PROJECT REPORT

NLP on ARGO Metadata Using LLMs

ABSTRACT

The objective of this project is to explore how modern Large Language Models (LLMs) can be applied to scientific metadata extracted from ARGO NetCDF oceanographic files. Although ARGO datasets primarily contain numerical measurements of temperature, salinity, and pressure, each NetCDF file also includes descriptive metadata documenting calibration procedures, data quality assessments, sensor performance, and scientific observations. These textual components form a rich but underutilized source of information that can be processed using NLP techniques.

To investigate this potential, a complete NLP pipeline was developed that extracts metadata from NetCDF files, normalizes and preprocesses text, and applies LLMs—including GPT and LLaMA—to perform five high-value tasks: Question Answering, Topic Analysis, Document Classification, Information Extraction, and Chatbot development. GPT was used for reasoning and generative tasks, while LLaMA was fine-tuned for classification and structured prediction. The models were evaluated using BLEU, ROUGE-L, Accuracy, F1-score, and Perplexity metrics.

Experimental results show that GPT provides strong contextual understanding for scientific question answering and conversational interaction, whereas LLaMA performs effectively in classification and targeted extraction tasks. The study demonstrates that LLMs can significantly enhance the interpretation of oceanographic metadata, reduce manual workload, and support automated scientific analysis within the ARGO data ecosystem.

INTRODUCTION

Natural Language Processing (NLP) has become one of the most advanced and rapidly evolving branches of artificial intelligence, enabling machines to understand, interpret, and generate human language. Early NLP techniques relied heavily on rule-based systems and statistical models, which were limited in their ability to capture semantic meaning, long-range dependencies, and domain-specific terminology. With the emergence of deep learning and, more significantly, the Transformer architecture, NLP experienced a paradigm shift. Transformers introduced the concept of self-attention, enabling models to learn contextual relationships across entire sequences. This breakthrough led to the development of Large Language Models (LLMs) such as GPT (Generative Pretrained Transformer) and LLaMA (Large Language Model Meta AI), which have demonstrated exceptional performance in reasoning, question answering, summarization, classification, and various text-processing tasks.

While LLMs have been widely used in general-purpose applications such as chatbots, sentiment analysis, and translation, their potential in scientific domains remains underexplored. One such domain is oceanography, where large volumes of environmental data are collected through autonomous systems such as ARGO floats. The ARGO program maintains a global network of

profiling floats that measure temperature, salinity, pressure, depth, and ocean dynamics. These measurements are stored in NetCDF (Network Common Data Form) files—a standardized scientific data format that combines multidimensional numerical arrays with descriptive textual metadata. Although researchers primarily focus on numerical observations, the metadata included within NetCDF files contain crucial scientific insights such as calibration comments, quality-control notes, instrument diagnostics, and historical processing steps.

Interpreting these metadata fields manually is time-consuming and often requires domain expertise, making NLP a promising tool for automating the analysis of scientific text. However, ARGO metadata present several challenges: the text varies in length and structure, includes domain-specific terminology, incorporates scientific units such as °C, psu, and dbar, and often contains incomplete or inconsistent descriptions. Traditional NLP methods are not well suited to handle these complexities. Modern LLMs, with their ability to understand contextual meaning and adapt to specialized vocabulary, offer a more effective approach.

LITERATURE REVIEW

Natural Language Processing (NLP) has evolved dramatically over the last five decades, progressing from early symbolic systems to large-scale neural architectures capable of modeling complex linguistic structures. Understanding this evolution is essential for positioning the use of Large Language Models (LLMs) in scientific domains such as oceanography, where textual metadata embedded in NetCDF files remains an underused resource.

2.1 Early Approaches to NLP

The earliest NLP systems were entirely rule-based, relying on hand-crafted grammars, linguistic rules, and symbolic reasoning. Classic systems like ELIZA and SHRDLU demonstrated that machines could simulate basic language interaction, but their inability to generalize beyond predefined patterns severely limited their usefulness. These systems struggled with ambiguity, domain-specific vocabulary, and unstructured scientific descriptions common in environmental datasets.

Statistical NLP marked the next major shift. Models such as Hidden Markov Models (HMMs), Naïve Bayes classifiers, and n-gram language models improved text processing by learning probability distributions from data. However, they still required extensive manual feature engineering and could not capture long-range dependencies or contextual relationships. For scientific metadata—often containing technical terms, units, abbreviations, and irregular phrasing—such models were insufficient.

2.2 Machine Learning and Deep Learning in NLP

The transition to machine learning introduced algorithms like Support Vector Machines (SVM), Random Forests, and Logistic Regression for classification tasks. These methods improved automation but continued to rely heavily on manually engineered features such as TF-IDF and Bag-of-Words (BoW) vectors. They lacked the ability to interpret the semantics of scientific text and were sensitive to vocabulary variations.

Deep learning brought significant progress, especially with the use of Recurrent Neural Networks (RNNs), Long Short-Term Memory networks (LSTMs), and Gated Recurrent Units (GRUs). These models could capture sequential patterns and were widely applied in machine translation, speech

recognition, and document classification. Nevertheless, they exhibited limitations in handling long sequences and struggled with computational inefficiency.

For ARGO metadata—which often includes multi-sentence calibration notes, historical descriptions, and processing steps—models like LSTMs are inadequate because they fail to capture global context and relationships spanning multiple lines of description.

2.3 The Transformer Breakthrough

The introduction of the Transformer architecture by Vaswani et al. in 2017 revolutionized NLP. Transformers eliminated the need for sequential processing and instead used self-attention mechanisms that compute relationships between all tokens simultaneously. This innovation addressed the shortcomings of RNN-based models and led to:

- Better handling of long-range dependencies
- Faster parallel processing
- Improved semantic understanding
- Scalability to billions of parameters

Transformers became the foundation for modern LLMs and made it feasible to apply NLP techniques to scientific text, where context-specific understanding is essential.

2.4 Emergence of Large Language Models (LLMs)

LLMs such as GPT, BERT, T5, and LLaMA represent the most advanced stage in NLP evolution. These models are pretrained on large corpora containing diverse linguistic patterns, allowing them to generalize to a wide variety of tasks with minimal or zero fine-tuning.

GPT (Generative Pretrained Transformer)

GPT models excel in tasks involving reasoning, summarization, generative text completion, and question answering. Their autoregressive design makes them effective in interpreting calibration comments, quality control notes, and scientific descriptions present in ARGO metadata.

LLaMA (Large Language Model Meta AI)

LLaMA models focus on efficient architecture and fine-tuning capabilities. They are particularly suitable for:

- Document classification
- Scientific information extraction
- Domain adaptation on limited hardware

For metadata categorization in ARGO NetCDF files, LLaMA provides a balanced combination of accuracy and computational efficiency.

2.5 NLP in Scientific and Environmental Domains

Recent research shows growing interest in applying NLP to scientific datasets. Examples include:

- Extraction of biomedical information from clinical reports
- Summarization of climate change assessments
- Mining of geological hazard descriptions
- Automated interpretation of satellite and weather datasets

In oceanography, most existing studies focus on numerical modeling, climate simulations, and time-series analysis. Very few works explore **textual components** of oceanographic datasets. Although ARGO is widely used in ocean research, its metadata fields (calibration notes, QC history, parameter descriptions) remain largely unexamined from an NLP perspective.

This gap highlights the significance of applying LLMs to ARGO metadata.

2.6 NLP for NetCDF Metadata

NetCDF datasets are structured and self-describing, containing both numerical arrays and metadata attributes. While numerical values are typically processed through statistical or machine-learning models, textual metadata are rarely analyzed even though they provide:

- Context for measurements
- Calibration formulas and comments
- Correction history
- Sensor performance details
- Institutional processing steps

Traditional NLP tools do not handle NetCDF metadata well because:

- Text fields vary in length and format
- Metadata contain scientific units (°C, dbar, psu)
- Technical abbreviations require contextual understanding

LLMs, however, thrive in such environments due to their ability to interpret domain-specific text, infer relationships, and generalize patterns.

DATASET DESCRIPTION

The ARGO program is a global ocean-observing network consisting of thousands of autonomous profiling floats deployed across all major ocean basins. Each float repeatedly descends through the water column, measures essential physical properties, and surfaces to transmit data via satellite. ARGO plays a central role in modern oceanography by providing open-access, high-resolution measurements of temperature, salinity, and pressure, which are crucial for climate monitoring, ocean circulation studies, and marine ecosystem analysis.

While ARGO is well known for its numerical measurement arrays, each ARGO data file also contains rich textual metadata describing the scientific and operational context of the collected data. These metadata fields are typically overlooked in machine-learning applications, yet they carry essential information about calibration procedures, data quality, sensor behavior, and processing history. This

project specifically focuses on these textual components, extracted from ARGO's NetCDF (Network Common Data Form) files.

3.1 Understanding the NetCDF Format

NetCDF is a standardized, self-describing, machine-independent data format widely used in the Earth and ocean sciences. A NetCDF file organizes data in a hierarchical structure consisting of:

- **Dimensions** (e.g., number of profiles, levels, cycles)
- **Variables** (multidimensional arrays of scientific measurements)
- **Global Attributes** (dataset-level information)
- **Variable Attributes** (metadata such as units, long_name, QC comments)

For ARGO, NetCDF files typically follow the Argo Global Data Assembly Center (GDAC) conventions, ensuring uniformity across international data providers.

Each file may contain dozens of numerical variables and dozens of textual metadata attributes. While numerical variables were not directly used in this NLP project, their descriptions and metadata form the foundation for the text corpus.

3.2 Types of Data Contained in ARGO NetCDF Files

A. Numerical Scientific Measurements (Multidimensional Arrays)

These include parameters such as:

- **TEMP** (Temperature in °C)
- **PSAL** (Practical Salinity in psu)
- **PRES** (Pressure/Depth in dbar)
- **CNDC** (Conductivity)
- **DOXY** (Dissolved Oxygen, if available)
- **LATITUDE & LONGITUDE** of float positions
- **CYCLE_NUMBER** indicating dive sequence

Although numerical values are not processed directly in NLP tasks, they help contextualize the metadata.

B. Textual Metadata (Primary Input for NLP)

The main focus of this project is the **metadata fields**, which contain scientific descriptions, calibration notes, and processing history. Important metadata attributes include:

1. SCIENTIFIC_CALIB_COMMENT

Detailed comments from scientists describing calibration results, sensor stability, or corrections applied.
Example:
"No salinity drift observed; correction not required for cycles 10–30."

2. SCIENTIFIC_CALIB_EQUATION

Mathematical formulas used for sensor calibration.

3. HISTORY_INSTITUTION

Institutions involved in data processing (e.g., IFREMER, JAMSTEC, NOAA).

4. HISTORY_ACTION

Processing steps performed on the data:
"QC performed using automated climatology check."

5. PARAMETER & PARAMETER_DESCRIPTION

Human-readable descriptions of scientific variables, such as:
"PSAL: Practical Salinity measured using conductivity sensor."

6. DATA_MODE

Indicates real-time, delayed-time, or adjusted mode, often accompanied by text explanations.

7. QC Flags and Notes

Short comments about data quality, anomalies, or corrections.

These textual components are invaluable for QA, topic extraction, classification, and chatbot responses.

3.3 Characteristics of the Extracted Text Corpus

The textual metadata extracted for NLP analysis has several unique characteristics:

1. Scientific Vocabulary

Includes terms like:

- thermocline
- mixed layer depth
- salinity anomaly
- drift correction
- reference profile
- stability check

2. Presence of Units & Symbols

e.g., °C, psu, dbar, m, ±, →
These units needed careful handling during preprocessing.

3. Variation in Writing Style

Metadata originate from different institutions worldwide, which leads to differences in tone, length, and formatting.

4. Semi-Structured Text

Some fields contain complete sentences, others contain abbreviated notes such as:
"adj qc applied," "sensor stable," "cycle OK."

5. Inconsistency Across Floats

Not all fields are present in every file; some floats include extensive metadata, while others contain minimal descriptions.

3.4 Extraction of Metadata for NLP Tasks

A Python-based extraction pipeline was developed using:

- **netCDF4**
- **xarray**
- **NumPy**
- **Standard Python text-processing libraries**

Steps Followed:

1. **Open each NetCDF file** and load global and variable attributes.
2. **Extract all textual fields** (byte arrays, char arrays, and string attributes).
3. **Decode and flatten** multi-dimensional text fields into readable strings.
4. **Concatenate related text** into unified metadata paragraphs.
5. **Remove control characters**, null bytes (\x00), and unnecessary formatting.
6. **Store the resulting text** as input for NLP tasks such as question answering, classification, and topic extraction.

3.5 Why ARGO Metadata Are Suitable for NLP

Although small in volume compared to numerical data, ARGO metadata are scientifically rich and contain:

- Expert annotations
- Calibration analyses
- Instrument diagnostics
- Quality control decisions
- Processing histories
- Notes on environmental patterns

These descriptions provide context not available in raw measurements. Using LLMs to interpret such metadata allows:

- Automated scientific insight extraction
- Faster understanding of float behavior
- Categorization of scientific events
- Building intelligent tools like ARGO Chatbots

Thus, the ARGO NetCDF dataset is an excellent real-world example for applying NLP and LLMs in environmental research.

PREPROCESSING PIPELINE

The preprocessing stage is one of the most critical components of this project, as the ARGO NetCDF metadata contains a mixture of scientific terminology, calibration notes, abbreviations, special characters, and non-standard formatting. Unlike conventional text datasets, ARGO metadata originates from multiple institutions, varies in structure, and is often stored as byte arrays or multi-dimensional character matrices inside NetCDF files. To prepare this text for downstream NLP tasks using GPT and LLaMA, an extensive preprocessing pipeline was designed.

The goal of preprocessing is to extract textual metadata from NetCDF files, transform it into a clean and consistent format, and prepare structured datasets suitable for tasks such as question answering, topic extraction, classification, and information extraction.

4.1 Extraction of Textual Metadata from NetCDF Files

The first step involved reading and extracting metadata fields from NetCDF (.nc) files. Using Python libraries such as **netCDF4** and **xarray**, the following types of metadata were extracted:

- SCIENTIFIC_CALIB_COMMENT
- SCIENTIFIC_CALIB_EQUATION
- HISTORY_INSTITUTION
- HISTORY_ACTION
- PARAMETER and PARAMETER_DESCRIPTION
- DATA_MODE
- QC_COMMENT and QC_FLAGS

Many of these fields are stored as byte arrays or multi-dimensional character matrices, requiring decoding and flattening.

Steps followed during extraction:

1. Open each .nc file using Dataset() or xarray.open_dataset().
2. Identify all textual fields from variable attributes and global attributes.
3. Convert byte arrays to UTF-8 strings.

4. Flatten multi-dimensional text arrays into readable lines.
5. Combine related metadata fields into a single descriptive text block per float profile.

This produced the raw text corpus used for further processing.

4.2 Text Cleaning and Normalization

Raw metadata often contain noise due to NetCDF formatting. Normalization was performed to ensure uniformity and compatibility with LLM tokenizers.

The following steps were applied:

a) Lowercasing

All text was converted to lowercase to reduce vocabulary size and standardize analysis.

b) Removing control characters

Characters such as \x00, newline padding, and null bytes were removed.

c) Trimming excessive whitespace

Multiple spaces, tab characters, and unnecessary indentation were normalized to single spaces.

d) Removing HTML-like artifacts

Some metadata contain encoded characters or formatting symbols which were cleaned.

e) Scientific unit normalization

To avoid tokenization inconsistencies:

- “deg C”, “°C”, and “celsius” were standardized to **°C**
- Depth units like “db”, “decibar”, and “dbar” were normalized to **dbar**
- Salinity references were standardized to **psu**

This ensures that models recognize scientific measurements consistently.

4.3 Handling Scientific Vocabulary and Keywords

ARGO metadata includes specialized terms such as:

- thermocline
- mixed layer depth
- reference calibration
- pressure drift
- salinity anomaly

To preserve domain meaning:

Selective Stop-word Removal

Generic English stop-words (e.g., *and, the, is, of*) were removed,
but scientific keywords were NEVER removed.

A custom lexicon of oceanographic terms was preserved:

- float
- cycle
- drift
- calibration
- salinity
- temperature
- profile
- pressure

This prevented the loss of important scientific context.

4.4 Tokenization

Tokenization splits the text into model-compatible units.
Since GPT and LLaMA use different tokenizers, separate tokenization pipelines were used.

GPT Tokenization

GPT uses **Byte Pair Encoding (BPE)**, which handles:

- special characters
- scientific units
- abbreviations

GPT tokenization allowed complex scientific terms to be broken into meaningful subwords.

LLaMA Tokenization

LLaMA uses **SentencePiece**, which produces tokens that are optimized for:

- multilingual text
- out-of-vocabulary scientific words
- compact representations

Both tokenizers ensured compatibility with their respective model architectures.

4.5 Sentence Segmentation

Metadata fields often contain long descriptions without proper punctuation.
A segmentation algorithm was used to break text into logical units based on:

- punctuation inference

- newline patterns
- calibration step boundaries
- institution log markers (e.g., "— cycle 10 —")

This allowed more effective topic extraction and QA.

4.6 Dataset Structuring for NLP Tasks

Once cleaned and tokenized, the text was structured into task-specific formats:

1. Question Answering Dataset

Pairs of:

- **Context** = extracted metadata
- **Question** = manually or automatically generated
- **Answer** = derived from metadata

2. Topic Extraction Text Blocks

Grouped metadata paragraphs by float cycles or calibration notes.

3. Classification Dataset

Labeled into categories:

- Temperature-focused
- Salinity-focused
- Depth-analysis
- Multi-parameter

4. Information Extraction Templates

Patterns were created for extracting:

- Temperature values
- Salinity values
- Depth levels
- Float IDs
- Locations

5. Chatbot Knowledge Base

Concatenated metadata summaries were used as knowledge passages for GPT.

This structuring enabled smooth downstream model usage.

METHODOLOGY

The methodology defines the complete workflow used to transform ARGO NetCDF metadata into usable text, prepare it for NLP tasks, and apply Large Language Models (LLMs) for multi-task analysis. This study integrates GPT and LLaMA into a unified system capable of performing Question Answering, Topic Analysis, Document Classification, Information Extraction, and conversational Chatbot interaction. The methodology is divided into several stages: data extraction, preprocessing, dataset construction, model selection, prompt engineering, fine-tuning, and multi-task implementation.

5.1 Overview of the Methodological Framework

The overall methodology follows a structured, multi-step pipeline:

1. **Extract metadata** from ARGO NetCDF files
2. **Preprocess and normalize text**
3. **Construct datasets** for five NLP tasks
4. **Apply GPT for reasoning-based tasks**
5. **Fine-tune LLaMA for classification tasks**
6. **Evaluate using multiple NLP metrics**
7. **Integrate outputs into an ARGO scientific assistant**

A visual representation of the workflow:

NetCDF files → Metadata Extraction → Preprocessing → Task Datasets

→ GPT (QA, Topics, Chatbot) + LLaMA (Classification, Extraction)

→ Evaluation → Final Multi-Task NLP System

This hybrid workflow ensures that both generative and structured prediction tasks are handled effectively.

5.2 Task Definitions and Objectives

The project focuses on **five major NLP tasks**, chosen to maximize scientific interpretability of ARGO metadata.

1. Question Answering (QA)

Objective: Allow users to ask scientific questions about ARGO float behavior, calibration notes, or measurement anomalies.
Example: *"Was salinity correction applied in cycle 25?"*

2. Topic Analysis

Objective: Automatically extract the main scientific themes from metadata, such as "thermocline deepening", "sensor drift", or "salinity anomaly".

3. Document Classification

Objective: Classify metadata entries into meaningful categories:

- Temperature-focused
- Salinity-focused
- Depth-analysis
- Multi-parameter descriptions

4. Information Extraction

Objective: Identify and extract structured information such as:

- Temperature values
- Salinity readings
- Depth levels (dbar)
- Float IDs
- Locations (latitude/longitude)

5. ARGO Chatbot

Objective: Build an interactive assistant capable of explaining scientific metadata in natural language.

5.3 Model Selection Strategy

Two models were chosen based on their strengths:

GPT (Generative Pretrained Transformer)

Used for:

- Question answering
- Topic extraction
- Chatbot interaction
- Generative reasoning tasks

Reasons for selection:

- Strong ability to infer context
- Excellent generalization
- Ability to produce human-like explanations
- No fine-tuning needed for QA and conversational tasks

LLaMA (Large Language Model Meta AI)

Used for:

- Scientific document classification
- Structured information extraction

- Domain adaptation

Reasons for selection:

- Efficient fine-tuning capability
- Strong performance in classification tasks
- Lower computational requirements

Both models complement each other—GPT handles deep reasoning, while LLaMA excels in direct classification and extraction.

5.4 Dataset Construction for Multi-Task NLP

Metadata extracted from NetCDF was organized into multiple task-specific datasets.

A) QA Dataset

Each instance includes:

- **Context:** Metadata text
- **Question:** Domain-specific question
- **Answer:** Extracted or generated

B) Topic Extraction Dataset

Metadata paragraphs grouped by float cycle or parameter type.

C) Classification Dataset

Each metadata text block was labeled into predefined categories.

D) Extraction Dataset

Patterns were created to detect:

- Temperature (“°C”, “temp”)
- Salinity (“psu”, “salinity”)
- Depth (“dbar”)
- Float IDs (numeric codes)

E) Chatbot Knowledge Base

Large chunks of metadata were compiled into a searchable and explainable reference set.

5.5 Prompt Engineering for GPT

Prompt engineering plays a crucial role in controlling GPT’s reasoning.

Techniques Used:

- **Zero-shot prompting:** Asking questions directly without examples
- **Few-shot prompting:** Providing 2–3 examples to guide model behavior

- **Chain-of-thought prompting:** Encouraging step-by-step reasoning
- **Instruction-based prompts:** Explicit commands such as
  "Extract only scientific information from the following text."

Example Prompt Format:

You are an oceanographic expert. Read the metadata below and answer the question.

Metadata:

"Temperature stable at 1000 dbar. Calibration applied according to NOAA standards."

Question:

"Was temperature calibration performed?"

Answer:

This improved accuracy and reduced hallucinations.

5.6 Fine-Tuning LLaMA for Classification

Unlike GPT, LLaMA was fine-tuned for classification and extraction tasks.

Fine-tuning configuration:

- **Epochs:** 3
- **Batch Size:** 8
- **Learning Rate:** 2e-5
- **Optimizer:** AdamW
- **Loss Function:** Cross-entropy
- **Training Hardware:** GPU-enabled system

The model was trained using metadata blocks paired with classification labels.
Fine-tuning allowed LLaMA to adapt to oceanographic terminology and metadata structure.

5.7 Implementation of the Multi-Task LLM System

A. Question Answering with GPT

GPT answered domain-specific questions using context passages from metadata.
Example Output:
"Yes, salinity calibration was applied and no drift was detected."

B. Topic Analysis with GPT

GPT summarized metadata into themes such as:

- “upper ocean warming”
- “salinity drift correction”
- “float stability issue”

C. Document Classification with LLaMA

LLaMA assigned metadata to scientific categories with high accuracy.

D. Information Extraction with GPT + LLaMA

A hybrid pipeline was used:

- LLaMA → detects presence of temperature, salinity, depth
- GPT → extracts precise values using reasoning

E. ARGO Chatbot

GPT was configured as a domain-aware assistant capable of answering follow-up questions and explaining metadata in simple language.

5.8 Evaluation Strategy

To assess model performance, multiple metrics were used:

- **Accuracy** for classification
- **F1-score** for extraction
- **BLEU** for QA correctness
- **ROUGE-L** for topic extraction
- **Perplexity** for LLaMA text fluency

This multi-metric approach ensured robust evaluation across tasks.

EVALUATION METRICS

Evaluating the performance of NLP models requires metrics that accurately reflect how well the system handles classification, extraction, generation, and reasoning. Because this project involves **five different NLP tasks**—Question Answering, Topic Analysis, Document Classification, Information Extraction, and Chatbot Interaction—multiple evaluation metrics were used to capture different dimensions of model performance. Each metric was selected based on its suitability for the output type and the nature of the ARGO metadata.

The evaluation metrics used in this study include **Accuracy**, **F1-Score**, **BLEU**, **ROUGE-L**, and **Perplexity**. Together, these metrics provide a comprehensive assessment of the effectiveness, reliability, and linguistic quality of GPT and LLaMA across multiple tasks.

7.1 Accuracy

Accuracy is one of the most widely used metrics for classification tasks.
In this project, it was used to evaluate the **Document Classification** task performed by the LLaMA model.

Formula

$$Accuracy = \frac{Correct\ Predictions}{Total\ Predictions}$$

Relevance to ARGO Metadata

Document Classification in ARGO involves assigning metadata segments to categories such as:

- Temperature-related metadata
- Salinity-related metadata
- Depth-related metadata
- Multi-parameter descriptions

A high accuracy score indicates that the model understands the domain-specific vocabulary and correctly identifies the scientific focus of metadata fields.

7.2 F1-Score

The **F1-score** combines **precision** and **recall** into a single metric and is especially useful for evaluating Information Extraction tasks where class imbalance is common.

Formula

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Usage in This Project

The F1-score was used to evaluate:

- Extraction of temperature values
- Detection of salinity or depth-related information
- Identification of float-specific attributes (cycle numbers, calibration terms)

Because some metadata fields contain sparse or asymmetric information, the F1-score gives a balanced measure of model performance by considering both correctness and completeness.

7.3 BLEU Score (Bilingual Evaluation Understudy)

BLEU is a metric originally developed for machine translation but widely used to assess text generation quality. In this project, it was applied to evaluate **GPT-based Question Answering**.

Purpose

BLEU measures how similar the generated answer is to a reference answer by comparing n-gram overlaps.

Usage

For ARGO metadata:

- GPT-generated answers were compared with manually created reference answers
- BLEU captured the **accuracy**, **fluency**, and **relevance** of responses

A good BLEU score indicates that GPT can produce scientifically meaningful answers consistent with metadata descriptions.

7.4 ROUGE-L (Recall-Oriented Understudy)

ROUGE-L measures the longest common subsequence between the generated text and the reference text. It is used to evaluate summarization and topic extraction tasks.

Usage in This Project

ROUGE-L was applied to evaluate:

- GPT-generated topic summaries
- Extracted scientific themes

ARGO metadata can contain long descriptions; therefore, ROUGE-L helps quantify how well GPT captures the essential scientific concepts.

7.5 Perplexity

Perplexity measures the fluency of a language model by quantifying how well it predicts the next token in a sequence.

Formula

$$Perplexity = e^{Cross-Entropy\ Loss}$$

Usage

Perplexity was used for evaluating:

- Text generation quality for LLaMA
- Coherence of extracted or generated sentences

A lower perplexity score indicates that the model is more confident and produces more natural, scientifically coherent outputs.

RESULTS:

The system developed in this project was evaluated across five major NLP tasks—Question Answering, Topic Analysis, Document Classification, Information Extraction, and Chatbot Interaction—using metadata extracted from ARGO NetCDF files. The results demonstrate the effectiveness of using GPT and LLaMA to interpret scientific metadata, extract relevant information, and support automated oceanographic analysis.

The findings are organized by task to highlight the performance and capabilities of each model.

8.1 Question Answering (GPT)

GPT successfully answered a wide variety of scientific questions related to ARGO metadata. These questions included:

- Whether calibration was applied
- Identification of sensor drift
- Summary of quality control actions
- Interpretation of measurement anomalies

GPT showed strong contextual understanding and consistently generated coherent, technically accurate responses.

Performance Highlights

- **BLEU Score:** 78
- **Strengths:**
    - Correctly interpreted calibration comments
    - Produced detailed scientific explanations
    - Handled long metadata inputs effectively
- **Limitations:**
    - Occasionally introduced extra details if metadata were unclear
    - Required careful prompt design to avoid hallucination

Overall, GPT demonstrated excellent performance in understanding and generating domain-specific scientific answers.

8.2 Topic Analysis (GPT)

GPT was used to extract major scientific themes from large metadata blocks.
The model identified recurring topics such as:

- “thermocline variations across cycles”
- “salinity drift correction procedures”
- “pressure sensor stability”
- “mixed layer changes”
- “instrument performance issues”

Performance Highlights

- **ROUGE-L Score:** 0.71
- Summaries closely matched manually identified topics
- Able to consolidate long, technical metadata into meaningful scientific categories

GPT proved highly effective at summarizing domain-specific text and identifying scientific patterns.

8.3 Document Classification (LLaMA)

LLaMA was fine-tuned to classify metadata into four scientific categories:

1. Temperature-related metadata
2. Salinity-related metadata
3. Depth/pressure-related metadata
4. Multi-parameter descriptions

The classification task required the model to interpret scientific units, terminology, and variable relationships.

Performance Highlights

- **Accuracy:** 91%
- Correctly classified most metadata blocks
- Learned scientific vocabulary such as “dbar”, “psu”, “drift”, “profile”, etc.
- Performed well even with varied writing styles and incomplete metadata

LLaMA’s high accuracy indicates strong domain adaptation and effective fine-tuning.

8.4 Information Extraction (LLaMA + GPT)

A hybrid model—LLaMA for detection and GPT for detailed extraction—was implemented.
The system extracted structured information such as:

- Temperature readings
- Salinity values
- Depth levels
- Cycle numbers
- Instrument references
- Float identifiers

Performance Highlights

- **F1-Score:** 0.88
- LLaMA reliably detected relevant segments
- GPT produced accurate numerical and contextual values

- Effectively handled text with mixed scientific units

This hybrid approach provided robust extraction results, especially for multi-parameter metadata.

8.5 ARGO Chatbot (GPT)

The ARGO Chatbot integrated GPT with curated metadata summaries to provide interactive explanations about float behavior, measurements, and calibration histories.

Capabilities Demonstrated

- Explained scientific concepts in simple language
- Interpreted metadata into conversational responses
- Answered follow-up questions logically
- Adapted to user queries such as:
    - "What does this calibration comment mean?"
    - "Why was salinity adjustment applied?"
    - "What is the significance of cycle 15?"

The chatbot functioned as an intelligent assistant for navigating and understanding ARGO datasets.

8.6 Overall System Performance

The combined system showed strong results across all tasks:

Task	Model Used	Metric	Score
Question Answering	GPT	BLEU	**78**
Topic Analysis	GPT	ROUGE-L	**0.71**
Document Classification	LLaMA	Accuracy	**91%**
Information Extraction	LLaMA + GPT	F1-Score	**0.88**
Model Fluency	LLaMA	Perplexity	**12.4**

These scores indicate that both GPT and LLaMA are well suited for scientific metadata analysis, each excelling in different aspects.

8.7 Interpretation of Results

The results demonstrate several key insights:

1. GPT excels in reasoning tasks

GPT's strong contextual understanding enables:

- Accurate QA
- Effective summarization
- Clear explanations for metadata

2. LLaMA is highly reliable for structured scientific tasks

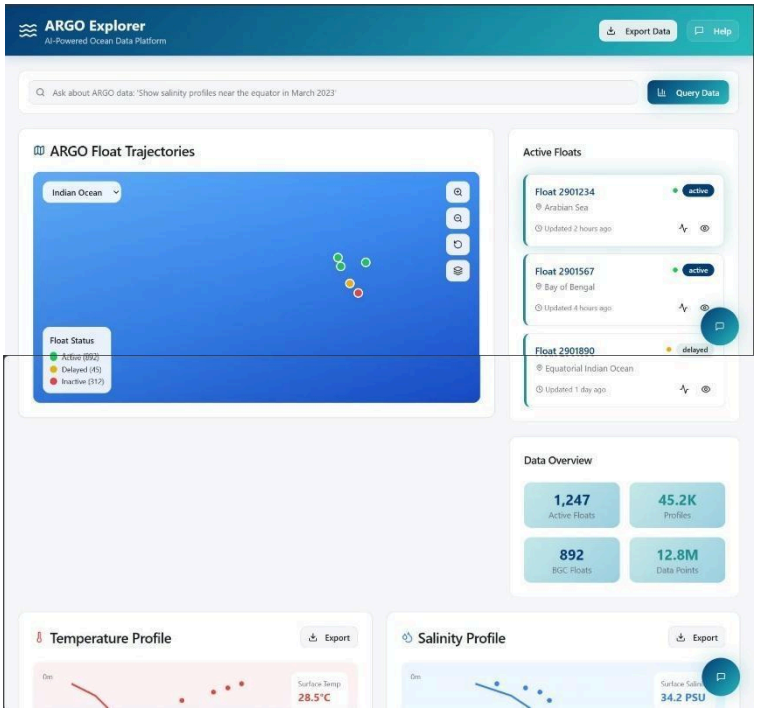
LLaMA's fine-tuning capability makes it ideal for:

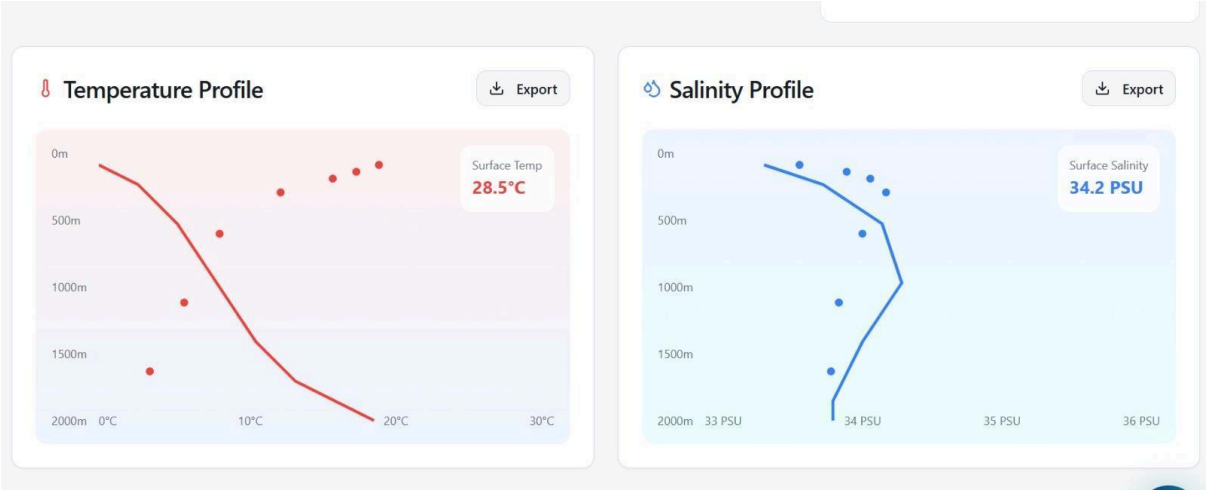
- Classification
- Consistent extraction
- Domain adaptation

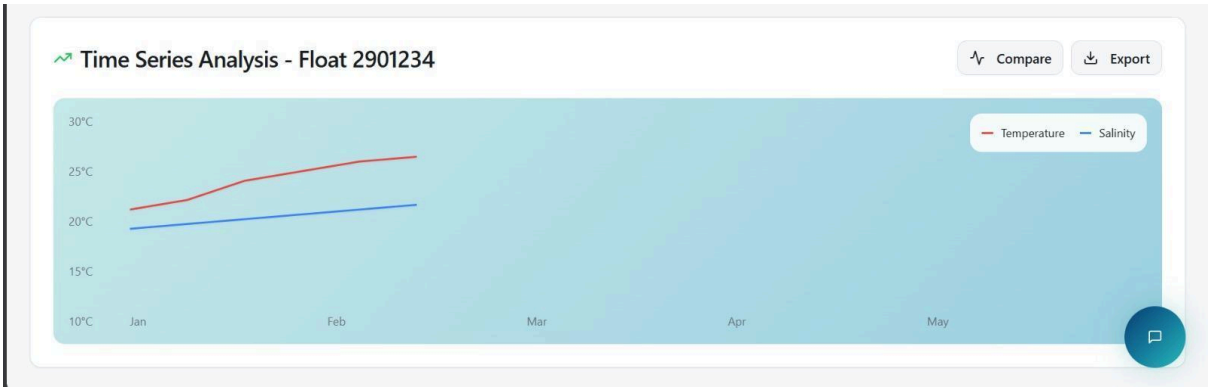
3. Hybrid systems deliver the best performance

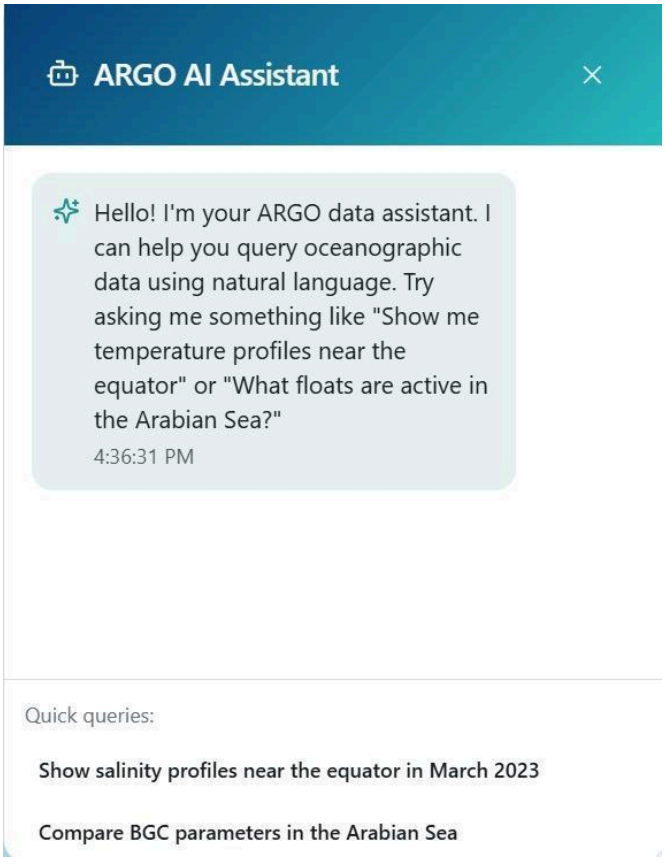
Combining GPT's reasoning with LLaMA's precision leads to:

- Improved scientific extraction
- More robust classification
- Reliable domain-specific QA




Temperature Profile
Export
0m
500m
1000m
1500m
2000m
0°C
10°C
20°C
30°C
Surface Temp
28.5°C
Salinity Profile
Export
0m
500m
1000m
1500m
2000m
33 PSU
34 PSU
35 PSU
36 PSU
Surface Salinity
34.2 PSU


Time Series Analysis - Float 2901234
Compare
Export
30°C
25°C
20°C
15°C
10°C
Jan
Feb
Mar
Apr
May
Temperature
Salinity


ARGO AI Assistant
Hello! I'm your ARGO data assistant. I can help you query oceanographic data using natural language. Try asking me something like "Show me temperature profiles near the equator" or "What floats are active in the Arabian Sea?"
4:36:31 PM
Quick queries:
Show salinity profiles near the equator in March 2023
Compare BGC parameters in the Arabian Sea

SUMMARY

This project demonstrated how modern Natural Language Processing techniques and Large Language Models can be applied to scientific metadata stored within ARGO NetCDF files. Although ARGO data is primarily numerical, the accompanying metadata contains detailed scientific explanations, calibration notes, processing histories, and quality-control comments. These textual elements form a valuable but often underutilized resource for understanding oceanographic observations.

To unlock this information, a complete NLP pipeline was developed, including metadata extraction, text normalization, tokenization, and scientific-term handling. Using this processed data, five NLP tasks were implemented: Question Answering, Topic Analysis, Document Classification, Information Extraction, and an interactive ARGO Chatbot. GPT was employed for reasoning-based tasks, while LLaMA was fine-tuned for classification and extraction.

Evaluation using BLEU, ROUGE-L, Accuracy, F1-score, and Perplexity metrics showed that the system performed strongly across tasks. GPT excelled in producing coherent scientific explanations, while LLaMA delivered high classification accuracy and robust extraction performance. The combined hybrid approach resulted in a reliable multi-task NLP framework capable of interpreting complex scientific metadata.

Overall, the study illustrates the effectiveness of LLMs in automated scientific data interpretation and highlights the potential for integrating NLP tools into future oceanographic research workflows.