IS460: Machine Learning and Applications:

LLM-based analysis for Maritime Incidents

G2

Group Members:

Solaiy Meyapan	*01415654*
Pang Ji Cheng	01420517
Ang Zhen Yue	01465453
Princess Sophie Montalban Pangan	01466764
Emmanuel Justin Koh	01437415

Table of Contents

Problem Statement

In 2020, when the Covid-19 pandemic hit, the global supply chain was heavily disrupted as many countries went into lockdown. The lockdowns slowed and stopped the flow of goods between countries causing significant slowdown to the growth of companies as they were both unable to receive raw materials or export finished goods.

These supply chain issues revealed deep vulnerabilities in maritime logistics. Port closures, crew shortages and increased shipping times further intensified the problem, leaving businesses unable to accurately depict shipping durations or maintain stock levels. In the post-pandemic landscape, ongoing geopolitical tensions, regulatory shifts, and natural disruptions continue to destabilize supply chains, particularly within maritime routes. In addition, there were also major disruption events that were occurring such as the closure of the Suez Canal due to a blockage.

Another example is in 2024 shipping bottlenecks at the Port of Singapore, spurred by unrest in the Red Sea and a spike in exports, showcase how external disruptions can cause significant delays, off-schedule arrivals and vessel bunching at major ports (*Singapore Business Review,* 2024). These disruptions have cascading effects on global supply chains, impacting both the cost and availability of goods.

Given these challenges, there is an urgent need for improved risk categorisation and proactive planning to mitigate the effects of maritime disruptions. Using Large Language

Models (LLMs) to analyse historical and real-time disruption data can enable companies to better anticipate risks and make more informed decisions. By categorising risks such as port closures, weather events, or geopolitical tensions, companies can tailor their logistics strategies to minimise delays and reduce costs.

Our project aims to address these challenges by leveraging LLM-based data exploration techniques to categorise maritime disruption risks. By building a dynamic model that incorporates data on previous and current disruptions, the goal is to offer a robust solution that improves supply chain resilience. We aim to create a model that would be able to analyze a new disruption event and cateogorize it into one of the categories as well as give it a severity rating. Additionally, using LLMs, we aim to be able to provide a brief summary of the impact as well as possible responses for the port authorities in Singapore.

Motivation

With an increasingly interconnected economy that has been spruced by the prevalence of free trade agreements (FTAs), the world has not been more reliant on its global supply chain than ever before. From the McKinsey cooperation metrics chart below (Figure 2.1) we can see that the cooperation between nations has been increasing steadily since 2014. In areas like trade and capital, the score for 2022 is the newest high. Indicating a trend of increase cooperation and interdependence between the nations.

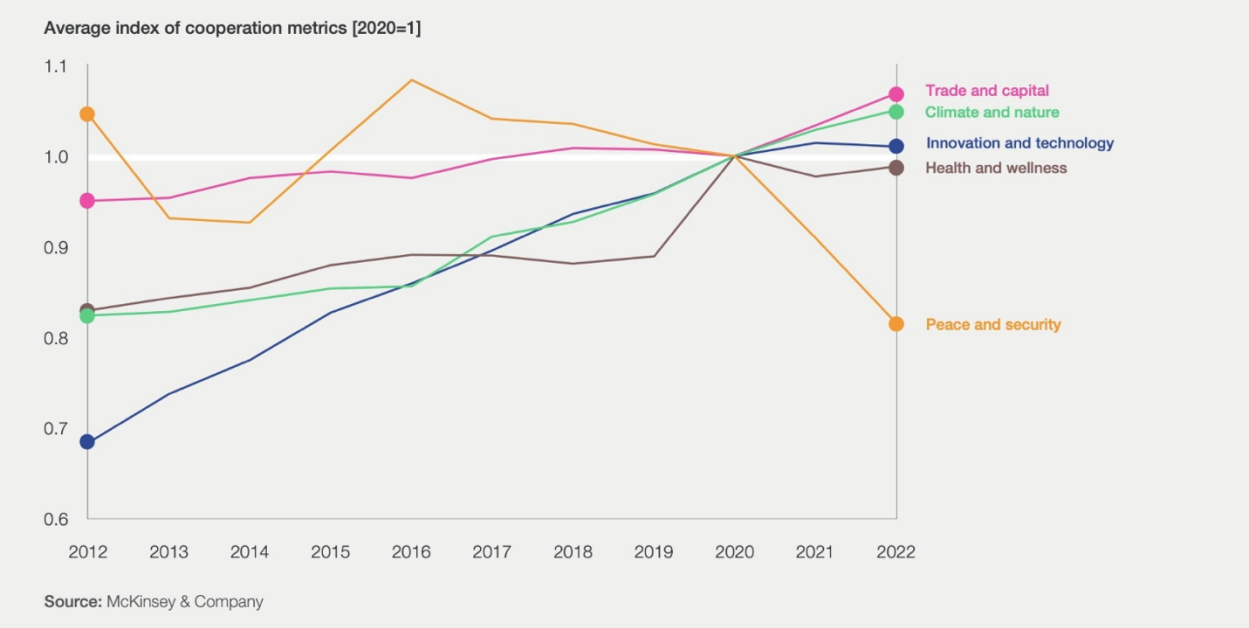


Figure 2.1: Cooperation metrics for different aspects of international cooperation (World Economic Forum, 2024)

Singapore alone has 15 bilateral and 12 regional FTAs include some of the largest combined trade agreements in the ASEAN-China, ASEAN-India, and ASEAN-Hong Kong trade blocs (Singapore International Free Trade Agreement, n.d.). These free trade agreements bring about a myriad of benefits for Singaporean companies including tariff concessions, preferential access to certain sectors, faster entry into markets and Intellectual Property

protection (Ministry of Trade and Industry, n.d.). Additionally, Singapore was able to save over SGD$1.1 billion in tariffs as a result of the free trade agreements as shown in Figure 2.2.

Figure 1.2: Infographic by Ministry of Trade and Industry (Ministry of Trade and Industry, 2018)

On top of free trade agreements, Singapore is in and of itself heavily reliant on the supply chain to bring in a lot of goods and services. Being built on little to no natural resources, Singapore has become overly reliant on the need for imports. In 2022, CNBC reported that Singapore imports over 90% of its food (Ong, 2022). This makes Singapore, susceptible to global supply chain issues, if there is an incident somewhere that disrupts the steady import supply of food into Singapore this could result in a shortage of food in Singapore and with food being one of the necessities for survival this has the potential to become a major problem for Singapore.

Case in point, in 2022, Malaysia declared a ban of chicken exports (Andres, 2022). At this time, Singapore's largest source of chicken imports was Malaysia, with over a third of Singapore's chicken being brought in from Malaysia which accounts for 73,000 tones in chicken sales (The Straits Times, 2022). In this case, Singapore was able to react well to the sudden ban from Malaysia and import chicken from other neighboring countries. But this highlights the gravity of the Singapore's over-reliance on its imports.

One of the largest sources of imports for Singapore, is through the maritime sector and it even accounts for 7% of the Gross Domestic Product (GDP) of Singapore (Ministry of Trade and Industry, 2018). In fact, even worldwide, 80% of world trade is carried by sea (Ministry of Education, n.d.). With such heavy reliance on the maritime sector in Singapore and internationally, it is important that we analyze the risks attributed with the maritime industry

so that we can better prepare ourselves for any impact on it due to various global disruption events. This is the pressing need that the project is trying to address.

Literature Review

Maritime Incident Information Extraction using Machine and Deep Learning Techniques (A. Mackenzie et al., 2021)

In this paper, they reviewed several natural language processing (NLP) techniques to extract information about piracy from unstructured maritime news articles. The proposed pipeline architecture is shown below in Figure 3.1.

The first method explained was article segmentation based on dates used in the article. To quote, “When an article mentions several incidents, the author typically uses a date as a clue that the context is switching from one incident to another”. The method, Using the SUTime CoreNLP date parser, uses dates as they are a very machine-recognisable piece of information. In the event an article describes multiple incidents in the same day, the method will look for transitional words such as ‘separately’ which signals a change in topic. The algorithm achieved an F1 score of 0.82.

The second method was Vessel Metadata extraction which was to determine the names and types of all ships involved in an incident. Traditional NER libraries proved to be ineffective as ship names could vary from arbitrary alphanumeric strings to common English words and the libraries could not pick them out. The method detailed in the report was a simple NER

engine that identified common vessel prefixes (MV, HMS) and vessel types (tanker, frigate) that allowed them to identify name such as ‘USS Bainbridge’. Vessels that did not follow these rules were identified using a list of international vessels, a database of 64K ships. However, this database was often outdated as ships changed names when they were bought by other companies. When a vessel could not be found in the database, they assigned each ship the type that occurred closest in the text to a mention of the ship’s name. This algorithm achieved an F1-score of 0.89 when tested with 82 articles. It was 85% accurate in determining the vessel type (misidentified 3% and the other 12% was due to the pipeline not offering any guesses at all).

The third method was incident metadata extraction which is to determine the incident type, severity of the incident and the response taken by the victim, and the eventual impact of this response. They trained a binary model to classify an article into all relevant labels by converting articles to bag-of-words format. Using Weka’s CfsSubsetEval function, a set of 300 words that were highly correlated with the label of interest and least correlated amongst themselves were selected. They were then ran through scikit-learn’s implementations of multinomial Naïve Bayes, Bernoulli Naïve Bayes, logistic regression and random forest, along with CatBoost. Convolutional Neural Networks were also included in the experiment. The models were trained on 90% of the data and 10-fold cross-validation accuracy was chosen as a metric. CatBoost achieved the highest classification accuracy in all cases but one. One problem their model faced was mistakes when basic reasoning or common-sense

knowledge was required. Suggested improvements included pretraining a model on a wide variety of English documents.

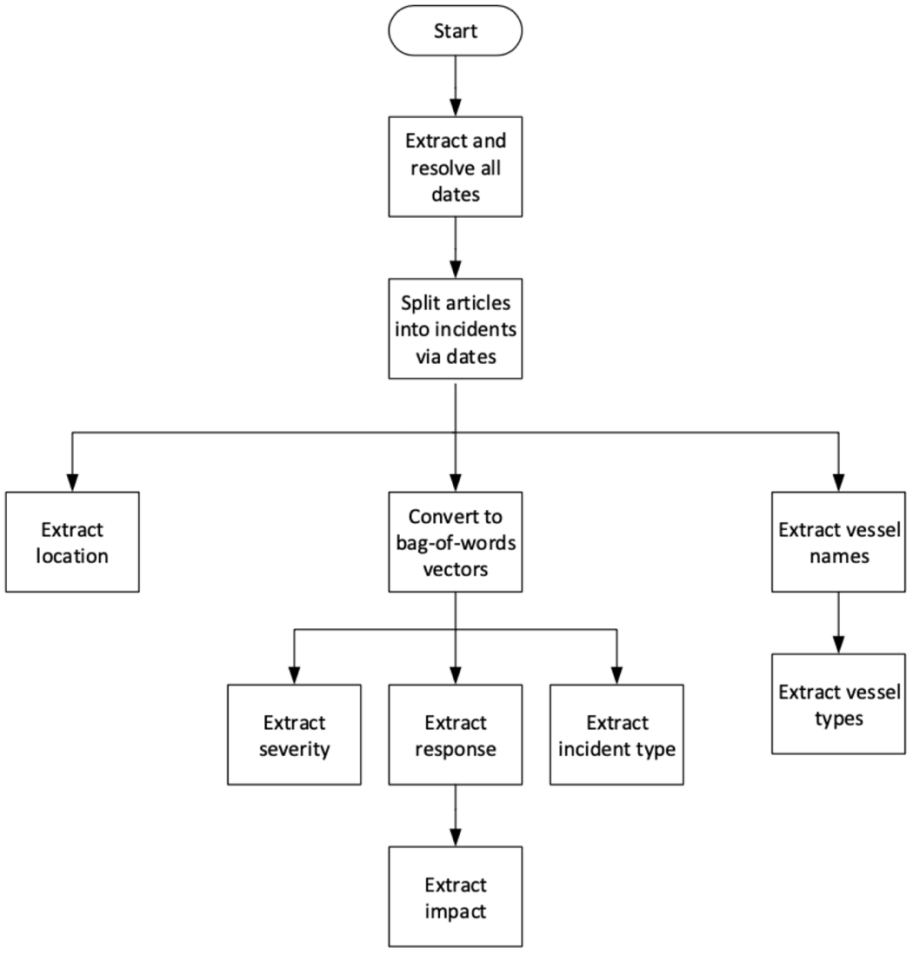


Figure 2.1: Proposed Pipeline Architecture

Enabling Maritime Risk Assessment using NLP based Deep Learning techniques (V. Jidkov et al., 2020)

This paper discussed the benefits of analysing high-risk areas in seas to help ships reroute. They analysed factors extracted from online articles and used them as training datasets for

a machine learning (ML) model. The ML model used Natural Language Processing (NLP) Deep learning (DL) techniques to identify specific areas around the world that are considered a risk to voyages. The proposed pipeline architecture is shown below in Figure 3.2.

This article used two variants of word embeddings: Word2Vec and FastText for their document classification tasks. It was observed that FastText captured word structure similarity better than contextual similarity, hence Word2Vec was prioritised. As a second embedding method, they used BERT due to the high sentiment classification accuracy results shown.

To identify articles containing mentions of maritime incidents, they used multiple binary classification models with a single output node. This way allowed them to determine if an article contained mentions of multiple different incidents. The three types of models used were ANN, CNN and LSTM, two of each model was constructed: one for word2vec and the other for BERT embedding. Loss function used was binary cross-entropy and the optimiser was Keras' Adam. The best-performing LSTM model achieved a result of 94.4% accuracy while the best-performing Weka Logistic Regression model was 77.56% accuracy.

These six models were then used for both document classification and incident classification tasks. It was observed that BERT performed poorly on highly unbalanced datasets but came highly close on the most balanced ones, BERT models were also never

recorded to have the highest scores. Hence, we also chose not to explore the use of BERT, not only did they have a poor accuracy with BERT but they mentioned the unbalanced data set as a reason for the poor performance which was an issue we would have as well. CNN outperformed every other model in most cases.

The next step was to extract information, proposed solution used DL NER libraries such as SpaCy and Stanford NLP library to extract important date information. They extracted the publishing date of the article and cases that failed to gather a publishing date were considered failed. Through analysis, it was deduced that the first mention of a date in articles were usually the correct date of the incident. The Stanford NLP library was used to parse dates (e.g. last Tuesday) relative to the published date. This method achieved an accuracy of 61.8%.

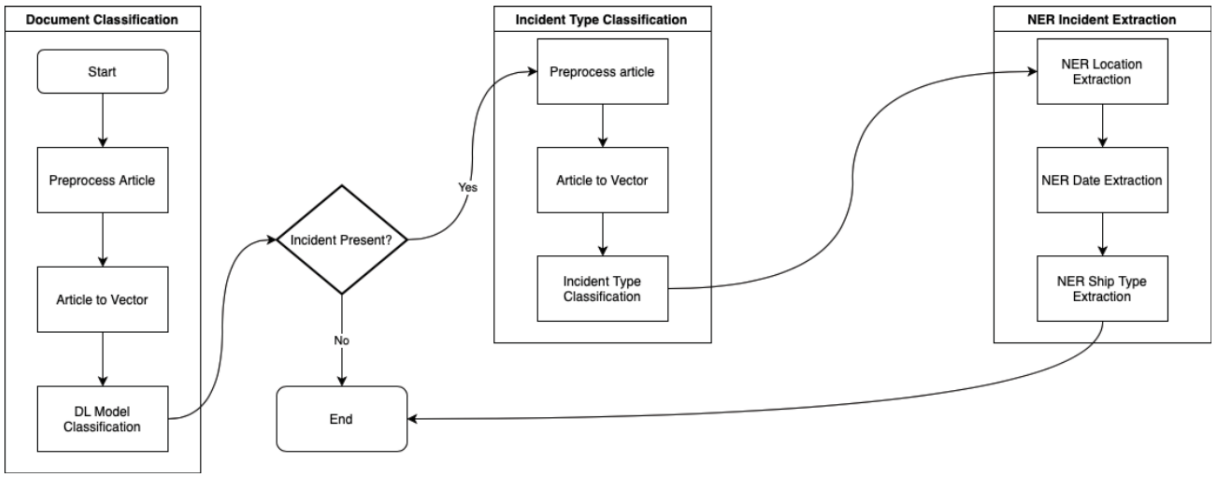


Figure 3.2: Proposed Pipeline Architecture

Automatic Identification of Maritime Incidents from unstructured articles (A.Teske et al., 2018)

This paper presented two NLP techniques for identifying maritime incidents and their respective locations described in unstructured articles. The proposed pipeline architecture is shown below in Figure 3.3.

The dataset of maritime incidents was created from downloading, reading and labelling 602 articles from online sources and labelled them: 0 for non-maritime incidents and 1 for maritime incidents. To find articles that are actually maritime incidents, they used Weka's CsfSubsetEval to search for subsets of attributes that are highly correlated to the class while having low intercorrelation. They also tallied and retained the top 300 words to achieve a good balance between the size and the accuracy of the vector.

To obtain the location of the maritime incident, they proposed the algorithm get_location which contains 6 steps. After processing the data, they used Named Entity Recognition Location Detector and Regular Expression Location Detector to detect named entities while filtering for locations. If 2 locations are detected, it proceeds to the 'Select Location' step. Otherwise, it will proceed to the 'Relax search conditions step' if there is a condition to relax, else it will terminate. The algorithm uses 2 conditions to focus the search: First three sentences and incident sentences. If both of these conditions do not yield any results, it will return 'Unknown'. For the last stage, if there are more than 1 candidate location, the most specific location will be chosen and the rest will be discarded. We explored doing this in our

own project, but one issue we found was when articles mentioned two difference locations or two different vessels. This caused wrongful identification.

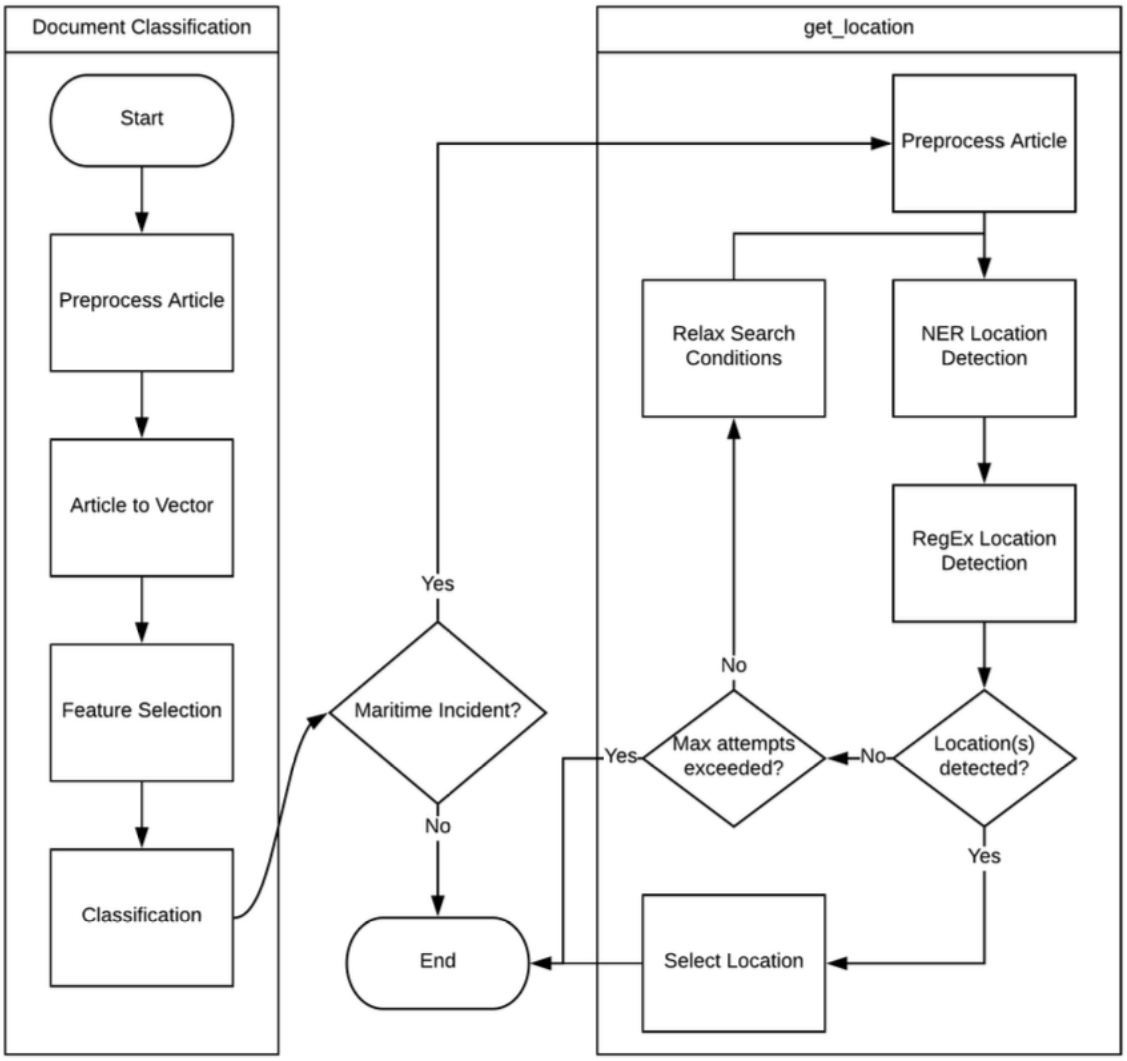


Figure 3: Proposed Pipeline Architecture

Datasets

1	Data.gov (MPA) Link (Source)	
2	Vessel Arrival (>75T) Annual Link	Potential correlation with major world events
3	Vessel Calls (>75T) Monthly Link	5 different purposes in the dataset: 1. Supplies 2. Cargo 3. Repairs 4. Bunkers 5. Others
4	Vessel Arrivals (>75T) Breakdown, Monthly Link	9 Vessel type: 1. Tug 2. Barges 3. Tanker 4. Container 5. Freighter 6. Passenger 7. Bulk Carrier 8. Miscellaneous
5	Tanker Arrivals (>75T) Breakdown, Monthly Link	Tanker types: 1. Oil Tanker 2. Chemical Tankers 3. LNG & LPG Tankers
6	External sources: Link Port performance 2024: Link / Link	

Datasets/ API	World News: 1. https://huggingface.co/datasets/AiresPucrs/News-Category-Dataset 2. 1. Vessel Delay Portcast (Congestion tracker API) 2. Vessel Accident 3. Maritime Piracy or Terrorism Risk 4. Port or important route congestion 5. Port Criminal Activities 6. Cargo Damage and Loss 7. Inland transportation risks 8. Environmental impact and Pollution 9. Natural extreme events and extreme weather 10. Cargo or ship detainment 11. Unstable regulatory and political environment 12. Maritime-related but not covered by existing categories 13. Not maritime-related

Milestone / Timeline

Week	Project Milestone	Activity	Remarks
4	Requirement Gathering and Data Exploration.	- Data Sourcing - EDA - Data Preparation - Literature Review and Research	
5	Data Labelling and Synthesizing Datasets. Creation of Article Summarizer	- Using LLM to generate datasets - Synthesizing datasets and labelling training sets - Using contextual embedding generation - Using existing LLM's to do keyword and topic extraction from articles o Article summarizer will retain critical information	
6	Initial training of LLM to do risk classification	- LLMs for article enhancement o Additional information, facts, related topics o Bias Detection o Sentiment Classification o Contextual understanding o Entity Identification - Supervised training of LLM with training set, performing dynamic label generation to classify Risk from articles	
7	Calibration of LLM improves accuracy of risk classification. Training of LLM for severity measurement	- LLM to generate severity through sentiment analysis - Probability and impact analysis, risk matric calculation - Weight Risk assessment o e.g. Financial, Operational, Reputational Cost - Analyse proposed mitigation strategies and assessed its reduction of a potential risk impact	Project proposal presentation
8	Training of LLM to assess Impact	- On Selected 2 categories - Risk Severity Analysis - Probability and Impact	

Phase 1: Data Collection

For phase 1 of the problem, we aim to collect any related or useful datasets that are related to maritime disruptions or incidents. We planned to use the data that was provided by ASTAR to classify and assess the severity of incidents. For the category classification, we scraped it with BeautifulSoup, however for the LLM corpus, we found out about Jina.ai, a webcrawler that is a free API. All you need to do is send a http request of the URL of the website you want to scrape to the API endpoint, and it will return to you the website in markdown format in the form of a string. This proved to be very useful for the LLM phase as LLMs process markdowns much better than most web scraped outputs.

Classifier Data Collection

For the articles scraped for the classifier, we mainly scraped them from Google News, gCaptain and the Maritime-Executive website. We input a query that is predetermined by us and took each article these websites gave us and scraped it with BeautifulSoup. The queries were the categories provided by ASTAR which are:

1. Vessel Delay
2. Vessel Accident
3. Maritime Piracy or Terrorism Risk
4. Port or important route congestion
5. Port Criminal Activities
6. Cargo Damage and Loss
7. Inland transportation risks

8. Environmental impact and Pollution
9. Natural extreme events and extreme weather
10. Cargo or ship detainment
11. Unstable regulatory and political environment
12. Maritime-related but not covered by existing categories
13. Not maritime-related

In the end we collected about 60 articles of data per category for each of the ones listed above.

Phase 3: Classification

For phase 3 of the problem, we aim to classify a news article input into one of the 13 given categories:

1. Vessel Delay
2. Vessel Accident
3. Maritime Piracy or Terrorism Risk
4. Port or important route congestion
5. Port Criminal Activities
6. Cargo Damage and Loss
7. Inland transportation risks
8. Environmental impact and Pollution
9. Natural extreme events and extreme weather

10. Cargo or ship detainment
11. Unstable regulatory and political environment
12. Maritime-related but not covered by existing categories
13. Not maritime-related

Data Pre-Processing

The data that we collected was web-scraped data and it contains tags such as ‘/t’ and ‘/n’. Hence, we created a function that would clean the text by removing these tags.

The function is listed below:

```
def clean_text(text):
    # Remove non-alphabetic characters
    text = re.sub(r'\W', ' ', text)
    # Lowercase the text
    text = text.lower()
    # Tokenize and remove stopwords
    tokens = [word for word in text.split() if word not in stop_words]
    return ' '.join(tokens)
```

Additionally, we used the ntlk python library to get a list of stop words. This is to remove words that are misspelled. It is also able to remove a whole laundry list of filler words as shown below:

Pronouns:

i, me, my, myself, we, our, ours, ourselves, you, your, yours, yourself, he, him, his, himself, she, her, hers, herself, it, its, itself, they, them, their, theirs, themselves

Articles and Determiners:

a, an, the, this, that, these, those

Prepositions:

at, by, for, with, about, against, between, into, through, during, before, after, above, below, to, from, up, down, in, out, on, off, over, under, again, further, then, once, here, there, when, where, why, how, all, any, both, each, few, more, most, other, some, such, no, nor, not, only, own, same, so, than, too, very, s, t, can, will, just, don, should, now

Conjunctions:

and, or, but, if, because, as, until, while

Auxiliary and Modal Verbs:

am, is, are, was, were, be, been, being, have, has, had, having, do, does, did, doing, would, could, should, ought

Sample Pre-Processing data:	SINGAPORE : The world's second busiest container port at Singapore has seen a major spike in congestion, forcing carriers to stretch charter agreements and to build container fleets in preparation for an elongated peak season. According to Hong Kong analyst Linerlytica significant new port congestion has added to the already over-stretched container market that is struggling to cope with shortages of container equipment and vessel space, mainly as a consequence of the Red Sea diversions. "The global port congestion indicator hit the 2m TEUs mark, accounting for 6.8% of the global fleet with Singapore becoming the new congestion hotspot. The SCFI [Shanghai Containerized Freight Index] has jumped by 42% in the past month, with further gains to follow in June as carriers are adding new surcharges and rate hikes," commented Linerlytica in its latest weekly report. Carriers have been forced to secure new equipment and extend vessel charters beyond September "after their initial hesitation to commit too far ahead in the event that demand would falter after the summer peak

	season," claimed Linerlytica. As berthing delays continue to lengthen\xa0— industry sources say delays in Singapore have hit a week in some cases\xa0— container lines are increasingly bypassing the port\xa0to keep their vessel schedules as consistent as possible.\xa0\xa0 Market signals are "extremely bullish" and\xa0are reminiscent of the substantial rate increases that began in 2021 and continued throughout 2022. Port congestion at that time was caused by backed up freight in US ports with insufficient inland capacity to store or move containers, causing ships to be delayed waiting for cargo handling slots, with the knock-on effect that too few empty containers were being returned to Asia for loading. This year congestion has returned to container supply chains, with Singapore becoming the latest victim, as ships are returning to Asia out of schedule due to the extended journeys around the African Cape. Moreover, the carriers have insufficient tonnage to handle the much longer supply chains caused by the Cape diversions "There are berthing delays of up to seven days with the total capacity waiting to berth rising to 450,000 TEUs in recent days. The severe congestion has forced some carriers to omit their planned Singapore port calls, which will exacerbate the problem at downstream ports that will have to handle additional volumes," said the analyst. In addition, delays have resulted in vessel bunching, causing "spillover congestion" and schedule disruptions at downstream ports. Increasing port congestion has already taken more than 400,000 teu of vessel capacity out of circulation in the last week alone, with a further escalation to the current critical delays expected in the coming weeks as the peak season gathers pace. Copyrights © 2024, By India Shipping News | All Rights Reserved.'
Post-Processing output:	singapore world second busiest container port singapore seen major spike congestion forcing carriers stretch charter agreements build container fleets preparation elongated peak season according hong kong analyst linerlytica significant new port congestion added already stretched container market struggling cope shortages container equipment vessel space mainly consequence red sea diversions global port congestion indicator hit 2m teus mark accounting 6 8 global fleet singapore becoming new congestion hotspot scfi shanghai containerized freight index jumped 42 past month gains follow june carriers adding new surcharges rate hikes commented linerlytica latest weekly report carriers forced secure new equipment extend vessel charters beyond september initial hesitation commit far ahead event demand would falter summer peak season claimed linerlytica berthing delays continue lengthen industry sources say delays singapore hit week cases container lines increasingly bypassing port keep vessel schedules consistent possible market signals extremely bullish reminiscent substantial rate increases began 2021 continued throughout 2022 port congestion time caused backed freight us ports insufficient inland capacity store move containers causing ships delayed waiting cargo

	handling slots knock effect empty containers returned asia loading year congestion returned container supply chains singapore becoming latest victim ships returning asia schedule due extended journeys around african cape moreover carriers insufficient tonnage handle much longer supply chains caused cape diversions berthing delays seven days total capacity waiting berth rising 450 000 teus recent days severe congestion forced carriers omit planned singapore port calls exacerbate problem downstream ports handle additional volumes said analyst addition delays resulted vessel bunching causing spillover congestion schedule disruptions downstream ports increasing port congestion already taken 400 000 teu vessel capacity circulation last week alone escalation current critical delays expected coming weeks peak season gathers pace copyrights 2024 india shipping news rights reserved

The benefits of this phase is that we are able to standardize the input variables and remove words from affecting the output.

Vectorizer

The initial question to solve was which vectorizer to user. The popular ones from our research were CountVectorizer, tf-idf and Word2Vec. The CountVectorizer looks at the pure counts of each word when comparing a inputs, the tf-idf model looks at the frequency of occurances of each word, punishing the input for having excessive words that do not add value in terms of finding out which category it falls under. The Word2Vec vectorizer converts the words to vectors and uses the vectors to identify the meaning in the word. We initially tried to build a Naïve Bayes model with each vectorizer to see which one could give the best accuracy.

For the CountVectorizer we were able to get a Multinomial Naïve Bayes model as listed below:

Naïve Bayes Model Vectorizer: CountVectorizer			
Accuracy: 0.73	Precision: 0.78	Recall: 0.73	F1 Score: 0.74

For the tf-idf we were able to get a Multinomial Naïve Bayes model as listed below:

Naïve Bayes Model Vectorizer: tf-idf			
Accuracy: 0.66	Precision: 0.80	Recall: 0.66	F1 Score: 0.66

For the Word2Vec we attempted to create the Multinomial Naïve Bayes model but for a Multinomial Naïve Bayes model you would need a count-based input data and the Word2Vec vectorizer creates a vector for the inputs and this is not compatible with the Multinomial Naïve Bayes model. Hence, we built a Support Vector Machine (SVM) instead.

Support Vector Machine (SVM) Model Vectorizer: Word2Vec			
Accuracy: 0.52	Precision: 0.53	Recall: 0.52	F1 Score: 0.52

Summary

From our initial exploration of the baseline models, we found that the CountVectorizer was the best in terms of providing an accurate model. We rationalized that this could be because the benefit gained from the tf-idf model of punishing excess words was no longer beneficial since we used the ntlk library to remove words from the inputs. This garners the benefits of tf-idf vectorizer as no longer applicable.

When comparing the Word2Vec model, the accuracy of the Word2Vec could have been lower due to the vector space in which the words are located are already similar since most words reference terms commonly found in the maritime industry.

Hence, we concluded that the best vectorizer to use for our models in the categorization phase would be the CountVectorizer.

Models

We knew the baseline model to compare with for the CountVectorizer was the Multinomial Naïve Bayes model:

Naïve Bayes Model Vectorizer: CountVectorizer			
Accuracy: 0.73	Precision: 0.78	Recall: 0.73	F1 Score: 0.74

We explored a few models and the best one we found was a Logistic Regression Tree model. This model is a cross between a Logistic Regression and a Decision Tree. Traditionally, logistic regressions are good at handling binary splits, but it is possible to do a multi split with logistic regression. When we did the multi-class logistic regression, we realized the accuracy was not as good as a binary split logistic regression. The logistic regression tree is one way that you can continue to use a binary split logistic regression and still achieve a multi category split.

This is achieved by looking at a binary split of each class first. Let's take for example the class is 'vessel delay'. We take the data set and relabel the data to 0 for those that are not 'vessel delay' and 1 for those that are 'vessel delay' and then we split this into a train and test split, and we train a logistic regression model to identify this category using the input text as well as the categorization of 1 or 0. We also compare the final output not just on accuracy, precision, recall and F1 score but also look at the entropy of the output. For the entropy we look at the original categorizations of all 13 categories and see if there is an increase or decrease in entropy (information gain).

We do this for all 12 categories (the 13th being not any of the remaining categories). Then we compare the models entropy for each item and we choose the categorization that had the most information gain for the first level of the tree. We then use this logistic regression model at the first level of the tree, then we repeat the steps with the remaining categories for the next level of tree until we have reached the lowest level. The outputs we got are listed:

Level 1:

Logistic Regression Model Category: Vessel Delay Vectorizer: CountVectorizer				
Accuracy: 0.93	Precision: 0.92	Recall: 0.93	F1 Score: 0.92	Information Gain: 0.0210

Logistic Regression Model Category: Vessel Accidents Vectorizer: CountVectorizer				
Accuracy: 0.96	Precision: 0.96	Recall: 0.96	F1 Score: 0.96	Information Gain: 0.2942

Logistic Regression Model Category: Port Congestion Vectorizer: CountVectorizer				
Accuracy: 0.95	Precision: 0.95	Recall: 0.95	F1 Score: 0.95	Information Gain: 0.2076

Logistic Regression Model Category: Maritime Piracy and Terrorism Risk Vectorizer: CountVectorizer				
Accuracy: 0.95	Precision: 0.95	Recall: 0.95	F1 Score: 0.95	Information Gain: 0.2076

Logistic Regression Model Category: Port Criminal Activities Vectorizer: CountVectorizer				
Accuracy: 0.98	Precision: 0.98	Recall: 0.98	F1 Score: 0.98	Information Gain: 0.3128

Logistic Regression Model Category: Cargo Damage and Loss Vectorizer: CountVectorizer				
Accuracy: 0.93	Precision: 0.92	Recall: 0.93	F1 Score: 0.92	Information Gain: 0.1022

Logistic Regression Model Category: Inland Transportation Risk Vectorizer: CountVectorizer				
Accuracy: 0.94	Precision: 0.94	Recall: 0.94	F1 Score: 0.94	Information Gain: 0.1271

Logistic Regression Model Category: Natural extreme events and extreme weather Vectorizer: CountVectorizer

Accuracy: 0.98	Precision: 0.99	Recall: 0.98	F1 Score: 0.98	Information Gain: 0.1189

Logistic Regression Model Category: Environmental impact and pollution Vectorizer: CountVectorizer				
Accuracy: 0.95	Precision: 0.95	Recall: 0.95	F1 Score: 0.95	Information Gain: -0.1653

Logistic Regression Model Category: Cargo or ship detainment Vectorizer: CountVectorizer				
Accuracy: 0.98	Precision: 0.99	Recall: 0.98	F1 Score: 0.98	Information Gain: 0.3089

Logistic Regression Model Category: Unstable regulatory and political environment Vectorizer: CountVectorizer				
Accuracy: 0.97	Precision: 0.97	Recall: 0.97	F1 Score: 0.96	Information Gain: 0.0516

Logistic Regression Model Category: Maritime related but not covered by existing categories Vectorizer: CountVectorizer				
Accuracy: 0.90	Precision: 0.88	Recall: 0.90	F1 Score: 0.89	Information Gain: -0.2208

From the first level we identified the model that binary split for the category of Port Criminal Activities was the best one as it provided the most information gain. Hence this would act the root node in our decision tree. This would mean that for any new input we would first put it through this model and then if it is classified it would be put under the category of Port Criminal Activities and if it is not, it would go on to the next model on the tree.

It is worth noting that some models had high accuracy, precision, recall, f1-score but lead to an information loss. We conducted further investigation to understand what was causing the varied information gain and we identified that the main cause of the differences in information gain is how much the model was mixing the groups together. Take for example we have groups A, B and C. If model 1 and model 2 had the same accuracy, precision, recall and f1-score, it does not mean the two models would lead to the same information gain. If for example both models misclassified 5 of the test data points but if in model 1 it only wrongly classified items that are actually in Group B as Group A whereas in model 2 it classified some from Group B and some from Group C as Group A this would mean that the information gain from model 2 is lower as the amount of entropy is greater. This is what led to some models leading to information loss despite high accuracy as they often misclassified items from a variety of groups as opposed to some models that only misclassified those from 1 or 2 groups.

Level 2:

For the second level we only pass into the models the output from using the selected logistic regression model from level 1 which was the binary split of Port Criminal Activities. We then created a binary split for each category on the output and repeated the steps in level 1 for level 2. The models and their performances are listed below:

Logistic Regression Model Category: Vessel Delay Vectorizer: CountVectorizer

Accuracy:	Precision:	Recall:	F1 Score:	Information Gain:
0.93	0.92	0.93	0.93	0.0757

Logistic Regression Model Category: Vessel Accidents Vectorizer: CountVectorizer				
Accuracy:	Precision:	Recall:	F1 Score:	Information Gain:
0.97	0.97	0.97	0.96	0.3315

Logistic Regression Model Category: Port Congestion Vectorizer: CountVectorizer				
Accuracy:	Precision:	Recall:	F1 Score:	Information Gain:
0.96	0.96	0.96	0.96	0.3373

Logistic Regression Model Category: Maritime Piracy and Terrorism Risk Vectorizer: CountVectorizer				
Accuracy:	Precision:	Recall:	F1 Score:	Information Gain:
0.97	0.98	0.97	0.98	0.0304

Logistic Regression Model Category: Cargo Damage and Loss Vectorizer: CountVectorizer				
Accuracy:	Precision:	Recall:	F1 Score:	Information Gain:
0.94	0.94	0.94	0.94	0.1622

Logistic Regression Model Category: Inland Transportation Risk Vectorizer: CountVectorizer				
Accuracy:	Precision:	Recall:	F1 Score:	Information Gain:
0.95	0.95	0.95	0.94	0.0905

Logistic Regression Model Category: Natural extreme events and extreme weather Vectorizer: CountVectorizer				
Accuracy:	Precision:	Recall:	F1 Score:	Information Gain:
0.97	0.99	0.97	0.97	0.2456

Logistic Regression Model Category: Environmental impact and pollution Vectorizer: CountVectorizer				
Accuracy: 0.97	Precision: 0.97	Recall: 0.97	F1 Score: 0.97	Information Gain: 0.3030

Logistic Regression Model Category: Cargo or ship detainment Vectorizer: CountVectorizer				
Accuracy: 0.97	Precision: 0.97	Recall: 0.97	F1 Score: 0.97	Information Gain: 0.2742

Logistic Regression Model Category: Unstable regulatory and political environment Vectorizer: CountVectorizer				
Accuracy: 0.99	Precision: 0.99	Recall: 0.99	F1 Score: 0.99	Information Gain: 0.3064

Logistic Regression Model Category: Maritime related but not covered by existing categories Vectorizer: CountVectorizer				
Accuracy: 0.94	Precision: 0.94	Recall: 0.94	F1 Score: 0.94	Information Gain: 0.1010

The model selected for the second level of the tree was a binary split on the category of Port Congestion.

Level 3:

Logistic Regression Model Category: Vessel Delay Vectorizer: CountVectorizer				
Accuracy: 0.94	Precision: 0.94	Recall: 0.94	F1 Score: 0.94	Information Gain: 0.2003

Logistic Regression Model Category: Vessel Accidents Vectorizer: CountVectorizer

Accuracy:	Precision:	Recall:	F1 Score:	Information Gain:
0.93	0.93	0.93	0.93	0.2288

Logistic Regression Model Category: Maritime Piracy and Terrorism Risk Vectorizer: CountVectorizer				
Accuracy:	Precision:	Recall:	F1 Score:	Information Gain:
1.00	1.00	1.00	1.00	0.3687

Logistic Regression Model Category: Cargo Damage and Loss Vectorizer: CountVectorizer				
Accuracy:	Precision:	Recall:	F1 Score:	Information Gain:
0.94	0.94	0.94	0.94	0.2128

Logistic Regression Model Category: Inland Transportation Risk Vectorizer: CountVectorizer				
Accuracy:	Precision:	Recall:	F1 Score:	Information Gain:
0.93	0.93	0.93	0.93	-0.0959

Logistic Regression Model Category: Natural extreme events and extreme weather Vectorizer: CountVectorizer				
Accuracy:	Precision:	Recall:	F1 Score:	Information Gain:
0.96	0.96	0.96	0.96	0.2831

Logistic Regression Model Category: Environmental impact and pollution Vectorizer: CountVectorizer				
Accuracy:	Precision:	Recall:	F1 Score:	Information Gain:
0.93	0.93	0.93	0.93	0.2474

Logistic Regression Model Category: Cargo or ship detainment Vectorizer: CountVectorizer				
Accuracy:	Precision:	Recall:	F1 Score:	Information Gain:
0.98	0.98	0.98	0.98	0.3360

Logistic Regression Model Category: Unstable regulatory and political environment Vectorizer: CountVectorizer				
Accuracy: 0.97	Precision: 0.97	Recall: 0.97	F1 Score: 0.97	Information Gain: 0.1988

Logistic Regression Model Category: Maritime related but not covered by existing categories Vectorizer: CountVectorizer				
Accuracy: 0.92	Precision: 0.92	Recall: 0.92	F1 Score: 0.91	Information Gain: 0.0561

The model selected for the third level of the tree was a binary split on the category of Maritime Piracy and Terrorism Risk.

Level 4:

Logistic Regression Model Category: Vessel Delay Vectorizer: CountVectorizer				
Accuracy: 0.94	Precision: 0.93	Recall: 0.94	F1 Score: 0.93	Information Gain: 0.1748

Logistic Regression Model Category: Vessel Accidents Vectorizer: CountVectorizer				
Accuracy: 0.91	Precision: 0.90	Recall: 0.91	F1 Score: 0.91	Information Gain: 0.1804

Logistic Regression Model Category: Cargo Damage and Loss Vectorizer: CountVectorizer				
Accuracy: 0.92	Precision: 0.92	Recall: 0.92	F1 Score: 0.92	Information Gain: 0.1889

Logistic Regression Model Category: Inland Transportation Risk Vectorizer: CountVectorizer

Accuracy: 0.93	Precision: 0.93	Recall: 0.93	F1 Score: 0.93	Information Gain: -0.2124

Logistic Regression Model Category: Natural extreme events and extreme weather Vectorizer: CountVectorizer				
Accuracy: 0.94	Precision: 0.93	Recall: 0.94	F1 Score: 0.93	Information Gain: -0.0880

Logistic Regression Model Category: Environmental impact and pollution Vectorizer: CountVectorizer				
Accuracy: 0.96	Precision: 0.96	Recall: 0.96	F1 Score: 0.96	Information Gain: 0.2312

Logistic Regression Model Category: Cargo or ship detainment Vectorizer: CountVectorizer				
Accuracy: 0.97	Precision: 0.97	Recall: 0.97	F1 Score: 0.97	Information Gain: 0.3332

Logistic Regression Model Category: Unstable regulatory and political environment Vectorizer: CountVectorizer				
Accuracy: 0.97	Precision: 0.97	Recall: 0.97	F1 Score: 0.97	Information Gain: 0.2438

Logistic Regression Model Category: Maritime related but not covered by existing categories Vectorizer: CountVectorizer				
Accuracy: 0.97	Precision: 0.97	Recall: 0.97	F1 Score: 0.97	Information Gain: 0.4331

The model selected for the fourth level of the tree was a binary split on the category of Maritime related but not covered by existing categories.

Level 5:

Logistic Regression Model Category: Vessel Delay Vectorizer: CountVectorizer				
Accuracy: 0.95	Precision: 0.95	Recall: 0.95	F1 Score: 0.95	Information Gain: 0.3063

Logistic Regression Model Category: Vessel Accidents Vectorizer: CountVectorizer				
Accuracy: 0.93	Precision: 0.94	Recall: 0.93	F1 Score: 0.93	Information Gain: 0.3429

Logistic Regression Model Category: Cargo Damage and Loss Vectorizer: CountVectorizer				
Accuracy: 0.93	Precision: 0.93	Recall: 0.93	F1 Score: 0.93	Information Gain: 0.2741

Logistic Regression Model Category: Inland Transportation Risk Vectorizer: CountVectorizer				
Accuracy: 0.91	Precision: 0.90	Recall: 0.91	F1 Score: 0.90	Information Gain: 0.0994

Logistic Regression Model Category: Natural extreme events and extreme weather Vectorizer: CountVectorizer				
Accuracy: 0.94	Precision: 0.94	Recall: 0.94	F1 Score: 0.94	Information Gain: -0.0614

Logistic Regression Model Category: Environmental impact and pollution Vectorizer: CountVectorizer				
Accuracy: 0.95	Precision: 0.95	Recall: 0.95	F1 Score: 0.95	Information Gain: 0.3050

Logistic Regression Model Category: Cargo or ship detainment Vectorizer: CountVectorizer

Accuracy: 0.95	Precision: 0.95	Recall: 0.95	F1 Score: 0.95	Information Gain: 0.3050

Logistic Regression Model Category: Unstable regulatory and political environment Vectorizer: CountVectorizer				
Accuracy: 0.95	Precision: 0.95	Recall: 0.95	F1 Score: 0.95	Information Gain: 0.2757

The model selected for the fifth level of the tree was a binary split on the category of Unstable regulatory and political environment.

Level 6:

Logistic Regression Model Category: Vessel Delay Vectorizer: CountVectorizer				
Accuracy: 0.97	Precision: 0.97	Recall: 0.97	F1 Score: 0.97	Information Gain: 0.3811

Logistic Regression Model Category: Vessel Accidents Vectorizer: CountVectorizer				
Accuracy: 0.94	Precision: 0.93	Recall: 0.94	F1 Score: 0.93	Information Gain: -0.0649

Logistic Regression Model Category: Cargo Damage and Loss Vectorizer: CountVectorizer				
Accuracy: 0.90	Precision: 0.89	Recall: 0.90	F1 Score: 0.89	Information Gain: 0.3102

Logistic Regression Model Category: Inland Transportation Risk Vectorizer: CountVectorizer				
Accuracy: 0.94	Precision: 0.93	Recall: 0.94	F1 Score: 0.93	Information Gain: -0.1873

Logistic Regression Model Category: Natural extreme events and extreme weather Vectorizer: CountVectorizer				
Accuracy: 0.94	Precision: 0.93	Recall: 0.94	F1 Score: 0.93	Information Gain: -0.2871

Logistic Regression Model Category: Environmental impact and pollution Vectorizer: CountVectorizer				
Accuracy: 0.95	Precision: 0.95	Recall: 0.95	F1 Score: 0.95	Information Gain: 0.3637

Logistic Regression Model Category: Cargo or ship detainment Vectorizer: CountVectorizer				
Accuracy: 0.96	Precision: 0.96	Recall: 0.96	F1 Score: 0.96	Information Gain: 0.3613

The model selected for the sixth level of the tree was a binary split on the category of Vessel Delay.

Level 7:

Logistic Regression Model Category: Vessel Accidents Vectorizer: CountVectorizer				
Accuracy: 0.89	Precision: 0.89	Recall: 0.89	F1 Score: 0.88	Information Gain: 0.3987

Logistic Regression Model Category: Cargo Damage and Loss Vectorizer: CountVectorizer				
Accuracy: 0.89	Precision: 0.89	Recall: 0.89	F1 Score: 0.88	Information Gain: 0.2636

Logistic Regression Model Category: Inland Transportation Risk Vectorizer: CountVectorizer

Accuracy: 0.92	Precision: 0.93	Recall: 0.92	F1 Score: 0.92	Information Gain: 0.2687

Logistic Regression Model Category: Natural extreme events and extreme weather Vectorizer: CountVectorizer				
Accuracy: 0.92	Precision: 0.92	Recall: 0.92	F1 Score: 0.92	Information Gain: -0.3653

Logistic Regression Model Category: Environmental impact and pollution Vectorizer: CountVectorizer				
Accuracy: 0.92	Precision: 0.92	Recall: 0.92	F1 Score: 0.92	Information Gain: 0.1713

Logistic Regression Model Category: Cargo or ship detainment Vectorizer: CountVectorizer				
Accuracy: 0.98	Precision: 0.99	Recall: 0.98	F1 Score: 0.98	Information Gain: 0.5043

The model selected for the seventh level of the tree was a binary split on the category of Cargo or ship detainment.

Level 8:

Logistic Regression Model Category: Vessel Accidents Vectorizer: CountVectorizer				
Accuracy: 0.90	Precision: 0.90	Recall: 0.90	F1 Score: 0.90	Information Gain: 0.0930

Logistic Regression Model Category: Cargo Damage and Loss Vectorizer: CountVectorizer				
Accuracy: 0.88	Precision: 0.88	Recall: 0.88	F1 Score: 0.88	Information Gain: 0.2001

Logistic Regression Model Category: Inland Transportation Risk Vectorizer: CountVectorizer				
Accuracy: 0.93	Precision: 0.94	Recall: 0.93	F1 Score: 0.93	Information Gain: 0.2240

Logistic Regression Model Category: Natural extreme events and extreme weather Vectorizer: CountVectorizer				
Accuracy: 0.92	Precision: 0.92	Recall: 0.92	F1 Score: 0.92	Information Gain: -0.3653

Logistic Regression Model Category: Environmental impact and pollution Vectorizer: CountVectorizer				
Accuracy: 0.93	Precision: 0.94	Recall: 0.93	F1 Score: 0.93	Information Gain: 0.4849

The model selected for the eight level of the tree was a binary split on the category of Environmental impact and pollution.

Level 9:

Logistic Regression Model Category: Vessel Accidents Vectorizer: CountVectorizer				
Accuracy: 0.87	Precision: 0.87	Recall: 0.87	F1 Score: 0.87	Information Gain: 0.3079

Logistic Regression Model Category: Cargo Damage and Loss Vectorizer: CountVectorizer				
Accuracy: 0.91	Precision: 0.91	Recall: 0.91	F1 Score: 0.91	Information Gain: 0.2867

Logistic Regression Model Category: Inland Transportation Risk Vectorizer: CountVectorizer

Accuracy: 0.87	Precision: 0.87	Recall: 0.87	F1 Score: 0.86	Information Gain: 0.0319

Logistic Regression Model Category: Natural extreme events and extreme weather Vectorizer: CountVectorizer				
Accuracy: 0.96	Precision: 0.96	Recall: 0.96	F1 Score: 0.96	Information Gain: -0.4858

The model selected for the ninth level of the tree was a binary split on the category of Vessel Accidents.

Level 10:

Logistic Regression Model Category: Cargo Damage and Loss Vectorizer: CountVectorizer				
Accuracy: 0.86	Precision: 0.86	Recall: 0.86	F1 Score: 0.86	Information Gain: 0.5105

Logistic Regression Model Category: Inland Transportation Risk Vectorizer: CountVectorizer				
Accuracy: 0.86	Precision: 0.86	Recall: 0.86	F1 Score: 0.86	Information Gain: 0.2576

Logistic Regression Model Category: Natural extreme events and extreme weather Vectorizer: CountVectorizer				
Accuracy: 0.94	Precision: 0.95	Recall: 0.94	F1 Score: 0.95	Information Gain: 0.5289

The model selected for the tenth level of the tree was a binary split on the category of Natural extreme events and extreme weather.

Level 11:

Logistic Regression Model Category: Cargo Damage and Loss Vectorizer: CountVectorizer				
Accuracy: 0.89	Precision: 0.92	Recall: 0.89	F1 Score: 0.89	Information Gain: 0.5030

Logistic Regression Model Category: Inland Transportation Risk Vectorizer: CountVectorizer				
Accuracy: 0.75	Precision: 0.75	Recall: 0.75	F1 Score: 0.75	Information Gain: 0.2519

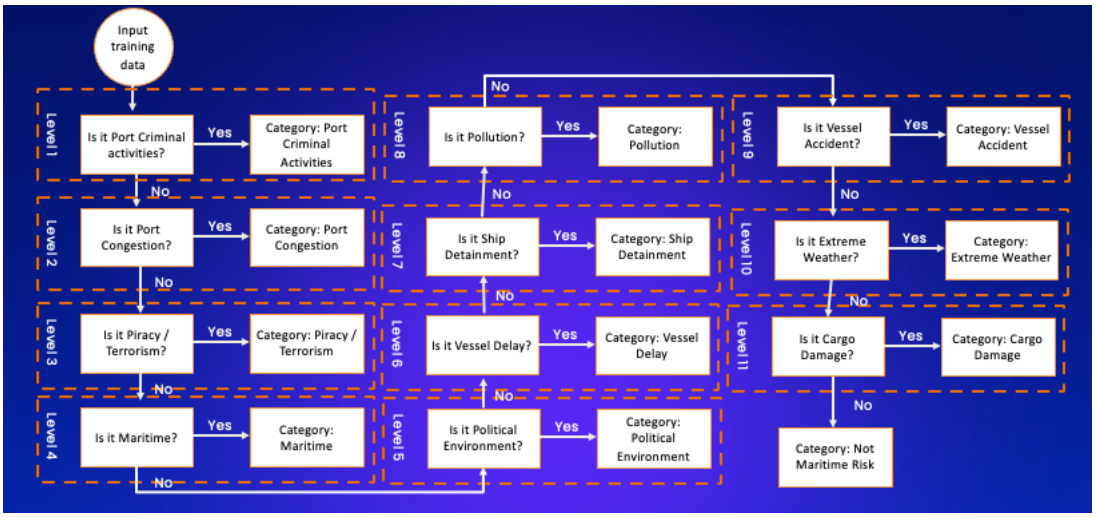
The model selected for the eleventh level of the tree was a binary split on the category of Cargo Damage and Loss.

Level 12:

Logistic Regression Model Category: Inland Transportation Risk Vectorizer: CountVectorizer				
Accuracy: 0.80	Precision: 0.79	Recall: 0.80	F1 Score: 0.79	Information Gain: -0.1359

For this level, there is only one category left to categorize but since the information gain is negative, we decided that it is better to not include the model and to not categorize in terms of inland transportation risk instead. This means that if the item passes through all levels it is none of the given categorize and instead will be labelled as not maritime related risk and we have included the category of inland transportation risk into this not maritime related category.

After completing all levels, we got the following tree:



When creating test inputs, one thing to note is that it should also be put through the same pre-processing that was used on the text at the start of the training phase, including putting it through the text cleaning function and running it against the stop words from the ntlk library.

Phase 4: Severity

In this phase we tried to build models to identify the severity of the incident. In this case we are categorizing into one of 3 severities:

1. Minor
2. Moderate
3. Severe

The choice of the categories comes from the data set provided to us by ASTAR. Since we do not know what metrics was used to determine the severity rating, we could only use the dataset provided to us as training data for the model.

Data Pre-Processing

Since this data was provided by ASTAR, a lot of the data was already largely cleaned. We still used the ntlk stop words library to remove filler words.

We did analyze if there would be a benefit to cleaning the data. To do this we compared 2 models with the same vectorizer and all variables kept constant with the only difference being the input X variable where for one we used the same clean_content function from phase 3 and for the other we did not. We compared the results below:

Multinomial Naïve Bayes Model Without clean_content function Vectorizer: CountVectorizer			
Accuracy: 0.59	Precision: 0.59	Recall: 0.59	F1 Score: 0.56

Multinomial Naïve Bayes Model With clean_content function Vectorizer: CountVectorizer			
Accuracy: 0.59	Precision: 0.57	Recall: 0.59	F1 Score: 0.55

Whilst there is a difference, the difference is marginal and not worth noting. Hence, to standardize the procedures from Phase 3, for this phase we continued with the use of the clean_content function mentioned earlier.

Vectorizer

Again, we decided to build a Multinomial Naïve Bayes model to compare the different vectorizers available which are CountVectorizer, tf-idf vectorizer and Word2Vec to identify which of the vectorizers would work best with the severity categorization.

Naïve Bayes Model Vectorizer: CountVectorizer			
Accuracy: 0.59	Precision: 0.57	Recall: 0.59	F1 Score: 0.55

Naïve Bayes Model Vectorizer: tf-idf vectorizer			
Accuracy: 0.56	Precision: 0.47	Recall: 0.56	F1 Score: 0.47

Naïve Bayes Model Vectorizer: Word2Vec			
Accuracy: 0.56	Precision: 0.59	Recall: 0.56	F1 Score: 0.54

From the above, we can see that the best vectorizer to be used for severity classification is the Word2Vec vectorizer.

The difference to that of the classification model in Phase 3 could be because the types of words used in the different training sets for the severity model has a different degree of significance which shows up in the differing vector space. This allows the Word2Vec vectorizer to give the best output.

Models

Initially, we tried building a logistic regression model. Hoping that we would see similar performance gains as we did in the first model:

Logistic Regression Vectorizer: Word2Vec			
Accuracy: 0.53	Precision: 0.59	Recall: 0.53	F1 Score: 0.37

But the model ended up performing significantly worse that even the baseline models, especially in terms of F1 score. The reasons behind the significantly reduced performances in this phase will be discussed at the conclusion section of this phase.

We also tried to use the Support Vector Machine:

Support Vector Machine Vectorizer: Word2Vec			
Accuracy: 0.53	Precision: 0.59	Recall: 0.53	F1 Score: 0.37

Yet again, the scores were equally poor.

We then tried to move away from the Word2Vec vectorizer and repeated the previous steps, for this segment we tried using tf-idf Vectorizer as well as the CountVectorizer:

Logistic Regression Vectorizer: tf-idf Vectorizer			
Accuracy: 0.56	Precision: 0.47	Recall: 0.56	F1 Score: 0.49

Support Vector Machine Vectorizer: tf-idf Vectorizer			
Accuracy: 0.56	Precision: 0.59	Recall: 0.56	F1 Score: 0.53

Logistic Regression Vectorizer: CountVectorizer			
Accuracy: 0.57	Precision: 0.58	Recall: 0.57	F1 Score: 0.53

Support Vector Machine Vectorizer: CountVectorizer			
Accuracy: 0.58	Precision: 0.57	Recall: 0.58	F1 Score: 0.56

Yet again, the CountVectorizer seems to perform the best, but unlike before, we were not able to get very high scores for the model. So, we tried to identify just Severity = ‘Severe’ for the binary split logistic regression.

Logistic Regression Severity = Severe Vectorizer: CountVectorizer

Accuracy:	Precision:	Recall:	F1 Score:
0.91	1.00	0.06	0.11

For this model, we initially got very high accuracy and precision scores, however, the recall and F1 score is low indicating poor external validity of the model. This also means that the model is being too conservative generally classifying things as not severe hence getting it a high accuracy and precision but it fails to correctly classify any severe ones leading to the low recall.

Having reaching a roadblock, we explored other models as shown below which focused on trying to handle the class imbalance in the provided dataset:

Random Forrest with Class Weighting Vectorizer: CountVectorizer			
Accuracy:	Precision:	Recall:	F1 Score:
0.85	0.60	0.11	0.19

Gradient Boosting Vectorizer: CountVectorizer			
Accuracy:	Precision:	Recall:	F1 Score:
0.76	0.71	0.71	0.71

In the end, the most optimum solution we could find was the gradient boosting with CountVectorizer which gave an F1 score of 0.71 which is still very low and not accurate enough to be used consistently.

RAG Pipeline for Severity

As the results above from the models were not ideal, we decided to try to leverage the power of LLMs and use one to categorize the severity level. We hence made a RAG pipeline that used the data provided by A*STAR as corpus data. We first embedded the “description” column of the data provided and populated our corpus with that using the all-MiniLM-L6-v2 embedding model. Then the agent would take in the raw text of the incident and embed it using the same model, allowing for semantic search. It would then retrieve the severity levels of similar incidents and provide them as context for the LLM which is Claude 3.5 Haiku to categorize the severity of the incident either “minor”, “moderate” or “severe”. Below in Fig. 4 is the pipeline of the agent.

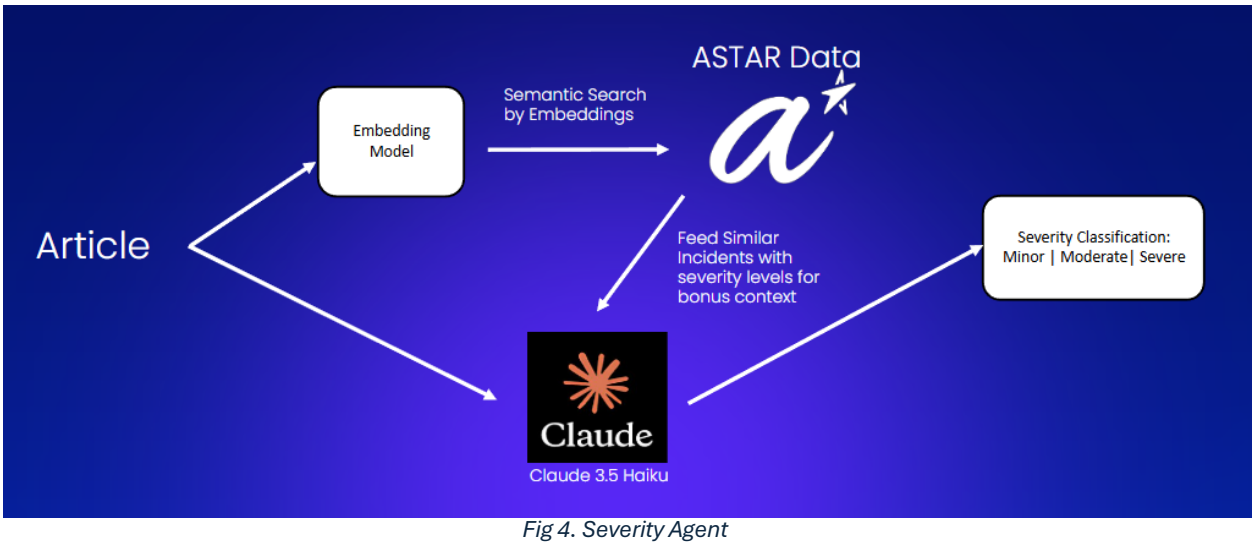


Fig 4. Severity Agent

We tested the agent against the sample data provided by the A*STAR and the results were not any better and the agent was especially terrible at predicting the “severe” category.

Category	Value
Overall Performance	

Sample Size	29
Overall Accuracy	0.517
Performance by Severity	
Minor: Accuracy	0.125
Minor: Cases	8
Minor: Correct Predictions	1
Severe: Accuracy	0.000
Severe: Cases	4
Severe: Correct Predictions	0
Moderate: Accuracy	0.824
Moderate: Cases	17
Moderate: Correct Predictions	14
Confusion Matrix	
True/Pred: Minor	Minor: 1, Moderate: 6, Severe: 1
Confidence Analysis	
Avg Confidence (Correct)	0.400
Avg Confidence (Incorrect)	0.643

Thus, we decided to assign weights to each category. To optimize these weights, we used Kfold cross validation. This partitioned the datasets and cross-checked to optimize the weights, and these were the results we landed upon.

Category	Value
Optimized Thresholds	
Minor Threshold	1.4587
Moderate Threshold	2.5644
Overall Accuracy	0.588
Per-class Performance	
Minor: Precision	0.514
Minor: Recall	0.679
Minor: F1-score	0.585
Minor: Support	28.0
Moderate: Precision	0.667
Moderate: Recall	0.612
Moderate: F1-score	0.638
Moderate: Support	49.0
Severe: Precision	0.333
Severe: Recall	0.125
Severe: F1-score	0.182
Severe: Support	8.0

The results were only slightly better. The overall accuracy increased from 0.517 to 0.588 which is not a significant improvement but at least the accuracy for the Severe category is no longer a 0.

Issues

In summary, we were not able to get a model that would be able to accurately predict the severity. We believe these could be due to one of 2 reasons:

1. Datapoints used in the classification of severity was not in the dataset provided to us
2. There were no clear metrics used during the classification of the severity which causes the model to be unable to identify the severity accurately.
3. Imbalance of training data provided as there were far fewer data points for severe maritime incidents.

As a result, we are unable to get a good model for this phase to be able to predict the severity based on the input text mainly due to poor data provided for the severity classification. Additionally, doing our own classification with our own data would not make sense here as they would need it to be classified into the different severity levels of their dataset. Hence the external validity of the model would not be good if we were to train it using our own data.

Phase 5 & 6 : LLM for Impact and Mitigation

In phase 5 & 6, the goal was to use large language models to assess the impacts maritime incidents could have on Singapore and to suggest possible mitigation strategies that can

help Singapore to lessen the impacts of such incidents. The model our group created is a retrieval-augmented generation (RAG) pipeline that allows users to load a pre-existing database of maritime articles or documents, retrieve the articles based on user queries and generate a structured response as a reply. Figure 5 is the planned outline of the RAG pipeline.

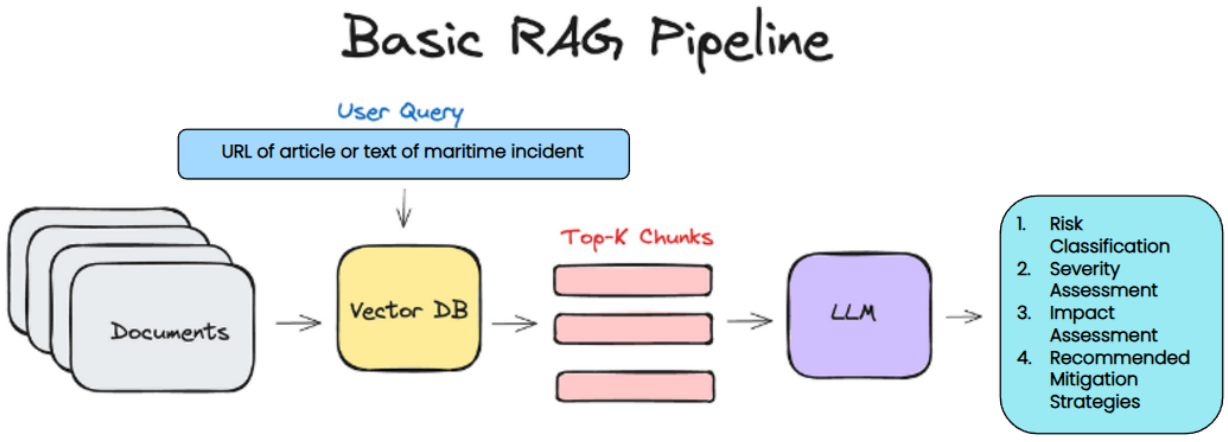


Figure 5. RAG Pipeline

Data Collection

To create a database of highly relevant articles pertaining to maritime incidents, we created a web scraping program using Jina.AI to scrape three main websites: Google News, Maritime Executive and gCaptain. To ensure relevance, we passed a list of queries as shown below. The Jina.Ai reader api will scrape the website and return the website in markdown format in a single string.

Queries: "Vessel delay", "Vessel accidents", "maritime piracy", "terrorism risk", "port congestion", "important route congestion", "port criminal activities", "cargo damage and loss", "inland transportation risks", "maritime environmental impacts and pollution", "natural extreme events and extreme weather maritime", "cargo and ship detainment", "maritime unstable regulatory and political environment", "Maritime port incident response

strategies", "Best practices in port accident mitigation", "Port risk management and incident prevention", "Recent port disruptions and recovery strategies", "Mitigation strategies for oil spills at ports", "Response to hazardous material spills in maritime ports", "Fire safety protocols in maritime ports", "Cybersecurity threats and response in maritime logistics", "Port infrastructure resilience during natural disasters", "Structural failure prevention strategies in ports", "Resilient port design to mitigate environmental incidents", "Use of AI in maritime incident prevention and response", "Port automation technologies for incident mitigation", "Predictive maintenance in port operations to avoid incidents", "Global policies on maritime incident response", "Port authority protocols for incident mitigation", "Impact of international maritime regulations on port safety".

There were about 2700 articles scraped which were then saved to a dataframe which is then converted to a csv file. One of the more important features of the webscraping script is that it allows the fetching of multiple articles simultaneously, making it more efficient. We also ensured that we did not scrape duplicated articles by cross referencing the urls of articles.

Creating the database

We then build a program to read and analyze the documents. We used several external libraries for this step:

1. Pandas
2. Json
3. Tqdm
4. Chromadb from langchain

5. HuggingFaceEmbeddings from langchain
6. RecurisveCharacterTextSplitter from langchain

We used pandas to read the csv and used json to parse the data so that it would be readable by the models. Using the RecursiveCharacterTextSplitter, we split the articles into chunks of 1000 characters with a 200-character overlap. We then used the HuggingFaceEmbeddings to generate embeddings for the chunks of text which are then stored in a vector database. For this project, we used Chromadb as the vector database and all-MiniLM-L6-v2 as the embedding model. This sentence-transformers model maps sentences and paragraphs to a 384-dimensional dense vector space and is used for clustering or semantic search. As we are dealing with semantic search, we have chosen this model.

Pipeline

We built a Retrieval Augmented Generation pipeline which retrieves relevant information from a knowledge base (Chromadb vector database) before generating a response to the user's query. The model searches for semantically similar documents using embeddings which ensures relevance. This step allows the model to provide factually and contextually accurate answers. We used many libraries from langchain, an open-source framework that helps to build LLMs:

1. ChatAnthropic from langchain_anthropic
2. Chroma from langchain_chroma

3. HuggingFaceEmbeddings from langchain_huggingface
4. RetrievalQA , ConversationalRetrievalChain from langchain.chains
5. ConversationBufferMemory from langchain.memory
6. PromptTemplate from langchain.prompts

We also integrated what we have done in Phase 3 and 4 in the pipeline. For the risk categorization, we pickled the models created and allowed it to take in the raw text of the incident as the input. The text of the incident would then be parsed into the clean_text function and then ran through the models, outputting a risk category. This would be then provided to the LLM in our RAG for further context provision.

For phase 4, we integrated the severity agent instead of the models as it gave slightly better results although it was still not ideal. So, similarly to how the base RAG pipeline works, the agent would take the raw text of the incident and embed it using the same model as the one for our vector database. It would then compare it to the A*STAR data's "description" column and fetch the severity levels. It would then categorize the severity level of the incident using the severity levels of similar incidents. The severity agent would then pass the severity level along with its sources to the main pipeline's LLM for additional context.

For the main pipeline, the 'MaritimeRAGPipeline' class which makes use of several libraries: ChatAnthropic, RetrievalQA, ConverstionalBufferMemory, ConversationalRetrievalChain which supports conversational AI and retains dialogue history. The model used in our RAG

Pipeline is the Claude 3.5 Haiku. The result of the pipeline is that it would take the risk category, severity level along with its sources and articles of similar incidents from the vector database, to suggest an insightful impact level and recommend mitigation strategies. Below in Fig 6. is a visualisation of the pipeline.

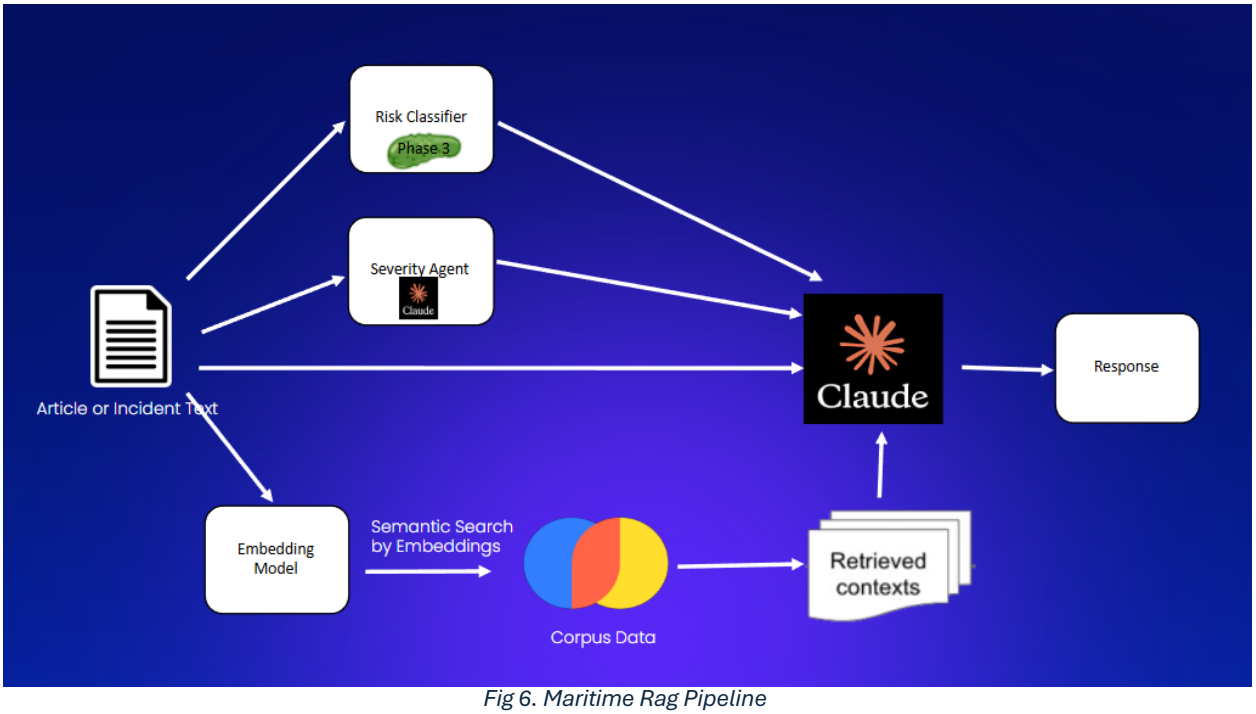


Fig 6. Maritime Rag Pipeline

Limitations and Improvements to the Model

For this project, one of the limitations of our model is that we are using a fixed chunking method for simplicity and to cut down on computational power. However, it might not be the best method of chunking as it can cut off sentences or paragraphs thus leading to a loss of context or coherence. This loss of context and coherence could negatively affect the pipeline. A better chunking method that could be used is semantic chunking where it takes the meaning of the information into consideration. Semantic chunking relies on language processing to detect boundaries and avoid cutting logical sequences, resulting in chunks that retain full context and

are better suited for natural language processing tasks. This method of chunking, however, is computationally intensive, and the enhanced context might not be worth the computational costs (Nayak, 2024).

Another limitation is vector search which compresses the information into a single vector which can cause information loss, this can lead to suboptimal results when the RAG pipeline is retrieving vectors related to the query. An example of suboptimal results could be that the vector documents retrieved is actually missing the relevant information. To combat this, we could add a reranker model which would rank the vector documents retrieved based on their relevance to the query. After all the documents that were retrieved are reranked, the model would then choose the most relevant to be used in answering the query. This can yield many benefits such as higher quality inputs for the LLM, higher relevance content matching and also reduced noise for the model. Thus, introducing a reranker model can increase the accuracy of our model and enhance the quality of the responses (*Rerankers and Two-Stage Retrieval*, n.d.).

Another improvement that could be made is to add a hallucination check in our pipeline to allow the model to reflect to ensure higher accuracy and relevancy of the pipeline's output. This makes sure that the output is consistent with the data retrieved. While we do make the pipeline out its confidence level, we believe that another method to critically examine its output and reiterate an answer would significantly improve the pipeline. We believe that G-eval (Liu et al., 2023) would be a suitable method for our pipeline as it allows us to customize the evaluation criteria

leveraging LLMs. LLM-based metrics generally outperform reference-based and reference-free baseline metrics in terms of correlation with human quality judgments.

The severity categorization could be significantly improved as well. As we already have an agent that can classify based on the corpus data provided, we believe improving the data provided would significantly improve results. Hence, we could explore collecting data and coming up with metrics to classify the severity levels of maritime incidents. After collecting, processing and classifying incidents, we could use them as corpus data for our severity agent.

Conclusion

The project tackles the critical need to improve resilience in global supply chains by leveraging Large Language Models (LLMs) to analyze and categorize maritime disruptions. With supply chains increasingly destabilized by external factors such as geopolitical tensions, natural disasters, and regulatory shifts, the project focuses on identifying and mitigating risks within maritime logistics. By combining robust classification models, advanced vectorization techniques, and the innovative use of retrieval-augmented generation (RAG) pipelines, we were able to achieve progress in risk categorization and impact analysis.

The project successfully categorizes disruptions, such as port closures, vessel delays, and environmental impacts, to help businesses adapt their logistics strategies. However, predicting severity levels, especially for "severe" disruptions, presented challenges due to imbalanced and

insufficient data. While approaches such as gradient boosting models and severity agents showed promise, they require further refinement to achieve higher accuracy and reliability. These challenges highlight the importance of robust datasets and clear metrics for defining and classifying incident severity.

Despite these limitations, the framework demonstrates significant potential to support proactive planning and reduce the cascading effects of maritime disruptions on supply chains. By tailoring strategies to specific risks, the solution can minimize delays and reduce costs, making it an invaluable tool for global supply chain resilience.

Looking ahead, addressing the limitations of each model will be key to enhancing the system's overall performance. For the logistic regression tree, improving its ability to handle more complex classification tasks will strengthen its effectiveness in multi-class scenarios. To improve severity predictions in the gradient boosting model, better handling of imbalanced data and rare cases, such as severe disruptions, is essential. The severity agent, meanwhile, requires clearer definitions of severity levels and higher-quality training data, which could be achieved through collaboration with industry stakeholders. Enhancing text analysis methods to capture deeper contextual meanings will also improve performance, particularly for severity predictions. Finally, creating comprehensive, well-labeled datasets and integrating real-time data from maritime technologies will ensure the system becomes more adaptable and responsive, making it an invaluable tool for global supply chain resilience. This project thus illustrates the critical role of

LLMs in mitigating vulnerabilities in critical global logistics networks and serves as a steppingstone to creating a more adaptive, data-driven framework for managing risks and ensuring resilience in global supply chains.

References

Andres, G. (2022, June 14). Malaysia bans chicken exports: What you need to know. CNA. https://www.channelnewsasia.com/singapore/malaysia-bans-chicken-exports-singapore-supply-price-consumers-2703071

Building resilient maritime logistics in challenging times. (2022, August 11). UNCTAD. https://unctad.org/news/building-resilient-maritime-logistics-challenging-times

Cao, X., & Lam, J. S. L. (2019). Catastrophe risk assessment framework of ports and industrial clusters: a case study of the Guangdong province. International Journal of Shipping and Transport Logistics, 11(1), 1-24.

How Singapore's Maritime and Port Authority is crafting the vessel management system of the future. (n.d.). https://govinsider.asia/intl-en/article/how-singapores-maritime-and-port-authority-is-crafting-the-vessel-management-system-of-the-future

Jidkov, V., Abielmona, R., Teske, A., & Petriu, E. (2020). Enabling maritime risk assessment using natural language processing-based deep learning techniques. 2021 IEEE Symposium Series on Computational Intelligence (SSCI). https://doi.org/10.1109/ssci47803.2020.9308441

Liu, Y., Iter, D., Xu, Y., et al. (2023, May 23). G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment. *ArXiv*.

https://arxiv.org/pdf/2303.16634

Mackenzie, A., Teske, A., Abielmona, R., & Petriu, E. (2021). Maritime Incident Information Extraction using Machine and Deep Learning Techniques. 2021 IEEE Symposium Series on Computational Intelligence (SSCI). https://doi.org/10.1109/ssci50451.2021.9659863

Magli, D. (2023, July 26). PSA implements AI-based digital solution in Singapore. Port Technology International. https://www.porttechnology.org/news/psa-implements-ai-based-digital-solution-in-singapore/

Ministry of Education. (n.d.). *Maritime*. MOE. https://www.moe.gov.sg/sgis/sponsoring-organisations/industries/maritime

Ministry of Trade and Industry. (2018, Jan 12). *Sea Transport Industry Transformation Map*. https://www.mti.gov.sg/-/media/MTI/ITM/Trade-Connectivity/Sea-Transport/Sea-Transport-ITM---Press-release.pdf

Ministry of Trade and Industry. (n.d.). *Free trade agreements*. https://www.mti.gov.sg/Trade/Free-Trade-Agreements

Nayak, P. (2024, April 24). Semantic chunking for RAG - the AI Forum - medium. Medium. https://medium.com/the-ai-forum/semantic-chunking-for-rag-f4733025d5f5

Ong, C. (2022, June 21). *Singapore imports more than 90% of its food. Here's how it's dealing with rising food inflation*. CNBC. https://www.cnbc.com/2022/06/21/singapore-imports-90percent-of-its-food-how-is-it-coping-with-inflation.html

Rerankers and Two-Stage retrieval. (n.d.). Pinecone. https://www.pinecone.io/learn/series/rag/rerankers/

Singapore's International Free Trade Agreements - Singapore Guide | Doing Business in Singapore. (n.d.). https://www.aseanbriefing.com/doing-business-guide/singapore/why-singapore/singapore-s-international-free-trade-and-tax-agreements

Singapore Business Review. (2024). *Singapore port congestion may persist until year-end.* https://sbr.com.sg/shipping-marine/exclusive/singapore-port-congestion-may-persist-until-year-end#:~=He%20said%20shipping%20lines%20have,of%20Singapore%20at%20Arthur%20D.

Teske, A., Falcon, R., Abielmona, R., & Petriu, E. (2018). Automatic identification of maritime incidents from unstructured articles. 2018 IEEE Conference. https://doi.org/10.1109/cogsima.2018.8423975

The Maritime Port Authority of Singapore Sets Sail with Its First Artificial Intelligence and Machine Learning Hub, Built On AWS. SG About Amazon. (2024, April 17). https://www.aboutamazon.sg/news/aws/the-maritime-port-authority-of-singapore-sets-sail-with-its-first-artificial-intelligence-and-machine-learning-hub-built-on-aws

The Straits Times. (2022, June 2). Malaysia bans chicken exports from June 1: How Singapore consumers, businesses are coping. *The Straits Times*. https://www.straitstimes.com/singapore/malaysia-bans-chicken-exports-from-june-1-how-singapore-consumers-businesses-are-coping

World Economic Forum. (2024, March). *Global cooperation and trade pacts* [Image]. World Economic Forum. https://www.weforum.org/agenda/2024/03/global-cooperation-trade-pacts/