Leveraging Large Language Models (LLM) for AIS Vessel Trajectory Prediction

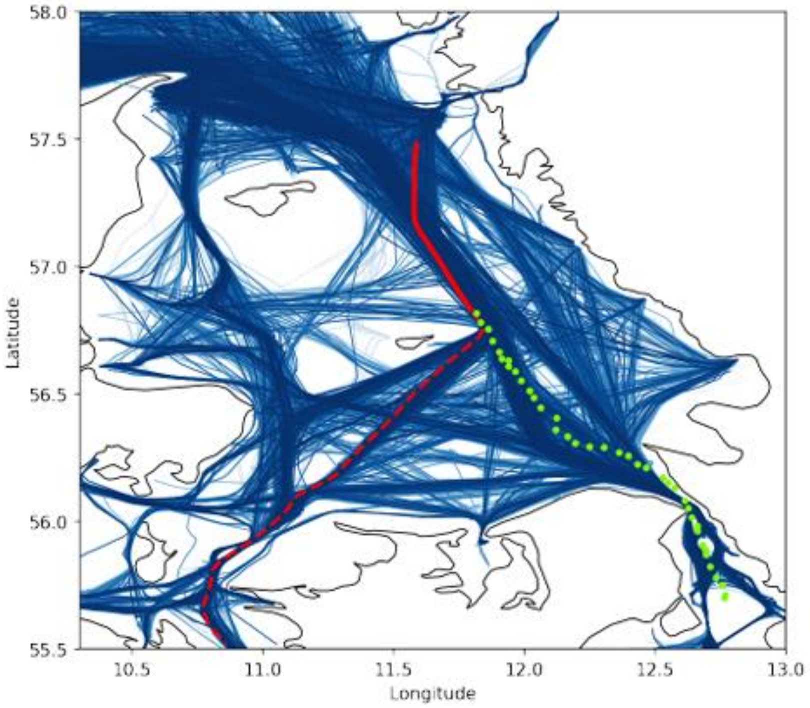
CSCI 6917 - Guided Research Grad I

Ryan Gross (G47667332)

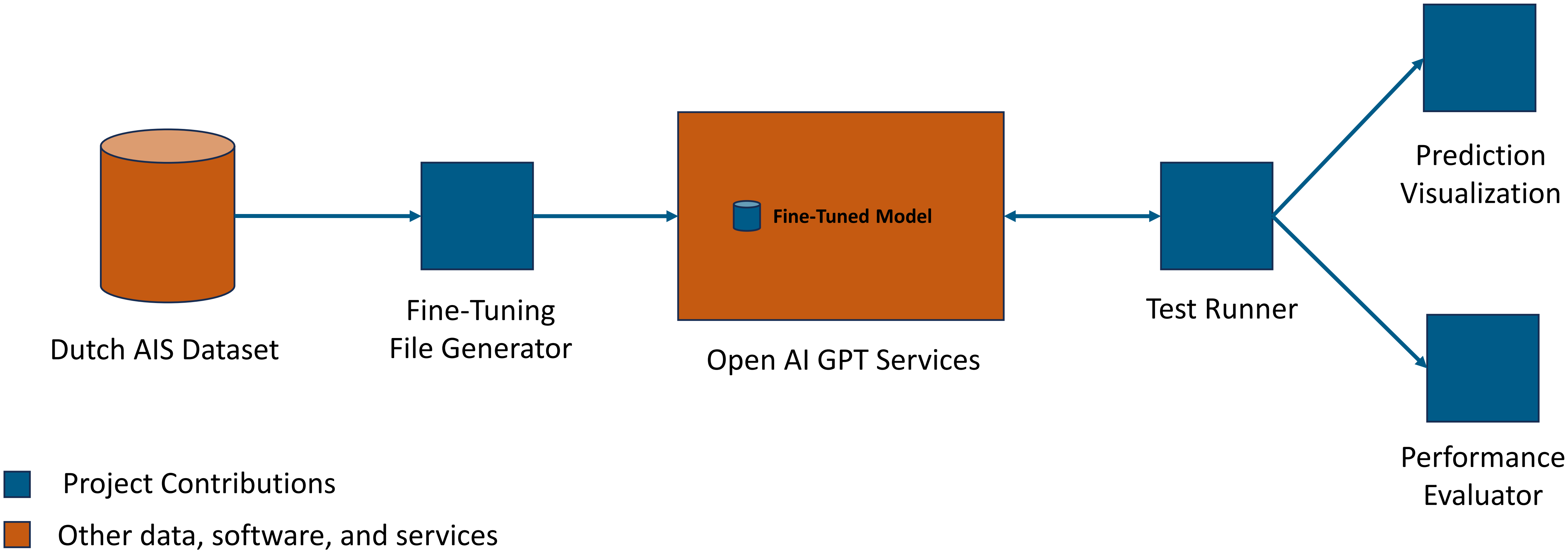
Summer Semester - 2023

Project Objective

- What are you trying to do? Articulate your objectives using absolutely no jargon.
  - **Predict the future trajectory of cargo vessels.**
- How is it done today, and what are the limits of current practice?
  - **Various seq2seq approaches such as RNN, LSTM, and transformers are used, but implementation is complex.**
- What is new in your approach and why do you think it will be successful?
  - **The approach treats the problem as a language translation task, which LMMs are good at, allowing pre-trained LLMs to solve the problem with little effort and complexity from the software engineer.**
- Who cares? If you are successful, what difference will it make?
  - **EMS, Coast Guard, Navy, and other maritime forces could better locate distressed or malign vessels.**
- What are the risks?
  - **LLM may not properly format messages, LMM may not pick up on vessel patterns.**
- How much will it cost? How long will it take?
  - **$12 and a summer semester.**
- What are the mid-term and final “exams” to check for success?
  - **Mid-term: Ensure LLM can issue predictions with the correct data format**
  - **Final: Performance beats or is near state-of-the-art**

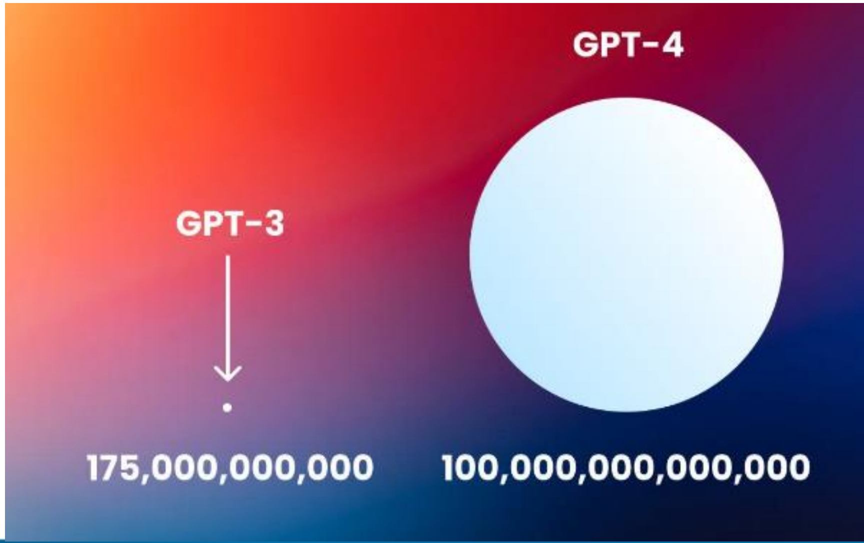


Technical Approach - Architecture



Technical Approach – Key Steps

1. Find and gather dataset – Needed to be free and large.
2. Clean data – Remove paths over land, non-moving vessels, impossibly fast trips, etc
3. Find best base model – Run Few-Shot learning on each model to determine best model to fine-tune
    1. Format data using prompt engineering
    2. Evaluate
    3. Visualize
    4. Compare
4. Create trained model – Create custom model by fine-tuning best base model
    1. Format data in GPT required JSONL for training
    2. Ensure time and cost of training meets requirements/limits
    3. Run fine-tuning job
5. Evaluate trained model – Run test set and measure accuracy loss
6. Visualize – Plotting
    1. Individual predictions to better understand results
    2. Overall accuracy loss across test set

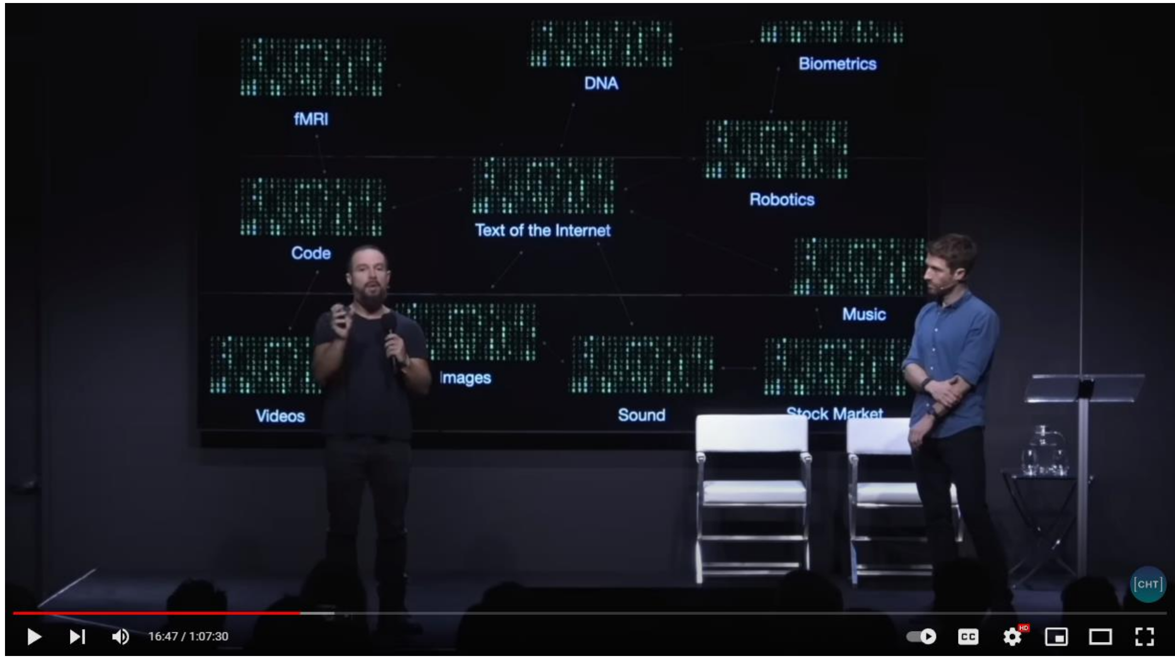


Technical Approach – Innovation

- This research further investigates the intuition that a large portion of prediction problems can be thought of as seq2seq, generative language problems.
- Examples:
  - Text phrase -> Python code (Code generation)
  - Text phrase -> Image (Image generation)
  - Image -> Next image (Image sequence generation)
  - English phrase -> French phrase (Speech translation)
  - Research paper text -> Research paper abstract (Text summarization)
  - Stock market historical data -> Stock market buy & sell tasks
  - Python code -> Code comments and documentation
  - Historical GPS locations -> Future GPS locations (This project)

- Leveraging existing, expertly trained models, could reduce the cost and complexity of solving many prediction problems.

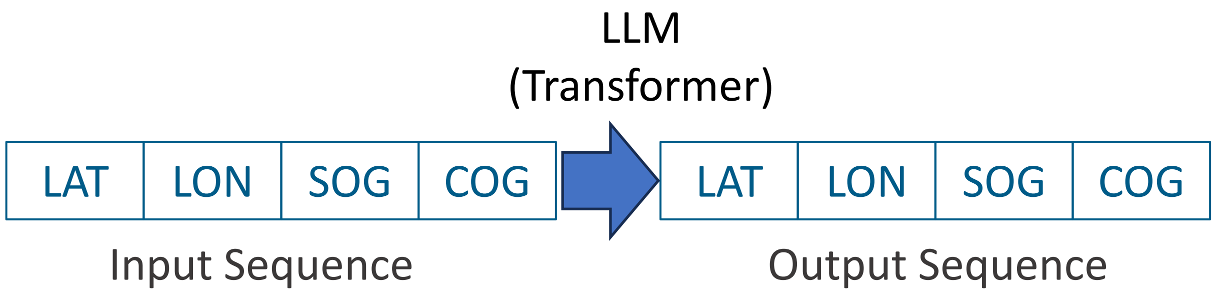
"Treat everything as language"



https://youtu.be/xoVJKj8lcNQ?t=967

Technical Approach – Innovation Cont.

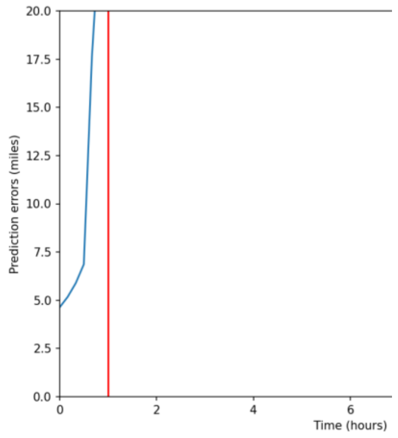
- The dataset is prepared to contain four input fields:
    1. Latitude
    2. Longitude
    3. Speed over Ground (SOG)
    4. Course over Ground (COG)
- The first half of a vessel trip is the input to the model, and the second half of the trip is the output the model should predict.
- After enough input output pairs, the LLM learns to complete the vessel's trip only given the input. This is similar to text completion in email or texts.
- Why LLMs instead of just using a transformer?
    - Training a transformer from scratch requires learning both data format and data content patterns.
    - LLMs are already trained to detect text format patterns. This is why "fine-tuning" LLMs has become popular.



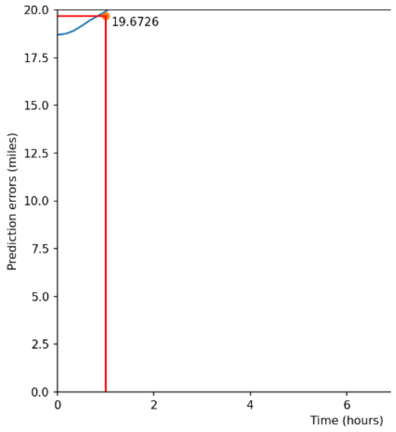

```
INPUT:
[[0.902284800000001, 0.000241481481481024, 0.3366666666666667, 0.1908333333333335],
[0.9062707999999958, 0.0185555555555442, 0.35, 0.246111111111109],
[0.907581920117l441, 0.03881266653955673, 0.3369999999999997, 0.2066666666666667],
[0.9119572001057434, 0.05661437692847809, 0.346969696969697, 0.1893686868686865],
[0.916883999999989, 0.07477777777777721, 0.35, 0.17],
[0.9206420000000008, 0.09309074074074032, 0.3433333333333334, 0.2525]]
OUTPUT:
[[0.9207287999999977, 0.1128807407407403 5, 0.32, 0.24000000000000002],
[0.9204009092225505, 0.1327053198908483, 0.356060606060606l, 0.2618181818181818],
[0.9197055001310531, 0.1526477469552080 8, 0.3399999999999997, 0.2629629629629629 5],
[0.917578066792810 8, 0.1717417593857311, 0.3430555555555556, 0.2952083333333335],
[0.9138304001005337, 0.1907566976679734, 0.3461111111111111, 0.3034490740740740 7],
[0.909522254634959, 0.208258417651993, 0.3196969696969696 5, 0.3178282828282828 5]]
```

Results

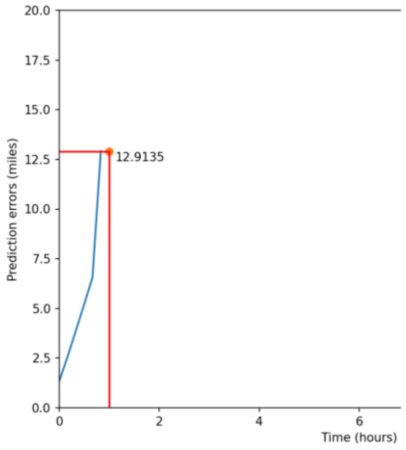
GPT 3.0
Curie
Few Shot



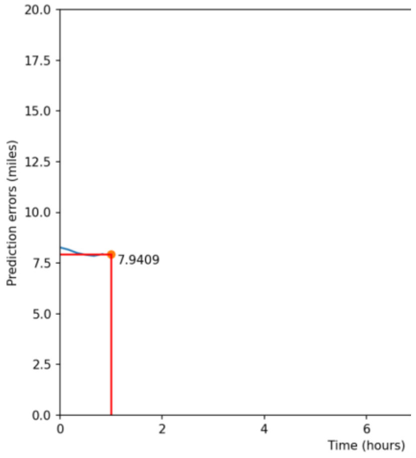
GPT 4.0
Few Shot

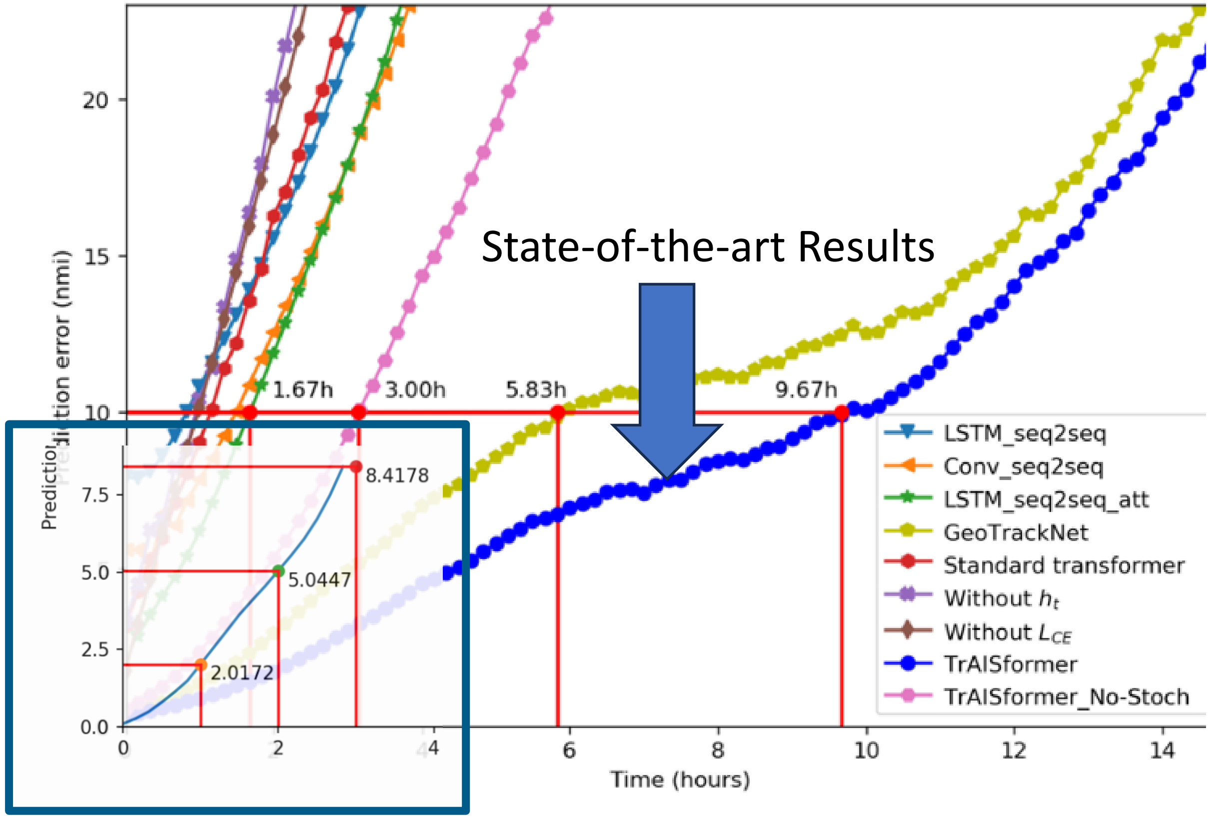


GPT 3.0
Davinci
Few Shot



GPT 3.0
Davinci
Fine-Tuned



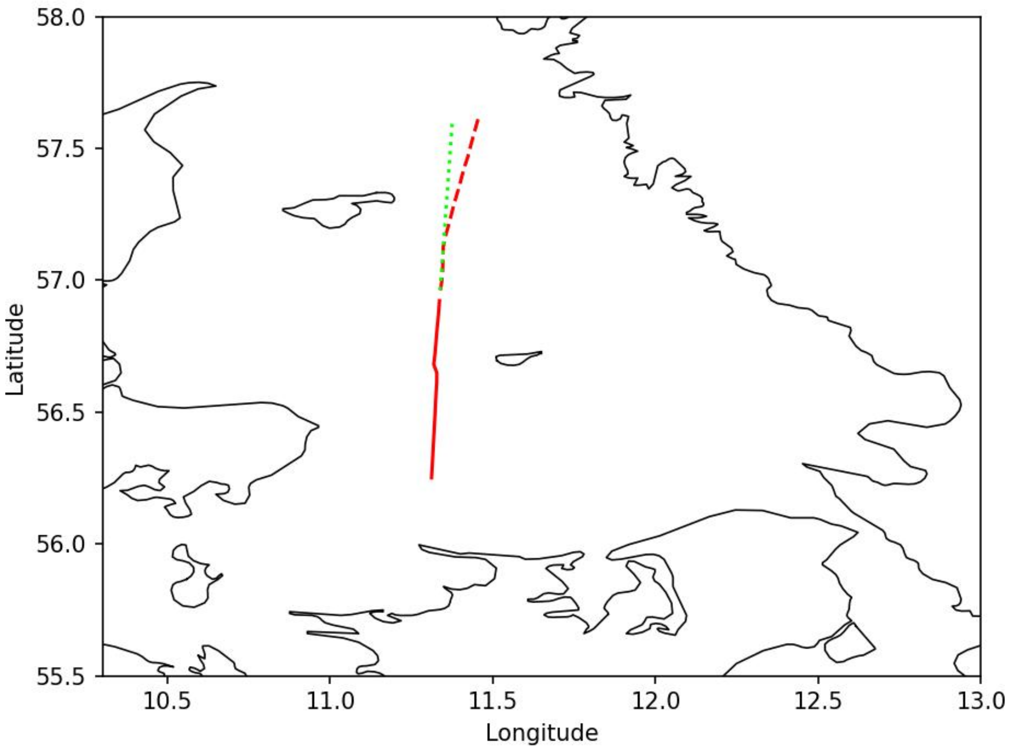


Best Results from Project
(GPT 3.5 Turbo, Few Shot)

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

Conclusion

- GPT 3.5 Turbo
    - With very few lines of code and “few-shot” training, GPT 3.5 Turbo outperforms many non-transformer algorithms.
    - GPT 3.5 Turbo seemed to perform about as well as a human, drawing a strait line from the end of the input.
    - GPT 3.5 Turbo picked up on LAT and LON patterns but did not pick up on SOG & COG patterns.
    - Fine-tuning not available
- GPT 3.0
    - Accepts too few characters as input and output to effectively pick up on patterns.
    - Training is expensive. Original 26.3MB training file had to be reduced to 177KB to stay under $10 billing limit.
- GPT 4.0
    - Strangely, performed about as well as GPT 3.0.
    - Fine-tuning not available
- While the original intuition and approach remains valid, OpenAI’s services allow for very limited customization or fine-tuning. Additionally, the cost to fine-tune is prohibitive to research students.



GPT 3.5 Turbo few-shot prediction in green, actual is dashed red, input is solid red.

Future Work

- Graduate Thesis Level
  - Continue to work with GPT 4.0 prompt engineering to try to increase performance.
  - Train an array of open source LLMs to compare results to state-of-the-art.
  - Compare LLMs with transformers trained from scratch.
  - Try different combinations of input and output features.

- PhD Dissertation Level
  - Understand how the transformer is learning trajectory patterns.
  - Understand how to optimize transformer/LLM parameters for trajectory prediction (heads, layers, embedded layers, learning rate, warm up tokens).
  - Build and demonstrate a transformer/LLM model that is state-of-the-art in location prediction.
  - Demonstrate applicability and versatility of model on ground, maritime, and air domains, given domain specific features.

References

[1] Vaswani, Ashish et al. “Attention is All you Need.” NIPS (2017).
[2] Nguyen, Duong and Ronan Fablet. “TrAISformer-A generative transformer for AIS trajectory prediction.” ArXiv abs/2109.03958 (2021): n. pag..
[3] D. Nguyen, R. Vadaine, G. Hajduch, R. Garello, and R. Fablet, “A Multi-task Deep Learning Architecture for Maritime Surveillance using AIS Data Streams,” in 2018 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Oct. 2018.
[3] D. Nguyen, R. Vadaine, G. Hajduch, R. Garello, and R. Fablet, “GeoTrackNet-A Maritime Anomaly Detector using Probabilistic Neural Network Representation of AIS Tracks and A Contrario Detection,” IEEE Transactions on Intelligent Transportation Systems, Feb. 2021.
[5] Ouyang, Long et al. “Training language models to follow instructions with human feedback.” ArXiv abs/2203.02155 (2022): n. Pag.
[6] Brown, Tom B. et al. “Language Models are Few-Shot Learners.” ArXiv abs/2005.14165 (2020): n. Pag.