as a mixture of words. This assumption results in a significant overlap of topics across documents instead of creating distinctive groups of documents.

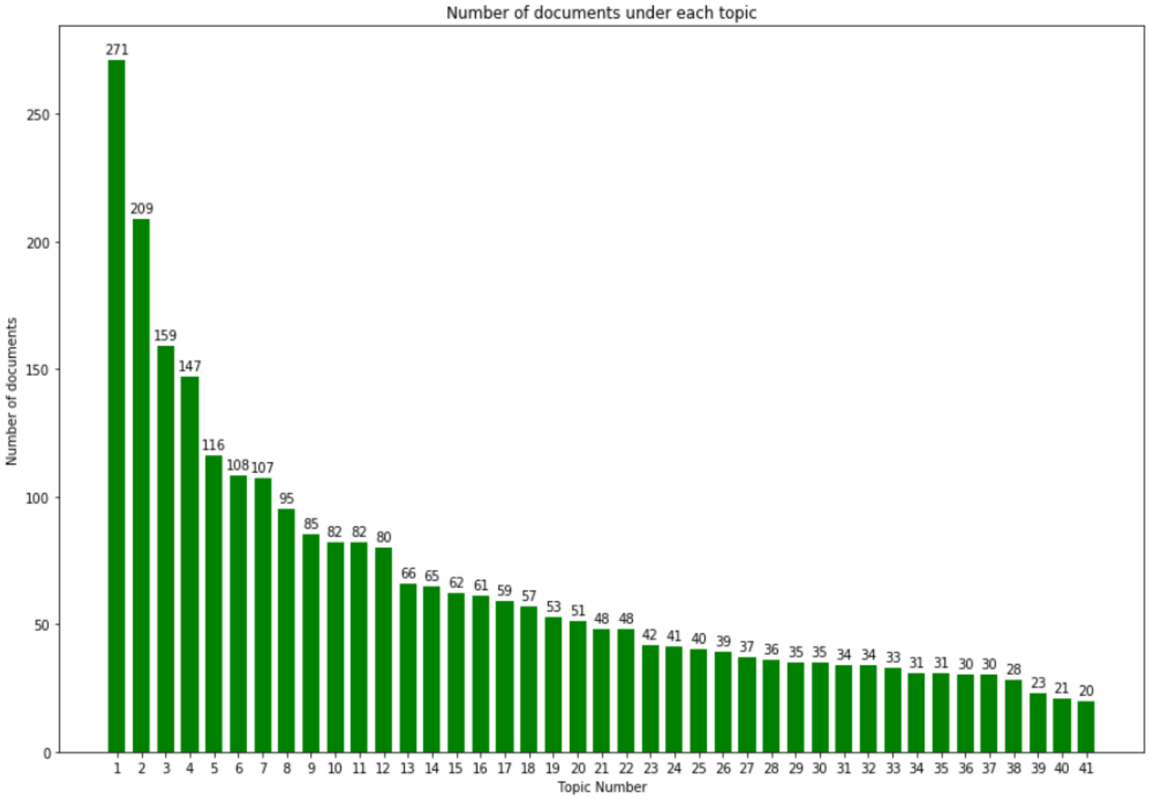


Figure 5.2: Document distribution over topics by Top2Vec model

Major drawbacks of LSA

1. Initially created for dimensionality reduction, LSA became a popular topic modeling technique that requires huge training data, which is decomposed to produce topics. This algorithm is not recommended for small datasets.

2. LSA assumes that documents and words are normally distributed, which may not be appropriate for all observed data.

3. LSA model also takes bag-of-words or TF-IDF as input, which only considers term frequencies and not the order of words in the corpus.

Major drawbacks of Top2vec with PTM

1. Even though recent developments in PTMs have revolutionalized the NLP community, most of the models are trained on general-purpose English text. Using

this for domain-specific downstream tasks is not advisable because of differences in vocabulary terms. Unfortunately, to our knowledge, there are no pretrained models trained on drilling reports.

2. Words and abbreviations can have different meanings based on the usage, a.k.a, polysemy. For example, *well* means *good* and a *drilling pit*, PVT is an abbreviation for *private* and *pressure*, *volume*, *temperature*. Corpus trained on standard corpus will assign embeddings to these terms based on the context, which might be different for domain-specific corpus, leading to improper representations.

Advantages of using Top2vec

1. Top2vec does not require a user to specify the number of prominent topics, as it finds by clustering similar documents together in a common vector space.

2. When trained from scratch, this model also learns representations for unique domain-specific terms, including abbreviations and n-grams.

3. The representative words for each topic are basically n-closest words in vector space, indicating fewer chances of a single term dominating in many topics. This increases the diversity of topics.

4. Top2vec uniformly distributes documents across topics through HDBSCAN clustering algorithm. Even though the parameter "min count per cluster" was set to 15, the smallest cluster has 20 data points.

5. Top2vec also finds the most representative topic for new documents without changing the existing word, document or topic vectors. If the new documents count is more than the trained documents count or has a vast new vocabulary, the model is re-trained from the start for best results.

6. Top2vec internally uses UMAP and HDBSCAN for finding density clusters in a lower-dimensional space. This model allows users to control various hyperparameters like $nearest\_neighbours$, *embedding_metric* and *embedding_dimension* for UMAP, *min_cluster_size* for HDBSCAN.

After a thorough analysis of the above-mentioned topic modeling techniques and their generated topics, top2vec results were finalized and sent to the domain experts to check human interpretability and assign labels. A detailed description of topic labeling is given in chapter 4.

Chapter 6

Conclusion

This thesis proposes implementation of topic modeling on a noisy, domain-specific corpus to produce human interpretable word (or phrase) groups. Our proposed data preparation pipeline can handle various complexities of textual datasets and bring structure for a better document management system. Most of these reports date back to the 1970s or 80s and are handwritten or printed by multiple authors. Due to the poor quality of physical records, performing OCR led to high amounts of unwanted noise and loss of valuable information. Hence, along with conventional preprocessing techniques, data preparation comprises more advanced spell correction, character replacement and language detection methods. Daily drilling reports play a vital role in documenting everyday activities that take place on a drilling rig. On-site project engineers maintain DDRs by entering valuable information like geological conditions, unfavourable events, best practices and lessons learned during every stage of the project. For new projects, project engineers often refer to these old related reports for making better decisions. Preserving this knowledge can help anticipate drilling risks and mitigate unwanted surprises beforehand, significantly reducing development costs and saving time for future projects. Manually going through massive amounts of unorganized reports is tedious and time-consuming. This research aimed to use topic modeling to organize these reports so engineers can access them effectively.

Traditional topic models like LDA, PLSA and LSA use BoW or TF-IDF, which does not capture the contextual information. An embedding-based topic model called top2vec was trained with two different embedding techniques and compared with traditional models. A thorough quantitative analysis was conducted using various evaluation metrics to determine the best topic model. Top2vec trained using a doc2vec embedding showed the best results, and industry experts approved these topics as the most human interpretable. Topic models only discover word groups that often occur together in a text corpus. Assigning appropriate topic labels that can best summarize the topics can help users quickly find relevant documents. Manually going through extracted topics can be time-consuming and biased, especially for domain-specific datasets where all topics resemble each other. For automatic labeling, we propose associating the daily drilling reports with oil industry journal articles to create a list of candidate labels. Oil and gas SMEs were consulted to review generated topics and score candidate labels based on their relevance to topic words. The survey helped in excluding trash topics and assigning suitable labels to good ones. This research aimed and succeeded in transforming a small unlabeled noisy domain-specific dataset into organized groups of documents represented by high-ranking terms, which are in turn represented by topic labels. These topics and document clusters can be used for various applications like query expansion, document classification and text summarization.

6.1 Future Work

As discussed earlier, inferred topics can be integrated with the existing information retrieval system to expand user queries. Grouped documents can be used to build a document classification system that the topic labels can represent. The vector representations of all jointly embedded entities can be further handled to create ontologies [9] and knowledge graphs. Each drilling report is associated with an offsite well and contains information like container identity, operator name, test type and location. This information was lost during the OCR text extraction process as no special precautions were taken to target this data. With this valuable information, ontologies can be created to identify relationships between offsite wells and observed topics. Ontology-based representations can also be used to understand complex hierarchical connections necessary for a semantic interpretation. It can help answer questions like the role of a geophone (topical word) in calculating seismic processing (label).

Well reports also have release dates or spud dates (the date when actual drillings commence). These dates can be used to examine the evolution of topics over time. Dynamic topic models [6] belong to a family of time series models take sequentially organized documents as input and analyze the evolution of trends of topic words. DDRs are dated from 1970 to 2016, and they primarily talk about the struggles and challenges engineers face during multiple stages of drilling. Dynamic topic modeling calculates the topic progression and can help project engineers investigate how various challenges have unfolded over time and compare the best practices.

Bibliography

[1] Dimo Angelov. Top2vec: Distributed representations of topics, 2020.

[2] Sethupathi Arumugam, Shebi Rajan, and Sanjay Gupta. Augmented Text Mining for Daily Drilling Reports using Topic Modeling and Ontology. volume Day 4 Wed, April 26, 2017 of *SPE Western Regional Meeting*, 04 2017. D041S011R003.

[3] J.R. Bellegarda. Latent semantic mapping [information retrieval]. *IEEE Signal Processing Magazine*, 22(5):70–80, 2005.

[4] Shraey Bhatia, Jey Lau, and Timothy Baldwin. Automatic labelling of topics with neural embeddings. 12 2016.

[5] David Blei, Andrew Ng, and Michael Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 05 2003.

[6] David M. Blei and John D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 113–120, New York, NY, USA, 2006. Association for Computing Machinery.

[7] Jordan Boyd-Graber and David Blei. Multilingual topic models for unaligned text. *arXiv preprint arXiv:1205.2657*, 05 2012.

[8] Jordan Boyd-Graber, Yuening Hu, and David Mimno. Applications of topic models. *Foundations and Trends® in Information Retrieval*, 11:143–296, 01 2017.

[9] Paul Buitelaar, Philipp Cimiano, and Bernardo Magnini. *Ontology Learning from Text: Methods, Evaluation and Applications*. 01 2005.

[10] Michael Burch, Steffen Lohmann, Daniel Pompe, and Daniel Weiskopf. Prefix tag clouds. In *2013 17th International Conference on Information Visualisation*, pages 45–50, 2013.

[11] Ricardo J. G. B. Campello, Davoud Moulavi, and Joerg Sander. Density-based clustering based on hierarchical density estimates. In Jian Pei, Vincent S. Tseng, Longbing Cao, Hiroshi Motoda, and Guandong Xu, editors, *Advances in Knowledge Discovery and Data Mining*, pages 160–172, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[12] Canadian Association of Petroleum Producers. Canada's economic contribution — Canada natural resources & GDP, 10 2021.

[13] Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Universal sentence encoder, 2018.

[14] Allison Chaney and David Blei. Visualizing topic models. *ICWSM 2012 - Proceedings of the 6th International AAAI Conference on Weblogs and Social Media*, 05 2012.

[15] Jonathan Chang, Jordan Boyd-Graber, Sean Gerrish, Chong Wang, and David M. Blei. Reading tea leaves: How humans interpret topic models. In *Proceedings of the 22nd International Conference on Neural Information Processing Systems*, NIPS'09, page 288–296, Red Hook, NY, USA, 2009. Curran Associates Inc.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding, 2019.

[17] E. L. Dillon. Information Storage and Retrieval Systems. volume All Days of *World Petroleum Congress (WPC)*, 04 1967. WPC-12146.

[18] S. T. Dumais, G. W. Furnas, T. K. Landauer, S. Deerwester, and R. Harshman. Using latent semantic analysis to improve access to textual information. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '88, page 281–285, New York, NY, USA, 1988. Association for Computing Machinery.

[19] Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. Understanding back-translation at scale, 2018.

[20] J. R. Firth. A synopsis of linguistic theory 1930-55. 1952-59:1–32, 1957.

[21] Wolf Garbe. SymSpell, 6 2012.

[22] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741, 1984.

[23] Tedros Adhanom Ghebreyesus. WHO Director-General's opening remarks at the media briefing on COVID-19 — 11 March 2020, 03 2020.

[24] Danny Haydon, Neeraj Kumar, and Jacob Bloom. Industries Most and Least Impacted by COVID-19 (Probability of Default Perspective) Recovery Insights: March 2021 Update, 03 2021.

[25] Marti Hearst. Untangling text data mining. *Proc 37th annual meeting of the Association for Computational Linguistics*, 05 2002.

[26] Julio Hoffimann, Youli Mao, Avinash Wesley, and Aimee Taylor. Sequence Mining and Pattern Analysis in Drilling Reports with Deep Natural Language Processing. volume Day 3 Wed, September 26, 2018 of *SPE Annual Technical Conference and Exhibition*, 09 2018. D031S033R004.

[27] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, page 50–57, New York, NY, USA, 1999. Association for Computing Machinery.

[28] Ah hwee Tan. Text mining: The state of the art and the challenges. In *In Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases*, pages 65–70, 1999.

[29] IBISWorld. Global fastest growing industries by revenue growth in 2021. `https://www.ibisworld.com/global/industry-trends/fastest-growing-industries/`.

[30] Paul Jaccard. The distribution of the flora in the alpine zone.1. *New Phytologist*, 11(2):37–50, 1912.

[31] Fanghong Jian, Jimmy Xiangji Huang, Jiashu Zhao, Tingting He, and Po Hu. A simple enhancement for ad-hoc information retrieval via topic modelling. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, page 733–736, New York, NY, USA, 2016. Association for Computing Machinery.

[32] Sergey Karpovich, Alexander Smirnov, Nick Teslya, and Andrei Grigorev. Topic model visualization with ipython. volume 776, pages 131–137, 04 2017.

[33] Timothy King. 80 Percent of Your Data Will Be Unstructured in Five Years, 05 2021.

[34] Varun Kumar, Alison Smith-Renner, Leah Findlater, Kevin D. Seppi, and Jordan L. Boyd-Graber. Why didn't you listen to me? comparing user control of human-in-the-loop topic models. *CoRR*, abs/1905.09864, 2019.

[35] Jey Lau, Karl Grieser, David Newman, and Timothy Baldwin. Automatic labelling of topic models. pages 1536–1545, 01 2011.

[36] Jey Han Lau and Timothy Baldwin. An empirical evaluation of doc2vec with practical insights into document embedding generation. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 78–86, Berlin, Germany, August 2016. Association for Computational Linguistics.

[37] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents, 2014.

[38] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach, 2019.

[39] Julie Beth Lovins. Development of a stemming algorithm. *Mech. Transl. Comput. Linguistics*, 11:22–31, 1968.

[40] Hongfang Lu, Lijun Guo, Mohammadamin Azimi, and Kun Huang. Oil and gas 4.0 era: A systematic review and outlook. *Computers in Industry*, 111:68–90, 2019.

[41] Hans Peter Luhn. Auto-encoding of documents for information retrieval systems. *Charles Babbage Institute Archives*, 1958.

[42] Edward Ma. NLP Augmentation. https://github.com/makcedward/nlpaug, 2019.

[43] Zheren Ma, Ali Karimi Vajargah, Hanna Lee, Rami Kansao, Hamed Darabi, and David Castineira. Applications of Machine Learning and Data Mining in SpeedWise® Drilling Analytics: A Case Study. volume Day 2 Tue, November 13, 2018 of *Abu Dhabi International Petroleum Exhibition and Conference*, 11 2018. D022S147R001.

[44] Leland McInnes and John Healy. Accelerated hierarchical density based clustering. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 33–42, 2017.

[45] Leland McInnes and John Healy. Accelerated hierarchical density based clustering. *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, Nov 2017.

[46] Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction, 2020.

[47] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3:861, 09 2018.

[48] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[49] David Mimno, Hanna M. Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. Optimizing semantic coherence in topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, page 262–272, USA, 2011. Association for Computational Linguistics.

[50] Thomas Müller, Ryan Cotterell, Alexander Fraser, and Hinrich Schütze. Joint lemmatization and morphological tagging with lemming. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2268–2274, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[51] David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic evaluation of topic coherence. HLT '10, page 100–108, USA, 2010. Association for Computational Linguistics.

[52] Christine Noshi. A Brief Survey of Text Mining Applications for the Oil and Gas Industry. volume Day 1 Tue, March 26, 2019 of *IPTC International Petroleum Technology Conference*, 03 2019. D011S010R001.

[53] The Canadian Press. Canada's unemployment rate reaches record 13.7%, 06 2020.

[54] Satyam Priyadarshy, Aimee Taylor, Ashwani Dev, Suresh Venugopal, and Geetha Gopakumar Nair. Framework for Prediction of NPT causes using Unstructured Reports. volume Day 4 Thu, May 04, 2017 of *OTC Offshore Technology Conference*, 05 2017. D041S046R006.

[55] Stephen Rassenfoss. Mining Daily Driller's Reports Looking for Telling Patterns. *Journal of Petroleum Technology*, 67(06):70–71, 06 2015.

[56] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.

[57] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[58] Stephen V. Rice, Frank R. Jenkins, and Thomas A. Nartker. The fourth annual test of OCR accuracy. Technical report, Technical Report 95, 1995.

[59] Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the space of topic coherence measures. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, page 399–408, New York, NY, USA, 2015. Association for Computing Machinery.

[60] Frank Rosner, Alexander Hinneburg, Michael Röder, Martin Nettling, and Andreas Both. Evaluating topic coherence measures, 2014.

[61] Mark Sanderson and W. Croft. The history of information retrieval research. *Proceedings of The IEEE - PIEEE*, 100:1444–1451, 05 2012.

[62] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, 2020.

[63] SethGrimes. Unstructured Data and the 80 Percent Rule, 08 2008.

[64] Nakatani Shuyo. Language detection library for java, 2010.

[65] F.E. Starratt, P.M. Gagnon, and D.G. Bruneau. Applying an information retrieval system to well data. *Journal of Petroleum Technology*, 19(08):1005–1010, 1967.

[66] J. Steele and N. Iliinsky. *Beautiful Visualization: Looking at Data through the Eyes of Experts*. Theory in practice series. O'Reilly Media, 2010.

[67] William Webber, Alistair Moffat, and Justin Zobel. A similarity measure for indefinite rankings. *ACM Trans. Inf. Syst.*, 28(4), November 2010.

[68] Wikipedia contributors. Siamese neural network — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Siamese_neural_network&oldid=1020522415`, 2021. [Online; accessed 26-November-2021].

[69] Wikipedia contributors. Tf–idf — Wikipedia, the free encyclopedia, 2021. [Online; accessed 12-October-2021].

[70] Wenkuang Wu, Xiaoguang Lu, Ben Cox, Guoqiang Li, Lihua Lin, and Qing Yang. Retrieving Information and Discovering Knowledge from Unstructured Data Using Big Data Mining Technique: Heavy Oil Fields Example. volume All Days of *IPTC International Petroleum Technology Conference*, 12 2014. IPTC-17805-MS.

[71] Fengwei Yang and Sai Gu. Industry 4.0, a revolution that requires technology and national strategies. *Complex & Intelligent Systems*, 7, 01 2021.

[72] Weiwei Yang, Jordan Boyd-Graber, and Philip Resnik. A multilingual topic model for learning weighted topic links across corpora with low comparability. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1243–1248, Hong Kong, China, November 2019. Association for Computational Linguistics.

[73] Xing Yi and James Allan. A comparative study of utilizing topic models for information retrieval. pages 29–41, 04 2009.

[74] Qing Zeng-Treitler, Doug Redd, Thomas Rindflesch, and Jonathan Nebeker. Synonym, topic model and predicate-based query expansion for retrieving clinical documents. *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, 2012:1050–9, 11 2012.

[75] Hongbao Zhang, Yijin Zeng, Hongzhi Bao, Lulu Liao, Jian Song, Zaifu Huang, Xinjin Chen, Zhifa Wang, Yang Xu, and Xin Jin. Drilling and Completion Anomaly Detection in Daily Reports by Deep Learning and Natural Language Processing Techniques. volume Day 2 Tue, July 21, 2020 of *SPE/AAPG/SEG Unconventional Resources Technology Conference*, 07 2020. D023S027R004.