SphereOI Studios
Sphere of Influence, Inc.
1420 Spring Hill Rd., Suite 525
McLean, VA 22102
Tel: 703-584-5405

Phase I Trade Study Report
"Use Machine Learning to Prioritize Inspections"

SphereOI.AI

An AI engineering company
When human intelligence isn't enough

Order Number: 31310023P0005

CAGE: 338K9
UEI: LDQAKL4UDN27

Sam.gov: Registered and up to date
Business Type: Small business
Period contract Type: Firm Fixed Price

Sphere of Influence, Inc.
1420 Spring Hill Rd., Suite 525
Tysons Corner, Virginia 22102
www.SphereOI.ai

Contracts: Sanda Klacar, Asst. Treasurer
SKlacar@sphereoi.com
703-548-5406 T
703-842-8479 F

Program Manager: Scott Pringle, EVP
SPringle@sphereoi.com
301-919-9393 T

We are pleased to present this trade study report as the Phase I deliverable for the "Use Machine Learning to Prioritize Inspections" effort.

Sphere of Influence (SphereOI) creates intelligent systems that engage AI/ML to raise the bar for efficiency, speed, and sustainability. In our digital factory, our proprietary SpeedShift design-engineering cadence accelerates tool selection and product development, turning years into months by delivering custom, defining technologies on small budgets. Our past performance covers a wide array of innovations that span precision agriculture, missile defense, and medical and consumer products.

Sanda Klacar, Assistant Treasurer
Sphere of Influence, Inc.

April 5, 2023
Date

SphereOI Studios
Sphere of Influence, Inc.
1420 Spring Hill Rd., Suite 525
McLean, VA 22102
Tel: 703-584-5405

Contents

SphereOI Studios
Sphere of Influence, Inc.
1420 Spring Hill Rd., Suite 525
McLean, VA 22102
Tel: 703-584-5405

Use Machine Learning to Prioritize Inspections

A. Executive Summary

The Azure cloud environment was selected and will be used to compare two different approaches, Latent Dirichlet Allocation (LDA) and Neural Topic Modeling (NTM). The LDA approach will leverage the Azure visual programming environment and the NTM approach will use BERTopic in a Jupyter notebook using an on-prem desktop environment. The Jupyter Notebook can be easily moved to an Azure compute instance for future use. These two approaches will be tuned and tested in Phase II. The most promising approach will be used to complete the Proof of Concept (PoC) for the final delivery.

B. Task 1: Machine Learning Tool Evaluation

Most of the data associated with this effort is in the form of textual reports. These reports are provided in both a PDF format and an extracted CSV format with primarily text-based fields. Given this, a few approaches are under consideration for the study. These approaches include Latent Dirichlet Topic Modeling and Neural Topic Modeling. In addition to common tasks like ingestion and visualization, the ML toolsets will be evaluated for support for these modeling approaches. Each ML toolset will provide their own version of topic modeling, and its performance against the target dataset will be considered. Note that the major cloud provided toolsets (Azure, AWS, and Google) as well as Matlab, all support the use of AI/ML notebooks (like Jupyter Notebook). These notebooks allow most Python libraries to be incorporated into a solution. Support for the notebook approach is also considered in the evaluation.

We will examine needs beyond Proof of Concepts (PoCs) to include fieldability and human factors. It is important to consider the programming languages and tools already in use as well as the types of problems being solved. In environments where data scientists and ML engineers are doing most of the work, the choice of toolset tends to enable more control over nuances. In environments where civilian data scientists, or scientists and engineers from other disciplines, are doing most of the work, then the choice of toolset tends to emphasize a simplified user experience. For the NRC, we will examine technical and ethnographic priorities to ensure a proper fit.

B.1 Tool Evaluation Objective

Evaluate the suitability of commercially available ML systems for the NRC. The specific use case of applying unsupervised clustering to the historical archive of safety inspection reports will be used to assess the relative merits of each platform.

SphereOI Studios
Sphere of Influence, Inc.
1420 Spring Hill Rd., Suite 525
McLean, VA 22102
Tel: 703-584-5405

Given the starting list of systems to evaluate (AWS Sagemaker, Azure ML, Google AI, and Matlab), we know from experience that all are suitable for the task. The clustering algorithms in each of these platforms have been used successfully by our engineers and scientists in the past. Assessing the nuances of this dataset and the goals for this study will allow us to differentiate between the algorithms offered by the various environments.

Numerous tools and methods are available for unsupervised machine learning and clustering. Choosing the toolset, architecture, and environment that will yield the best fit within the NRC for this problem will be at the center of the trade study.

B.2 Clustering

Given the dataset for this study, two primary clustering methods have been identified – Topic Modeling and Neural Topic Modeling.

B.3 Trade Study Method

In order to assess the 4 toolset candidates, we have identified the two clustering techniques mentioned above, Topic Modeling and Neural Topic Modeling. For Topic Modeling the industry standard is Latent Dirichlet Allocation (LDA). For Neural Topic Modeling several competing approaches exist. Bidirectional Encoder Representations from Transformers (BERT) Topic has been selected based on its performance and availability.

B.3.1 Describe the problem

Using information provided in text-based inspection reports, define safety clusters and associate reactors with those clusters. Safety clusters should be discovered using unsupervised machine learning approaches.

B.3.2 Search literature

The toolset candidates suggested in the RFP are sufficient to determine the capabilities currently offered in the industry. Azure ML, AWS Sagemaker, Google AI, and Matlab will be the candidates assessed in the study.

For the unsupervised machine learning, two approaches have been identified: Latent Dirichlet Allocation (LDA) and Neural Topic Modeling. These approaches are most effective when dealing with text-based data and unsupervised topic clustering. Based on the literature, the candidates support these modeling approaches and the trade study will provide side by side comparisons of the support provided.

B.3.3 Select candidates

Azure ML, AWS Sagemaker, Google AI, and Matlab will be the candidates assessed in the study.

SphereOI Studios
Sphere of Influence, Inc.
1420 Spring Hill Rd., Suite 525
McLean, VA 22102
Tel: 703-584-5405

B.3.4 Select evaluation factors

Model Support:

- LDA Topic Modeling – ability to support this modeling approach
- Neural Topic Modeling – ability to support this modeling approach
- Other relevant approaches for future use/experiments with text data - Text classification, Summarization, key phrase extraction, Named entity recognition and linking, Question / Answering

Processing Support:

- Notebook Integration - ability to launch and scale a python notebook on the platform.
- Text Extraction - ability to extract text from PDFs.
- Text Pre-processing - ability to clean text data before passing it to a machine learning algorithm remove punctuation, emails, URLs, specific string patterns, correct spelling errors filter stop words, stemming & lemmatization, extract linguistic features.
- Text embedding - ability to represent a text document as a vector of numbers using various embedding techniques and pre-trained models.
- Visual Programming – support for no-code and low code approaches including drag and drop interfaces.

Python Library Support

With Notebook Integration, these libraries are all supported. Without Notebook Integration, the use of Python is severely limited and visibility into the algorithms used is also limited.

- Pandas
- Numpy
- BeautifulSoup
- NLTK
- Spacy
- Gensim
- pyLDAvis
- Matplotlib
- Bertopic
- hdbscan
- scikit-learn
- scipy
- huggingface transformers
- Pytorch
- sentence-transformers

B.3.5 Develop evaluation factor weighting

Using pairwise comparisons, and eliciting feedback from the NRC, the following weights were developed:

SphereOI Studios
Sphere of Influence, Inc.
1420 Spring Hill Rd., Suite 525
McLean, VA 22102
Tel: 703-584-5405

Criteria	Weight
Neural Topic Modeling	.27
LDA	.17
Visual Programming	.14
Text Pre-processing	.11
Text Embedding	.10
Other Text Approaches	.09
Notebook Integration	.08
Text Extraction	.03

B.3.6 Define evaluation factor ranges

For these evaluation factors, a scale of 1 to 10 with 10 being the best score was selected. Candidates were evaluated on the ease of use, availability, and flexibility with respect to each category.

B.3.7 Cost

Cost was not considered as an evaluation criterion. In our experience, costs across cloud providers are similar. While the pricing models vary, the ultimate cost is roughly the same. The volume of data and compute resources for this study are relatively small. In fact, we were able to perform the studies using desktop machines. Cost for this type of effort would be on the order of $1000 per month while it is in use and could be shut down between studies. Below is some price analysis to assist for future planning.

The experiments for the NRC project have been on a desktop machine and similar compute resource on Azure should suffice. For reference, the machine has 8 cores and 64 GB memory, and the NVIDIA GeForce RTX 3070 Ti GPU with 6144 cuda cores and 8GB memory.

For LDA in the visual no-code Azure Machine Learning Studio, an ML studio workspace ($10/month) and then Studio usage ($1/hour). A general-purpose CPU compute instance (~$0.76/hour for D16 v3 with 16 vCPU and 64GB) for running LDA will be needed. Azure ML is using the scikit learn implementation of LDA which may not be as fast as the Gensim parallelized LDA implementation used in local experiments (~10 mins/experiment). The expected run time should be in the order of minutes for each experiment for the shorter text strings, maybe hours if we use full documents. We can start with one of the cheaper options and see how the runtime is for the shorter text.

Aure ML Studio and Notebooks quick start:

SphereOI Studios
Sphere of Influence, Inc.
1420 Spring Hill Rd., Suite 525
McLean, VA 22102
Tel: 703-584-5405

https://learn.microsoft.com/en-us/azure/machine-learning/quickstart-create-resources
https://learn.microsoft.com/en-us/azure/machine-learning/quickstart-run-notebooks

Azure ML LDA reference:
https://learn.microsoft.com/en-us/azure/machine-learning/component-reference/latent-dirichlet-allocation
https://scikit-learn.org/stable/modules/decomposition.html#latentdirichletallocation

For neural topic modeling in a notebook, the pricing will depend on the results of the experiments we do over the next few weeks and what the vision is for the future (result replication will require ~2 hours of compute, model updates with new data overtime will require ~2 days of compute, or continued experiments for model improvement will require ~2 weeks of compute).

So far, we have only been using pre-trained models in BERTopic for embedding the text and for representing the topics, and GPUs will do these steps in seconds as opposed to 10-15 minutes on the CPU. If future needs require finetuning any models with the NRC text or explore other options (like the CTM library) that must train a neural network for topic modeling (unlike BERTopic) then training time and memory requirements based on the language model's size & token limits must be considered. These advanced approaches are not anticipated. For Azure GPU we recommend starting with the NCas_T4_v3 Series ($0.752/hour for NC8as T4 v3 with 8 vCPUs, 56GB RAM and 1 NVIDIA Tesla T4 GPU)

Azure CPU, GPU Pricing:
https://azure.microsoft.com/en-us/pricing/details/machine-learning/

Azure ML compute options and standard recommendations:
https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/ready/azure-best-practices/optimize-ai-machine-learning-cost
https://learn.microsoft.com/en-us/azure/machine-learning/concept-compute-target#supported-vm-series-and-sizes

Various GPUs and their typical use-case:
https://learn.microsoft.com/en-us/azure/virtual-machines/sizes-gpu

Recommended for AI/Deep Learning:
https://learn.microsoft.com/en-us/azure/virtual-machines/nct4-v3-series
https://learn.microsoft.com/en-us/azure/virtual-machines/ncv3-series

Azure ML studio and the CPU and GPU compute resources mentioned above in their pricing calculator here with a day of compute time:
https://azure.microsoft.com/en-us/pricing/calculator/?&ef_id=EAIaIQobChMIhO-3h7mQ_gIVM8qUCR0nHg7xEAAYASAAEgIZ0_D_BwE:G:s&OCID=AIDcmm5edswduu_

SphereOI Studios
Sphere of Influence, Inc.
1420 Spring Hill Rd., Suite 525
McLean, VA 22102
Tel: 703-584-5405

SEM_EAIaIQobChMIhO-3h7mQ_gIVM8qUCR0nHg7xEAAYASAAEgIZ0_D_BwE:G:s&gclid=EAIaIQobChMIhO-3h7mQ_gIVM8qUCR0nHg7xEAAYASAAEgIZ0_D_BwE

C. Perform Assessments

C.1 Azure

- Many AI/ML Products: Applied AI Services, Cognitive Services, Form Recognizer, Cognitive Search, OpenAI Services, Machine Learning and more.
- Various environments supported: visual no-code (Azure ML Designer), code-first (Jupyter Notebooks, Python SDK, CLI).
- Azure Machine Learning
  - Text extraction: supported through Form Recognizer if needed, else import data from an Azure Datastore and transform via Designer.
  - Text pre-processing: remove stop words, regular expressions for string matching, lemmatization, case normalization, remove special characters, patterns, emails, or URLs.
  - Text embedding: N-gram features, Word2Vec, FastText, GLoVe
  - Topic modeling: LDA.
  - Other relevant offerings: text classification and named entity recognition via Designer; many NLP tasks (key phrase extraction, entity recognition and linking, summarization, question-answering) supported via Cognitive Services.
- Basic NLP functionalities and LDA topic modeling with a visual no-code environment or notebook approach to explore more advanced methodologies.

C.2 AWS

- Many AI/ML Products: Comprehend, Textract, Augmented AI and more.
- Various SageMaker Environments: visual no-code (Canvas), code-first (Studio, Notebook Instances, Studio Lab).
- AWS SageMaker Built-in functionalities
  - Text extraction: supported through Amazon Comprehend if needed, but data can be provided in various formats (text, csv, json) for use within SageMaker.
  - Text pre-processing: not built-in but can be imported from libraries as needed.
  - Text embedding: BlazingText (for learning CBOW, skip-gram or batch skip-gram embeddings with Word2Vec; learning character n-gram embeddings), Object2Vec (for learning embeddings with sentence pairs).
  - Topic modeling: LDA and Neural Topic Modeling.

SphereOI Studios
Sphere of Influence, Inc.
1420 Spring Hill Rd., Suite 525
McLean, VA 22102
Tel: 703-584-5405

Other relevant offerings: text classification, summarization, entity recognition and relationship extraction, question-answering.

- Some advanced NLP functionalities, LDA and neural topic modeling with a visual no-code environment or notebook approach to explore more advanced methodologies.

C.3 Matlab

- Simple function calls and interactive notebook like environment.
- Ability to call python libraries from MATLAB environment and import/export deep learning frameworks with Open Neural Network Exchange (ONNX) format.
- Text Analytics Toolbox
  - Text extraction: supports extractions from various formats (text, PDF, HTML, CSV, Excel, and Word).
  - Text pre-processing: remove punctuation, URL, correct spelling errors, filter stopwords, stemming & lemmatization, extract linguistic features.
  - Text embedding: word and n-gram counting, word2vec, CBOW, FastText, GloVe.
  - Topic modeling: LDA.
  - Other relevant offerings: document summarization, text classification, and keyword extraction with limited deep learning models.
- Basic NLP functionalities offered by MATLAB, but more advanced methods will likely be needed to obtain actionable results from technical text data.
- Any code written will be specific to MATLAB and not easily portable to other platforms or a python notebook.

C.4 Google

- Many AI/ML Products: Vertex AI, Natural Language AI, Document AI, Contact Center AI.
- Code-first (Jupyter Notebooks, Python SDK, CLI).
- Google AI Built-in functionalities
  - Text extraction: Collect structured data from unstructured text data.
  - Text pre-processing: not built-in, but can be imported from libraries as needed.
  - Text embedding: Supervised learning tasks for image, tabular, text, and video– not clear embedding for neural analysis.
  - Topic modeling: Only for customer / agent conversation analysis.
  - Other relevant offerings: Sentiment analysis, Entity analysis, Entity Sentiment analysis, Syntactic analysis, Content classification.
- No out of the box support for unsupervised topic modeling.

D. Report Results

SphereOI Studios
Sphere of Influence, Inc.
1420 Spring Hill Rd., Suite 525
McLean, VA 22102
Tel: 703-584-5405

Criteria	Weight	Azure	AWS	Matlab	Google
Neural Topic Modeling	0.27	0	4	0	0
LDA	0.17	7	7	7	0
Visual Programming	0.14	7	6	0	0
Text Pre-processing	0.11	8	0	8	7
Text Embedding	0.10	9	9	9	0
Other Relevant Approaches	0.09	9	7	6	3
Notebook Integration	0.08	10	10	8	10
Text Extraction	0.03	7	7	7	7
Weighted Score		**84**	83	64	30

E. Conclusions

The Azure and AWS environments both support the modeling needs for this study. The relative scores do not differentiate between these two technologies. However, the NRC is familiar with the Azure environment, giving it an advantage.

Algorithms and models supplied by the environments are limited and somewhat inflexible. Python libraries used in a notebook provide many advantages. A comparison table is provided below.

Platform Supplied Algorithms	Python Library Algorithms
Pros • Integrated into user experience. • Can be combined into standard pipelines. • Work well in common cases (social media).	Pros • Flexibility to select from multiple algorithms. • Flexibility to select from multiple pre-trainings. • Advanced models available. • Flexibility to address highly technical domains. • Flexibility to leverage internal parameters. • Work well in Machine Learning Notebooks. • Can be deployed in any cloud or on premises. • Limited coding knowledge required.
Cons • Do not handle technical domains well. • Black box algorithms (unpublished models). • Difficult to tailor pipelines. • Limited selection of algorithms. • Limited selection of pre-training datasets. • Advanced algorithms are not available. • Strengths in one area (Topic Modeling) may not translate to other areas (Classification, clustering, regression, anomaly detection).	Cons • Requires some knowledge of Python. • Algorithms must be researched and selected.

SphereOI Studios
Sphere of Influence, Inc.
1420 Spring Hill Rd., Suite 525
McLean, VA 22102
Tel: 703-584-5405

Topic modeling in highly technical domains requires extensive tuning and analysis. Technical terms and domain specific language, including the names of specific devices, hardware, processes, and technical discussions, are not handled well by pre-trained machine learning models that were trained on generic text from literature. Re-training the models is expensive and requires massive amounts of data. To address this, alternate modeling methods can be used, in this case, Neural Topic Modeling (NTM) performs better than “bag of words” type models, including Latent Dirichlet Allocation (LDA), which is the most common model used in Topic Modeling. NTM can give better results than the 'bag of words' approaches because the embeddings will retain the semantic meaning of the text & the context of the words that is lost when we view a document as a bag-of-words.

Rapid advances in many areas of machine learning provides access to numerous algorithms and methods, but only when using a notebook style approach – the platforms and environments take time to incorporate the newest models, and often do not publish the underlying models making it difficult to assess the quality of the model provided.

F. Recommendation

In early results using operational data, Neural Topic Modeling seems to out-perform LDA Topic modeling when using titles and item introductions text from Inspection Reports. In particular, the default BERTopic model configuration finds interesting and cohesive topics. The BERTopic library itself provides significant modularity to swap and tune every stage of the topic modeling pipeline and offers many variants of topic modeling that can be implemented with minimal code.

For the contemplated study, we recommend using BERTopic, and plan to make a side-by-side comparison to Azure’s no-code LDA offering. This will allow the relative merits of the algorithms to be compared in the context of complexity of implementation. Using BERTopic will allow us to vary the representation of the textual data and the discovered topics, as well as explore unsupervised and semi-supervised topic modeling variants to find meaningful Safety Clusters that resonate well with the Subject Matter Experts.

The Azure no-code experiments will be performed using Azure cloud resources. The BERTopic modeling will be performed using a notebook approach with SphereOI on-prem resources. The notebook created for the experiments can be easily loaded into cloud instances for future runs.

G. SOW Tasks

Below is the list of subtasks associated with Phase 1 of the effort. All subtasks have been completed.

G.1 Trade Study – Phase I

SphereOI Studios
Sphere of Influence, Inc.
1420 Spring Hill Rd., Suite 525
McLean, VA 22102
Tel: 703-584-5405

Schedule / Duration:

Start: 3/6/2023
End: 4/9/2023
Duration: 34 calendar days

Subtasks:

- ✓ Weekly progress demonstrations
- ✓ Describe the problem
- ✓ Search the literature
- ✓ Select candidates
- ✓ Select evaluation factors
- ✓ Develop evaluation factor weights
- ✓ Define evaluation factor ranges
- ✓ Perform assessments
- ✓ Report Results
- ✓ Deliver Trade Study Report

G.2 SOW Deliverables

Phase I required Deliverable: Complete

1. Trade Study Report – all evaluation factors with weights and ranges, all candidate scores, candidate selection, and summary of findings.

Phase II Required Deliverable:

2. Study design – describe the study including key measurements.

Phase III Required Deliverable:

3. Project Report – including considerations and recommendations for the future.
    a. Proof-of-Concept (PoC) including ETL, feature extraction, and model code – software, models, and parameters to recreate the results.
    b. Project Documentation – includes weekly demonstrations presentations, progress reports, and solution descriptions.
    c. Study Report and Evaluation – provide the study report including roadblocks, complications, and considerations.

H. Glossary and Acronym List

Item	Description or Definition
AI	Artificial Intelligence
AWS	Amazon Web Services
BERT	Bidirectional Encoder Representations from Transformers
CBOW	Continuous Bag of Words
CLI	Command Line Interface
CPU	Central Processing Unit

SphereOI Studios
Sphere of Influence, Inc.
1420 Spring Hill Rd., Suite 525
McLean, VA 22102
Tel: 703-584-5405

Item	Description or Definition
CSV	Comma separated values
ETL	Extract, Transform, and Load
GPU	Graphical Processing Unit
LDA	Latent Dirichlet Allocation
ML	Machine Learning
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
NRC	Nuclear Regulatory Commission
NTM	Neural Topic Modeling
PDF	Portable Document Format
PoC	Proof of Concept
PWS	Performance Work Statement
RFP	Request for Proposal
SDK	Software Developer Toolkit
SME	Subject Matter Expert
SOW	Statement of Work