A case study on automated risk assessment of ships using newspaper-based event extraction

Jesper Hoeksema[1] and Willem Robert van Hage[2]

[1] Computer Science, Network Institute
VU University Amsterdam
De Boelelaan 1081a, 1108 HV
Amsterdam, The Netherlands
J.E.Hoeksema@vu.nl
[2] SynerScope B.V.
willem.van.hage@synerscope.com

Abstract. In this paper we describe an event-type extractor on top of a distributed search engine. We apply this event-type extractor in a case study concerned with assisting maritime security operators to assess potential risk factors of ships. Based on a corpus of maritime-related press releases we automatically investigate the history of ships as they enter an area of interest. The performance of the system is evaluated with a task-oriented focus on a set of vessels with known risk factors, and typical behaviour is evaluated by batch-processing a large set of vessels.

1 Introduction

In many safety and security-related tasks it is necessary to quickly investigate the background of an object under surveillance in order to see if its history raises any red flags. In this paper we analyse how a combination of techniques from event extraction, information retrieval, text processing and background knowledge can be used to support this task. Our aplication domain is maritime safety and security in the Dutch coastal area.

On average every thirty seconds a vessel leaves or enters the Netherlands Exclusive Economic Zone, an area of 154.011 km^2 in front of the Dutch Coast.[3] The Dutch coastguard employs 51 full-time operators who continuously monitor this area (which typically contains at any point in time around 1300 to 1400 ships) in order to predict and hopefully prevent events that threaten the law, the environment, or public safety. The current generation of naval vessel observation systems process most information retrieved from readily available sources automatically, such as vessel positions from radar and information broadcasts by the vessels themselves, and project this information on a map view to the operator. These systems, however, do not take any information from outside sources into account, such as news articles and other public information. If an operator wants to know more about a certain vessel, he or she has to search for this information manually, often on a second computer. This means that an

operator is not able to fully investigate all vessels in the area of interest, as the number of ships coming and going is too large to process manually in (near) real-time. Currently, operators circumvent this problem by prioritizing the ships to investigate, using data that is immediately accessible, such as their previous port of call, bearing, name and cargo, and only further investigating the vessels with the highest priorities.

As this process of elimination is inherently incomplete due to the fact that not all available information is taken into account, potential threats could possibly slip through. We propose a full prioritization by automating parts of the initial investigation using a combination of techniques that tie into the current state-of-the-art vessel observation systems. The focus of our research is to explore the possibilities of using a combination of relevance feedback, lexical databases and domain information to perform event type detection in the context of the surveillance task assigned to a maritime security operator. This means we aim to minimize the number of false negatives detected by the system. Due to the fact that a detected threat will not result in automatic actions being taken, but rather in an alert to the operator, minimizing false positives has a lower priority, as long as the number of alerts stays within a manageable rate. It is also important that the operator is able to trace back to the sources of the information that triggered an alert. An assumption in the every day work of such operators is that ships with a record are more likely to be involved in subsequent similar situations. This can be due to many, sometimes complex, causes, possibly having to do with the motivations of the crew or the owners, but in any case the correlation between a shady history and future trouble exists.

Throughout this paper we will use the (fictional) running example of the Very Large Crude oil Carrier Sirius Star entering the Dutch coastal waters. This ship has been involved in hijacking, kidnapping of the crew, parliamentary debate about they payment of a ransom sum, and participant in a lengthy and tumultuous aftermath of these events. We choose this ship and its history as an example, because many more suitable cases touch upon sensitive information, which we want to avoid, and yet this example has a clear press coverage. This would make it easy for us to detect the event descriptions in text, and therefore it sets a good lower bound on the performance we need to demonstrate.

This paper is structured as follows. We first discuss related work in Sec. 2. Then, in Sec. 3, we describe the composition of our system and the methods used. Sec. 4 provides a description of the setup we used to evaluate our system. Sec. 5 describes the results after using the system with a sample data set. Finally, these results are discussed in Sec. 6.

2 Related Work

This work falls inside the domain of the application of computational linguistics and information retrieval to the task of structured event extraction. A lot of existing research in these domains have been done using traditional NLP pipelines,

such as Gate [2] and Kyoto [9], that would require processing each document in the corpus first, before being able to say something about the history of a ship.

Atkinson *et al.* [1] state that news items, in particular from online media, are particularly interesting to exploit for gathering information about security-related events. They argue that information on certain events might not be available through other (official) sources, and even if they were, official sources often have a significant delay. They continue by presenting two approaches to extracting events related to border security events from on-line news articles: (i) a cluster-based approach looking at the title and first sentence of multiple articles at once, and (ii) an approach processing a single document at a time. These approaches both try to match specific patterns of words to the text, exploiting the fact that news articles are often written in a distinctive style. Variables inside these patterns are then filled to find the various properties of an event. Both these approaches use the articles themselves as starting point, thus requiring to pre-process all articles as they come in.

Turney *et al.* [7] stress that leveraging the respresentation of documents as term vectors, as used by many search engines, is a powerful paradigm that should be employed in many AI-related topics, such as word sense disambiguation, word clustering, spelling correction, and information extraction. The Term Saliency module in our system is an application of this paradigm, using term frequencies represented as vectors to find salient words, rather than employing natural language processing techniques.

3 Approach

Our implementation architecture, shown in Fig. 1, consists of a set of custom-built modules, an installation of WordNet, a modified ElasticSearch[3] cluster, and a database of ship names. The ElasticSearch cluster is filled with a corpus of approximately 25,000 maritime-related press releases. We will first provide an overview of the general system, and then proceed to describe its components in detail in the following subsections.

All international vessels above 300 gross tonnage, all national vessels above 500 gross tonnage, as well as all passenger ships are required to broadcast their position, name and destination using the Automatic Identification System[4] every few seconds. This, along with radar information allows the coastguard's vessel observation systems to track their position. Whenever a vessel enters the area of interest defined by the operator, an event is fired by the vessel observation system, depicted by *Mission Management* in Figure 1. This event is picked up by our risk assessment system.

From that event, a query is formulated by the *Query Builder* to retrieve relevant documents about that ship, taking special care to exclude ships with similar names. This query is fired against the ElasticSearch cluster twice by the *Term Saliency* module - once for retrieving a set of documents relevant

[3] http://www.elasticsearch.org/
[4] http://en.wikipedia.org/wiki/Automatic_Identification_System

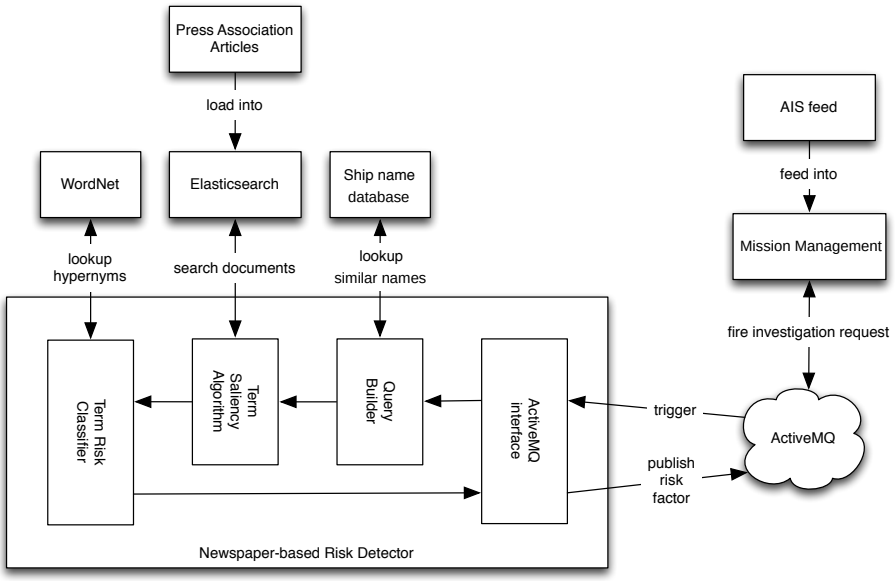

Fig. 1. System Architecture

to the query, and once to retrieve a set of documents that do not match the query. The Rocchio algorithm[6] is then executed against the term vectors of the two results, in essence calcuating a prototype document for the ship under investigation. From this prototype document, every term that on average occurs more than once in every five documents is lemmatized and matched to one or more WordNet synsets by the *Term Risk Classifier*. The selected cut-off point of five is an ad hoc choice, based on our experience with the technique and data set, and has a marginal effect on the quality of the results, but a big influence on the processing speed. All these synsets are then compared to a set of pre-defined synset trees that are each mapped to a specific event type in order to compile an evidence score for the ship for each event type detected.

3.1 Query Builder

ElasticSearch is a distributed search and analytics engine built on top of Apache Lucene. We use a modified version of ElasticSearch that allows us to retrieve the indexed term counts for each indexed document. The search cluster has been filled with approximately 25,000 maritime-related press releases from the Press Association (essentially all articles with the meta-data term "sea" of the past 10 years) and detailed records of about 40,000 ships from IHS FairPlay. These ship records contain, for example, details about ship owners and current and previous names of ships.

The name of each ship in our area of interest is broadcast by its Automatic Identification System (AIS) transmitter, which allows us to formulate a query to find all press releases in which said vessel is mentioned. Due to the fact that ship names often consist of multiple words, and names of different ships can be quite similar, we first search the detailed ship records for other ships that have names that contain the name of the ship being investigated. We can then take extra care to exclude these other names from our search. For example, this allows us to exclude documents about the *Queen Mary* when searching for articles about the ship *Mary*, which otherwise would have matched and been returned. In the case of the Sirius Star, there are no ships with a name that contains the phrase "Sirius Star" other than the Sirius Star itself. So we illustrate the query builder with the example of the Mary and the Queen Mary. The JSON ElasticSearch query constructed to fetch documents about the Mary while excluding documents about the Queen Mary is shown below.

```
query : {
  bool : {
    must : { text_phrase_prefix : { text : "mary" } },
    must_not : [ { text_phrase_prefix : { text : "queen mary" } } ]
  }
}
```

Once the query has been constructed we retrieve the term vectors of all documents that are returned by the query, and the term vectors of a sample of 100 documents that do not match the query. As an example, if we would investigate the Sirius Star, this would result in a set of term vectors about the hijacking of the Sirius Star, as well as a set of term vectors about a number of different arbitrary vessels.

3.2 Term Saliency Algorithm

The Rocchio Algorithm[6] is a relevance feedback technique. This algorithm is applied over the two sets of term vectors, in order to reshape these into one term vector that best describes the documents about the investigated vessel with respect to the other documents in the corpus. The algorithm is described in Equation 1, with $\overrightarrow{Q_m}$ being the modified query vector, $\overrightarrow{Q_o}$ the original query vector, D_r the set of term vectors of related documents, and D_{nr} the set of term vectors of non-related documents. a, b and c are weights, in this case set to 0, 1 and 1 respectively. In our Sirius Star example, D_r would contain terms such as *hijack*, *pirates*, *ship* and *captain*, while D_{nr} would contain *ship*, *captain*, *sea* and *engine*. The resulting vector $\overrightarrow{Q_m}$ would then result in *hijack*, *pirates*, *ship* and *captain*, but with significantly higher weights attached to the first two terms than the last two terms.

$$\overrightarrow{Q_m} = \left(a * \overrightarrow{Q_o}\right) + \left(b * \frac{1}{|D_r|} * \sum_{\overrightarrow{D_j} \in D_r} \overrightarrow{D_j}\right) - \left(c * \frac{1}{|D_{nr}|} * \sum_{\overrightarrow{D_k} \in D_{nr}} \overrightarrow{D_k}\right) \quad (1)$$

This essentially calculates *the query that should have been asked to the search engine in order to get the most number of relevant documents and the least number of irrelevant documents*, which essentially is a list of terms specific about the vessel being identified, along with weights. All dimensions of the $\overrightarrow{Q_m}$ that are lower than 0.2 are removed in order to speed-up subsequent processing.

3.3 Term Risk Classifier

WordNet is a large lexical database of English, with words grouped into sets of cognitive synonyms (synsets), interlinked by means of semantic and lexical relations[5].

To relate terms to event types, we have created a set of pre-defined concepts for each event type we wanted to detect. For example, the concept set for *accident* contains synsets like *hit*, *collide*, *explode*, *sink*, etc. All hyponyms of these synsets are automatically included as well.

Each term in the term vector $\overrightarrow{Q_m}$ is first lemmatized. WordNet is then searched for synsets that contain the term's lemma. If any of these synsets is a match to any of the predefined concept sets, the vessel is considered to have participated in at least one event of the matching event type.

To compensate for ambiguous words, each event type is assigned a score, consisting of the sum of the score for each matching term in the event type's concept set. This term score is in turn calculated by dividing its term frequency by the number of synsets in WordNet that contain the term's lemma. Each found event type with a score lower than 0.1 is pruned for not having enough evidence. In our example, both *hijack*, and *pirates*' lemmatized form *pirate* are a match to the *thievery and piracy related events* event type, which consequenly gets a score derived from both these terms.

3.4 Simple Event Model

To keep the output simple, we assume each matched event type with evidence (in our case *thievery and piracy*) corresponds to one distinct event, with its score as a measure of supporting evidence. These events are represented in RDF, using the Simple Event Model ontology (SEM), which is a light-weight event ontology designed with a minimum of semantic commitment to guarantee maximal interoperability[8]. Each event is modeled as an anonymous event with a *sem:eventType* type corresponding to the matched event type. The ship under investigation is linked to the event as an Actor. All other event properties (Place, Time, other actors) are left unspecified as insufficient information is available to specify these, but the schema does allow specifying them later on, either by extensions to our system or by an outside tool.

The W3C standard provenance ontology PROV[4] is used to link the event back to the documents from which it was originally derived. This allows the operator to manually read the news articles for ships that trigger an alert in order to confirm the potential threat.

The resulting events are then sent back to the vessel observation system, where they are prioritized based on the detected event types and their evidence scores.

For the moment we ignore the date of the past events (apart from the limit of 10 years in the past imposed by the coverage of the news corpus). Possible performance improvements could be obtained by investigating the order and time distribution of the events.

4 Evaluation

To evaluate our system's performance, we have compiled a gold standard, consisting of ships with known risk cases. We then let the system investigate these vessels, and evaluated at three points:

- *E1:* Given the name of a ship, does the system provide us with relevant documents that relate to the ship being investigated? This is done by manually reviewing the documents that are retrieved, checking whether they are relevant for the given ship.
- *E2:* Does the system classify the correct event types that a human annotator would also find when *only* looking at the documents retrieved in *E1*?
- *E3:* Given a ship with known risk behaviour in the past, does the system classify this ship correctly and completely? This is essentially a combination of *E1* and *E2* with a slightly different gold standard, which was constructed before running the evaluation.

Behavior for non-remarkable vessels was also evaluated qualitatively by classifying a large set of around 76000 known ship names and looking for anomalies in the results. A thorough evaluation would include a comparison to actual decisions made by coast guard personnel with and without the assistance of the tool. This remains future work for the moment.

5 Results

The evaluation results for evaluation criteria *E1*, *E2* and *E3* can be found in Table 1. After batch evaluating 76696 vessels, 3064 triggered an alert on at least one category.

6 Discussion

From Table 1 - in particular the difference in recall between *E2* and *E3* - we can see that the system performs quite well for those ships that are actually mentioned in news articles in our corpus. The drop in recall from *E2* to *E3* can for the most part be explained by the lack of news articles found for the affected vessels (see the D_F column for *E1*). A larger and more up-to-date corpus of news articles should hopefully improve these results.

Table 1. Evaluation results for evaluation criteria E1, E2 and E3 described in Section 4. D_F denotes number of documents found by the system, D_R represents which of these documents were actually relevant to the vessel. For both *E1* and *E2*, T_{TP} indicates the number of true positive classified event types, T_{FP} represents false positives, and T_{FN} denotes the number of false negatives. P and R denote Precision and Recall respectively.

	E1			E2					E3				
Vessel	D_F	D_R	P	T_{TP}	T_{FP}	T_{FN}	P	R	T_{TP}	T_{FP}	T_{FN}	P	R
Exxon Valdez	34	8	0.24	3	0	0	1.00	1.00	3	0	0	1.00	1.00
Probo Koala	0	0	1.00	0	0	0	1.00	1.00	0	0	1	1.00	0.00
Costa Concordia	0	0	1.00	0	0	0	1.00	1.00	0	0	2	1.00	0.00
Estonia	136	80	0.59	0	0	1	1.00	0.00	0	0	1	1.00	0.00
Herald of Free Enterprise	95	52	0.55	1	1	0	0.50	1.00	1	1	0	0.50	1.00
Sirius Star	46	44	0.96	2	0	0	1.00	1.00	2	0	0	1.00	1.00
Vindo	26	26	1.00	2	0	0	1.00	1.00	2	0	0	1.00	1.00
Edinburgh Castle	4	0	0.00	0	1	0	0.00	1.00	0	1	1	0.00	0.00
Zeldenrust	1	1	1.00	2	1	0	0.67	1.00	2	1	0	0.67	1.00
Scandinavian Star	9	9	1.00	1	3	0	0.25	1.00	1	3	0	0.25	1.00
Lady Azza	0	0	1.00	0	0	0	1.00	1.00	0	0	2	1.00	0.00
Ronin	2	2	1.00	2	1	0	0.67	1.00	2	1	0	0.67	1.00
Union Pluto	1	1	1.00	2	1	0	0.67	1.00	2	1	0	0.67	1.00
Achille Lauro	72	48	0.67	0	0	3	0.00	0.00	0	0	2	1.00	0.00
Viking Victor	23	22	0.96	0	1	1	0.00	0.00	0	1	1	0.00	0.00
Astree	4	4	1.00	1	2	0	0.33	1.00	1	2	0	0.33	1.00
Total	453	297	0.66	16	11	5	0.59	0.76	16	11	10	0.59	0.62

Of the ships that were mentioned in at least one news article, the system only failed to raise the correct red flags for three instances, one of which did trigger an alert but for an incorrect event type. For the other two, the system was most probably thrown off by the fact that these vessels (Estonia and Achille Lauro) were mentioned a lot in news articles about other events involving different ships. One could say that these ships might have been 'too famous' to be correctly picked up.

The false positives generated by the system seem to mostly originate from the fact that, in addition to the correct event type, sometimes additional types are triggered by the documents that describe the correct event type. For example, in the case of smuggling, the smuggled goods are often *seized* by the authorities after being discovered, which in turn triggers the *thievery and piracy* category, as the system in this case canot discern between the legal interpretation of *seizing of goods*, and the illegal one.

Out of the 76696 batch-evaluated ships, the system did not detect any risk factors for 73532. This means that, with our system in use, the operator will receive an alert and has to confirm approximately 4% of all vessels. If we assume this is a representative sample of ships, this will cause the operator to have to look at approximately 5 vessels each hour for the Netherlands Exclusive Economic Zone (compared to 120 when manually assessing all ships).

When manually reviewing the ships that trigger alerts, a considerable number of them either have names that refer to a place ('baltic sea', 'brasilia', 'brooklyn', 'casablanca'), or are named after words that have something to do with the exact threats we are looking for ('buccaneer', 'dealer', 'robin hood'). Due to the search engine only looking at words in the press releases without actually disambiguating them, the queries formed from the names of these ships most probably return articles about entirely different ships. These false positives, however, can be very quickly dismissed by the operator, as one glance at the documents should be enough to see they are not about said ship.

In this paper we wanted to focus on the statistical saliency algorithm and the term risk classification part of the entire event detection pipe line. We assume that a thorough NER tool would solve many of the cases discussed above if properly retrained with domain-specific terms such as ship vessels and port names.

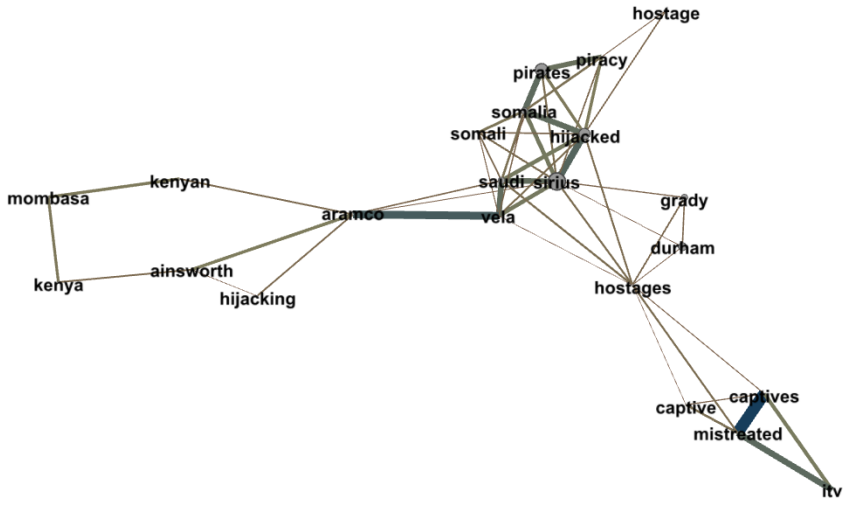

Fig. 2. Term Network for the Sirius Star

7 Conclusion

In this paper, we have described an event-type extractor on top of ElasticSearch, and applied this system in a case study concerned with assisting maritime se-

curity operators to assess potential risk factors of vessels. Our main objectives were to investigate if such a system, based on a combination of relevance feedback, lexical databases and domain information would yield results useful for the surveillance task assigned to maritime security operators.

With a task-oriented focus, we have evaluated the performance of our system using a set of vessels with known risk factors, and concluded that, given that news articles about certain events actually exist in the system's database, the system can raise red flags about ships with a suspicious history fairly accurately, and does not produce enough false negatives to overload the operator.

We will continue this line of research by using co-occurrence metrics to form term networks in order to detect clusters of terms that may point to separate distinct events. An example of such a network is shown in Figure 2. Natural Language Processing tools will then be employed to further fill in the rest of the event properties such as other actors, places and times.

Acknowledgements

We wish to thank Thomas Ploeger for his work in the initial stages of this research, and the Press Association for providing us with a set of maritime-related press releases. This publication was supported by the Dutch national program COMMIT. The research work was carried out as part of the Metis project under the responsibility of the Embedded Systems Innovation by TNO with Thales Nederland B.V. as the carrying industrial partner.

References

[1] M. Atkinson, J. Piskorski, E. Goot, and R. Yangarber. Multilingual real-time event extraction for border security intelligence gathering. In U. K. Wiil, editor, *Counterterrorism and Open Source Intelligence*, volume 2 of *Lecture Notes in Social Networks*, pages 355–390. Springer Vienna, 2011.

[2] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. Gate: an architecture for development of robust hlt applications. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 168–175. Association for Computational Linguistics, 2002.

[3] T. Hendriks and P. van de Laar. Metis: Dependable cooperative systems for public safety. *Procedia Computer Science*, 16:542–551, 2013.

[4] T. Lebo, S. Sahoo, D. McGuinness, K. Belhajjame, J. Cheney, D. Corsar, D. Garijo, S. Soiland-Reyes, S. Zednik, and J. Zhao. Prov-o: The prov ontology. *W3C Recommendation, http://www. w3. org/TR/prov-o/(accessed 30 Apr 2013)*, 2013.

[5] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. Introduction to wordnet: An on-line lexical database*. *International journal of lexicography*, 3(4):235–244, 1990.

[6] J. J. Rocchio. Relevance feedback in information retrieval. 1971.

[7] P. D. Turney, P. Pantel, et al. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37(1):141–188, 2010.

[8] W. R. Van Hage, V. Malaisé, R. Segers, L. Hollink, and G. Schreiber. Design and use of the simple event model (sem). *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(2):128–136, 2011.

[9] P. Vossen, E. Agirre, N. Calzolari, C. Fellbaum, S.-k. Hsieh, C.-R. Huang, H. Isahara, K. Kanzaki, A. Marchetti, M. Monachini, et al. Kyoto: a system for mining, structuring and distributing knowledge across languages and cultures. In *LREC*, 2008.