Defence Research and Development Canada | Recherche et développement pour la défense Canada

CAN UNCLASSIFIED

DRDC | RDDC
technologysciencetechnologie

# Arctic Maritime Awareness for Safety and Security (AMASS)

*Final Report*

Scott Buffett, Colin Cherry, Chengbi Dai, Alain Désilets, Harry Guo, Daniel McDonald, Jiang Su, Dan Tulpan
National Research Council

Prepared by:
National Research Council
Project number: CSSP-2015-CP-2092
Technical authority: Brian Greene, Defence Scientist
Contractor's date of publication: September 2017

**Defence Research and Development Canada**
**Contract Report**
DRDC-RDDC-2017-C319
March 2018



Canada

**IMPORTANT INFORMATIVE STATEMENTS**

This document was reviewed for Controlled Goods by Defence Research and Development Canada (DRDC) using the Schedule to the *Defence Production Act*.

# Arctic Maritime Awareness for Safety and Security (AMASS)

# Final Report

**Prepared by:**
*Scott Buffett, Colin Cherry, Chengbi Dai, Alain Désilets, Harry Guo, Daniel McDonald, Jiang Su, Dan Tulpan*
*National Research Council*

Project: **CSSP-2015-CP-2092**

September 29, 2017

# Table of Contents

# 1. Introduction

The Arctic Maritime Awareness for Safety and Security (AMASS) project adapted, configured and demonstrated maritime domain awareness capabilities for ship detection, classification and tracking, route/destination prediction, suspicious vessel alerts and oil spill monitoring for exploitation by User Government Partners (UGPs), viz. the Canadian Coast Guard (CCG), Transport Canada (TC), the RCMP, Environment Canada (EC) and the Department of Fisheries and Oceans (DFO). The ship detection, classification and tracking capability included detection and classification of "dark" ships, i.e. vessels that do not self-report through services such as the Automatic Identification System (AIS), Long Range Identification and Tracking (LRIT) and the Vessel Monitoring System (VMS), but can nevertheless be detected in satellite imagery. Small vessels are generally dark ships because (with some exceptions) they have no regulatory requirements for self-reporting. In addition, we developed a capability for assessing a vessel's intent and threat level from open web sources. Thus the project supported Investment Priority #3, Arctic Domain Awareness, #9, New Surveillance Imagery Technology, and #28, Multi-Organization Decision Making, as well as Canada's Northern Strategy goals of exercising sovereignty, protecting the environment and improving and devolving Arctic governance. The project also implemented proof of concept interfaces to the Multi Agency Situational Awareness System (MASAS) and to Emergency Management systems capable of reading National Information Exchange Model (NIEM) messages, thereby supporting Priority #21, NIEM Interfaces for Emergency Management, and Priority #22, MASAS-Compatible Information Fusion for Emergency Operations Centres (EOCs).

Prior to the AMASS project, MDA Systems Ltd. (MDA) had already applied its expertise in radar and optical remote sensing satellite technology and intelligence data processing to develop the above maritime domain awareness capabilities under contracts from DND and the Canadian Space Agency (CSA). The Lead Government Department, NRC, had considerable expertise in developing Natural Language Processing (NLP) techniques for processing unstructured text, such as open web sources, applicable to assessing vessel intent and threat level. The project enhanced the maritime domain awareness capabilities above, with NLP-based capabilities to assess intent and threat level.

To achieve the project objective of developing capabilities to assess the intent and threat level of a vessel from open web sources, advances in the field of Natural Language Processing (NLP) were used to capture, process and present data that may be of interest to analysts or investigators. Two main categories of data sources were targeted: structured data, i.e. data that exhibits a high level of organization, such as a table, and free text, such as that seen in news articles, stories or blog postings.

The net result of these efforts is the development of a robust information extraction system that is capable of consuming both structured and unstructured data, collecting and identifying the data of interest and extracting and storing pertinent information of interest related to that data. This information then is represented in a form that allows quick and automatic updates provided to an analyst who might be investigating a particular vessel, port or person at sea, and could later be

consumed by a reasoning system to answer complex queries regarding nautical situations or circumstances that could be of interest to authorities.
To help guide project efforts toward producing relevant results, a number of User Government Partners (UGPs) (see section 6 for list) were directly involved at each stage of development. Six user conferences were held at regular intervals through the project to update the partners on recent achievements and to solicit feedback on results and future plans. Also individual interviews were conducted with each of the partners early on in the project to get a sense of their individual needs and a deeper look at their daily duties and challenges.

## 2. Background

# 2.1 Structured vs Unstructured Text Data

Data is said to be structured when it exhibits a high degree of organization, repetition, or otherwise has some predictable nature in which the data is presented or represented. Examples of structured data representations include lists, tables or markup languages such as XML and HTML. For example, Figure 1 shows a structured page that provides information about a single vessel. Here, the various pieces of data about the vessel (its name, its flag, etc.) can be located by looking at specific locations inside the page's Document Object Model (DOM) structure. The page in Figure 1 is one of approximately 200K pages on the same web site that provide information on different vessels, and a given type of data is always located at the same place in the pages' DOM tree.

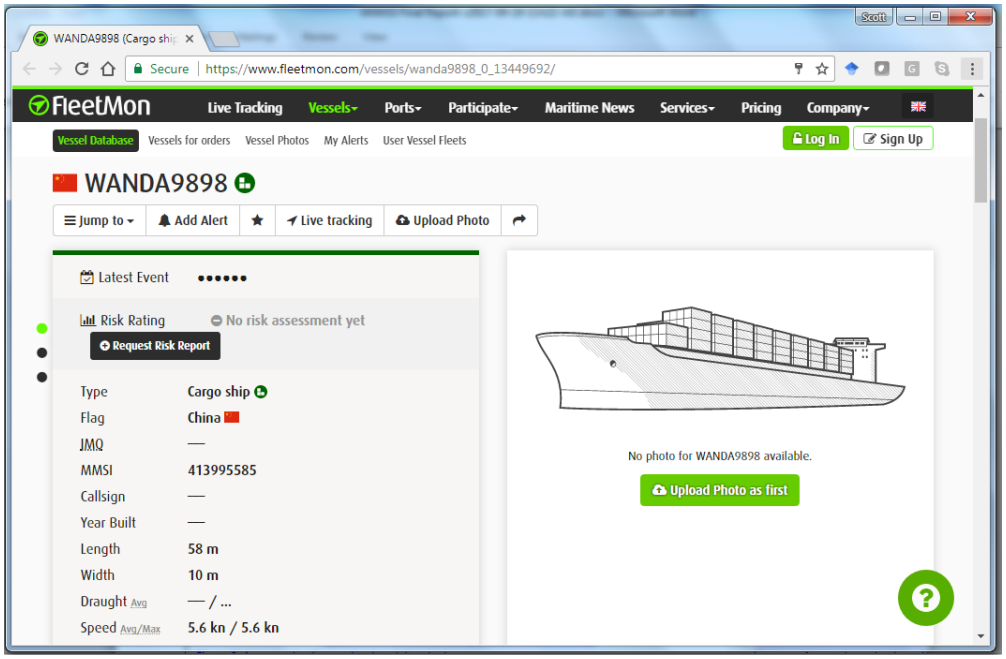



Figure 1: An example of a single entry structured page[1]

[1] Retrieved from https://www.fleetmon.com/vessels/wanda9898_0_13449692/ on 09/20/2017

Unstructured data, or *free text*, on the other hand, has no predefined data model. Processing free text is inherently more challenging as one cannot take advantage of regularities in structure to identify the desired data and extract relations between entities, but rather must rely on partial, noisy cues in the text. Figure 2 provides an example of a free text page, where information about the ship is buried in the text of the article. Although the same site may contain thousands of news items about vessels, we cannot rely on the pages' DOM structure to focus on information such as the ship that is being discussed, the places where events occurred, etc.

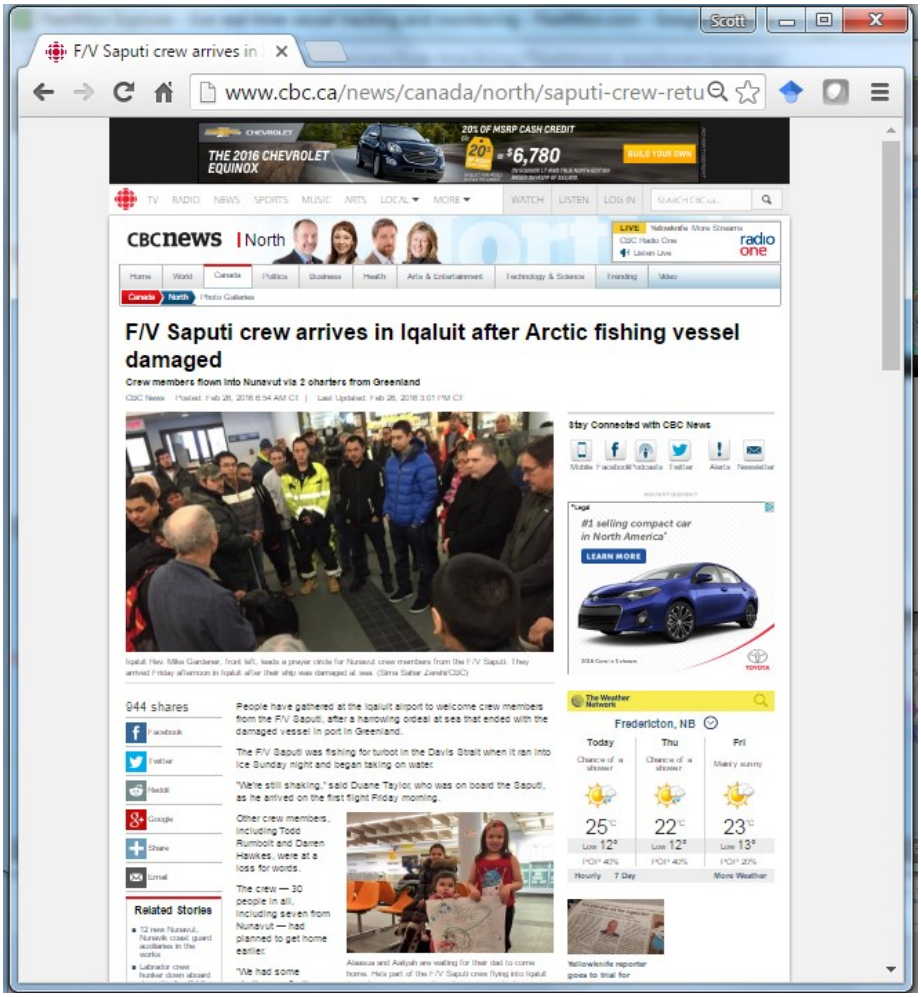


Figure 2: An example of an unstructured, free text page[2]

[2] Retrieved from http://www.cbc.ca/news/canada/north/saputi-crew-return-iqaluit-friday-1.3464995 on 09/20/2017

## 2.2 Web Scraping

Web scraping is a subfield of data scraping that focusses on extracting data in text-based markup languages such as HTML. These languages format the data in a way that enables it to be displayed for human consumption, which does not necessarily make the data's relational structure clear for machine consumption. Therefore, techniques such as DOM parsing or wrapper induction are often employed to overcome the associated challenges.

In the context of AMASS, interviews revealed that users commonly access structured pages from different sites to obtain the same kind of information. For example, ship detentions can be reported by different organizations located in different parts of the world. Unfortunately, different sites do not necessarily use the same labels for equivalent fields (for example: "Detained by" versus "Detaining Authority"), which makes it hard to conduct consolidated searches on scraped information that comes from different, yet equivalent sources. Performing this consolidation is a form of data fusion, using techniques such as schema matching or merging and table interpretation.

## 2.3 Information Extraction

Information extraction refers to the task of extracting structured information from unstructured text, such as that found in news articles, documents, etc., using techniques from the area of Natural Language Processing (NLP). The goal is to simplify text to create a structured view of the information that will be easily machine-readable, facilitating the ability to conduct logical reasoning to draw inferences from the data. Information Extraction is a broad topic. For the purposes of this project, we focus on the following subtasks:

- *Named Entity Recognition (NER)*: The NER process seeks to identify words or sentence fragments in the text that refer to an entity of interest, such as a ship, person, organization, place or date.

- *Entity Linking (aka Entity Grounding)*: Entities can be presented in different ways in the text (e.g. "National Research Council" and "NRC"), and thus must be recognized as referring to the same entity. Also, anaphoric links must be identified in order to "ground" sentence fragments to the absolute entity that is being referenced (e.g. that "the vessel" referred to a ship named "Catalina" or that "last Tuesday" referred precisely to August 29, 2017)

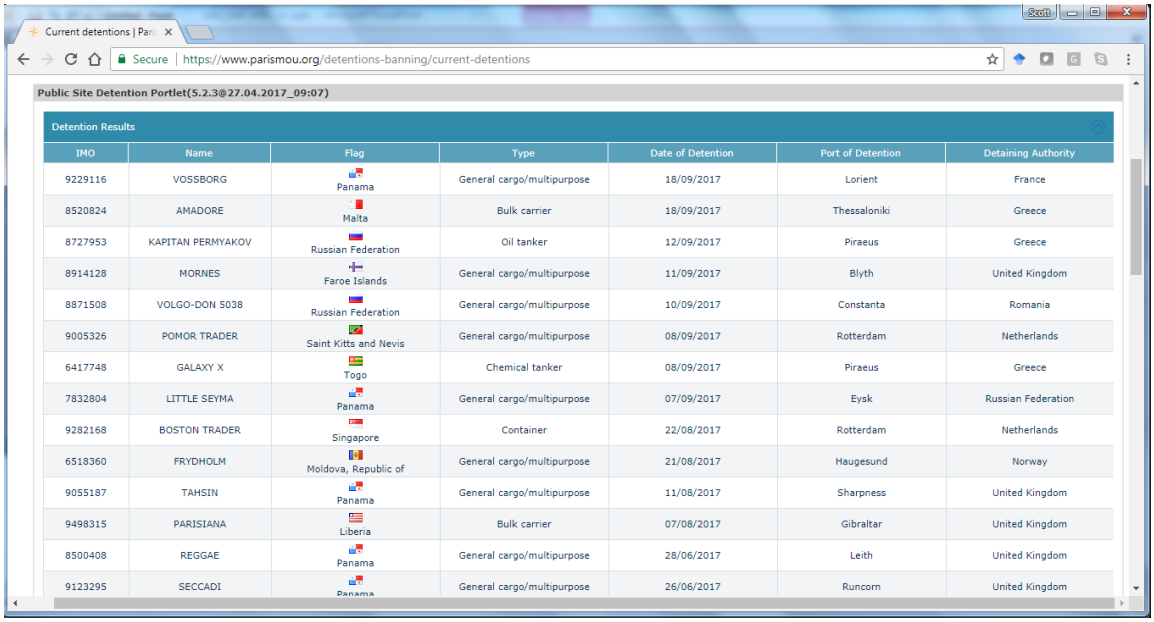
- *Relation Extraction*: Once the entities are identified and grounded, relation extraction is used to draw information amongst them. Relations often take the form of subject-verb-object triples yielding such information as Ship(Catalina)-arrived_at-Port(Montreal). This type of data can then be placed in a Resource Description Framework (RDF) store to enable downstream inference or retrieval.

Closed information extraction refers to the task of identifying relations among entities using a fixed, pre-established set of relation types. For example, the system may only be able to find Place Of Birth (relating people to places) and Employment relations (relating people to organizations). Open information extraction, on the other hand, attempts to identify any and all possible relations existing in the text, and often relies on the identification of verb phrases as the anchor for a relation between two entities in the same sentence.

## 3. Research Advances

# 3.1 Structured Data

To showcase the technology developed for structured data extraction, an adaptive web scraper was developed. The scraper can easily be trained by a user to extract whichever fields might be of interest, either from a series of pages with the same structure (Figure 1) or from a single page containing one or more tables where each row of the tables has the same structure (Figure 3). Unlike existing technologies that require the user to have existing knowledge of HTML in order to specify DOM paths to the information of interest, this web scraper allows a user to simply point to the desired fields, while the system then learns the DOM paths automatically. Besides being easier to use, this learn-by-example approach also results in DOM paths that are less brittle and more robust to small changes in the HTML structure of the pages over time. The scraper was packaged with a GUI, named “Kiliutaq”, and offered as part of the AMASS system.




| IMO | Name | Flag | Type | Date of Detention | Port of Detention | Detaining Authority |
|---|---|---|---|---|---|---|
| 9229116 | VOSSBORG | Panama | General cargo/multipurpose | 18/09/2017 | Lorient | France |
| 8520824 | AMADORE | Malta | Bulk carrier | 18/09/2017 | Thessaloniki | Greece |
| 8727953 | KAPITAN PERMYAKOV | Russian Federation | Oil tanker | 12/09/2017 | Piraeus | Greece |
| 8914128 | MORNES | Faroe Islands | General cargo/multipurpose | 11/09/2017 | Blyth | United Kingdom |
| 8871508 | VOLGO-DON 5038 | Russian Federation | General cargo/multipurpose | 10/09/2017 | Constanta | Romania |
| 9005326 | POMOR TRADER | Saint Kitts and Nevis | General cargo/multipurpose | 08/09/2017 | Rotterdam | Netherlands |
| 6417748 | GALAXY X | Togo | Chemical tanker | 08/09/2017 | Piraeus | Greece |
| 7832804 | LITTLE SEYMA | Panama | General cargo/multipurpose | 07/09/2017 | Eysk | Russian Federation |
| 9282168 | BOSTON TRADER | Singapore | Container | 22/08/2017 | Rotterdam | Netherlands |
| 6518360 | FRYDHOLM | Moldova, Republic of | General cargo/multipurpose | 21/08/2017 | Haugesund | Norway |
| 9055187 | TAHSIN | Panama | General cargo/multipurpose | 11/08/2017 | Sharpness | United Kingdom |
| 9498315 | PARISIANA | Liberia | Bulk carrier | 07/08/2017 | Gibraltar | United Kingdom |
| 8500408 | REGGAE | Panama | General cargo/multipurpose | 28/06/2017 | Leith | United Kingdom |
| 9123295 | SECCADI | Panama | General cargo/multipurpose | 26/06/2017 | Runcorn | United Kingdom |

Figure 3: An example of a multi-entry structured page

The web scraper that was integrated into the AMASS system falls in the category of *supervised* web scrapers, where a user is required to indicate to the system where the fields of interest reside on a sample page. This allows the system to learn how to scrape fields, and assemble them into untyped relations (called "Flex Relations"), based on typically 1-2 (in some cases more) user-provided examples.

The scraper was shown to be effective for two different models of structured data representation:

- *Single entity pages*: Each page contains a series of fields that provide information about a single "object" (e.g. a vessel). After being shown the fields on a single sample page, the scraper is then able to scrape those same fields on other pages with the same DOM structure. See Figure 1 for an example.

- *Multi-entry pages*: A page that contains one or more tables, where all rows have the same fields. After being shown the fields of two rows, the scraper is able to extract fields for all remaining rows on that page, and for any other page that contains tables with the same DOM structure. See Figure 3 for an example.

The system learns how to scrape a given field by generating a short list of xpath-like patterns which:

- Catch all the examples provided for that field
- Contain as few paths as possible
- Includes as many wildcards as possible, so as to be more resilient to changes in the DOM structure, and also to be able to catch other values of the field that have not been provided by the user (in the case of a multi-entry table page).

To generate flex relations, the system collects scraped fields together in the order that they appear in the DOM, until it meets a field whose name has already been included in the current relation (at which points it starts a new relation containing that one field). The scraper may also employ user-provided relation prototypes in order to decide when a new field signals the start of a new relation.

In addition to the supervised scraper that was delivered for AMASS, we experimented with two other approaches:

- **Label-only supervised scraper**: The supervised scraper that was delivered in AMASS (and described above) required the user to provide both the labels and values of the fields to be extracted. We also experimented with an approach where the user only needs to provide the names of the fields he wants to extract, as they are displayed on the page. Besides requiring fewer inputs from the user, this approach has the advantage of being even more robust to changes in the HTML structure of the page. In principle, the model learned by the scraper should continue to work as long as the labels displayed on the page for the various fields do not

change. Intuitively, we would expect those labels to change less frequently than the DOM structure.

- **Unsupervised scraper**: This scraper is able to extract label and values of all fields contained on a page, whether it be single-entry or multi-entry, without requiring the user to provide any examples. This system works by solving two subproblems: first it finds table structures on the page and converts them into a standardized format. Then it attempts to determine the orientation of the table; that is, is the table intended to have relational entities described in its rows or columns? It does so by taking advantage of the fact that the name (or header) of a field will tend to have a different statistical signature from the values of that field ("Province" versus "Nova Scotia" and "Ontario"). Because the system can quickly derive a page's structure from scratch, it is resilient to changes in the DOM over time. In other words, if the unsupervised scraper is able to successfully scrape a given page today, chances are that it will always be able to scrape it in the future. The downside of the approach is that, without any user provide clues about how to scrape a given page, it is more likely that the unsupervised scraper will never be able to scrape that page at all. It also requires the scraped table to have meaningful headers in order to determine orientation, as opposed to the supervised scraper, which can scrape tables without headers using information provided by a human.

We also implemented a simple Schema Merger, which is able to recognize when two fields scraped from different sites are actually referring to the same concept (e.g. recognize that the field "Detained by" scraped from one site means the same thing as the field "Detaining Authority" scraped from another site).

For each scraped field, the Schema Merging algorithm assumes that we have the following:

- The label assigned to that field by the person who trained the scraper
- A sample of at least 5 values for that field

The sample of values may have been scraped from several pages with the same structure (in the case of a single-entry page), or from different rows scraped from a single multi-entry page.

Using that information, the system creates a signature for the field, which takes the form of a short pseudo-document. Each line in the document corresponds to one of the sample values and has the following structure:

user-provided-label: value (value-characteristics)

Here, the user-provided-label is the name for the field that the user has provided at training time, the value is one of the actual sample values that was scraped for that field and value-characteristics is a semi-colon separated list of special tokens that capture key aspects of that value, such as:

- Total number of characters in the value
- The number of alphabetical and numerical characters in the value
- Whether the value can be parsed as an integer or floating point number
- Whether the value as a whole is a particular type of entity (Time, Place, Ship, Person), or might strictly contain such an entity

For example, a sample value of "2017-08-13" scraped for a field with user-given label "Arrival" would yield the line:

Arrival : 2017-08-13 (ContainsTime:TimeCoversWholeValue; )

The pseudo-documents for the various fields are indexed with Lucene[3], and when the user creates a new field while training the scraper, we retrieve the pseudo-documents that are most similar to that new feature's document, using Lucene's "More Like This" query. The most similar features would then be shown to the user in order to nudge him towards reusing existing labels whenever applicable. When two labels have been confirmed to mean the same thing, their pseudo-documents are merged and re-indexed in Lucene.

# 3.2 Unstructured Data

### 3.2.1 Named Entity Recognition (NER)

In the context of the AMASS project, NER serves three main purposes:

*Entity typing*: Entities are found both in free text and in structured pages, and their types can be used to provide semantic hooks into the AMASS database. For example, if the web scraper identifies a table row with three fields, one identifying a boat, another a date of detention, and another the location of detention, then correctly typing these fields enables a user to later issue a query like, "Find all boats detained in this region last month."

*Search space reduction*: NER reduces the search space for entity normalization or linking; that is, the task of transforming recognized names into canonical forms.

*Relation extraction*: NER provides valuable hooks for relation extraction, which can be viewed as a classification problem that attempts to identify the correct relation for pair of entities, where

[3] https://lucene.apache.org/core/

"No relation" is one of the classes under consideration. NER enables the training of the relation classifier, and indicates when the classifier should be used at runtime.

NER software was developed based on NRC tagging technology [1] for the AMASS project with a focus on the maritime domain. Six entity types were detected:

- Person: a named person, often captains, boat owners and journalists
- Location (called Place in the AMASS codebase): all non-port locations, including cities, countries and named buildings
- Organization: all non-shipping-company organizations
- Ship: a named boat or vessel, referring specifically to the vessel's name and not to another type of vessel identifier such as IMO or MMSI numbers
- Port: a known port, often sharing a name with a city
- Shipping Company: an organization that owns ships

Our NER tagger uses a Conditional Random Field model that is based on features such as:

- words contained in the region to be tagged
- words that precede and follow that region
- shape of the words (capitalized, contains digits, etc.)
- words that appear in a dictionary of known instances of a given entity type
- each word's membership at different levels of an automatically induced hierarchy of words, as suggested by Miller et al. [3].

To train the NRC tagger, two corpora of articles were collected. The first, pilot corpus was collected from the following sources:

- Fleetmon Newsroom, for articles on shipping companies and shipping accidents. Collected on October 29, 2015.
- Paris MOU News, for articles on vessel inspection and detainment. Collected on Feb 19, 2016.
- Port Metro Vancouver News, for articles on the events and politics that surround a port, and to introduce some Canadian content. Unfortunately, this turned out to be a fairly irrelevant source. Collected on Feb 18, 2016.

From a crawl of those three sources, we selected 212 articles that were estimated to have high entity density according to a preliminary automatic tagging scheme. These articles were humanly annotated for person, location, organization, ship, port, shipping company, coordinate, date, event, IMO number and port authority. Ultimately, only person, location, organization, ship, port and shipping company were identified as good targets for statistical tagging. Annotators were instructed to note any file-wise duplicates. Annotation was initially divided equally among the AMASS development team, and then one

team member was selected to resolve contradictions and inconsistencies across articles. After eliminating duplicates, we were left with 165 annotated articles.

Performance of the tagger on this pilot corpus is described in detail in section 5.2.

After some experimentation, we discovered that this pilot corpus was heavily biased towards:

- Pages that mention places and people in the Vancouver area (because one of the three sources was focused solely on that area)
- Pages that discuss the shipping industry in general, as opposed to news about specific ships (which is the type of news that UGP users are most interested in).

To avoid those issues, we created a second corpus using Fleetmon Newsroom and Paris MOU News, plus crawls of BC Shipping News, Cruise Critic, Marine Link, Noon Site, World Maritime News and Yachting World. Furthermore, we added Google News Items found through specific queries that the RCMP uses on a routine basis (according to the UICD), for example, Google News Search for "fishing vessels".

For each of these data sources, we manually selected stories for which we could tell that the news item was referencing a specific boat. This resulted in a total of 239 pages, which were then human annotated. These were divided into a 199-page training set, and a 40-page held-out test set. The test set was constructed to represent all crawled sources. Combining the new 199-page training set with our 165-page pilot set results in a total of 364 training pages.

Performance of the tagger on this second corpus is described in detail in section 5.2.

### 3.2.2 Time Expression Recognition

Recognizing time expressions in running text, such as “tonight”, “last Tuesday”, “September 25, 2017” or simply 09/25/2017 was handled by a specialized instance of our NER tagger, supplemented by a rule-based recognizer.

Our statistical tagger used the same features as described above, but with time-specific training data. As time recognition is a fairly well studied problem, and because time expressions do not tend to be domain dependent, we opted not to create our a new labeled training set, but instead extracted labeled time expressions from the version of TimeBank provided for the TempEval 3 evaluation. In doing so, we follow Bethard [4], who showed that a discriminative tagger trained on TimeBank alone outperforms the combination of TimeBank with other labeled corpora such as ACQUAINT. TimeBank provided us with 183 labeled documents, containing 2,659 sentences and 1,243 labeled time expressions. We later supplemented this with 33 AMASS-motivated sentences. Other than the use of this specialized training data, the only modification made to the tagger was to give it access to the features indicating which words were found to be parts of time expression by the rule-based recognizer described subsequently. The accuracy of our statistical recognizer is described in Section 5.2.

We built a rule-based time recognizer using regular expressions to supplement the statistical recognizer. These rules were designed to cover formulaic time expressions ("2017-09-25", "Monday, September 25, 2017") that were not well represented in the TimeBank training data. Crucially, these rules allowed us to react quickly to unanticipated date formats, without requiring us to create new training data and re-train the statistical recognizer.

### 3.2.3 Entity Linking and Normalization

Free text documents abound with ambiguous references to objects, for example: "the vessel", "on Sunday", "in Victoria". For the system to be able to reason about this information, or retrieve it in response to a user query, we need to ground those ambiguous references to a specific object.

The subsections below explain how this was done for ambiguous Time and Place as well as anaphoric Ship references. Performance of these algorithms is provided in section 5.2.

#### 3.2.3.1 Time Expression Resolution

Once a time expression has been recognized, it needs to be grounded to a canonical form to be useful to downstream applications. We carried out an extensive literature review of existing time expression parsing options, and eventually selected the Apache-licensed TimeNorm [5][4] library for its flexibility, accuracy and permissive license. TimeNorm uses a synchronous context free grammar to resolve time expressions. The use of a context free formalism makes the ruleset both compact and powerful, capable of handling arbitrary levels of nesting, i.e.: "the Friday the 13th following the 15th day of the 3rd month of 1985." We used the TempEval3 version of TimeBank, which not only annotates time expressions in text, but also provides gold-standard resolutions for those expressions, to confirm TimeNorm's resolution accuracy at 77% (assuming each ambiguous date appears in the past). TimeNorm does not resolve ambiguities in time expressions, but instead provides a list of feasible interpretations of the temporal expression, each in the TIMEX3 extension of the ISO8601 date format. It also requires a document-creation time to resolve expressions such as "tomorrow" or "next Thursday." This leaves us with the following tasks:

- Recognizing temporal expressions, described in Section 3.2.2
- Finding the document creation time (DCT), described below
- Selecting the correct interpretation from the list of feasible answers, most often selecting between a past or future interpretation of expressions like "Sunday" or "October 3," also described below.

For the first point, TimeNorm assumes that the DCT is known a-priori and will be fed as an input to the algorithm. But in the context of AMASS, our user interviews revealed that most of the documents that

[4] https://github.com/clulab/timenorm

they might feed into the free text components would come from open ended Google queries, and that the page's creation time is not provided as part of such query results.

To determine the Document Creation Date automatically, we therefore adopted the following heuristic:

> *Choose the first unambiguous date that occurs in the document (whether in the body or in the page's HTML headers). If this is the same date that the page was downloaded, then the next unambiguous date is chosen if one exists.*

The clause that prevents the download date from being chosen was introduced because many news websites display the current date at the top of the page's body, which often lead the system to mistake it for the DCT.

This simple heuristic turned out to be highly effective, as long as we are dealing with "yesterday's news" (i.e. news that has been published at least one day in the past relative to the Download Date). Again, our user interviews revealed that this was the most common situation in which users analyse news pages. See section 5.2 for performance evaluation.

To decide whether the date is in the future or past, we used a series of heuristics. Our single most powerful heuristic was to always select the interpretation in the past. This is right the vast majority of the time in typical news articles. Our next strongest heuristic was motivated by the fact that interpretations far in the past or future are less likely than closer interpretations. Therefore, we extended the "always past" heuristic to select the past interpretation unless the two options are a year apart, in which case we pick whichever interpretation is closer to our document creation time. Concretely, if a document mentions "September 1," then the two interpretations are in two different years; therefore, we select a past interpretation for documents written on September 3, and a future interpretation for documents written on August 29. We also experimented with heuristics involving the part-of-speech of nearby verb phrases, hoping indicators such as "will" and present tense verbs (English has no future tense) would allow us to identify future events. Unfortunately, errors in the part-of-speech tagger, coupled with inconsistent use of present tense in newswire articles, limited the effectiveness of this strategy.

Finally, we also implemented a suite of tools for interpreting and displaying the TIMEX3 dates and times that are returned by TimeNorm. This is non-trivial, as TIMEX3 intentionally retains the ambiguity inherent in verbal communications of time; for example, "summer" has different interpretations depending on whether you ask a teacher, a meteorologist, a Wiccan or an Australian, so TIMEX3 just notes that it was summer for some definition of the word. Therefore, we made several arbitrary choices to resolve these ambiguities in a consistent manner. Our tools allow us to reason about the start and end dates of resolved times, and to generate a human-readable description for a given ambiguous time expression. For example, given the expression "next Fall," TimeNorm may yield the TIMEX3 expression of 2015-FA, from which we derive the start and end dates of 2015-09-01 and 2015-11-30, and the description "Fall of 2015."

#### 3.2.3.2 Location Linking

Place names can often be ambiguous. For example, "Victoria" may refer to a city in BC or Hong Kong, as well as several locations spread over the former British Empire.

We implemented an algorithm that can ground such ambiguous location names to a specific entry in the GeoNames database[5]. This is a database of place names that provides additional information such as latitude-longitude, population, and granularity (e.g. country, state, county or city).

Given a particular mention of a place name in a free text document, the algorithm finds all likely matches for it in the GeoNames database, and sorts them according to the following criteria:

- Is the candidate place in Canada?
- Is the candidate place in North America?
- Is the candidate near the Arctic?
- Population of this place
- Geographical distance to neighbouring places in the text (e.g. in "Victoria, BC", choose the "Victoria" entry that is geographically closest to an entry with name "BC")

The relative weights of these criteria were adjusted manually in an ad-hoc fashion.

Note that while the first three criteria are specific to the context of AMASS, the last two are generic and could be applied in any situation.

#### 3.2.3.3 Anaphoric Reference to Ships

Maritime news abound with ambiguous references to ships such as: "The vessel arrived…". We implemented a simple heuristic that grounds these anaphoric references by resolving them to the named ship that lies closest to the anaphora in the text.

### 3.2.4 Relation Extraction

#### 3.2.4.1 Closed Relation Extraction

Closed relation extraction was conducted on the following set of fixed relations:

- SHIP ArrivesAt PLACE at time TIME

[5] http://www.geonames.org/

- SHIP DepartsFrom PLACE at time TIME
- SHIP WasLocatedAt PLACE at time TIME
- SHIP WasCaughtIllegallyHarvestingBy ORGANIZATION at PLACE and TIME

We explored two approaches for this.

The first approach was rule-based, and developed using Ruta[6], which is part of the UIMA[7] standard. Hand-crafted rules were created to deal with all of the above relation types. See section 5.2 for a report on performance.

The second approach, which was not integrated into AMASS, used a statistical classifier to determine if a particular sentence implies a particular type of relation. This was only implemented and tested for the the relation "SHIP SuspectedOfIllegallyHarvesting" relation type. This is not properly speaking a relation, as it only involves one entity, but we decided to treat it as 1-ary relation, with the "object" side of the relation set to null.

Our statistical classifier for the SuspectedOfIllegallyHarvesting relation uses a simple bag-of-words approach to determine whether a sentence includes a SuspectedOfIllegallyHarvesting relation. If it is, a relation is emitted for each vessel mentioned in the sentence. Future work will make it possible to tell which of the ships mentioned in a given sentence were actually suspected of illegal fishing. See section 5.2 for a report on performance.

#### 3.2.4.2 Open Relation Extraction

A statistical recognizer for open relations was developed based on the REVERB method for open information extraction [6]. The algorithm employs a two-pass approach. In the first pass, the system identifies short verb phrases, called "open relation anchors", which might imply some meaningful relationship between two specific entities. These are found using regular expressions over part-of-speech tags, which are designed to detect verb phrases ("was intercepted") and light verb constructions ("made a deal with"). The closest entities appearing before and after such an anchor are then attached to the anchor to generate a subject-verb-object relation candidate such as: "SHIP :: was intercepted at :: PLACE". In the second pass, we use statistical classification methods, trained on a labeled set of good relations, to assign a confidence score to each candidate relation. We developed two versions of open relation extraction for AMASS. The first was a straight re-implementation of REVERB, using the tag patterns provided in their paper for anchor detection, and using the features suggested in their paper in a maximum entropy classifier for confidence estimation. Our baseline version was AMASS-specific only in that it was trained on AMASS data, and it allowed relations to be extracted only from sentences where NER had recognized a Ship. In the next version, we modified both the anchor rules and the

[6] https://uima.apache.org/ruta.html
[7] https://uima.apache.org/

feature set based on our observations of how the method performed on AMASS data. In the remainder of this section, we will describe our confidence training data and the modifications implemented in version 2 of our REVERB implementation. Open relation extraction results are presented in section 5.2.

To train our confidence classifier, we ran NER, time expression recognition, and the REVERB anchor detection rules on 91 news documents found by querying the Bing Search API for maritime terms such as “vessel” or “illegal fishing.” For each anchor detected in each document, we collected its subject and object to create a complete relation triple. Each relation was then shown to a human judge, along with the sentence from which it was drawn. The judge was asked to decide whether the relation was correct or incorrect. Correct relations were defined to have correct (not misleading) and interpretable anchors, with the correct entities attached. Put another way, one should be able to read the subject-verb-object triple as a meaningful (but not necessarily grammatical) fact, which is entailed by the sentence from which it was drawn. A maximum entropy classifier was then trained to predict, for any proposed relation, the probability that it is correct.

We extended our REVERB implementation by improving both the rule-based anchor detector and the confidence classifier. The anchor detector was improved in two ways: first we restricted the anchors to disallow light verb constructions using nouns that were also detected as entities. This meant that we never lost a good relation because its subject or object was absorbed by its anchor. We made an exception for temporal entities, which were allowed to appear in anchors, enabling for example, “spent the last three months aboard.” Second, we further altered the light verb construction pattern to allow for internal prepositions. In the original REVERB patterns, light verb constructions had to end with a preposition but could not otherwise contain a preposition. We found that allowing internal prepositions made for longer, more meaningful anchors, extending for example “SHIP :: damaged by :: PLACE” to “SHIP :: damaged by fire in :: PLACE”. This extension was necessitated by the fact that we only allow detected entities to appear as subjects or objects of relations, while the original REVERB allows for any noun phrase to play a subject/object role. That is, in the example above, if “fire” had been detected as an entity of interest, we would not have needed to expand the anchor to include it.

We also extended our REVERB confidence classifier’s feature set. Most significantly, we changed REVERB’s features that look at sentence length to look instead at distance in words between subject and anchor; anchor and object; and the length of the anchor itself. The original REVERB was specifically designed to seek out simple (short) sentences, and extract high-confidence relations from them, while extracting relations from text from across the entire web. Because it saw so much text, it could afford to wait for simple sentences. In our setting, we are interested in getting as much actionable information as possible from a single document; therefore, we can not afford to wait for simple sentences; but instead, we had to seek out the clauses that match our simple patterns within otherwise complex sentences. Such clauses are characterized by short distances between anchors and entities. We also included a few lexicalized features, which down-weighted anchors that contained words unlikely to yield interpretable relations between our entities of interest. In particular, since we have no “statement” entity, any anchor headed by the words “says” or “said” is unlikely to yield a useful relation. Finally, to account for the fact that only certain noun phrases are allowed to participate in relations, we used part-of-speech features

to indicate when we passed over another (non-named-entity) noun when traveling from an anchor to a subject or object entity. It is possible that the passed noun was the true subject or object, which should reduce confidence.

## 3.3 Dictionaries

In our statistical NER tagger, we made use of different vessel dictionaries for the ship tagging. We started with the Lloyd's registry of ships: 51,776 vessel records, manually curated by Lloyd's of London, and provided to us by MDA. It includes name, IMO, MMSI, call sign, flag, and many other details of the ship.

First experiments using the Lloyds ship dictionary for NER purposes produced many false positives for ships. Error analysis showed that a large portion of those were caused by confusion with a place name or a fairly common word like "courage". Indeed, for every place of significance one can think of, there is usually a ship by that name. Similarly, most common words in the English language have ships by that name.

In order to minimise these types of false positives, we filtered the ship dictionary to remove:

- Any of the ports from our ports dictionary
- Any of the locations found in the GeoName database
- Any name that consists of a single word that is in Google's list of 20K most commonly used English words[8]
- We also augmented the Lloyds dictionary with 153,040 ship names scraped from the web site marinetraffic.com (done before carrying out the above filtering).

# 4. Technologies Developed

The following technologies were implemented in the project and either integrated into AMASS or prototyped for proof of concept, demonstrations or testing:

## 4.1 NRC Technologies

- UIMA wrappers around the web scraper and IE components, which allows them to be pipelined flexibly using that industry standard.

- A basic user interface for operating the web scraper, which requires users to enter the values and the occurrence numbers of fields to be scraped.

[8] https://raw.githubusercontent.com/first20hours/google-10000-english/master/20k.txt

- Schema Merger: Allows the system to tell a user when two fields coming from two separate scraper models refer to the same concept even though they have different user-provided names (e.g.: "Arrives in" versus "Destination").

- A user interface mock-up showing how the Schema Merger can be leveraged to nudge end users towards using existing field names (when appropriate) for new scraper models, instead of creating new names.

- MARS (MARitime Summarizer), a simple web demo that shows how the aforementioned named entity recognition and grounding capabilities can be used to provide maritime-oriented summaries of news.

- Mock-up for a Query Composer. Allows users to compose complex queries that leverage relations that were obtained from different web sources, whether they be free text or structured sources. The mockup was validated with end users, and eventually implemented by MDA.

- A real-time structured data parser and reporting system for pirate attacks grabbed from https://icc-ccs.org/piracy-reporting-centre/live-piracy-report. The following info is automatically extracted: attack number, date, lat/long, location, description. This scraper uses hand-crafted xpath rules to parse the pages.

- A ship info database and front-end for www.marinetraffic.com data. Displays: vessel name, IMO, MMSI, call sign, flag AIS type

- A real-time unsupervised rule-based ship info finder. Able to detect potential vessel names, IMO and MMSI from any URL.

## 4.2 MDA Technologies

- “Kiliutaq”, MDA's UI to the scraper, which allows users to simply point at fields to be scraped.

- Integration of scraped data into BlueHawk. Demonstrates how scraped data for a ship can be displayed on that ship's data card.

- Advanced, federated search, which implements a version of the NRC Query Composer mock-up.

- Collection of port visit data from AIS, and the Port Disclosure Report

- Webcams and user Insertion of vessel positions and images

## 5. Performance

# 5.1 Corpora and Tools for Test Bed Generation

- Corpus of 49 structured web pages that are known to be relevant to AMASS users. Each page comes with results that the scraper is expected to produce. Can be used to evaluate NRC's scraper algorithm, or any other competing alternatives.
    - For 9 of those pages, we also have human annotations that specify when a specific field from one page is equivalent to a specific field from another page. These pages can be used to evaluate NRC's Schema Merger algorithm, or any competing alternative.

- Human-annotated corpus of 384 free-text, maritime oriented news articles. Annotated with SHIP, PLACE, PORT, PERSON, ORGANIZATION, SHIPPING COMPANY, resulting in 12,773 entity labels over 9,798 sentences

- Human-annotated corpus of closed relations for the types mentioned in section 3.2.4.1.

- Human-annotated corpus of Open Relations. Captures the pairs of named entities that have a meaningful relation, as well as the phrases that would be appropriate to use as the relation anchor

- Human-annotated corpus for the closed relation type "SHIP SuspectedOfIllegallyHarvesting"

- Useful tools for collecting and annotating above corpora:
    - Tool for downloading results of a Google news query, and running them through the NRC IE pipeline, in order to produce a preliminary annotation to be corrected by humans.
    - Tool for exporting the results of the NRC IE pipeline to BRAT, a standard system for viewing and correcting annotations.

- 39,644 vessel records including name, IMO, MMSI, call sign, flag, AIS, tonnage, year built, lat/long, destination, last known port, draught, max/avg speed, deadwight, breadth, length, etc. Collected from www.marinetraffic.com.

- 153,034 vessel records include name and IMO. More info is available for each record (acquired) but not parsed. Collected from www.maritime-connector.com.

# 5.2 Results

### 5.2.1 Structured Data

5.2.1.1 Supervised Web Scraping

The scraper that was integrated into AMASS (label-value supervised scraper) was tested on 26 pages from our 49 web page corpus, collected from the project specification document and user interviews.

In 23 of those 26 cases, the scraper was able to successfully extract all the expected fields in the page (87% success rate).

Even for the 3 unsuccessful pages, the scraper was still able to extract most of the fields in that page. Our inability to scrape this handful of problematic fields is caused by the fact that those fields have "exceptionally inconsistent" formatting, As indicated by our performance statistics, such situations are relatively rare.

5.2.1.2 Unsupervised Web Scraping

The following table compares label prediction performance of the NRC scraper (PDT) with and open source scraper (IDE), and shows that the NRC scraper performs significantly better in terms of both precision and recall.

| Site | Size | Leaves | Test Pages | Percent | PDT-precision | PDT-recall | IDE-precision | IDE-recall |
|---|---|---|---|---|---|---|---|---|
| www.shipspotting.com | 3390000 | 429 | 9917 | 0.15 | **1.0** | 1.0 | 0.71 | 1.0 |
| www.shipais.co.uk | 211000 | 144 | 6228 | 0.30 | **0.61** | **1.0** | 0.47 | 0.96 |
| www.world-ships.com | 195000 | 181 | 4830 | 0.48 | **1.0** | 1.0 | 0.48 | 1.0 |
| directory.marinelink.com | 94800 | 121 | 9997 | 0.20 | 0.98 | 1.0 | 0.98 | 1.0 |
| www.wcpfc.int | 65300 | 398 | 4286 | 0.03 | **1.0** | 1.0 | 0.34 | 1.0 |
| www.marhisdata.nl | 50300 | 558 | 9994 | 0.01 | **0.99** | **1.0** | 0.0 | 0.0 |
| www.shipnumber.com | 39500 | 37 | 427 | 0.85 | **0.86** | **1.0** | 0.59 | 0.64 |
| www.shippingdatabase.com | 25200 | 183 | 66 | 0.42 | 0.82 | 1.0 | **0.84** | 1.0 |
| www.hafen-hamburg.de | 24700 | 265 | 1565 | 0.22 | **1.0** | **1.0** | 0.0 | 0.0 |
| www.ship-hunters.be | 18100 | 168 | 2951 | 0.47 | **0.86** | 1.0 | 0.85 | 1.0 |
| www.helderline.nl | 16100 | 204 | 9575 | 0.11 | **0.56** | **1.0** | 0.0 | 0.0 |
| www.shipphotos.ru | 5420 | 293 | 2858 | 0.07 | **0.42** | **0.97** | 0.22 | 0.84 |
| www.ferry-site.dk | 4680 | 218 | 4823 | 0.59 | **0.78** | **1.0** | 0.45 | 0.26 |
| superyachts.agent4stars.com | 3830 | 445 | 941 | 0.32 | **0.83** | **1.0** | 0.68 | 0.86 |
| www.tuapseport.ru | 3660 | 82 | 916 | 0.14 | **0.90** | 1.0 | 0.51 | 1.0 |
| ship-photo-roster.com | 1860 | 323 | 3173 | 0.33 | **0.89** | **0.99** | 0.75 | 0.97 |
| www.ivanegeriis.dk | 1560 | 74 | 664 | 0.34 | **0.97** | **1.0** | 0.89 | 0.88 |
| www.zeeschepenophetharingvliet.nl | 1040 | 329 | 418 | 0.16 | **0.76** | **0.76** | 0.08 | 0.28 |
| www.vanaalstmarine.com | 933 | 576 | 960 | 0.29 | **1.0** | **1.0** | 0.95 | 0.99 |
| www.globalcruiseship.com | 676 | 300 | 624 | 0.21 | **0.99** | 0.81 | 0.52 | **1.0** |
| csmru.com | 407 | 194 | 406 | 0.32 | **0.58** | **1.0** | 0.0 | 0.0 |
| sotonships.uk | 399 | 68 | 80 | 0.97 | 1.0 | 1.0 | 1.0 | 1.0 |
| mareud.blogg.se | 227 | 132 | 239 | 0.94 | 0.99 | **0.95** | **1.0** | 0.92 |
| Mean | 180638 | 248 | 3301 | 0.22 | 0.86 | 0.97 | 0.53 | 0.72 |

5.2.1.3 Schema Merger

We performed preliminary tests on the Schema Merger using 9 vessel monitoring web pages from different sites, taken from our corpus of 49 web scraping pages templates. These are pages that provide basic information about ships, such as: ship name, flag, type, registration numbers, last known location, etc. While most of the fields displayed on the different pages were equivalent to each other, the labels used differed from one page to the next (e.g. “Type”, “Ship Type”, “AIS Vessel Type”) as well as the format used to display the values (e.g. “10.0, “10.0 knots” or “10.0 kn” for speed) . Tests performed on this data shows that, when a new field is equivalent to a field that was previously seen, the Schema Merger is able to recognize it in 81% of the cases.

### 5.2.2 Named Entity Recognition (NER) Performance

Tagging results measured using 10-fold cross-validation on our 165 pilot documents are described in detail in [1], but that same report also describes how our pilot data draws from too few sources to be predictive of performance on general maritime news. Therefore, we will use this space to describe the performance of various iterations of our system on our held out test set of 40 documents that reflect diverse sources (described in section 3.2.1). We measure precision, recall and balanced F-measure. We will compare systems using these metrics micro-averaged across all entity types, and just for ships (the most impactful entity type). We test three systems:

- amass-tools 1.1.0: NER trained on our 165 pilot documents, and using Lloyd’s registry of ships without modification as a ship dictionary
- amass-tools 1.2.0: NER trained on our 165 pilot documents, and using an improved ship dictionary described in section 3.3 which filters place names and common words while adding ships scraped from maritime connector
- amass-tools 1.3.2: NER trained on our more diverse set of 364 documents, also using the improved ship dictionary from 1.2.0

The following comparison shows how entity detection in general and ship detection in particular improved over time:

| **Over all entities** | **Precision** | **Recall** | **F-Measure** |
|---|---|---|---|
| amass-tools 1.1.0 | 68.27 | 52.15 | 59.13 |
| amass-tools 1.2.0 | 71.60 | 54.41 | 61.83 |
| amass-tools 1.3.2 | 80.33 | 74.75 | 77.44 |

| **Ship entities only** | **Precision** | **Recall** | **F-Measure** |
|---|---|---|---|
| amass-tools 1.1.0 | 46.28 | 23.73 | 31.37 |
| amass-tools 1.2.0 | 54.61 | 35.17 | 42.78 |
| amass-tools 1.3.2 | 74.09 | 60.59 | 66.67 |

The performance of our final system (amass-tools 1.3.2) over all entities is as follows:

| Entity Type | Precision | Recall | F-Measure |
|---|---|---|---|
| PER | 92.24 | 93.02 | 92.63 |
| LOC | 82.87 | 87.75 | 85.24 |
| ORG | 61.35 | 50.29 | 55.27 |
| SHIP | 74.09 | 60.59 | 66.67 |
| PORT | 86.55 | 78.03 | 82.07 |
| SHIPCO | 77.27 | 38.64 | 51.52 |

### 5.2.3 Time Expression Recognition Performance

As described in Section 3.2.1, we trained a time expression recognizer on TimeBank data using our existing NER tagger. As estimated by 4-fold cross-validation on its training data, it achieves a precision of 84.78, a recall of 80.96 and a balanced F-measure of 82.83, which is comparable with the current state of the art. By design, the rule-based time expression recognizer, which supplements the statistical tagger, only recognizes date formats not present in TimeBank, so it was not formally evaluated.

### 5.2.4 Entity Linking Performance

5.2.4.1 Ambiguous Date Grounding

The heuristic for finding an article's Document Creation Date (aka Focus Date) had an accuracy of 93% over a test set of 100 articles authored prior to the download date, whereas accuracy was reduced to 66% for articles published on the download date. The decision was made to focus on older articles as this is what end users are more likely to look at.

To decide whether an ambiguous time expression, such as “Sunday” or “October 1” is in the future or past, we tested three heuristics:

1. Past: always pick the past interpretation.
2. Distance: as above, but choose the closest date to the document creation time when alternatives are a year apart.
3. Tense: as above, but select the future interpretation when the closest verb is in modal or present tense, and only when the future option would be closer to the document creation time.

Using TimeBank, we extracted 620 time expressions that were ambiguous with respect to past-future interpretations, along with labels for the correct interpretation (past or future). We used this data to evaluate the three heuristics described above. We gave the resolution code access to the original sentence and the gold-standard document creation time (both available from TimeBank). Heuristic 1, “Past” scored 84% accuracy, Heuristic 2, “Distance” scored 92% accuracy, while Heuristic 3, “Tense” scored 95% accuracy. It is difficult to measure statistical significance for rule-based methods, but given

how complex we had to make the "Tense" rule in order to see any improvement at all, we suspect that the difference between heuristics 2 and 3 is unlikely to generalize to new datasets.

5.2.4.2 Place Grounding

We evaluated the accuracy of our place-grounding heuristics on two sets of ambiguous place names extracted from our second free text corpus:

- 50 place names, about half of which occur near the Arctic
- A subset of 40 place names, which excluded bodies of water. This set contained 40% of place names that occur near the Arctic.

Our heuristic for grounding places to coordinates had an accuracy of 90% over the 50 places, and 92.5% over the 40 place set.

We compared our heuristic to the commercially available HERE API. We used HERE in different configurations:

- Providing only the text of the location as it appeared in the sentence.
- Restricting the results to be in Canada or US.
- Restricting the results to be in a rectangular region that loosely corresponds to the North American Arctic.
- Favoring Canadian locations while allowing other countries
- Combinations of the above

HERE Geocoding API Results:

| Settings | Accuracy (all Locations) | Accuracy (on Land Only) | Errors on Prominent Locations |
|---|---|---|---|
| Restrict to North American Arctic, and Canadian or U.S. locations | 52% | 65% | Most countries and non-North American Cities, Fredericton, Vancouver Island |
| Restrict to Canadian or U.S. locations | 56% | 67.50% | Most countries and non-North American cities, Vancouver Island, Arctic |
| Restrict to North American Arctic locations | 58% | 72.50% | St. John's, Russia, England, Paris, Fredericton, Vancouver Island |
| Restrict to North American Arctic, and preference for Canada (but locations in other Countries allowed) | 58% | 72.50% | St. John's, Russia, England, Paris, Fredericton, Vancouver Island |
| No restriction nor preference | 66% | 80% | St. John's, Vancouver Island, Arctic |
| Preference for Canada (but locations in other Countries allowed) | 68% | 82.50% | St. John's, Vancouver Island |

AMASS Location Disambiguation Results:

| Settings | Accuracy (all Locations) | Accuracy (on Land Only) | Errors on Prominent Locations |
|---|---|---|---|
| Text search only | 36% | 27.5% | Numerous mistakes: most countries, Toronto, Victoria, etc. |
| Preference towards Population | 82% | 87.5% | Victoria |
| Preference towards population and Canadian locations | 88% | 90% | |
| Preference towards population and Canadian or U.S. locations | 88% | 90% | Arctic |
| Preference towards population and within North American Arctic | 88% | 90% | Victoria |
| Preference towards population, Canada, and within North American Arctic | 90% | 92.5% | |
| Preference towards population, Canada or U.S., and within North American Arctic | 90% | 92.5% | Arctic |

To summarize, the best result from HERE was 68%. Seeing how poorly HERE performed on bodies of water, further tests were conducted over a smaller test set with bodies of water removed, and AMASS had an accuracy of 92.5% over HERE's best result of 82.5%.

### 5.2.5 Relation Extraction

5.2.5.1 Closed Relations

The Rule-based Closed Relations Tagger was tested on a corpus of 32 articles. Testing based on perfect NER tagging gives an F1 score of .346, while testing based on the AMASS NER tagging gives an F1 score of .235, though we did not implement some of the relation types expected in the corpus, which leads to lower scores.

The Statistical Closed Relation Tagger approach was tested for a single relation type: SuspectedOfIllegalFishing. The tagger was trained on a corpus of news story collected through the following Google news query:

> Fishing vessel AND (illegal OR unreported OR unregulated OR iuu OR ghost OR “dark ship” OR crime OR criminal OR supiscious OR suspect OR Interpol OR law OR court OR plead OR guilty OR violation OR violate OR intercept OR seize OR seized OR detain OR detention OR warning shot OR gunfire OR arrest OR capture OR “without a license”)

A certain number of the resulting stories were passed through our statistical NER tagger, and any sentence that contained at least one ship was included in the training set. This resulted in 3074, of which, 467 were positive examples of a SuspectedOfIllegalFishing relation.

We trained on 90% of the dataset, and evaluated the model on the remaining 10%. The tagger achieved an accuracy of 94% on the retained dataset.

5.2.5.2 Open Relations

We carried out two evaluations to measure the performance of our open relation extraction, one quantitative and one qualitative.

The quantitative evaluation was based on 24 documents: 19 derived by searching the web for news about boats deemed of interest to MDA, and 5 derived by searching the web with RCMP-inspired queries and manually selecting interesting articles mentioning individual boats. These documents were then labeled using an intermediate version of our open relation extractor, and its output was transformed into a format to be displayed and corrected in the BRAT annotation tool. One human annotator (the lead NRC researcher) looked at each document, and proceeded to make whatever changes to the annotation were necessary to extract what he deemed to be correct open relations. This could include changes to the NER, the time expression recognition, the anchor detection algorithm, the heuristic that attaches entities to anchors, and the anchor confidence component. We then compared the relations extracted by two versions of our AMASS tools to this gold standard, deeming a relation to be correctly extracted only if it exactly matches both entities (checking the entity label and the location in the text) and the relation (checking the anchor's location in the text). This is a very strict evaluation metric, therefore we expect scores to be low. We compared two systems:

- amass-tools 1.3.1: NER as in amass-tools 1.2.0 (pilot training data with improved dictionary) and using our direct re-implementation of REVERB for open relations
- amass-tools 1.3.2: NER as in amass-tools 1.3.2 (diverse training data with improved dictionary) and using our extended version of REVERB, described in Section 3.2.4.2

The following table summarizes the results of this comparison:

| **AMASS version** | **Precision** | **Recall** | **F-Measure** |
|---|---|---|---|
| amass-tools 1.3.1 | 25.00 | 15.75 | 19.32 |
| amass-tools 1.3.2 | 51.56 | 51.16 | 51.36 |

The second, qualitative evaluation was based on a set of the documents obtained by taking the news articles from the search results of running the following query through the MARS UI:

"ship name" AND '(ship OR boat OR vessel)'

where “ship name” was an MDA boat of interest. We then looked at the Open Relations extracted by AMASS for each document, and where appropriate, comment on whether or not the primary, or most important relation from a document was extracted. Most news documents have one main event that they are centred around, and this was designed to test whether we are getting that event. We compared amass-tools 1.3.1 and 1.3.2 with this methodology, resulting in the following comparison, where the table shows how many of the 9 ships identified in the search fall into each error category:

| Category | 1.3.1 count | 1.3.2 count |
|---|---|---|
| Main relation correctly extracted | 1 | 4 |
| Correct relation and entities, but wrong ship linked to unnamed ship | 1 | 1 |
| Correct entities, incorrect anchor | 1 | 1 |
| Incorrect entity, correct anchor | 1 | 1 |
| Main relation not generated | 5 | 2 |

## 6. User Interviews/Conferences

NRC and MDA worked closely over the course of the AMASS project with five Canadian government departments who acted as User Government Partners (UGPs):

- Canadian Coast Guard (CCG)
- Transport Canada (TC)
- Royal Canadian Mounted Police (RCMP)
- Environment Canada (EC)
- Fisheries and Oceans (DFO)

Six user conferences were held at regular intervals throughout the project to update the UGPs on recent achievements and to solicit feedback on results and future plans:

- Nov 2015 (NRC Ottawa)
- Mar 2016 (MDA Richmond)
- Jul 2016 (Online)
- Nov 2016 (NRC Ottawa)
- May 2017 (MDA Richmond)
- Aug 2017 (Online)

Individual interviews were conducted with each of the partners early on in the project to get a sense of their individual needs and a deeper look into their daily duties and challenges. These interviews were conducted on:

- Mar 15, 2016 (Group interviews: RCMP, TC, EC, DFO, CG)

- Mar 16, 2016 (Individual interviews: RCMP, EC, CG, DFO)
- July 12, 2016 (CG)
- July 20, 2016 (TC)

## 7. Outreach

Presentations:

- *AMASS*, Canadian Forces Maritime Warfare Center, CFB-Stadacona, Halifax, NS, November 2, 2016, Op LIMPID meeting.
- *AMASS*, Presentation to the Mayor of Fredericton, NB, May 8, 2017.
- *AMASS in a Nutshell*, Presentation to the Canadian Forces Joint Operations Command, September 7, 2017.

## 8. Conclusions

This project has built, tested and validated several tools that enable MSOC analysts to leverage relevant maritime information that is publicly available on so-called Open Web Sources.

We built a lightly supervised scraper that can learn to extract information from most relevant structured web sites, with just a few user-provided examples. The scraper also helps users standardize the extracted information so that information scraped from different sources can be consolidated, thus enabling cross-source searching.

We also built a solid free text Information Extraction pipeline that can extract structured facts from unstructured news texts. The system is able to recognize maritime-relevant entities such as: ships, places (ports in particular), organisations (in particular shipping companies), people and times with reasonable accuracy. It can also ground those entities to unambiguous identifiers (e.g. latitude-longitude for places, YYYY-MM-DD dates for times).

The free text Information Extraction pipeline is also able to identify relations between entities, whether those relations are of a type known a-priori (so-called Closed Relations) or not (so-called Open Relations). On the Closed Relation front, we built and tested a Statistical Classifier that achieved 94% accuracy on a single relation type (SuspectedOfIllegalFishing). On the Open Relations front, we developed a system that identifies verb-phrases that capture the nature of any entity relation with 52% precision and 51% recall.

Finally, MDA and NRC built several user-facing functionalites and demos that show how information extracted with the above tools can be useful to customers.

As a result of the project, one UGP partner (DFO) has expressed interest in licensing the BlueHawk system to deploy to its users.

## 9. Future Directions

Directions for future research include the following:

Current, Label-Value supervised Scraper

- Deal with optional fields in tables
- Deal with relations that don't follow a flat table structure
    - Multiple tables on same page, each table being prefixed with some additional data (ex: multiple tables, each prefixed with a ship name, where the table specifies the itinerary for that ship)
    - Tables within tables
    - Tables where the number of columns is variable (e.g. a table specifying the quantity of fish caught for different species, where the list of species is provided as column labels, and can vary from page to page. In this case, we need to scrape the species labels, whereas normally, the labels fixed and provided by the user instead of scraped)

Label-only supervised Scraper

- Evaluate the Label-only supervised scraper on our 49 sample web pages.
- If it performs almost as well as the Label-Value supervised one, then it may be a better option as it will be even more resilient to changes in page structure.

Unsupervised Scraper

- Likewise, try it on our 49 sample pages, and if it is just as good, go for that one as it will be pretty much 100% robust to changes in the page layout.

Scraper Self-repair

- If neither the Label-only supervised scraper nor the unsupervised scraper work well on our sample of 49 pages, we need to come up with a self-repair strategy for the Label-Value supervised scraper.

Schema Merger

- Evaluate on a larger and more diverse set of pages
- Evaluate the accuracy when we accept any of the top N suggestions (e.g. N = 5) as being correct.
- Make the system able to say "this new field does not exist in the current schema". At the moment, we only evaluate the system in situations where there exists a field in the current schema. In all other cases, we assume that the system giving some suggestions when it should not is not bad (the user just has to ignore them).

Ship recognition

- Improvements on performance

Download date

- We need to do a better job because although accuracy is fairly high, when an error occurs, the date can be off by as much as 12 months.
    - Train a statistical classifier to recognize if a date is the focus date based on features like:
        - Rank of the date in the list of all dates found
        - Was this date in the document headers (HTTP or HTML) or in the body?
        - Absolute offsets of the date
        - Relative offsets (i.e. normalized by document length)
        - Words in the date, and those that precede and follow it

Anaphoric ship references

- Use a co-reference algorithm to resolve those.

Closed Relation Tagging

- Test the Statistical Closed Relation Tagging approach on other types of relations such as: ArrivesAt, DepartsFrom, SuspectedOfSmuggling, DetainedIn.

Open Relations

- Improve Precision and Recall beyond the current 0.50 values, by employing
    - Open-anchor simplification and generalization
    - Non-continguous anchors
- Open-anchor clustering to identify anchors that describe the same type of relation using different wordings

User Studies on Deployed System

- Should one or more of the UGPs purchase the AMASS system, it would be interesting to do a user study to see how they use the system, and how it could be improved to better meet their needs.

**References**

[1] Cherry, C. 2016. Named Entity Recognition for Maritime Domain Awareness. NRC Technical Report.

[2] Su, J., Cherry, C. and Guo, H. 2016. Word Frequency for Web Table Understanding. NRC Technical Report

[3] Miller, S., Guinness, J., & Zamanian, A. (2004). Name Tagging with Word Clusters and Dis- criminative Training.. In Proceedings of the Human Language Technology Conference of the NAACL (HLT-NAACL), pp. 337–342.

[4] Bethard S. ClearTK-TimeML: A minimalist approach to TempEval 2013. InSemEval@ NAACL-HLT 2013 Jun 14 (pp. 10-14).

[5] Bethard S. A Synchronous Context Free Grammar for Time Normalization. InEMNLP 2013 (pp. 821-826).

[6] Fader A, Soderland S, Etzioni O. Identifying relations for open information extraction. InProceedings of the Conference on Empirical Methods in Natural Language Processing 2011 Jul 27 (pp. 1535-1545).

## DOCUMENT CONTROL DATA

*Security markings for the title, authors, abstract and keywords must be entered when the document is sensitive

1. ORIGINATOR (Name and address of the organization preparing the document. A DRDC Centre sponsoring a contractor's report, or tasking agency, is entered in Section 8.)

National Research Council Canada
https://www.nrc-cnrc.gc.ca/

2a. SECURITY MARKING (Overall security marking of the document including special supplemental markings if applicable.)

CAN UNCLASSIFIED

2b. CONTROLLED GOODS

NON-CONTROLLED GOODS
DMC A

3. TITLE (The document title and sub-title as indicated on the title page.)

Arctic Maritime Awareness for Safety and Security (AMASS): Final Report

4. AUTHORS (last name, followed by initials – ranks, titles, etc., not to be used)

Buffett, S.; Cherry, C.; Dai, C.; Désilets, A.; Guo, H.; McDonald, D.; Su, J.; Tulpan, D.

| 5. DATE OF PUBLICATION (Month and year of publication of document.) | 6a. NO. OF PAGES (Total pages, including Annexes, excluding DCD, covering and verso pages.) | 6b. NO. OF REFS (Total references cited.) |
|---|---|---|
| September 2017 | 31 | 6 |

7. DOCUMENT CATEGORY (e.g., Scientific Report, Contract Report, Scientific Letter.)

Contract Report

8. SPONSORING CENTRE (The name and address of the department project office or laboratory sponsoring the research and development.)

DRDC – Centre for Security Science
Defence Research and Development Canada
222 Nepean St., 11th Floor
Ottawa, Ontario K1A 0K2
Canada

| 9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.) | 9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.) |
|---|---|
| | |

| 10a. DRDC PUBLICATION NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document.) | 10b. OTHER DOCUMENT NO(s). (Any other numbers which may be assigned this document either by the originator or by the sponsor.) |
|---|---|
| DRDC-RDDC-2017-C319 | |

11a. FUTURE DISTRIBUTION WITHIN CANADA (Approval for further dissemination of the document. Security classification must also be considered.)

Public release

11b. FUTURE DISTRIBUTION OUTSIDE CANADA (Approval for further dissemination of the document. Security classification must also be considered.)

| 12. KEYWORDS, DESCRIPTORS or IDENTIFIERS (Use semi-colon as a delimiter.) |
| --- |
| Maritime Domain Awareness; Natural Language Processing |
| 13. ABSTRACT/RESUME (When available in the document, the French version of the abstract must be included here.) |
| |