Proceedings of Proceedings of the ASME 2021 40th International & Conference on Ocean, Offshore and Arctic Engineering OMAE2021 June 21-30, 2021, Virtual, Online

OMAE2021-62979

SEMANTIC SEARCH IN OFFSHORE ENGINEERING WITH LINGUISTICS AND NEURAL PROCESSING PIPELINES

Flavio Jaime Pol Gonçalves [1], Vinicius Cleves de Oliveira Carmo [1], Vinicius Toquetti de Melo [1], Rodrigo da Silva Cunha [1], Ismael H. F. Santos [2], Rodrigo Augusto Barreira [2], Carlos Eduardo Cugnasca [1], Fabio Gagliardi Cozman [1], Edson Satoshi Gomi [1]

[1] Universidade de São Paulo, São Paulo, Brazil
[2] Petrobras, Rio de Janeiro, Brazil

ABSTRACT

This paper presents a computing pipeline architecture for semantic search in the domain of Offshore Engineering. The proposed system combines modules such as document retriever, passage retriever, and answer extractor to produce textual responses to queries in natural language such as: "What FPSO motion is mostly affected by viscous damping?" Such responses are often needed in Offshore Engineering activities, and linguistic techniques such as those based on inverted indexes with a syntactic focus tend to perform poorly. Instead, this research explores semantic techniques that take into account the meaning of words in the domain of Offshore Engineering. This paper describes a Linguistic QA pipeline architecture built that provides a way to retrieve answers instantly from a collection of 13,000 unstructured technical documents about Offshore Engineering, reports the achieved results and future work. This paper also presents additional modules under construction that exploit Neural Networks and ontologies approaches for semantic search in the domain of Offshore Engineering.

NOMENCLATURE

QA Question Answering
SeSO Semantic Search on Offshore Engineering
POS Part-of-Speech tagging
TF-IDF Term Frequency–Inverse Document Frequency
ANN Artificial Neural Network
GPT-2 Generative Pre-trained Transformer 2
NLTK Natural Language Toolkit
MRR Mean Reciprocal Rank
BERT Bidirectional Encoder Representations from Transformers

INTRODUCTION

Offshore Engineering activities, like many other engineering endeavors, depends on technical data and information that is often buried in extensive collections of documents. Access to technical data and information is particularly critical in the Oil & Gas industry, where health, safety, and environmental risks associated with production and processing are a constant concern. Currently, one of the ways to predict events and the behavior of oil platforms is through the virtual representation of a physical object or process. These virtual representations are known as Digital Twins; they allow one to use data and predictive models to predict platform events and behavior. It is essential to give platform's operators ways to analyze unexpected events; in order to do that, operators have to be able to get answers from the collection of documents related to the offshore platform.

A platform's related documents usually consist of millions of files with unstructured text, tables, and figures. The problem is how to get answers from related information spread across multiple sources. It is common to run keyword-based queries over

Copyright © 2021 by ASME

unstructured data; popular tools, such as Solr or ElasticSearch,[1] operate with keywords and are often deployed to handle thousands or even millions of documents. These tools can profitably support Offshore Engineering, and indeed many existing information retrieval systems in the industry rely on keywords.

However, keywords are rather blunt when the user seeks a particular explanation or technical detail. It seems reasonable to allow users to express their queries in natural language and resort to retrieval techniques that consider the semantics (that is, the meaning) of words. In doing so, one might wish to handle questions that require some fine understanding of technical terms, such as

> How much is the storage capacity of the floating hydrocarbon storage facility?

This paper describes the steps we have pursued to build a semantic search engine customized for Offshore Engineering. The goal is to have a system that receives a query in natural language and finds the answer, if there is any, in a given stack of unstructured documents. There are several strategies that one can use in doing so. One is to explore a pipeline architecture where the incoming query is divided into a token that is further assigned meaning and then used to match the query to the available documents. Another strategy is to create a mapping from the query to the document, or the paragraph, that contains the answer; recent efforts have obtained surprising success by building mappings out of neural networks.

The research reported here aims at a system, called Semantic Search on Offshore Engineering, for short SeSO, where an incoming query is addressed by various techniques so that the final answer is an ensemble of possible solutions. This paper deals with one of these schemes, focusing on a pipeline architecture implemented and tested with technical documents on Offshore Engineering. The overall goal is to have a reliable searching engine that is well-tuned for Offshore Engineering design and operation of an offshore platform.

In the next section, the paper summarizes a few needed concepts and further comments on the broad strategies that can be applied to the search engine. It is followed by a description of the pipeline architecture as tested with a realistic document collection. Finally, the last section discusses several comments and directions for future work.

SEMANTIC SEARCH WITH NATURAL LANGUAGE QUERIES

The word "search" has a broad meaning: it can refer to the search for a specific item using a key, or to the search from information scattered in several documents. The latter task is usually labeled "information retrieval", broadly understood as finding material from within large collections of unstructured documents [1]. The last two decades have seen extraordinary gains in the performance of information retrieval systems, most of which are keyword-based: the user types in several words, and the retrieval system looks for documents that contain those words by resorting to indexes and scores.

This work focus on *semantic search* aimed at collections of technical papers and reports on the offshore domain. Broadly speaking, *semantic search* refers to processes that retrieve information from structured and unstructured documents by taking into account the *semantics* of the query and the document (that is, the meaning of linguistic of elements and their relationships). For instance, a useful technique with semantic content is *query expansion*, where synonyms and related words are added to a query. For example, a query asking for mooring line documents may lead to an expanded query that include berthing line or tie up words. A successful query expansion mechanism requires at least a previously built thesaurus. There are now rather large lexical datasets in the world-wide-web, such as the WordNet[2], that are often used for query expansion. However, a general thesaurus cannot provide the level of detail required by information retrieval in some specific domain, say within offshore platforms. Thus one must build a computational representation of the semantics of the particular domain of interest.

The semantic representation of a domain can be used beyond query expansion; it can rank documents when many documents are selected, prune wrong answers, or direct the search when many datasets are available. For instance, one may use an ontology(a set of concepts and categories in a domain that shows their properties and the relations between them) of terms in a particular domain when translating loosely formatted queries to a formal language such as SQL: then the meaning of words may be useful in determining the best translation [2]. A semantic understanding of queries helps the retrieval system in processing complex questions posed in natural language by mapping those questions to keywords. For instance, instead of just inserting keywords such as

> temperature, ocean, thermal,

one might ask directly

> "What is the minimum temperature difference required for ocean thermal energy conversion?"

Of course, the challenge is to convert such a question in natural language into a meaningful response.

As noted in the Introduction section, there are two broad strategies to do so. One of them springs mostly from linguistics, as it divides the query into tokens that are individually and collectively analyzed: they are divided into nouns and verbs, for exam-

[1] https://lucene.apache.org/solr/, https://www.elastic.co/elasticsearch/.

[2] http://wordnet.princeton.edu.

Copyright © 2021 by ASME

ple, and their connections with well-known acronyms and technical concepts are identified. This strategy relies on a pipeline architecture of modules that gradually build a representation of the query suitable for comparison with representations of the available documents. This sort of strategy is pursued in this paper; the next section presents it in more detail, and later the tests with the proposed implementation are summarized. The other strategy is to map the query to response using, for example, a neural network; the following section discusses the initial efforts in this direction right after it describes the pipeline architecture.

SEMANTIC SEARCH IN OFFSHORE ENGINEERING USING A LINGUISTIC QA PIPELINE ARCHITECTURE

This section presents the first implemented pipeline architecture within the Semantic Search on Offshore Engineering (SeSO) system. This prototype is based on a Linguistic QA pipeline architecture, which uses methods of information retrieval and additional modules for information recovery techniques to extract responses directly from over 13,000 offshore engineering text documents [3]. The objective of this section is to present the Linguistic QA pipeline architecture and, meanwhile, to explain the modules and techniques in use.

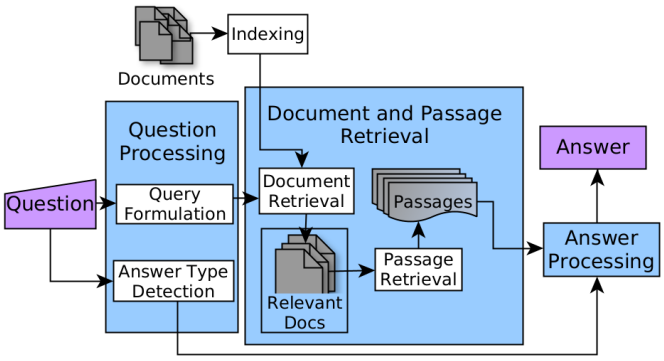

FIGURE 1: The three main stages of the Linguistic QA pipeline architecture: Question Processing, Passage Retrieval and Answer Processing. Image based on Ref. [4]

The Linguistic QA pipeline architecture consists of three main stages: Question Processing, Passage Retrieval, and Answer Processing. Figure 1 shows the three stages. Question Processing receives the question and generates a transformed query for the next stages; Passage Retrieval processes the offshore engineering text documents and retrieves the most relevant passages; Answer Processing extracts the answer. The Linguistic QA pipeline architecture answers factoid questions, this means that the answer is found within given text.

To test the Linguistic QA pipeline architecture we developed an evaluation dataset for the experiment consisting of 100 factoid questions in natural language. These questions were manually created by reading the documents and labeling the answer. This set of diversified questions that include abbreviations from the Offshore Engineering vocabulary, definition questions (What, Which, Why, and How), locations (Where), periods and dates (When), and individuals or organizations (Who). The list below depicts some questions from the evaluation dataset.

- What is the speed of the steel plate model for the ice test ?
- What FPSO motion is mostly affected by viscous damping ?
- Why is related to the gap resonances in the narrow gaps ?
- Who started using polyester ropes for mooring ?
- When did the 2015 COSL Innovator accident occurred ?
- Which are the sediments in the northern region of South China Sea ?
- How much is the storage capacity of floating hydrocarbon storage facility ?
- Where does have the same pore pressure ratio as South China Sea ?

Question Processing

The purpose of the Question Processing stage is to extract as much information from the question text as to feed the later stages. This stage receives the input question from the user in natural language, for example: "*What kind of model can detect mooring line failure ?*", then applies a technique called tokenization to split the question into individual pieces called tokens. The Penn Treebank project developed the tokenizer applied in this implementation (the Penn Treebank is a collection of documents with over 4.5 million American English words; it uses regular expressions to tokenize text present in Penn Treebank [5]). The result of tokenization to the question example is a list of tokens shown below.

```
['what', 'kind', 'of', 'models', 'can',
'detect', 'mooring', 'line', 'failure',
'?']
```

The next step in the Question Processing stage is to apply Part-of-Speech tagging (POS), or grammatical tagging, to mark up the tokens corresponding to a particular part of speech, for instance: nouns, verbs, adjectives, and adverbs [6]. The POS module applies statistical techniques to parse the text and tag the token. The result of POS tagging for the question example is shown below.

```
('What', 'WP'), ('kind', 'NN'), ('of',
'IN'),('model', 'NN'),('can', 'MD'),
('detect', 'VB'),('mooring', 'VBG'),
```


Copyright © 2021 by ASME

```
('line', 'NN'), ('failure','NN')]
```

The output shows each word and its tag encoded as a tuple. The reference tag in the example is defined by the Penn Treebank. The Penn Treebank initially used an automated POS assignment using a stochastic algorithm and after a manual correction of the tags by annotators. The first tuple `('WP')` refers to a wh-pronoun, the following `('NN')` to a noun, the `('IN')` to a preposition, the `('MD')` to a modal verb, the `('VB')` to a verb in its base form and the `('VBG')` to a verb in gerund form. This POS tag helps identify the keywords to formulate the search query and identify the answer type.

Another technique applied to process questions is Chunking. This technique identifies and classifies nouns and verb groups from unstructured text. Chunking is applied after POS tagging using the compound nouns and verbs found with the POS tagging output. Some approaches to Chunking use handwritten rules to parse, and others use a transformation-based learning mechanism to train a classifier (chunker) by using annotated data from a collection of documents as a training set. The result of Chunking are words grouped into chunks. For example, after applying Chunking, the nouns `[('line', 'NN'), ('failure','NN')]` are combine into a single noun group `('NN', 'line failure')`.

To process the search query it is essential to remove stopwords and apply Stemming. Stopwords removal cleans the search query by eliminating the most common words in a language. These words have little value and also increase processing in further stages. For this research it was implemented a customized list of stopwords for the Linguistic QA pipeline architecture. After removing stopwords, the question example becomes `kind models detect mooring line failure ?`. Stemming's main objective is to reduce inflectional forms and derivations related to a word in its base form. The implementation in this paper uses the Porter Stemming Algorithm [7]. For instance, the words *moors, mooring, moored* have as common base the word *moor*. In this implementation, Offshore Engineering documents do not need to use the words in their many variations because they have a similar meaning. The question example, after Stemming, is turned into `'kind', 'model', 'detect', 'moor', 'line', 'failure'`.

The Answer Type Detection module within the Question Processing stage determines the expected class of the answer. Such a class can be a person, organization, location, and date, for example. This module can be a set of handwritten linguistic rules and a classifier trained with machine learning techniques. This implementation uses handwritten rules to extract answer types such as dates, locations, person, quantity, and measurements for this implementation. Within the Question Processing stage, the Answer Type Detection module receives the question and outputs the answer type.

The Query Formulation module generates the search query to be used in the next Passage Retrieval stage to look for documents. This module applies a query expansion technique to select and add similar terms to the search query to improve document retrieval matching and retrieval performance [8]. The key idea behind query expansion is to reduce ambiguity by adding new meaningful terms. This idea enables the retrieval of more relevant documents.

Query expansion techniques can be classified as follows [9]:

- Manual Query Expansion: The user manually chooses the terms for query expansion;
- Automatic Query Expansion: The system automatically computes new terms without any user intervention;
- Interactive Query Expansion: The user and the system cooperate; the system returns new terms, and the user chooses the most meaningful term to add to the query.

One of the major benefits of query expansion is that it increases the chance to retrieve the most relevant documents. For the search query `'kind', 'model', 'detect', 'moor', 'line', 'failure'`, the output of the Query formulation module becomes:

```
['kind', 'type', 'sort', 'variety',
'form', 'models', 'manikin', 'poser',
'model', 'theoretical', 'framework',
'modeling', 'example', 'simulation',
'detect', 'observe', 'find', 'notice',
'discover', 'mooring', 'moor', 'up',
'moorage', 'mooring', 'berth', 'wharf',
'line', 'cable', 'production', 'failure']
```

Passage Retrieval

The Passage Retrieval stage goal is to select the sentences that contain the answer to the user's question. The first step is to build an index that contains the most relevant words for each document based on the Term Frequency–Inverse Document Frequency, for short TF-IDF.

The TF-IDF is a statistical measure that reflects the importance of a word in a document within a collection of documents, and it is used to give a higher weight to words that occur only in a few documents. The words limited to a few documents are useful for discriminating those documents from the rest of the collection; frequent terms across the entire collection are not helpful. The TF-IDF increases proportionally to the number of times a word appears in the document. This metrics is affected by the number of documents in the corpus (a collection of texts) that contain the word. TF-IDF (1) is the product of two expressions, Term Frequency (2) and Inverse Document Frequency (3):

$$tfidf_{(t,d,D)} = tf_{(t,d)} \cdot idf_{(t,D)}, \tag{1}$$

Copyright © 2021 by ASME

For the Term Frequency, in Expression (2), $tf_{(t,d)}$ is the raw count of the number of times that term t occurs in document d. A normalization is discussed by [3] to mitigate the anomaly observed in higher term frequencies in longer documents.

$$tf_{(t,d)} = 0.5 + 0.5 \cdot \frac{f_{t,d}}{\max\{f_{t',d} : t' \in d\}}. \quad (2)$$

The Inverse Document Frequency, in Expression (3), is a measure of how much information a word provides across all documents, where N is the total number of documents in the collection and the denominator is number of documents where the term appears.

$$idf_{t,D} = \log \frac{N}{|\{d \in D : t \in d\}|}. \quad (3)$$

The indexing of documents is a pre-stage of the Linguistic QA pipeline architecture, and it consists of creating indexing for the collection of Offshore Engineering documents by using the TF-IDF value of each word and organizing the index for its frequency in each document and paragraph. To run the tests, we selected 40 documents within a collection of 13,000 technical documents related to Offshore Engineering. This collection of 40 documents contain an average of 58 paragraphs per document.

In this implementation, the Passage Retrieval stage uses the search query to rank the three most relevant documents, and within these relevant documents, it ranks the five most relevant paragraphs. From the five most relevant paragraphs, the Passage Retrieval stage also ranks the five most relevant sentences. This ranking is achieved by calculating the cosine similarity between all the question terms and most relevant sentences. For the question example "*What kind of model can detect mooring line failure ?*", the five most relevant sentences are:

1. Two types of Neural Networks models can be developed;
2. The cases of each mooring Several methods have been investigated for detecting mooring line failure without knowing the environmental conditions;
3. The trained Artificial Neural Network (ANN) model can detect mooring line failure, even for vessel draft (mass), sea states, and environmental directions that are not included in the training data;
4. The ANN model's output indicates the status condition for the mooring lines (intact or failed);
5. The ANN model representation, in terms of its input variables, is based on assessing when changes in a platform's motion characteristics are in-fact indicators of a mooring line failure.

Answer Processing

In this implementation, the Answer Processing stage uses pattern extraction and N-gram tiling techniques to rank the candidate answers and extract the final string from the highest-ranked candidate answer that may contain the correct answer for the user's question. The pattern extraction technique uses information about the Question Processing stage's class answer type, along with regular expression patterns. These patterns expressions are in conjunction with information based on the number of matched question keywords, keywords distance, and sequences of question terms, for example. This implementation uses handwritten rules that combine these pattern expressions. The N-gram tiling technique compares the terms in the question with the most relevant sentences. It uses term-filtering to rank how similar the sentence is for the potential answer. The output is a string from the most relevant sentences containing the correct answer for the user's question. In the question example "*What kind of model can detect mooring line failure ?*" the final answer provided by the Linguistic QA pipeline architecture is "*ANN model*".

A NEURAL QA PIPELINE ARCHITECTURE

As noted in the Introduction, the Linguistic QA pipeline architecture described in the previous section is not the only natural language processing approach in semantic search. Another possibility is to resort to direct mappings from query to answer, usually learned from sets of queries/answers. Recent literature has reported excellent performance of neural networks in encoding such mappings; hence we are considering and implementing a neural-based QA pipeline architecture presented in this section.

The objective of the Neural QA System is to learn to answer a question from examples. It consists of two main stages: Passage Retrieval and Machine Reading. Passage Retrieval recovers a set of passages from the corpus (a collection of texts) that are expected to contain the answer, while Machine Reading takes this set of passages and pin-points the answer. In both stages, words from questions and document passages are represented using word embeddings. These are dense vector representations that map words in a semantic space. In the Passage Retrieval stage, question and passage are independently encoded by neural networks. A vector similarity score then measures the relation between passage and question. This decoupling of question and passage encodings allows pre-encoding all passages in execution time and enables scaling the retrieval to a large number of documents. In the Machine Reading stage, the question and the relevant passages are encoded together, one passage at a time. This encoding allows a better interaction between questions and passages. The result is a score for the passage's ranking containing the answer and a score for each span within the passage that is a candidate to be an answer.

Learning to answer questions from data is a data-hungry task. Large datasets with more than 100k examples are available

Copyright © 2021 by ASME

for some domains, like SQuAD[3], for Wikipedia, and NewsQA[4], for news articles. For most specific domains, though, large datasets for QA are not available and are expensive to create. On the other hand, the neural model suffers from a loss of accuracy when applied on distributions (in this case domains) different from the one it was trained for. This motivates using the existing resources on other domains and applying domain adaptation techniques for adapting to the Offshore Engineering domain. The proposed Neural-based QA pipeline architecture is trained on Wikipedia, and the research focus on automatically generated questions to transfer this model to offshore engineering domain. The question generation pipeline architecture is composed of three stages:

1. Answer Extraction, where segments from sentences that define an information need are identified;
2. Question Generation, where questions are generated given a passage and an answer segment within it, and;
3. Validation, where a trained QA model answers the generated question.

If it finds the expected answer, the triple (a passage, question, answer) is approved; otherwise, it is discarded. Answer extraction and Validation stages are implemented using BERT models [10], while Question Generation is implemented using a GPT-2 model [11]. All stages are trained using large existing datasets for QA. Once trained, they are applied on the offshore documents, generating questions to perform domain adaptation on the Neural QA System.

AN ONTOLOGY FOR OFFSHORE ENGINEERING

In this section the paper describes briefly the current work on building an ontology for Offshore Engineering. Such an ontology can expand queries and generate testing examples as described in the Introduction.

An ontology can be defined as an explicit specification of a domain. In other words, an ontology is a set of concepts, objects, processes, and other entities that are presumed to exist in some area of interest, as well as the relationships that occur between them. Building an ontology is time-consuming and costly since much of the process depends on the intervention of an expert. That is why the research focus on automatic and semi-automatic construction methods, using artificial intelligence tools, which require minimal human intervention. Figure 2 shows a very simplified part of the ontology, with a few entities and relations.

For the initial approach to implementing a semiautomatic method of building ontologies, we used the methodology described in [12], developing all the passages using the Python programming language and its libraries, according to the following steps:

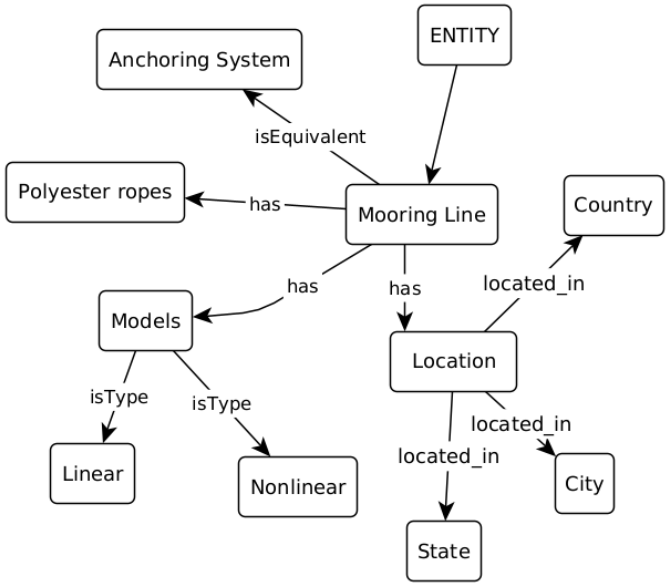

FIGURE 2: Partial view of a simplified ontology for Offshore Engineering

Step 1: Build corpora of texts that are representative of the domain whose ontology to be built.

Step 2: Use Natural Language Processing techniques to identify several linguistic components of interest in the corpora. This model was made using code developed with the Python language and the spaCy natural language processing library.

Step 3: Identify possible candidates for concepts in the corpora. To perform this step, an algorithm was built using Python and the NLTK library. This algorithm can be modified to search for prominent terms in the corpora; we use the most basic strategy, which consists of searching for the terms most frequently present in the corpora. We intend to use other strategies such as comparing the relative frequency of terms in other corpora, but this strategy is already sufficient for a first analysis.

Step 4: Identify in the corpora relationships between concepts. In this step, an algorithm was created to extract the relationships between the concepts found in step 3. It is still necessary to refine this algorithm so that the generated relationships are relevant and express the domain concepts assertively, but it is already possible to identify some relations of concepts. Table 1 shows some relations extracted for the concepts **Fatigue** and **Mooring**.

[3]https://rajpurkar.github.io/SQuAD-explorer/
[4]https://www.microsoft.com/en-us/research/project/newsqa-dataset/

Copyright © 2021 by ASME

TABLE 1: Extracted relationships for "Fatigue" and "Mooring".

Subject	Relation	Object
distribution	driving	fatigue life consumption
stresses	decrease	fatigue life
cyclic load range	affects	fatigue life
Torng	determined	total fatigue life
faults	increases	fatigue damage
rupture	decreases	fatigue resistance
mooring system	includes	mooring lines
supporting systems	includes	catenary mooring lines
internal turret	supports	3x3 mooring system
use	improve	mooring costs

Step 5: Introduce the user to the concepts and relationships discovered for their verification and adjustment. In this step, it is necessary to introduce the concepts and relations extracted for some specialists in the domain area, in this case, offshore engineering. The concepts and relationships extracted in steps 3 and 4 are still very simple and we can improve the quality of these terms before presenting them to a specialist. We are still organizing a way for experts to evaluate and approve the terms extracted.

Step 6: Represent these concepts and relationships in the chosen formal language. We are currently working on the development of an algorithm that converts the concepts and relationships extracted in steps 3 and 4 to formal representation. The language chosen for this formal representation is the Web Ontology Language (OWL).

This work intends to use ontology to improve the semantic search systems presented in the previous sections. The ontology can be used in the query formulation, expanding the list of keywords in the question, using synonyms or other ways of asking the same question. It is also possible to use ontology to translate necessary information to a standard database query language Structured Query Language (SQL) as described in [13]. Ontology helps in the data augmentation process, and it can expand the number of questions to train the QA systems.

An ontology can also help recover the relevant passages, incorporating the semantic meaning of the entities and improving the understanding of acronyms and abbreviations.

Finally, in the processing of the answer, it can increase the ability to distinguish which is the best answer to the question asked, using the entities' relationships and properties.

RESULTS

The typical evaluation metric for factoid QA systems is Mean Reciprocal Rank, or MRR [14]. The MRR metric assumes a rank list of answers or passages containing the correct answer. To evaluate a system returning a set of ranked answers for a test set consisting of N questions is

$$MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank_i}, \tag{4}$$

where *rank* is the position of the first correct answer in the list of all answers for the *i*th question generated by the Linguistic QA pipeline architecture.

For example, if the correct answer is located at the fourth position in the rank list, it receives a reciprocal rank score of 1/4. For the Linguistic QA pipeline architecture we considered a rank list of five candidate answers. The Linguistic QA pipeline architecture achieved 0.63 MRR for the 100 questions from the evaluation dataset. Table 2 shows the result for 100 questions, with 77 correct answers found, where 55 are the first candidate answer, and 22 varies from second to the fifth candidate answer. For 23 questions the Linguistic QA pipeline architecture did find the correct answer. Table 3 shows random questions from the evaluation dataset and their respective answer retrieved by the Linguistic QA pipeline architecture in the second column. The third column shows the text's paragraph from the document and the highlighted sentence where the answer was extracted.

TABLE 2: SeSO results for 100 factoid questions.

Item	Number of Questions
Correct 1^{st}	55
Correct 2^{nd} to 5^{th}	22
Incorrect	23

Copyright © 2021 by ASME

TABLE 3: Random questions from the evaluation dataset and their respective answer provided by the SeSO prototype with the Linguistic QA pipeline architecture.

Question	Answer from SeSO Classic QA system	Text Paragraph
What is the speed of the steel plate model for the ice test ?	1m/s	Sensitivity analysis on meshes is usually significant for numerical simulation. However, due to the constraint of large number of meshes in ship-ice and ship-ice-water model, it is very hard to decrease the mesh sizes. In order to simplify the calculation, we choose a similar ice-structure collision case to check the mesh sensitivity on ice loads. This simple model simulates the process of a steel plate extruding an ice cylinder with a hemisphere head. In the model, the steel plate is set as a rigid body and its material is same with the ship in section 2, while the material of the ice and its failure mode are the same with the ice in section 2. **The speed of the steel plate is 1m/s and the ice cylinder is fixed at its bottom.**
What FPSO motion is mostly affected by viscous damping ?	the roll motion	A ship-shaped or barge-shaped floating production platform such as FPSO and floating liquefied natural gas (FLNG) is one of the most common floaters for offshore field development due to large deck space and storage capacity. Given the hull form and displacement of these floaters, they may be subject to relatively large motions resulting from environmental actions. Since the motion responses of floaters directly affect other structure and equipment designs such as topsides and risers, the accurate and reliable prediction of motions are one of key design tasks. **Compared to other motions, the roll motion is greatly governed by the viscous damping in addition to wave radiation damping.**
Why is related to the gap resonances in the narrow gaps ?	the overestimation on the free surface elevation in the gaps	The numerical simulation also shows the accurate prediction of the transverse motions such as sway and roll. These transverse motions are mainly generated by hydrodynamic interactions and influenced by the gap resonances between the two floating hydrocarbon storage tanks (FHSTs). A noticeable peak can be found on the numerical results in sway at around 5.5s. **This peak is due to the overestimation on the free surface elevation in the gaps when the resonance occurs.** Similarly, the response amplitude operators (RAOs) in roll also show peaks at the same period. The RAOs in yaw show differences between numerical and experimental results in short waves and in long waves. In short waves, the yaw RAOs are influenced by the unrealistic gap resonances in the numerical model. In longer waves, the difference is possibly due to the difference of the inertia of the FHST. It will not cause problem as the yaw motion is quite small.
Who started using polyester ropes for mooring ?	Petrobras	**Polyester ropes were first introduced for offshore mooring applications in the mid 1990ies, piloted by Petrobras.** Nowadays polyester ropes are used worldwide, particularly for deep-water applications where catenary systems become heavy and inefficient. We also see many shallow-water applications where the compliance of the fibre ropes is required.
When did the 2015 COSL Innovator accident occurred ?	on December 30, 2015	The accurate prediction of the wave impact load on an offshore platform under a harsh environment is one of the most important issues because it is directly related with the survivability of the structure. Recently, horizontal wave impact load on semi-submersible and Tension Leg Platform (TLP) structure is a significant concern due to the accident of COSL Innovator in the North Sea. **On December 30, 2015, an accident occurred at the COSL Innovator, a drilling unit which had operated in the North Sea, due to extreme wave impact.** The freak wave struck the deck side and living quarter, causing damage to the side walls and injury of crew members. After this, Det Norske Veritas DNV-GL has proposed guidelines for air-gap prediction and horizontal wave impact loads evaluation for column stabilized units. Due to these circumstances, horizontal wave impact loads on column stabilized units becomes an important concern
Which are the sediments in the northern region of South China Sea ?	the fine-grained and coarse- grained	**Moreover, there are two different deposit modes for the sediment in the northern region of South China Sea, which are the fine-grained and coarse-grained govern deposit modes in deep and shallow water respectively.** It is found that the sleeve friction ratio of the cohesive sediment is very low in shallow water. The normalized values of the clay soil in deep water are consistent with those from the Gulf of Mexico and West Africa.
How much is the storage capacity of floating hydrocarbon storage facility ?	12500m3	The simplified system contains two FHSTs of smaller size and a rigid barge frame. **The storage capacity of the FHST is 12500m3.** The two FHSTs are connected to the surrounding barge frame through 8 flexible ropes. In the entire system, the barges are interconnected through flexible connectors. In the present study, the barge frame is considered as one rigid body. The interconnection loads between barges are measured in model test Phase-2. The moment of inertia is determined with reference to a coordinate system with the same axis definition as explained in the model test setup but with the origin at the COG. The barge frame is assumed to be free floating vertically. No motion is allowed in the surge, sway, and yaw in both numerical and experimental study. The horizontal motion of the barge frame is restrained by mooring dolphins as in the entire system. In the prototype, an articulated bridge is located in the barge center to protect the FHSTs. In the simplified system, this bridge has been moved to above of the free surface.
Where does have the same pore pressure ratio as South China Sea ?	Gulf of Mexico and West Africa	The paper researched the fundamental engineering characteristics of cohesive sediments in the northern region of South China Sea. The low plasticity index and the low clay particle content of the cohesive sediments in the northern shallow water region of South China Sea are prominent physical properties, which are obviously different from those of the clay soils in the northern deep water region of South China Sea, the Gulf of Mexico and West Africa. The different clay particle content of cohesive sediments is considered as the main reason of the significant difference of physical indexes of cohesive sediments in the shallow and deep water. There are two different deposit modes for the sediments of the northern region of South China Sea, which are the fine-grained and coarse-grained govern deposit modes in deep and shallow water respectively. The cone resistance ratio and excess pore pressure ratio calculated based on the Cone Penetration Test (CPT) data are little different by the comparison of cohesive sediments from the northern shallow and deep water region of South China Sea. But the sleeve friction ratio of the clay soil in shallow water is 1/3 or less of that in deep water. **The cone resistance ratio, pore pressure ratio and sleeve friction ratio for cohesive sediment from the deep water region of South China Sea are similar to those from the Gulf of Mexico and West Africa.** In the paper, tests data of the clay soil with greater than 1000m are analyzed from the only one oil and gas project. The authors are integrating and analyzing more tests data for the oil and gas projects in the northern region of South China Sea, which can be used to enrich and verify these conclusions.

Copyright © 2021 by ASME

CONCLUSION AND FUTURE WORK

The SeSO's prototype for the Linguistic QA pipeline architecture has achieved a good performance considering the nature of unstructured text from technical documents in offshore engineering. Since a big part of the Offshore Engineering scientific and technical knowledge is in textual form, the proposed Linguistic QA pipeline architecture is essential by providing a quick solution for Offshore Engineers to retrieve answers instantly from a collection of unstructured text documents and hence allowing quick decision making in the case of anticipated or unexpected events.

The Linguistic QA pipeline architecture struggles to find answers for questions containing acronyms, initialism, and abbreviations standard in offshore engineering documents, such as TESK, CDF, JPD, QUAD, VLFS. A future upgrade solution is to feed the model with a knowledge base of all the acronyms, initialism, and abbreviations within the Offshore Engineering industry. Another obstacle for the Linguistic QA pipeline architecture is related to documents containing paragraphs with line breaking or hyphenation. This obstacle occurs when it is preferable to break the word and continue on the next line instead of moving the entire word to the next line. The indexing of these documents containing the line breaking fails on these words and affect the answer's quality. A possible solution is to concatenate all the words containing hyphens within the QA pipeline architecture processing stages.

The implementation of an Interactive Query Expansion component to the Linguistic QA pipeline architecture is also under evaluation. It will improve document retrieval matching and retrieval performance by reducing the ambiguity and adding new meaningful terms to the search query. In the interactive technique, the user and the system cooperate to enhance the search query to return new and meaningful terms.

Future work within the SeSO Linguistic QA pipeline architecture includes:

1. The implementation of Interactive Query Expansion to include synonyms words from offshore engineering;
2. A knowledge-base of all the acronyms, initialism, and abbreviations;
3. The use an ontology for the offshore engineering domain, and;
4. The implementation of answer extraction from tables and graphs within the text documents.

ACKNOWLEDGMENT

This work was financed by ANP/PETROBRAS, Brazil (project 21721-6). Fabio G. Cozman is partially supported by the Conselho Nacional de Desenvolvimento Cientifico e Tecnológico (CNPq), grant 312180/2018-7, and acknowledges partial support from FAPESP grant 2019/07665-4.

REFERENCES

[1] Manning, C. D., Raghavan, P., and Schutze, H., 2009. *Introduction to Information Retrieval*. Cambridge University Press.

[2] Sen, J., Lei, C., Quamar, A., Özcan, F., Efthymiou, V., Dalmia, A., Stager, G., Mittal, A., Saha, D., and Sankaranarayanan, K., 2020. "Athena++: Natural language querying for complex nested sql queries". In Proceedings of the VLDB Endowement, VLDB Endowment, p. 2747–2759.

[3] Manning, C., Prabhakar, R., and Hinrich, S., 2008. "Introduction to information retrieval". *An Introduction To Information Retrieval,* ***151***(177), p. 5.

[4] Jurafsky, D., and Martin, J. H., 2009. Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition.

[5] Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B., 1993. "Building a large annotated corpus of english: The Penn treebank". *Comput. Linguist.,* ***19***(2), June, p. 313–330.

[6] Charniak, E., 1997. "Statistical techniques for natural language parsing". *AI Magazine,* ***18***(4), Dec., p. 33.

[7] Van Rijsbergen, C. J., Robertson, S. E., and Porter, M. F., 1980. *New Models in Probabilistic Information Retrieval*, Vol. 5587. British Library Research and Development Department London.

[8] Vechtomova, O., 2009. *Query Expansion for Information Retrieval*. Springer US, Boston, MA, pp. 2254–2257.

[9] Azad, H., and Deepak, A., 2017. "Query expansion techniques for information retrieval: a survey".

[10] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K., 2019. "Bert: Pre-training of deep bidirectional transformers for language understanding". In NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, Vol. 1, pp. 4171–4186.

[11] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I., 2019. "Language models are unsupervised multitask learners". *OpenAI blog,* ***1***(8), p. 9.

[12] Carvalheira, L. C. d. C., 2007. "Método semi-automático de construção de ontologias parciais de domínio com base em textos.". Mestrado em Sistemas Digitais, Universidade de São Paulo, São Paulo, Aug.

[13] Xiao, G., Calvanese, D., Kontchakov, R., Lembo, D., Poggi, A., Rosati, R., and Zakharyaschev, M., 2018. "Ontology-Based Data Access: A Survey". International Joint Conferences on Artificial Intelligence Organization, pp. 5511–5519.

[14] Soares, M. A. C., and Parreiras, F. S., 2020. "A literature review on question answering techniques, paradigms and systems". *Journal of King Saud University - Computer and Information Sciences,* ***32***(6), pp. 635–646.

Copyright © 2021 by ASME